Viktoria Stray
Klaas-Jan Stol
Maria Paasivaara
Philippe Kruchten (Eds.)

LNBIP 445

Agile Processes in Software Engineering and Extreme Programming

23rd International Conference on Agile Software Development, XP 2022
Copenhagen, Denmark, June 13–17, 2022, Proceedings

Springer

Lecture Notes in Business Information Processing 445

More information about this series at https://link.springer.com/bookseries/7911

Viktoria Stray · Klaas-Jan Stol ·
Maria Paasivaara · Philippe Kruchten (Eds.)

Agile Processes in Software Engineering and Extreme Programming

23rd International Conference on Agile Software Development, XP 2022
Copenhagen, Denmark, June 13–17, 2022
Proceedings

Springer

Editors

Viktoria Stray
University of Oslo
Oslo, Norway

Maria Paasivaara
LUT University
Lahti, Finland

Klaas-Jan Stol
University College Cork
Cork, Ireland

Philippe Kruchten
University of British Columbia
Vancouver, BC, Canada

ISSN 1865-1348 ISSN 1865-1356 (electronic)
Lecture Notes in Business Information Processing
ISBN 978-3-031-08168-2 ISBN 978-3-031-08169-9 (eBook)
https://doi.org/10.1007/978-3-031-08169-9

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 papers of XP 2022, the 23rd International Conference on Agile Software Development, held during June 13–17, 2022, at the IT University of Copenhagen, Denmark.

XP is the premier Agile software development conference combining research and practice. It is a unique forum where Agile researchers, practitioners, thought leaders, coaches, and trainers get together to present and discuss their most recent innovations, research results, experiences, concerns, challenges, and trends. The XP conference series provides an informal environment to learn and trigger discussions, welcoming both new and seasoned Agile practitioners.

Although the XP conference series originally focused on eXtreme Programming, it has since widened its scope to all things Agile. XP 2022 solicited contributions that address all modern Agile approaches, as well as the application of Agile in a variety of domains. While Agile methods have been successfully scaled up to large and distributed projects, we are now facing new challenges in the era of hybrid work. The COVID-19 pandemic has served as a catalyst for this trend. Hybrid work brings new challenges: not quite distributed and not quite co-located, instead individual developers are working from anywhere and touching base with the office intermittently. Thus, the theme for XP 2022 was "Agile in the Era of Hybrid Work."

The XP 2022 conference featured ten tracks, covering research papers, research workshops, experience reports, industry and practice, Agile in education and training, journal-first papers, leadership, Agile games, diversity and inclusion in Agile, and lightning talks. In total, we received 235 submissions, which demonstrates that the XP community continues to grow.

The research paper track invited submissions of previously unpublished high-quality research papers, full and short, related to Agile and lean software development. We welcomed submissions addressing topics across the full spectrum of Agile software development, broadly focused on Agile, on issues of interest to researchers or practitioners, or both.

The research track received a total of 40 submissions. Based on a thorough review process, 14 papers, 13 full and one short, were accepted which address a range of topics, including studies of Agile practices and processes, and how Agile scales "in the large."

We would like to extend our sincere thanks to all the people who contributed to XP 2022: authors, speakers, reviewers, sponsors, shepherds, chairs, and volunteers. Finally, we would like to express our gratitude to the XP Conference Steering Committee and the Agile Alliance for their ongoing support.

June 2022

Viktoria Stray
Klaas-Jan Stol
Maria Paasivaara
Philippe Kruchten

Organization

Conference Chair

Maria Paasivaara	LUT University and Aalto University, Finland

Program Co-chairs

Viktoria Stray	University of Oslo, Norway
Klaas-Jan Stol	Lero and University College Cork, Ireland

Publication Chair

Philippe Kruchten	University of British Columbia, Canada

Program Committee

Noura Abbas	Colorado Technical University, USA
Ademar Aguiar	University of Porto, Portugal
Craig Anslow	Victoria University of Wellington, New Zealand
Hubert Baumeister	Technical University of Denmark
Jan Bosch	Chalmers University of Technology, Sweden
Frank Buschmann	Siemens AG, Germany
Fabio Calefato	University of Bari, Italy
Daniela S. Cruzes	Norwegian University of Science and Technology, Norway
Torgeir Dingsøyr	Norwegian University of Science and Technology, Norway
Yael Dubinsky	StepAhead, Israel
Jutta Eckstein	IT communication, Germany
Steven Fraser	Innoxec, USA
Ilenia Fronza	Free University of Bozen-Bolzano, Italy
Juan Garbajosa	Universidad Politécnica de Madrid, Spain
Alfredo Goldman	University of São Paulo, Brazil
Peggy Gregory	University of Central Lancashire, UK
Eduardo Guerra	Free University of Bozen-Bolzano, Italy
Tomas Gustavsson	Karlstads universitet, Sweden
Orit Hazzan	Technion - Israel Institute of Technology, Israel
Helena Holmström Olsson	University of Malmö, Sweden
Fabio Kon	University of São Paulo, Brazil

Philippe Kruchten	University of British Columbia, Canada
Marco Kuhrmann	Reutlingen University, Germany
Casper Lassenius	Aalto University, Finland
Ville Leppänen	University of Turku, Finland
Kashumi Madampe	Monash University, Australia
Frank Maurer	University of Calgary, Canada
Tommi Mikkonen	University of Helsinki, Finland
Alok Mishra	Atilim University, Turkey
Nils Brede Moe	SINTEF, Norway
Parastoo Mohagheghi	Norwegian Labour and Welfare Administration, Norway
Jürgen Münch	Reutlingen University, Germany
Cécile Péraire	Carnegie Mellon University, USA
Alireza Pourshahid	University of Ottawa, Canada
Ken Power	Independent Consultant, Ireland
Rafael Prikladnicki	PUCRS, Brazil
Pilar Rodríguez	Universidad Politécnica de Madrid, Spain
Helen Sharp	The Open University, UK
Darja Šmite	Blekinge Institute of Technology, Sweden
Rasmus Ulfnes	SINTEF, Norway
Stefan Wagner	University of Stuttgart, Germany
Xiaofeng Wang	Free University of Bozen-Bolzano, Italy
Eileen Wrubel	Software Engineering Institute, USA

Steering Committee

Hubert Baumeister	Technical University of Denmark, Denmark
François Coallier	Ecole de technologie supérieure, Canada
Jutta Eckstein	IT communication, Germany
Steven Fraser	Innoxec, USA
Juan Garbajosa (Chair)	Universidad Politécnica de Madrid, Spain
Peggy Gregory	University of Central Lancashire, UK
Ellen Grove	Agile Alliance, USA
Casper Lassenius	Aalto University, Finland
Michele Marchesi	University of Cagliari, Italy
Maria Paasivaara	LUT University and Aalto University, Finland
Viktoria Stray	University of Oslo, Norway

Sponsoring Organization

Agile Alliance, USA
Teresa Foster and Ellen Grove

Contents

Agile Practices

Agile Processes

Agile in the Large

Agile Practices

Benefits of Card Walls in Agile Software Development: A Systematic Literature Review

Marc Sallin(✉) and Martin Kropp(✉)

University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland

marc.salin@outlook.com, martin.kropp@fhnw.ch

Abstract. Card walls are often used to visualize various aspects of the software development process. They are an essential and widespread agile practice. Despite the drawback of physical card walls, its digital version is often not considered a sufficient alternative. This paper aims to find the reason for this and suggests how to evolve digital card walls into a viable alternative. We conducted a systematic literature review and analyzed twenty-two studies. We identified which desirable effects agile teams get from card wall usage and derived a set of properties a card wall needs to achieve those effects. Furthermore, we suggested a typology of card walls to compare the benefits and challenges among them.

Keywords: Agile · Software development · Card wall · Task board · Scrum board · Information information radiator · Big visible chart · Systematic literature review

1 Introduction

Card walls play a central role when working in an agile team. According to the state of agile report [1], most agile teams use card walls for team collaboration and visualization of the project status. In this paper, the term *card wall* is used as a synonym for various kinds of boards to track and visualize the team's current work and progress. In the mentioned study, the usage of a Kanban board and a task board, in general, are the two highest-ranked tools in the analysis of agile tool usage. While there exists a variety of digital board soalutions, which offer a wide range of inherent benefits, physical card walls are still widespread [2], and agile teams decide explicitly to use a physical card wall over a digital one [3,4]. This raised the question of why agile teams still very often favor physical card walls over digital and what is necessary to make the digital solution more competitive with the physical ones. What makes the question especially interesting is the fact that the COVID-19 pandemic has served as a catalyst for the hybrid working trend, and many teams do not plan to come back in the office full-time [1]. This paper aims to describe how digital card walls need to be realized to offer the same benefits as a physical solution, especially concerning

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 3–18, 2022.
https://doi.org/10.1007/978-3-031-08169-9_1

hybrid-working. We examined the current state of research with a systematic literature review (SLR) to answer this question. Our main research question is:

RQ: How do digital card walls need to be implemented to be able to replace physical solution?

To answer this question and guide the SLR, we framed more granular research questions. First, we want to understand why and how agile teams use card walls. Understanding the benefits of applying this agile practice makes it possible to infer what characteristics are essential to replicate the desired experience. Second, we wondered why agile teams decided to use physical card walls instead of digital card walls. That means we wanted to understand the benefits and challenges of physical and digital card walls. This leads to the following two research questions.

RQ1: What makes card walls beneficial to agile teams?
RQ2: What are the challenges & benefits of physical/digital card walls?

The rest of the paper is structured as follows. The methodology of the SLR is described in Sect. 2, followed by the results in Sect. 3. In Sect. 4, the results are discussed with concrete suggestions about how digital card walls could be improved, and Sect. 5 contains the conclusions.

2 Research Method

We conducted a Systematic Literature Review (SLR) to answer the two research questions. We followed the recommended general steps for literature review [5–8]. After identifying the need for a systematic review, we derived the research questions. Then, we executed the search for relevant studies using a predefined search string to retrieve results from several databases. After cleaning up and eliminating duplicates, we screened the records and included studies based on the inclusion/exclusion criteria. Finally, we reviewed and analyzed the full text of the remaining studies. The described process is visualized in Fig. 1.

2.1 Search Process

We defined keywords to retrieve potentially relevant articles from the databases. To define the keywords, we looked at studies and non-scientific literature about agile software development and examined synonyms for describing card walls' usage in an agile context. The resulting keywords are shown below.

Agile: Agile, Scrum, Kanban, Scrumban, Extreme programming
Card wall: card wall, Scrum wall, Scrum board, status board, task board, story board, information radiator, Kanban board, wall board

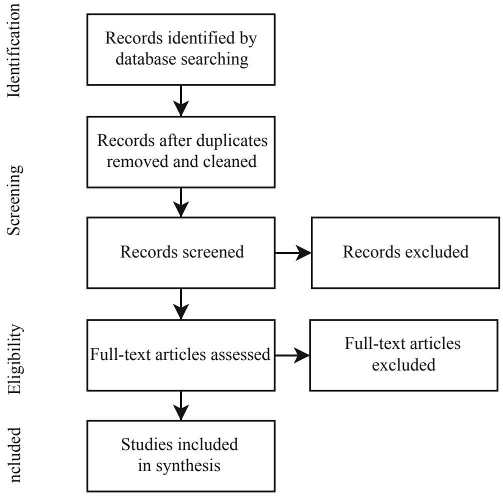

Identification
Screening
Eligibility
ncluded
Records identified by database searching
Records after duplicates removed and cleaned
Records screened
Records excluded
Full-text articles assessed
Full-text articles excluded
Studies included in synthesis

Fig. 1. Research methodology, adapted from PRISMA

Weidt and Da Silva recommend using six search engines to conduct an SLR [7]. However, Gusenbauer and Haddaway found that only three out of the stated six are suitable to be used as principal search engines [10]. Therefore, we used the following three search engines to search the literature for this study: ACM Digital Library[1], ScienceDirect[2], and Scopus[3]. Out of the identified keywords, we constructed the query string shown in Table 1. Table 2 shows the applied inclusion and exclusion criteria. The inclusion criteria define the topics we were looking for. If one or more of the criteria matched included a study. However, we excluded a study if one of the exclusion criteria matched.

2.2 Data Collection

We executed the search on April 10th, 2020. The initial search in the three databases returned 829 studies, from which 667 were candidates for further processing. Table 3 shows the results of every step in the identification process, and Fig. 2 shows the graphical representation of the search process[4] First, we did the initial search using the defined query string. Then, if the search engine offered refinement filters, we applied these as a second step according to the listed exclusion criteria. Finally, we filtered the results manually in the third step and excluded obvious false positives like whole journals or books. The only deviation from the protocol was that ScienceDirect could not process the whole

[1] portal.acm.org.

[2] sciencedirect.com.

[3] scopus.com.

[4] Notice that the table contains more detail than the visualization, and the steps do not directly match.

Table 1. Search query to retrieve studies

```
(agile OR scrum OR kanban OR scrumban OR “extreme programming”)
AND
((“scrum wall” OR “scrumwall” OR “scrum-wall” OR “scrum-board)” OR
(“scrum board” OR “scrumboard” OR “statusboard” OR “status board”) OR
(“status-board” OR “cardwall” OR “card-wall” OR “card wall”) OR
(“taskboard” OR “task-board” OR “task board”) OR
(“storyboard” OR “story-board” OR “story board”) OR
(“information radiator” OR “information-radiator”) OR
(“kanban board” OR “kanban-board” OR “kanbanboard”) OR
(“wallboard” OR “wall board” OR “wall-board”))
```

query string in one step. Therefore, we divided the query into three parts, merged the results, and removed duplicates.

Table 2. Inclusion and exclusion criteria.

Inclusion	Exclusion
• Card wall (digital or physical)	
•Communication	
• Visualization	
• Workspace	• Not written in English or German
• Process Monitoring/Project Management	• Not domain agnostic or not Software Engineering domain
• Global/Large scale organizations	
• Distributed teams	• Not empirical e.g. no manuals or guides
• Agile Practices/Adoption	
• Tools to support agile practices	

In the resulting recordset, we extracted the following data from each study to use in the screening process: Title, Authors, Abstract, Keywords, source (journal or conference), and complete reference. We then retrieved the full article and extracted the following metadata for the articles that passed the screening.

- The type of research.
- The agile methodology, which was the subject of the investigation.
- The main topic of the research.
- If card walls were the main topic of the research.
- The contribution of the study to the research about card walls.

We reviewed the title, abstract, and keywords of every record for the screening process. Of the 667 initial records, we classified 77 as definitely or potentially matching the defined inclusion criteria and retrieved the full text. After assessing the complete text, we excluded 55 articles because they did not match the

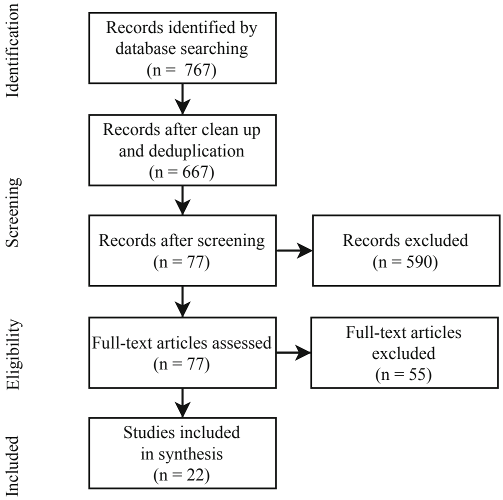

Identification
Records identified by database searching (n = 767)
Records after clean up and deduplication (n = 667)
Screening
Records after screening (n = 77)
Records excluded (n = 590)
Eligibility
Full-text articles assessed (n = 77)
Full-text articles excluded (n = 55)
Included
Studies included in synthesis (n = 22)

Fig. 2. Number of included/excluded records.

inclusion criteria or matched one of the exclusion criteria. Finally, we identified 22 articles to include in the synthesis (see Table 4). In 11 of the identified studies, the card wall is the research object. The other 11 studies have another main topic but contain important information for answering the research questions.

Table 3. Number of records from identification including source.

Database	Number of records			
	Initial	After refinement	After cleaning	Unique
ACM	484[a]	406[b]	390	**667**
ScienceDirect	121	110[b]	97	
Scopus	287[c]	251[d]	242	
Total	**892**	**767**	**729**	

[a] We searched "The ACM Guide to Computing Literature" database
[b] Include only periodicals, proceedings, and journals
[c] Search all text fields
[d] Include only English or German, conference papers or articles

2.3 Data Analysis

To answer our research questions, we were interested in the seen and experienced effects when working with the boards and the feedback from the users. Thus, we did not consider explanations about a methodology or practice taken from

Table 4. Studies included in the synthesis.

Id	Author/Study	Date	Agile Methodology	Main topic	Main topic card wall	Contribution to card wall research
S1	Ahmad et al. [11]	2018	Kanban	Kanban in SWE	No	Experience report
S2	Annosi et al. [12]	2020	Scrum/Kanban	Organizational learning	No	Experience report
S3	Anwar et al. [13]	2016	Scrum	Agile adoption	No	Experience report
S4	Azizyan et al. [2]	2011	Agile	Scrum tools	Yes	Tool usage
S5	Bakke & Agnar [14]	2019	Lean	Agile adoption	No	Experience report
S6	Bastarrica et al. [3]	2018	Agile	Agile adoption	No	Experience report
S7	Eckhart & Feiner [15]	2016	Scrum	Scrum tools	Yes	Requirements to card walls
S8	Hajratwala & Nayan [16]	2012	Scrum	Card wall usage	Yes	Requirements to card walls
S9	Hunt et al. [17]	2007	Agile	Workspace	No	Experience report
S10	Katsma et al. [18]	2013	Scrum	Card wall usage	Yes	Requirements to card walls, challenges with card wall, tool usage
S11	Kropp et al. [19]	2017	Scrum	Digital card wall	Yes	Requirements to card walls
S12	Liechti et al. [20]	2017	Agile	Actable Metrics	No	Benefits of physical card walls
S13	Mishra et al. [21]	2012	Agile	Workspace	No	Impact of card walls
S14	Nakazawa & Tanaka [22]	2016	Kanban	Digital Kanban Board	Yes	Impact of card walls
S15	Perry [23]	2008	Agile	Digital and physical card walls	Yes	Requirements to card walls, prod & cons of digital and physical
S16	Pikkarainen et al. [24]	2008	Agile	Agile practices	No	Impact of card walls
S17	Rola et al. [4]	2016	Scrum	Workspace	No	Impact of card walls, Benefits of physical card walls
S18	Rubart [25]	2014	Scrum	Digital card wall	Yes	Experiment with digital card wall
S19	Rubart & Freykamp [26]	2009	Scrum	Digital card wall	Yes	Benefits of physical card walls, requirements of digital card wall
S20	Sharp & Robinson [27]	2008	XP	Card wall usage	Yes	Impact of card walls, Benefits of physical card walls
S21	Sharp et al. [28]	2009	Agile	Physical artefacts	Yes	Requirements to Card Walls
S22	Wiklund et al. [29]	2013	Scrum	Agile adoption	No	Requirements to Card Wall (different boards)

a guide or recommendation. Instead, we looked for studies with interviews, surveys, observations, and experience reports. We applied an inductive data driven approach to develop thematic categories. We did this by scanning the identified literature for statements that help answer our research questions and highlighted those statements. That means, statements about benefits, challenges or the way of working with regards to card walls. In the next step, we worked out categories for the statements per research question and finally condensed the categories. The results are shown in tables 5 - 10, and presented and discussed in the next section. For RQ1, we did not distinguish between physical and digital card walls since we were interested in the general benefits of card walls. For RQ2, the type of card wall was considered to be able to list the benefits and challenges depending on the card wall type.

3 Results

In this section, we present the results of the SLR and the answers to the research questions. It is divided into two sections, one devoted to each research question.

3.1 RQ1: What Makes Card Walls Beneficial to Agile Teams?

Table 5 lists the benefits grouped by category why agile teams use card walls and also references the reporting literature[5]. The here listed benefits concern general benefits that are seen and experienced from card walls independent of their nature (physical or digital boards). On one side, the benefits concern visibility aspects of the board (visualization, always-on, transparency). On the other side, team aspects like decision making and communication, for example. In the following, the categories are explained in detail.

Table 5. Benefits of card wall usage.

Id	Category	Reporting studies
C1	Attention of team	[15,18,27]
C2	Collaboration and communication	[4,15,18,21,22]
C3	Decision making	[11,24]
C4	Focus	[12,23,24]
C5	Gathering place	[15,18,25,27]
C6	Knowledge dissemination	[4,13,21,22]
C7	Up to date information	[4,22,23,27]
C8	Physical interaction	[4,17,18,27]
C9	Progress tracking	[3,4,12,18,20,24]
C10	Transparency	[11,12,14,24]
C11	Visualize work	[11,12,16,17,23,27,29]
C12	WIP control	[11,16,22]

C1-Attention of team: The act of updating the card wall, i.e., walking to the card wall and interacting with it, raises the attention of other team members and thus helps to keep the team up to date [15,27]. Furthermore, a large wall, placed in a central place, which is always "on" catches everyone's attention by itself [18].

[5] The following Excel sheet shows the extracted segments of the studies and the assigned codes, which were later used to build the categories https://1drv.ms/x/s!ApmGN3k-vuHI1YAjDWozMovfryHukQ.

C2-Collaboration and communication: The visible interaction with the card wall encourages open communication and collaboration in the team [15,18]. Moreover, as there is only one single interface to the tool, it acts as a central meeting point and leads to more face-to-face communication [18].

C3-Decision making: With the visual nature and up-to-dateness, the card wall supports decision making like prioritization, dependencies, and resource allocation [11,24].

C4-Focus: In ceremonies like the daily stand-up, which are held in front of the wall, the team is more focused on talking about the currently relevant tasks [23]. Additionally, the usage of a card wall helps to keep focused on the tasks that one is working on [12,23]. In one study, it was reported that the card wall helps to increase the visibility of common short-term goals [24].

C5-Gathering place: The card wall becomes a gathering place, either to hold discussions [15] or also because ceremonies like daily stand-ups are held in front of it [25,27].

C6-Knowledge dissemination: It was reported that the card wall helps with knowledge dissemination even with no further explanation [13]. As a result of shared knowledge and understanding, redundancy and the overlapping of work are minimized [21]. With a broader view, team members are encouraged to grab tasks that are less related to them themselves [22]. Card walls support knowledge dissemination by the fact that they are used to communicate besides the cards, which represent tasks to work on [4].

C7-Up to date information: Several studies reported that the team members were motivated to keep the information on a card wall up to date [4,22,23,27].

C8-Physical interaction: The physical interaction with the card wall leads to a good feeling, which is a source of motivation. One aspect of the good feeling arises due to the visibility of the action by the team and the immediate feedback [4,17,18]. There were also other interactions mentioned which are related to the card wall. For example, the cards are pulled away from the wall when working on them, signifying responsibility. Furthermore, a card sometimes acts as a kind of token. Team members are pulling it from the wall and holding it while they are talking about it in daily stand-up meetings [27].

C9-Progress tracking: As all activity which the team currently works on is visible on the card wall and up to date, it makes it an excellent tool for tracking the progress [3,4,12,18,20,24].

C10-Transparency: The wall is placed at a prominent and visible place. Thus, the work and progress are transparent to everyone in the room. Furthermore, all tasks and their assignment are visible at a glance, which also makes transparent who currently works on which tasks [11,12,14,24].

C11-Visualize work: The aspect that a card wall is designed to visualize the work is considered a significant benefit. The mentioning of visualization as a benefit or instrument in a broad range of studies reflects this [11,12,16,17,20,22,23,27,29].

C12-WIP control: The card wall helps the team to track the current work in progress [11,16]. Due to the visual nature of presenting the cards, it becomes evident if there is too much work in progress, even without explicitly defining a work-in-progress limit [22].

The results show that card walls generally play an important role in agile team collaboration, especially concerning serving as an information radiator and for common decision-making.

3.2 RQ2: What Are the Challenges and Benefits of Physical/digital Card Walls?

With this research question, we wanted to analyze the benefits and challenges of physical and digital card walls and why teams still often prefer physical over digital card walls.

Table 6. Card wall types

Id	Type	Kind	Description
T1	Paper	Physical	Physical wall with paper and cards on it.
T2	Paper & Audio photo/Video	Physical	T1 but its shared/documented with video and/or photo.
T3	Software	Digital	Software which helps keep track of the task but with no special visualization nor physical appearance.
T4	Software with virtual card wall	Digital	T3 but replicating the visual appearance of a physical card wall.
T5	Software with non-interactive vertical screen	Digital	T4 but permanently displayed on a big visible screen.
T6	Software with interactive vertical screen	Digital	T5 but interactive screen, e.g., drag and drop the virtual cards

The benefits and challenges depend on the kind of card wall. Different types of digital card walls must be distinguished. Therefore, we created the typology of different card wall types shown in Table 6. This typology is based on the studies identified in this SLR, which aimed to replicate the aspects of the physical card wall: Scrumpy [15], Kanban Tool [22], Multi-touch-scrum task board [25], Cooperative Task Board [26], and aWall [19]. Furthermore, the usage scenarios from Katsma et al. [18] are taken into account. Unfortunately, it was impossible to extract the concrete used card wall type from the analyzed reports. The

Table 7. Reported benefits of physical card walls.

Id	Category	Subcategory
PB1	Physical presence	Big and visible [16,20,23]
		Publicly available [13,19,23]
		Meeting place [13,18,23,28]
		Availability [18,23,27]
		Attraction outside of team [13]
		Place for extra information [19]
PB2	Usability	Easy to modify [19,27]
		No process pre-defined [15,28]
		Ease of use [18,23,28]
PB3	Physical interaction	Responsibility [19,27]
		Communication Frequency [21,23]
		Motivation [17,18,23,24]
PB4	Overview	Makes sloppy tracking visible [4,23]
		Focused [19,23]
		Good overview [19]
PB5	Cost	Cheap [18,23]

Table 8. Reported challenges of physical card walls.

Id	Category	Subcategory
PC1	Physical presence	Fixed location [18,19]
		Sharing is hard [15]
		Cards can get lost [17,28]
PC2	Lack of automation	Keep up to date is hard [13]

included studies often do not contain enough details about what kind of tool the teams used. There are often statements like a “scrum tool” or a “digital task board”, which do not even allow to make a reasonable guess about the used card wall type. Thus, for the analysis of the challenges and benefits, we generally distinguish between physical and digital card walls.

Tables 7, 8, 9 and 10 list the summarized benefits and challenges of physical and digital card walls. The sub-categories are not explained further, as they are granular enough to be understandable on their own (see the footnote 9).

One of the main benefits of a physical card wall is its physical nature by itself: standing in the room draws attention, makes it visible to everybody, and fosters transparency. Another important aspect mentioned is its ease of use and haptic behavior (Table 7).

The advantage mentioned above of the physical nature is at the same time reported as one of the biggest challenges. Its physical presence is restricted to the place where it is standing (Table 8). The lack of automation covers the aspect of missing traceability or missing support of digital intelligence.

Table 9. Reported benefits of digital card walls.

Id	Category	Subcategories	
DB1	Location independent	Available at multiple locations	[18, 19, 23]
DB2	Automation	Reporting	[15, 18]
		Tracability	[19]
		Can archive data	[18, 23]
		Integration with other tools	[23]
		Automatic adjustments of cards	[15]

One of the main reported benefits of digital, typically Web-based, card walls is its location independence together with its digital support like traceability, archiving, and integration possibilities (Table 9). Amongst the most often reported challenges is the complexity of the systems, which makes them very hard to use, and the missing overview (Table 10).

Table 10. Reported challenges of digital card walls.

Id	Category	Subcategory
DC1	ICT	Possible outage [15, 23]
		Shifts focus from interactions to tools [23]
DC2	Ease of use	Inefficient overview [15]
		Too many features [19]
		Training required [23]

4 Discussion

In this section, the findings of the research questions are discussed, and the paper's main question is is addressed.

4.1 RQ1: What Makes Card Walls Beneficial to Agile Teams?

The first question aims to answer why teams even use card walls. Analyzing the retrieved studies resulted in twelve categories that reflect the stated reasons. Looking at the categories, each category is either a benefit of the card wall itself or an effect of using the card wall. The categories often influence each other and whether a card wall has the stated benefits heavily depends on how it is implemented. So to precisely answer this research question, more details about the causes and effects (why they are beneficial vs. how they are beneficial) would be required. Most of the studies do not explain in very detail how the card wall was implemented and used; also, most studies were not conducted experimentally. Although it is possible to make some inferences, e.g., that the team's attention is an effect of the physical interactions, it is not sure if this is the only effect or if there are some other interactions. However, the analysis seems to show that the location of a card wall has an important effect. For example, if a card wall is placed in its own room and other team members cannot see an individual's interaction with the card wall, this will not raise any attention, and thus, it will not increase the communication frequency. On the other side, if the card wall is put in a shared office room, its permanent visibility and the visibility of the interactions of others seem to be very beneficial for agile teams.

4.2 RQ2: What Are the Challenges & Benefits of Physical/digital Card Walls?

The analysis shows that each approach has its strength and weaknesses. The pure physical nature of physical card walls brings many benefits, especially serving as an information radiator and a meeting point. On the other side, digital solutions add a lot of new functionality to card walls due to their digital nature, which supports the teams in many aspects. A major benefit concerns the support for distributed work, especially in today's distributed world. We found that a binary classification between physical and digital card walls is not appropriated and defined six different types of card walls. Furthermore, it must be considered that the software used for a digital card wall also has a considerable influence. A digital card wall does not inherently offer all the stated benefits, it also depends on the specific software and which features it offers. Nonetheless, digital card walls seem to suffer from their high complexity.

4.3 How Do Digital Card Walls Need to Be Implemented to Offer the Same Benefits as a Physical Solution?

This question must especially also be seen under the aspect of the new hybrid work style. We will have more and more distributed and dispersed teamwork, a mixture of multiple teams distributed worldwide, and team members working at home. Card walls, as the major collaboration tool for agile teams, must be able to support such teams as efficiently and effectively as possible.

The two research questions formulated to guide the SLR were intended to gather the necessary knowledge to answer the main question of this paper. RQ1 resulted in a set of categories from which we derived the following properties, which lead to the benefits of card walls.

- Physical artifact
- Placed in a central location

- Big and visible
- Always available and visible to everyone
- Physical interaction necessary for task update
- Visualization instrument

Two aspects cannot be influenced by the card wall itself but need to be considered by a team implementing a card wall.

- Reflect the real process/state of work of the team.
- What is not on the wall does not exist.

RQ2 revealed that the card wall type T6 "Software with interactive vertical screen" has the most significant potential to replicate the benefits of a physical card wall. A digital card wall of type T6 can potentially have all the properties to be considered. Therefore, the stated benefits and challenges need to be addressed when implementing the software for the digital card wall. However, it is essential always to remember that the desired effects may result from specific properties. That also means that some stated challenges of physical card walls and benefits of digital card walls should not be addressed because this has a potentially harmful influence on the experience, which is necessary to replicate the benefits of a physical card wall. For example, the benefits stated for digital card walls are: available at multiple locations, interaction with other tools, and automatic adjustment of cards. Those three benefits could lead to a situation where a visible physical interaction with the card wall is not necessary anymore. However, this visible physical interaction seems to be a card wall property that leads to benefits. There are also certain aspects that are either not solvable by the current technology, available or contradictive. Thus, there are always certain trade-offs. An example of a contradiction is traceability (only possible with a defined process) vs. no pre-defined process. An example of an inherent problem with the current state of technology is that the risk of an outage with a digital card wall is higher than that one of a physical one.

The potential of type T6 was already mentioned by Sharp et al. in their paper "The role of physical artefacts in agile software development: Two complementary perspectives" [28], but they also point out the fact that it is important to be able to replicate the social context, not only the purely functional nature of a card wall. This is in line with the findings of this SLR because it was shown that it is not sufficient just to solve the mentioned challenges to replicate the experience. Further research should clarify which properties are critical to replicate the social context around a digital card wall and how they can be implemented while maintaining the desired advantages of digitalization.

4.4 Limitations

This study has several limitations related to the methods and the corpus of studies. First, this review summarizes research results in a field with a rapidly changing technological landscape. The oldest studies included are from the year

2008. The benefits of a card wall may not change, but the tools available to build digital solutions are. Second, despite the systematic approach, the body of literature discovered may not be exhaustive. We may not include important literature with our methodology, and we did not consider gray literature. Third, there were no experimental or quasi-experimental studies on this topic. Hence all stated causality must be seen as a hypothesis that needs to be checked. Furthermore, as the studies mainly were qualitative case studies with small sample sizes, they are subjective and may not be transferable to other fields or teams.

5 Conclusion

We created twelve categories that show the benefits arising from card wall usage in general. Additionally, we summarized the benefits and challenges of physical and digital card walls. An important finding is that the desired benefits of card walls depend on specific properties. Hence, the benefits are only achievable by considering those properties. This is independent of the nature of the card wall, i.e., if it is a physical or a digital one. Those properties are essential to replicate the benefits of a physical card wall with a digital card wall. Another finding is that it is often unclear what is meant by talking about a "digital card wall". Hence, we suggested a typology of card walls and used it to analyze the challenges and benefits differentiated. Although it is not always possible to classify every aspect clearly as a challenge or benefit because it depends on the viewpoint, it is clear which effects are desirable to replicate with a digital card wall. Bringing the results together showed that the most promising type of digital card wall so far may be the "Software with interactive vertical screen" as it has the potential of replicating most of the effects by imitating many aspects of a physical card wall. However, some aspects are impossible to imitate with digital card walls, with the currently available technology. Furthermore, some reported benefits and challenges, if implemented/solved, contradict the properties, which will potentially lead to the desired effects/experience of using the card wall.

Further research may clarify the hypothesis that a digital card wall of type "Software with interactive vertical screen" can replace a physical wall and replicate their effect while bringing some of the stated desired benefits and resolving all the technically resolvable challenges.

References

1. Digital.ai: State of Agile Report. Technical report, Digital.ai (2021)
2. Azizyan, G., Magarian, M.K., Kajko-Matsson, M.: Survey of agile tool usage and needs. In: 2011 AGILE Conference, pp. 29–38. IEEE, August 2011
3. Bastarrica, M.C., Espinoza, G., Marín, J.: Implementing agile practices: the experience of TSol. In: International Symposium on Empirical Software Engineering and Measurement, ESEM 2018, New York, Association for Computing Machinery (2018)
4. Rola, P., Kuchta, D., Kopczyk, D.: Conceptual model of working space for agile (Scrum) project team. J. Syst. Softw. **118**(C), 49–63 (2016)

5. MacDonald, J.: Systematic approaches to a successful literature review. J. Can. Health Libr. Assoc./Journal de l'Association des bibliothèques de la santé du Canada **34**(1), 46 (2014)
6. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering - a systematic literature review. Inf. Softw. Technol. **51**(1), 7–15 (2009)
7. Weidt, F., Neiva, Da Silva, L.D.S.: Systematic Literature Review in Computer Science - A Practical Guide. Technical Report November, Federal University of Juiz de Fora (2016)
8. Cook, D.A., West, C.P.: Conducting systematic reviews in medical education: a stepwise approach. Med. Educ. **46**(10), 943–952 (2012)
9. Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G.: Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. PLoS Med. **6**(7), e1000097 (2009)
10. Gusenbauer, M., Haddaway, N.R.: Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed, and 26 other resources. Res. Synth. Methods **11**(2), 181–217 (2020)
11. Ahmad, M.O., Dennehy, D., Conboy, K., Oivo, M.: Kanban in software engineering: a systematic mapping study. J. Syst. Softw. **137**, 96–113 (2018)
12. Annosi, M.C., Martini, A., Brunetta, F., Marchegiani, L.: Learning in an agile setting: a multilevel research study on the evolution of organizational routines. J. Bus. Res. **110**, 554–566 (2020)
13. Anwar, A., Kamel, A.A., Ahmed, E.: Agile adoption case study, pains, challenges & benefits. In: Proceedings of the 2nd Africa and Middle East Conference on Software Engineering, AMECSE 2016, New York, pp. 60–65. Association for Computing Machinery (2016)
14. Bakke, A.L., Johansen, A.: Implementing of lean - challenges and lessons learned. Procedia Comput. Sci. **164**, 373–380 (2019)
15. Eckhart, M., Feiner, J.: How scrum tools may change your agile software development approach. In: Winkler, D., Biffl, S., Bergsmann, J. (eds.) SWQD 2016. LNBIP, vol. 238, pp. 17–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-27033-3_2
16. Hajratwala, N.: Task board evolution. In: 2012 Agile Conference, pp. 111–116. IEEE, August 8 2012
17. Hunt, J., Hume, T., Lozdan, D.: On rabbits, space and cards: moving towards an informative workspace. Proc. AGILE **2007**, 212–217 (2007)
18. Katsma, C., Amrit, C., van Hillegersberg, J., Sikkel, K.: Can agile software tools bring the benefits of a task board to globally distributed teams? In: Oshri, I., Kotlarsky, J., Willcocks, L.P. (eds.) Global Sourcing 2013. LNBIP, vol. 163, pp. 163–179. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40951-6_10
19. Kropp, M., Anslow, C., Mateescu, M., Burkhard, R., Vischi, D., Zahn, C.: Enhancing agile team collaboration through the use of large digital multi-touch cardwalls. In: Baumeister, H., Lichter, H., Riebisch, M. (eds.) XP 2017. LNBIP, vol. 283, pp. 119–134. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57633-6_8
20. Liechti, O., Pasquier, J., Reis, R.: Beyond dashboards: on the many facets of metrics and feedback in agile organizations. In: 2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), CHASE 2017, pp. 16–22. IEEE, May 2017

21. Mishra, D., Mishra, A., Ostrovska, S.: Impact of physical ambiance on communication, collaboration and coordination in agile software development: an empirical evaluation. Inf. Softw. Technol. **54**(10), 1067–1078 (2012)
22. Nakazawa, S., Tanaka, T.: Development and application of kanban tool visualizing the work in progress. In: 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), pp. 908–913, July 2016
23. Perry, T.: Drifting toward invisibility: the transition to the electronic task board. In: Agile 2008 Conference, pp. 496–500. IEEE, August 2008
24. Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still, J.: The impact of agile practices on communication in software development. Empirical Softw. Eng. **13**(3), 303–337 (2008)
25. Rubart, J.: A cooperative multitouch scrum task board for synchronous face-to-face collaboration. In: ITS 2014 - Proceedings of the 2014 ACM International Conference on Interactive Tabletops and Surfaces, pp. 387–392 (2014)
26. Rubart, J., Freykamp, F.: Supporting daily scrum meetings with change structure. In: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, HT 2009, pp. 57–61 (2009)
27. Sharp, H., Robinson, H.: Collaboration and co-ordination in mature eXtreme programming teams. Int. J. Hum. Comput. Stud. **66**(7), 506–518 (2008)
28. Sharp, H., Robinson, H., Petre, M.: The role of physical artefacts in agile software development: two complementary perspectives. Interact. Comput. **21**(1–2), 108–116 (2009)
29. Wiklund, K., Sundmark, D., Eldh, S., Lundqvist, K.: Impediments in agile software development: an empirical investigation. In: Heidrich, J., Oivo, M., Jedlitschka, A., Baldassarre, M.T. (eds.) PROFES 2013. LNCS, vol. 7983, pp. 35–49. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39259-7_6

Moscow Rules: A Quantitative Exposé

Eduardo Miranda(✉)

Carnegie Mellon University, Pittsburgh, PA 15213, USA
mirandae@andrew.cmu.edu

Abstract. This article analyzes the performance of the MoSCoW method to deliver all features in each of its categories: Must Have, Should Have and Could Have using Monte Carlo simulation. The analysis shows that under MoSCoW rules, a team ought to be able to deliver all Must Have features for underestimations of up to 100% with very high probability. The conclusions reached are important for developers as well as for project sponsors to know how much faith to put on any commitments made.

Keywords: Agile planning · Release planning · Requirements prioritization · Feature buffers · MosCoW method

1 Introduction

MoSCoW rules [1], also known as feature buffers [2], is a popular method to give predictability to projects with incremental deliveries. The method does this by establishing four categories of features: Must Have, Should Have, Could Have and Won't Have, from where the MoSCoW acronym is coined. Each of the first three categories is allocated a fraction of the development budget, typically 60, 20 and 20 percent, and features assigned to them according to the preferences[1] of the product owner until the allocated budgets are exhausted by subtracting from them, the development effort estimated for each feature assigned to the category. By not starting work in a lower preference category until all the work in the more preferred ones have been completed, the method effectively creates a buffer or management reserve of 40% for the Must Have features, and of 20% for those in the Should Have category. These buffers increase the confidence that all features in those categories will be delivered by the project completion date. As all the development budget is allocated by the method, there are no white spaces in the plan, which together with incentive contracts, makes the method palatable to sponsors and management.

Knowing how much confidence to place in the delivery of features in a given category is an important concern for developers and sponsors alike. For developers it helps in formulating plans consistent with the organization's risk appetite, making promises they can keep, and in calculating the price of incentives in contracts as well as the risk of

[1] These preferences might induce dependencies that need to be addressed by the team, either by incorporating lower preference features in the higher categories or by doing additional work to mock the missing capabilities.

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 19–34, 2022.
https://doi.org/10.1007/978-3-031-08169-9_2

incurring penalties, should these exist. For sponsors, it informs them the likelihood the features promised will be delivered, so they, in turn, can make realistic plans based on it. To this purpose, the article will explore:

1. The probabilities of delivering all the features in each of the categories: Must Have, Should Have and Could Have, under varying levels of under and overestimation of the features' development efforts
2. The impact of features' sizes, dominance, number of features, and correlation between development efforts in said probabilities
3. The effect of budget allocations other than the customary 60/20/20 on them.

To calculate the probabilities of delivery (PoDs) we need to make suitable assumptions about the distribution of the efforts required to develop each feature since the single point estimate used in the MoSCoW method are insufficient to characterize them.

In this article, those assumptions are derived from two scenarios: a low confidence estimates scenario used to establish worst case[2] PoDs and a typical estimates scenario used to calculate less conservative PoDs.

The potential efforts required and the corresponding PoDs, are calculated using Monte Carlo simulations [3, 4] to stochastically add the efforts consumed by each feature to be developed.

The rest of the paper is organized as follows: Sect. 2 provides an introduction to the MoSCoW method, Sect. 3 introduces the Monte Carlo simulation technique and describes the calculations used for the interested reader, Sect. 4 discusses the two scenarios used in the calculations, Sect. 5 analyzes the main factors affecting the method's performance, Sect. 6 discuss the method's effectiveness in each of the scenarios and Sect. 7 summarizes the results obtained.

2 The MoSCoW Method

The MoSCoW acronym was coined by D. Clegg and R. Baker [5], who in 1994 proposed the classification of requirements into Must Have, Should Have, Could Have and Won't Have. The classification was made on the basis of the requirements' own value and was unconstrained, i.e. all the requirements meeting the criteria for "Must Have" could be classified as such. In 2002, the SPID method [6] used a probabilistic backcasting approach to define the scope of three software increments roughly corresponding to the Must Have, Should Have and Could Have categories, but constraining the number of Must Have to those that could be completed within budget at a level of certainty chosen by the organization. In 2006, the DSDM Consortium, now the Agile Business Consortium, published the DSDM Public Version 4.2 [7] establishing the 60/20/20% recommendation although this, was probably used before by Consortium's members on their own practices. The current formulation of the MoSCoW prioritization rules is documented in the DSDM Agile Project Framework [1].

[2] Worst case, means that if some of the assumptions associated with the scenario were to change, the probability of delivering within budget would increase.

During the project planning phase, see Fig. 1.a, features are allocated to one of four sets: Must Have, Should Have, Could Have, and Won't Have on the basis of customer preferences and dependencies until the respective budgets are exhausted.

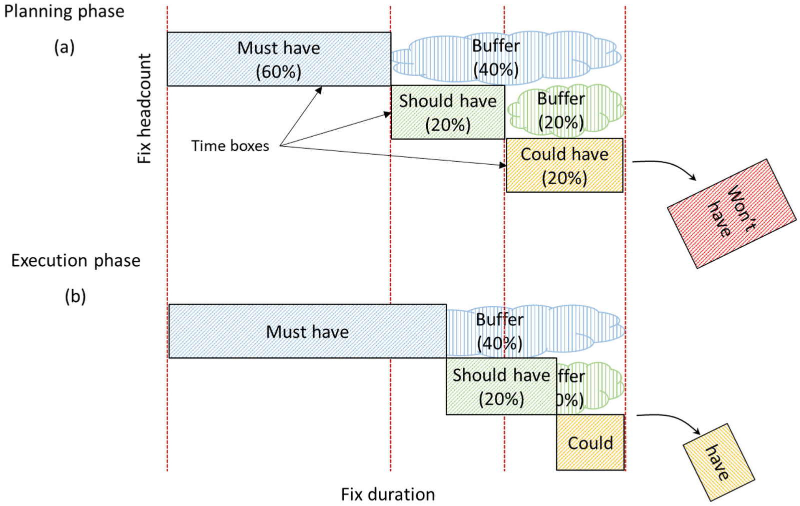


Fig. 1. MoSCoW rules at play: a) During planning, b) in execution

During execution, Fig. 1.b, features in the Must Have category are developed first, those in the Should Have second, and those in the Could Have, in third place. If at any time the work in any category requires more effort than planned, work on them will continue at the expense of those in the lower preference categories which will be pushed out of scope in the same amount as the extra effort required. The advantage for the project sponsor is that, whatever happens, he or she can rest assured of getting a working product with an agreed subset of the total functionality by the end of the project.

For the MoSCoW method to be accepted by the developer as well as by the sponsor of a project, the risk of partial deliveries must be shared between both of them through incentive contracts since approaches like firm fixed price or time and materials, that offloads most of the risk on only one of the parties could be either, prohibitive or unacceptable to the other. Contractually, the concept of agreed partial deliveries might adopt different forms. For example, the contract could establish a base price for the Must Have set, with increasingly higher bonuses or rewards for the Should Have and Could Have releases. Conversely the contract could propose a price for all deliverables and include penalties or discounts if the lower priority releases are not delivered. This way the incentives and disincentives will prevent the developer from charging a premium price to protect itself from not delivering all features while the sponsor, is assured the developer will do its best, in order to win the rewards.

3 The Monte Carlo Simulation

The Monte Carlo method is a random sampling technique used to calculate probability distributions for aggregated random variables from elementary distributions. The technique is best applied to problems not amenable to closed form solutions derived by algebraic methods.

The Monte Carlo method involves the generation of random samples from known or assumed elementary probability distributions, the aggregation or combination of the sample values according to the logic of the model been simulated and the recording of the calculated values for the purpose of conducting an ex-post statistical analysis.

The technique is widely used [3, 4] in probabilistic cost, schedule and risk assessments and numerous tools[3] exist to support the computations needed.

The results presented in the paper were calculated using @Risk 7.5. As these are the product of simulation runs, they might slightly differ from one run to another, or when using a different number of iterations or platforms.

The rest of the section explains the model used to generate the cumulative probability curves and calculate the PoD for each MoSCoW category: Must Have (MH), Should Have (SH) and Could Have (CH), with the purpose of allowing interested readers replicate the studies or develop their own simulations. Those not so inclined might skip it, with little or no loss in understanding the paper. The name of the parameters should make them self-explanatory however, conceptual definitions about its meaning and usage will be provided throughout the paper.

The probability of completing all features in a given category in, or under, an x amount of effort is defined as:

$$F_{MH}(x) = P(EffortRequired_{MH} \leq x)$$

$$F_{SH}(x) = P(EffortRequired_{MH} + EffortRequired_{SH} \leq x)$$

$$F_{CH}(x) = P(EffortRequired_{MH} + EffortRequired_{SH} + EffortRequired_{CH} \leq x)$$

The cumulative distribution functions: $F_{MH}(x)$, $F_{SH}(x)$ and $F_{CH}(x)$, are built by repeatedly sampling and aggregating the effort required by the features included in each category.

$$EffortRequired_{MH} = \sum\nolimits_{\forall i \in MH} EffortFeature_i$$

$$EffortRequired_{SH} = \sum\nolimits_{\forall j \in SH} EffortFeature_j$$

$$EffortRequired_{CH} = \sum\nolimits_{\forall k \in CH} EffortFeature_k$$

$$EffortFeature_i = \begin{cases} \text{Low confindence estimates: } RndUniform(Estimate_i, u \times Estimate_i, r) \\ \text{Typical estimates: } RndTriangular(0.8 \times Estimate_i, Estimate_i, u \times Estimate_i, r) \end{cases}$$

[3] @Risk by Palisade, Crystal Ball by Oracle, ModelRisk by Vose and Argo by Booz Allen among others.

similarly, for features j and k, and:

$$\text{u} = \begin{cases} 1.5 & & 50\% \\ 2.0 & \text{underestimation of up to} & 100\% \\ 3.0 & & 200\% \end{cases}$$

$$\text{r} = \begin{cases} 0 & & \text{independent estimates} \\ & \text{global correlation coefficient for} & \\ 0.6 & & \text{correlated estimates} \end{cases}$$

subject to the maximum allocation of effort for each category:

$$\sum\nolimits_{\forall i \in MH} Estimate_i \leq 0.6 \times DevelopmentBudget$$

$$\sum\nolimits_{\forall j \in SH} Estimate_i \leq 0.2 \times DevelopmentBudget$$

$$\sum\nolimits_{\forall i \in MH} Estimate_k \leq 0.2 \times DevelopmentBudget$$

The Probability of Delivery (PoD) of each category is defined as:

$$PoD_{MH} = F_{MH}(DevelopmentBudget)$$

$$PoD_{SH} = F_{SH}(DevelopmentBudget)$$

$$PoD_{CH} = F_{CH}(DevelopmentBudget)$$

All quantities are normalized for presentation purposes by dividing them by the *DevelopmentBudget*.

4 Low and Typical Confidence Scenarios

Figure 2 contrasts the two scenarios mentioned in the introduction. The low confidence scenario is characterized by the uniform distribution of the potential efforts required to realize each feature, with the lower limit of each distribution corresponding to the team's estimated effort for the feature and their upper to increments of 50, 100 and 200% above them, to express increasing levels of uncertainty. Since all values in the interval have equal probability, this scenario corresponds to a maximum uncertainty state [8]. This situation, however unrealistic it might seem, is useful to calculate a worst case for the PoD of each category. In the typical confidence scenario, the potential efforts are characterized by a right skewed triangular distributions, in which the team's estimates correspond to the most likely value of the distribution, meaning the realization of many features will take about what was estimated, some will take some more and a few could take less.

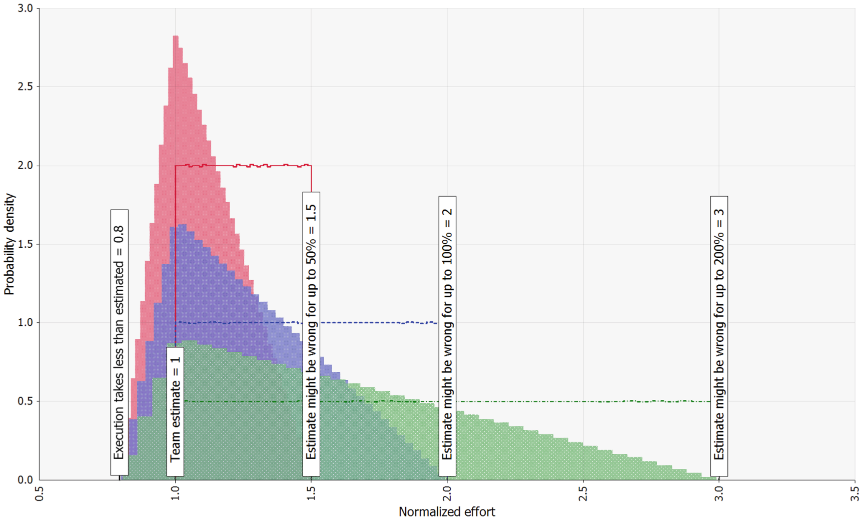


Fig. 2. Probability distributions for the effort required by each feature in the low (uniform distributions) and typical (triangular distributions) confidence scenarios

The right skewness of the typical estimate distributions is predicated on our tendency to estimate based on imagining success [9], behaviors like Parkinson's Law[4] and the Student Syndrome[5], which limit the potential for completing development with less effort usage than estimated, and the fact that the number of things that can go wrong is practically unlimited [10, 11]. Although many distributions fit this pattern, e.g. PERT, lognormal, etc., the triangular one was chosen for its simplicity and because its mass is not concentrated around the most likely point [12], thus yielding a more conservative estimate than the other distributions mentioned.

As before, the right extreme of the distribution takes values corresponding to 50, 100 and 200 percent underestimation levels. For the lower limit however, the 80 percent of the most likely value was chosen for the reasons explained above.

Considering this second scenario is important, because although having a worst case for the PoDs is valuable as they tell the lowest the probabilities could be, relying on them for decision making may lead to lost opportunities because of overcautious behaviors.

[4] Parkinson's Law, the 1955 assertion by British economist Cyril Northcote Parkinson, that "Work expands so as to fill the time available for its completion", regardless of what was strictly necessary.

[5] Student Syndrome, a term introduced by Eliyahu M. Goldratt in his 1997 novel Critical Chain to describe the planned procrastination of tasks by analogy with a student leaving working in an assignment until the last day before its due date.

5 Level of Underestimation, Correlation, Number of Features in a Category, Feature Dominance and Non-traditional Budget Allocations

Before calculating the PoDs for each MoSCoW category under the two scenarios, the impact of different factors on the PoD is explored with the purpose of developing an appreciation for how they affect the results shown, i.e. what makes the PoDs go up or down. Understanding this is important for those wanting to translate the conclusions drawn here to other contexts.

Although the analysis will be conducted only for the low confidence estimates for reasons of space, the same conclusions applies to the typical estimates scenario, with the curves slightly shifted to the left.

Figure 3 shows the impact of underestimation levels of up to 50, 100 and 200% of the features' individual estimates on the PoD of a Must Have category comprising 15 equal sized features, whose development efforts are independent from each other.

Independent, as used here, means the efforts required by any two features will not deviate from its estimates conjointly due to a common factor such as the maturity of the technology, the capability of the individual developing it or the consistent over optimism of an estimator. When this occurs, the efforts are correlated rather than independent. Having a common factor does not automatically mean the actual efforts are correlated. For example, a feature could take longer because it includes setting up a new technology, but once this is done, it doesn't mean other features using the same technology would take longer since the it is already deployed. On the other hand, the use of an immature open source library could affect the testing and debugging of all the features in which it is included.

The higher the number of correlated features and the stronger the correlation between them, the more individual features' efforts would tend to vary in the same direction, either requiring less or more of it, which would translate into higher variability at the total development effort level. This is shown by curves "r = 0.2", "r = 0.6" and "r = 0.8" in Fig. 4, becoming flatter as the correlation (r) increases.

Correlation brings good and bad news. If things go well, the good auspices will apply to many features, increasing the probability of completing all of them on budget. Conversely, if things do not go as well as envisioned, all affected features will require more effort, and the buffers would not provide enough slack to complete all of them.

Estimating the level of correlation between estimates is not an easy task, it requires assessing the influence one or more common factors could have on the items affected by them, a task harder than producing the effort estimates themselves. So while correlation cannot be ignored at risk of under or over estimating the safety provided by the method, the cost of estimating it, would be prohibitive for most projects. Based on simulation studies, Garvey et al. [13] recommend using a coefficient of correlation of 0.2 across all the estimated elements to solve the dilemma, while Kujawski et al. [14], propose to use a coefficient of 0.6 for elements belonging to the same subsystem, as these would tend to exhibit high commonality since in general, the technology used and the people building it would be the same, and 0.3 for elements on different subsystems, because of the lower commonality.

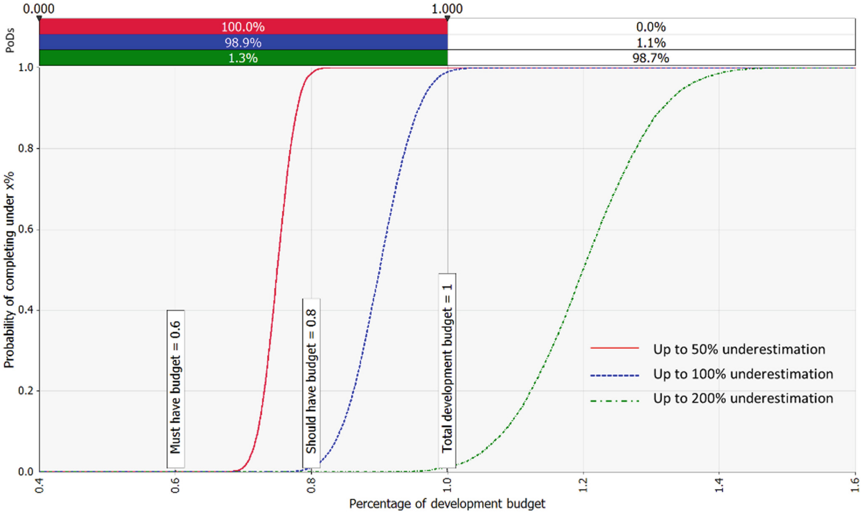


Fig. 3. Cumulative completion probabilities under increasing levels of underestimation. The simulation shows a PoD for the Must Have features of 100% for an underestimation level of up to 50%, of 98.9% at up to 100%, and of 1.3% for an underestimation in which each feature can require up to 200% of the estimated budget.

The PoDs are also affected by the number of features in the category as well as by the existence of dominant features, which are features whose realization requires a significative part of the budget allocated to the category. See Figs. 5 and 6.

As in the case of correlation, a small number of features and the presence of dominant features result in an increase in the variability of the estimates. Dominant features, contribute to this increase because it is very unlikely that deviations on their effort requirements could be counterbalanced by the independent deviations of the remaining features in the category. As for the increase of variability with a diminishing number of features, the reason is that with a fewer independent features, the probability of them going all in one direction, is higher than with many features.

The model in Fig. 7 challenges the premise of allocating 60% of the development budget to the Must Have category and explores alternative assignments of 50, 70 and 80% of the total budget. Reducing the budget allocation from 60 to 50% increases the protection the method affords at the expense of reducing the number of features a team can commit to. Increasing the budget allocation for the Must Have allows developers to promise more, but as will be shown, this is done at the expense of reducing the certainty of delivering it. For the 50% allocation level, there is a 100% chance of delivering the Must Have for underestimations of up to 100%, and of 68.2% for underestimations of up to 200%. At the 70% allocation level, the simulation shows that the PoD for the Must Have, when the possibility of underestimation is up to 50% still is 100%, but that it drops sharply to 34% when the underestimation level rises to up to 100%. For the 80% allocation level, the PoD for the Must Have falls to 49.7% for the up to 50% underestimation level and to 0 for the other two. The rest of the paper will then use the customary 60, 20 & 20% allocation scheme.

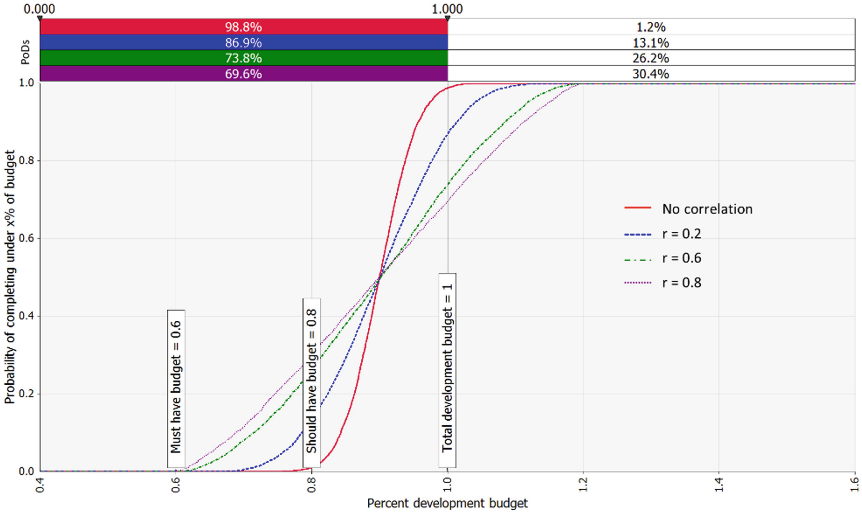


Fig. 4. Probability of completing all features in the Must Have category under a given percent of the budget when the underestimation level is up to 100% and the efforts are correlated (r > 0)

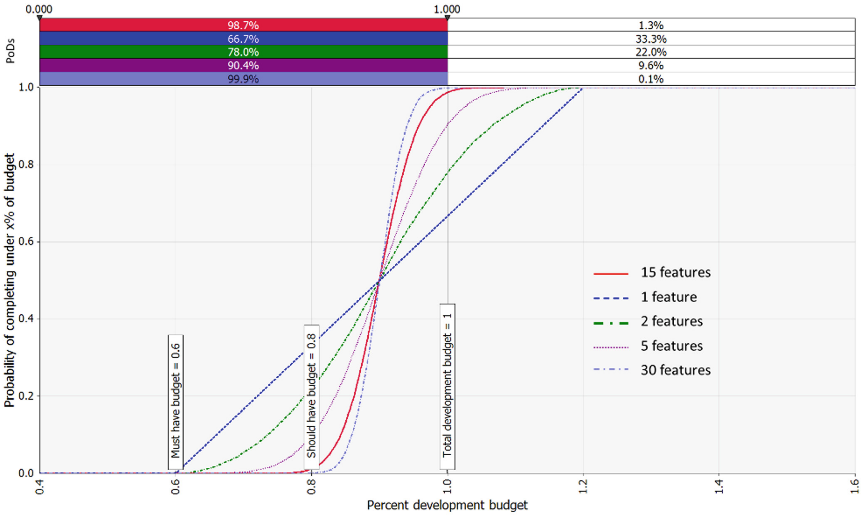


Fig. 5. Influence of the number of features on the PoD for a Must Have set containing the number of equally sized independent features indicated by the legend on the chart, with an underestimation level of up to 100%. The PoD offered by the method drops sharply when the set contains less than 5 features

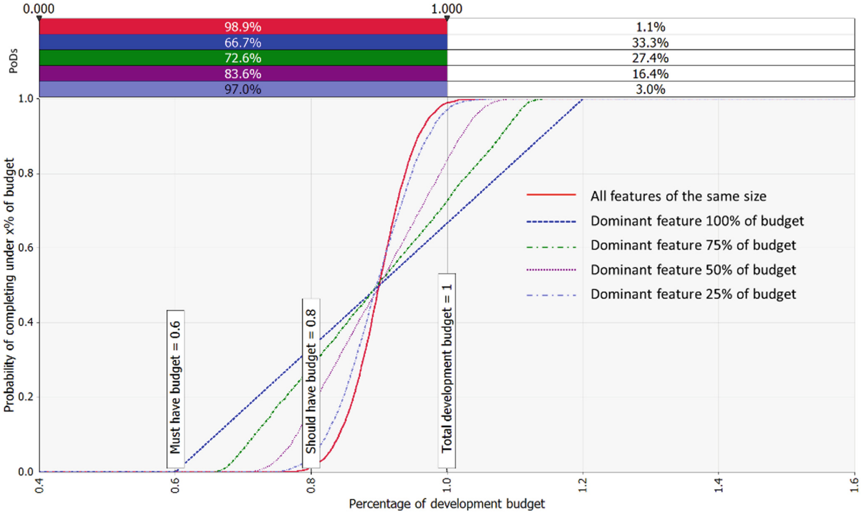


Fig. 6. Influence of a dominant feature on the PoD. Each set, with the exception of the dominant at 100%, contained 15 features, with the dominant feature assigned the bulk of the effort as per the legend in the chart with the remaining budget equally distributed among the other 14 features. The safety offered by the method drops sharply when a feature takes more than 25% of the budgeted effort for the category. Underestimation of up to 100% and independent efforts

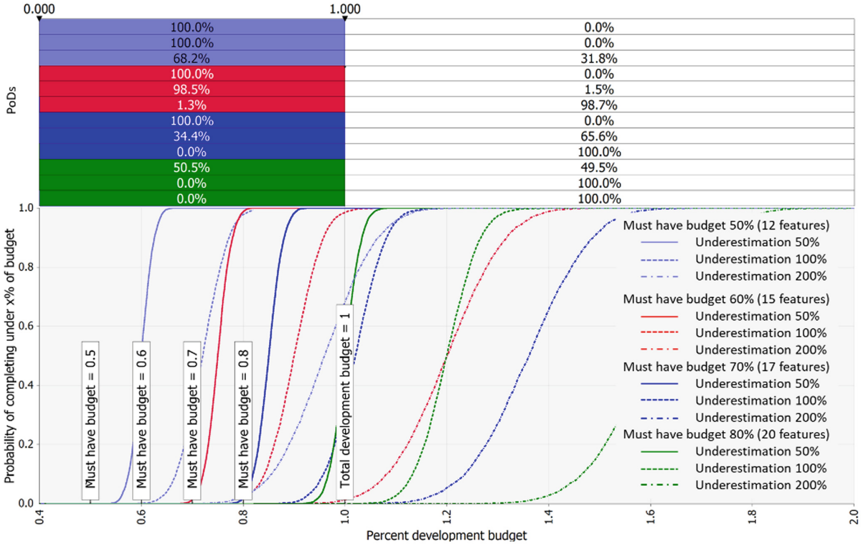


Fig. 7. Probability of delivering all Must Have features for Must Have budget allocations of 50, 60, 70 and 80% under different underestimation conditions. The respective number of Must Have features for each budget allocation were 12, 15, 17, and 20.

6 Probabilities of Delivery for Each MoSCoW Category

This section discusses the PoDs for each MoSCoW category: Must Have, Should Have and Could Have under the following conditions:

1. Low confidence estimation, independent efforts
2. Low confidence estimation, correlated efforts
3. Typical estimation, independent efforts
4. Typical estimation, correlated efforts

In all cases, the underestimations considered are of up to 50, 100 and 200% of the estimated effort, a 60/20/20 effort allocation scheme and a Must Have category comprising 15 equal sized features with Should and Could Have categories comprising 5 equal sized features each. These assumptions are consistent with the precedent analysis and with the small criteria in the INVEST [15] list of desirable properties for user stories. For the correlated efforts cases, the article follows Kujaswki's recommendation, of using an r = 0.6, as many of the attributes of an agile development project: dedicated small teams, exploratory work and refactoring, tend to affect all features equally.

6.1 Low Confidence, Independent Efforts

Figure 8 shows the PoDs for all MoSCoW categories for the low confidence, uncorrelated features, r = 0, model. At up to 50% underestimation, the probability of delivering all Must Have is 100%, as expected, and the probability of delivering all Should Have is 50.2%. At up to 100% underestimation, the probability of delivering all the Must Have still high, 98.9% but the probability of completing all the Should Have drops to 0. At up to 200% the probability of delivering all the Must Haves is pretty low, at 1.3%. In no case it was possible to complete the Could Have within budget.

6.2 Low Confidence, Correlated Efforts

As shown by Fig. 9, in this case the variability of the aggregated efforts increases, with the outermost points of the distribution becoming more extreme as all the efforts tend to move in unison in one or another direction. Comparing the PoDs for this case with those of the previous one, it seems paradoxical, that while the PoD for the Must Have at 100% underestimation level goes down from 98.9 to 74.0, the PoD for the same category at 200% underestimation level goes up from 1.3 to 26.9%! This is what was meant when it was said that correlation brought good and bad news.

To understand what is happening, it suffices to look at Fig. 10. Figure 10.a shows histograms of the Must Have aggregated independent efforts for uncertainty levels of 50, 100 and 200%. Because of the relatively lower upper limit and the tightness of the distribution spread afforded by the sum of independent efforts, the 100% uncertainty distribution fits almost entirely to the left of the total budget, scoring this way a high PoD. A similar argument could be made for the 200% uncertainty level, except that this time, the distribution is almost entirely to the right of the total budget, thus yielding a very low PoD. As could be seen in Fig. 10.b, when the efforts are correlated, the distributions

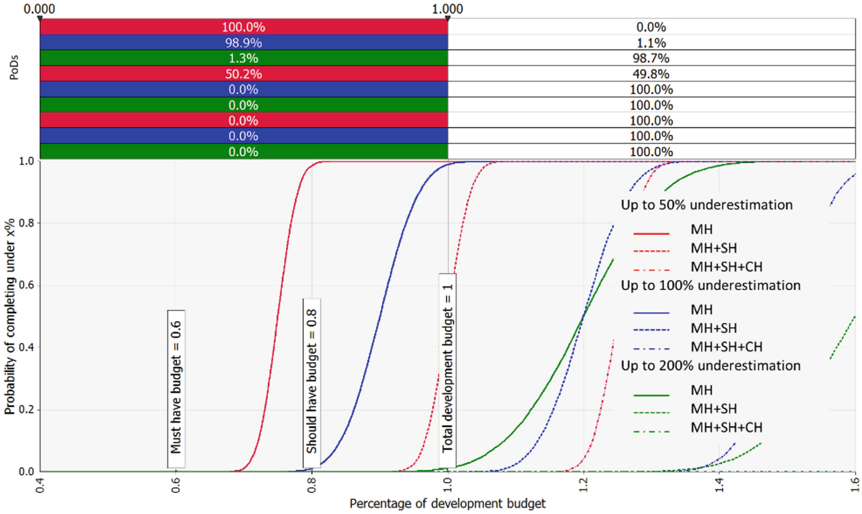


Fig. 8. Probability of delivering all features in a category in the case of low confidence estimates under different levels of underestimation when the efforts required by each feature are independent (r = 0)

spread more widely, making part of the 100% distribution fall to the right of the total budget line, reducing its PoD, and conversely, part of the 200% distribution might fall to the left of the line, thus increasing its PoD, which is what happened with this particular choice of parameter values.

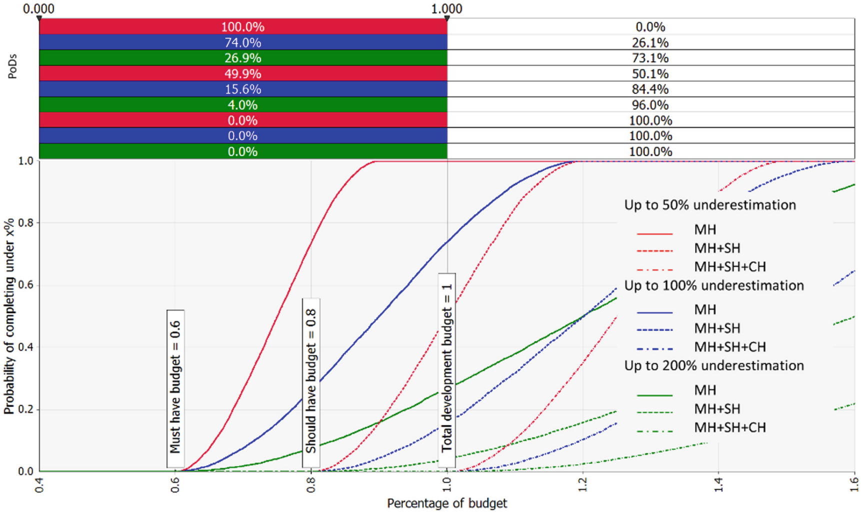


Fig. 9. Probability of delivering all features in a category in the case of low confidence estimates under different levels of underestimation when the efforts required by each feature are highly correlated (r = 0.6)

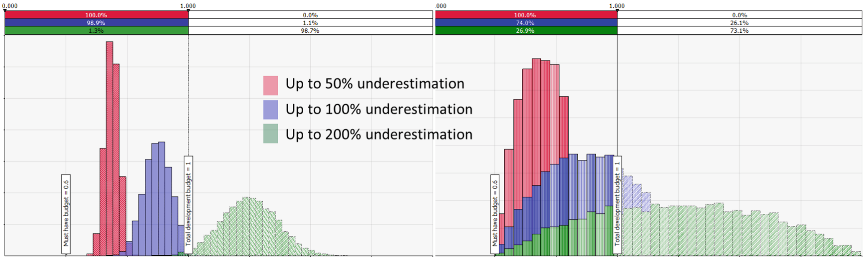


Fig. 10. Histograms for Must Have features' effort (a) left – independent efforts, (b) right – correlated efforts

6.3 Typical Estimates

Figures 11 and 12 show the typical estimates' PoDs for uncorrelated and correlated efforts respectively. As expected, all the PoDs in this scenario are higher than in the case of the low confidence estimates. In the case of independent efforts, at up to 50% underestimation, the PoDs for the Must Have and the Should Have are 100%. At up to 100% underestimation, the PoD for the Must Have is 100% with the PoD for Should Have dropping to 39.7%. At up to 200% the probability of delivering all the Must Haves still high, at 70.5%, but there is no chance of delivering the Should Have. In no case, any Could Have were completed. For the correlated efforts case, the respective probabilities at 50% underestimation are: 100% for the Must Have, 88.7% for the Should Have and 20.6% for the Could Have. At 100% underestimation: 96.4, 50.3 and 8.6% respectively and at 200% underestimation: 59.8, 20.5 and 3%.

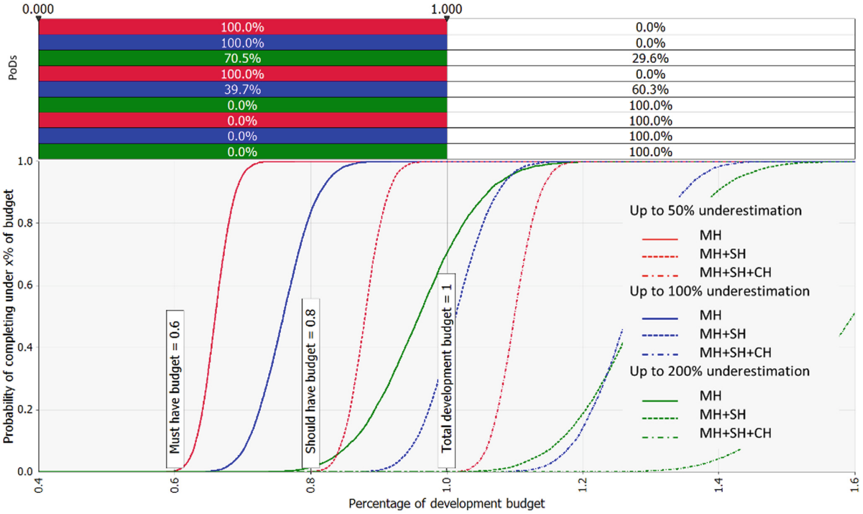


Fig. 11. Probability of delivering all features in a category in the case of typical estimates under different levels of underestimation when the efforts required by each feature are independent (r = 0)

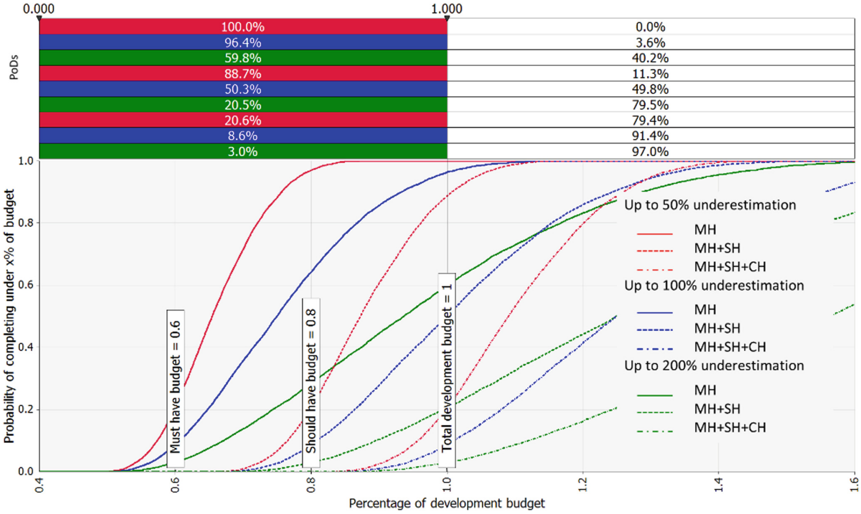


Fig. 12. Probability of delivering all features in a category in the case of typical estimates under different levels of underestimation when the efforts required by each feature are highly correlated (r = 0.6).

7 Summary

This article sought to quantitatively answer the following questions:

1. What are the probabilities of delivering all the features in each of the categories: Must Have, Should Have and Could Have, under varying levels of under and overestimation of the features' development efforts?
2. What is the influence of features' sizes, feature dominance, number of features, and correlation between development efforts in said probabilities?
3. What is the effect of budget allocations other than the customary 60/20/20 on them?

To answer question 1, it is necessary to look at Table 1 which summarizes the results for the low confidence and typical estimates scenarios, for the three levels of underestimation studied: 50, 100 and 200%.

Not surprisingly, the results indicate that the method consistently yields a high PoD for the Must Have features. What is noteworthy, is its resilience in face of up to 100% underestimation of individual features in the category. For the Should Have, the results are robust for up to 50% of underestimation and with regards to the Could Have, they should only be expected if destiny is smiling upon the project.

Question 2 is important for practitioners preparing release plans. For the method to offer these levels of certainty, the number of features included in each category should be at least 5 with none of them requiring more than 25% of the effort allocated to the category. If these conditions are not met, the safety offered by the method drops sharply. Correlation, as mentioned before, is a mixed blessing. Depending on which direction things go, it can bring the only possibility of completing all the features in the project.

Table 1. PoD summary for the three MoSCoW categories under different conditions

	Underestimation up to 50%				Underestimation up to 100%				Underestimation up to 200%			
	Independent efforts		Correlated efforts (r = 0.6)		Independent efforts		Correlated efforts (r = 0.6)		Independent efforts		Correlated efforts (r = 0.6)	
	Low conf	Typical	Low conf	Typical	Low conf	Typical	Low conf	Typical	Low conf	Typical	Low conf	Typical
Must have	100%	100%	100%	100%	98.9%	100%	74.0%	96.4%	1.3%	70.5%	26.9%	59.8%
Should have	50.2%	100%	49.9%	88.7%	0	39.7%	15.6%	50.3%	0	0	4.0%	20.5%
Could have	0	0	0	20.5%	0	0	0	8.6%	0	0	0	3%

Notice that in Table 1, all the Could Have can only be completed when the efforts are highly correlated since all of them must be low. Under the independence assumption, when some could be low and others high, there is no chance of completing them on or under budget.

With regards to question 3, the 60, 20, 20% allocation seems to be the "Goldilocks" solution, balancing predictability with level of ambition. As shown by Fig. 7, changing the allocation from 60 to 70%, has a dramatic impact on the safety margin which, at the up to 100% underestimation level, drops from 98.5 to 34%.

Finally, it is worth making clear, that the analysis refers to variations in execution times of planned work and not changes in project scope, which should be addressed differently.

The author gratefully acknowledges the helpful comments of Hakan Erdogmus. Diego Fontdevila and Alejandro Bianchi on earlier versions of this paper.

References

1. Agile Business Consortium: Chapter 10 MoSCoW Prioritization, January 2014. https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation. Accessed 10 Oct 2021
2. Cohn, M.: Agile Estimationg and Planning. Prentice Hall, Upper Saddle River (2006)
3. Garvey, P.: Elements of Cost Risk Analysis. MITRE Corporation (2014)
4. Magennis, T.: Managing software development risk using modeling and Monte Carlo simulation. In: Lean Software and Systems Consortium Conference, Boston (2012)
5. Clegg, D., Baker, R.: CASE Method Fast-track: A RAD Approach. Addison-Wesley, Reading (1994)
6. Miranda, E.: Planning and Executing Time Bound Projects. IEEE Comput. (2002)
7. DSDM Consortium: Not longer accessible. http://www.dsdm.org/version4/2/public/default.asp
8. Shu-Cherng, F., Tsao, H.: Entropy optimization: shannon measure of entropy and its properties. In: Encyclopedia of Optimization, Springer, Boston (2001). https://doi.org/10.1007/0-306-48332-7_119
9. Newby Clark, I., Buehler, R., Koehler, D., Griffin, D.: People focus on optimistic scenarios and disregard pessimistic scenarios while predicting task completion times. J. Exp. Psychol. Appl. (2000)

10. Halkjelsvik, T., Jørgensen, M.: Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life, Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74953-2
11. Kitchenham, B., Linkman, S.: Estimates, uncertainty and risk. IEEE Softw. (1997)
12. Hulett, D.: Practical Schedule Risk Analysis, Gower (2009)
13. Garvey, P., Book, S.C.R.: Probability Methods for Cost Uncertainty Analysis, 2nd. CRC Press (2016)
14. Kujawski, E., Edwards, W.R., Alvaro, M.L.: Incorporating psychological influences in probabilistic cost analysis. Syst. Eng. **7**(3) (2004)
15. Wake, B.: INVEST in Good Stories, and SMART Tasks. XP123, 17 8 (2003). https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/. Accessed 19 Nov 2021

Are Your Online Agile Retrospectives Psychologically Safe? the Usage of Online Tools

Dron Khanna(✉) and Xiaofeng Wang

Free University of Bozen-Bolzano, Bolzano 39100, Italy
{dron.khanna,xiaofeng.wang}@unibz.it

Abstract. One essential prerequisite for successful agile retrospective sessions is to accomplish a psychologically safe environment. Creating a psychologically safe environment for the co-located team is challenging. Further, it becomes more demanding with online agile retrospective teams. Literature sheds little light on creating a psychologically safe online environment for conducting agile retrospectives. Our study aims at addressing this knowledge gap and asks the research question: how does the usage of online tools influence psychological safety in online agile retrospectives? A single case study was conducted with a major software company's Research and Development team. We analysed a recorded online retrospective session of the team to identify patterns of the usage of online tools associated with the online meeting platform they used and how that usage influenced the psychological safety level of the team. Our findings show that retrospective participants are psychologically safe if they share opinions, make mistakes, raise a problem, ask questions, and show consent using online tools. Our study contributes online tools that influence psychological safety factors, corresponding levels and behaviours.

Keywords: Online retrospective · Agile retrospective · Psychological safety · Online tools · Online meetings

1 Introduction

Practising agile retrospectives helps the participants to reflect & learn from the experience [20], be more collaborative and contribute to work [18]. Also, it outlines the problems in workflow, makes transparent the work process [20] and overcomes efficiency loss challenges (rise in customer requirements, product complexity and prevention from competitive pressure) [6]. The new normality has pushed agile retrospectives in an online environment [4].

A psychologically safe environment is one key prerequisite for successful agile retrospective sessions, as indicated in the Prime Directive[1], widely embraced by agile software development teams. Safety is a state of mind that lets human

[1] https://retrospectivewiki.org/index.php?title=The_Prime_Directive.

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 35–51, 2022.
https://doi.org/10.1007/978-3-031-08169-9_3

beings sense their protectiveness from danger [31]. Psychological safety is a common belief where individuals or participants feel mentally and emotionally safe and willing to share their opinions with others in a group [8,16]. It ensures participants feel included, can be themselves and enhance their work engagement within a team [16]. Psychologically safe team participants are inclined to be efficient and act responsively in the meetings. They are actively collaborating, contributing and helping their peers to solve problems [8,9,16].

While it is challenging to create a psychologically safe environment when software development teams are co-located, it becomes more demanding when agile retrospectives are conducted fully online. In online agile retrospectives (OARs), team members use tools provided by the online meeting platform to communicate. The online tools include video or teleconferencing, breakout rooms, chat and digital boards [10,29]. Video or teleconferencing tools offer good support to run the online session [17]. The usage of these online tools during OARs can play a vital role in the psychological safety level of a team.

For example, a participant could use an audio or chat window to express opinions [9] on other participant's opinions about What went well? What did not go well? and What could be done? to obtain improved sprints [18]. Doing so reveals that the participant is psychologically safe, feels included, and contributes to the team. Then a vote or emoji as an online tool allows a team member [12] to express decisions and emotions about the sprint. In a parallel and efficient way, while the participant is speaking during the OAR, a team member could use (raise hand ✋) [9] to ask a question or raise a problem [1]. It provides the team to reflect, learn and express faster about the sprint [19] and ensures that participants are psychologically safe [16]. Whereas often, the unsafe participants are hesitant to express themselves. However, they can be anonymous and express their emotions with votes or emojis.

Few studies mentioned psychological safety explored during online meetings. A software engineering study mentions psychological safety in teams and the norm clarity. The paper outline importance of adopting various norms that could contribute to a safe psychological ambience [23]. Also, a recent study describes psychological safety impacts on agile software development team performance. It might be either directly or indirectly through team reflexivity [3]. Still, there is a lack of studies investigating psychological safety in OAR.

Hence, the research questions formulated for this study is: ***RQ: How does the usage of online tools influence psychological safety in online agile retrospectives?***

The paper is structured as follows. Section two describes the online agile retrospective, psychological safety levels, behaviours and factors. Also, the online tools influence the essence of psychological safety in online meetings. Then in section three, we describe software company information, the data collection and analysis procedure. Section four findings outline the five stages of OAR. In each stage, we found the usage of online tools that influences psychological safety factors, corresponding levels, and behaviours. Section five discusses the specificity of online agile retrospectives, including the meeting content conducted with online tools. Section six concludes the study with the inclusiveness of online meetings and their linkage to psychological safety as an interesting future study.

2 Background and Related Work

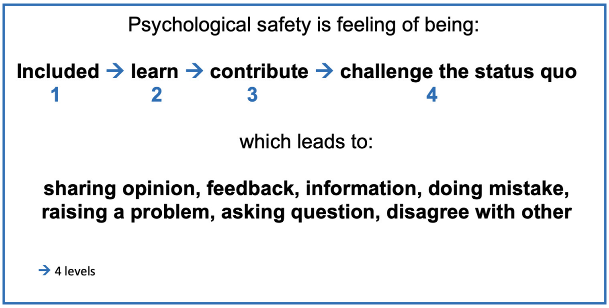
2.1 Online Agile Retrospective (OAR)

The idea of conducting a retrospective with participants is to collect information and notify those areas that need closer attention [18,19]. Hence, that improves the team's productivity and performance [20]. OAR help participants acquire knowledge gaps existing in the sprint before the next learning sprint begins [4] and insights about the learning activities [14]. During a retrospective session, the objectives or tasks are re-evaluated and then outlined in front of participants before the next iteration [18,26], which leads to an improved product or service development life cycle [19]. Online retrospective participants use video/teleconference channels to contribute to the reflection of the iteration with other participants. The participants also share the time, location and duration of the retrospective [26]. A team can learn from the experience and share learning with other participants [20]. In retrospect, asking questions and raising a problem is common to learn from other participants [18]. One participant to facilitate the meeting must be present during the retrospective. They help to moderate the communication between the satellite participants [26]. A crucial thing to note during the OAR is to schedule it in advance. OAR is planned previously in online settings, as participants could vary with the working hours and time zones [4]. Online retrospectives cannot be very spontaneous, as different time zones could vary in hours, and the setup of video/teleconference is mandatory. The online environment could require time to set up the internet and other online tools [26]. In OAR, participants must contribute to work by sharing an opinion or asking a question about the previous iteration cycle [11]. In doing so, the participant should feel safe presenting the work [31] and help peers learn better about the iteration [19].

2.2 Psychological Safety

Psychological safety is a shared or common belief where individuals are willing to share opinions, feedback, information, mistakes, raise a problem, ask a question, or even disagree with participants without fear [5,8,9]. Figure 1 provides psychological safety levels and behaviours. It is an unsaid belief within participants about feeling safe to be (1) *included*, (2) *learn*, (3) *contribute*, (4) *challenge the status quo* [5] while working with others.

1. **Included**: This initial psychological safety level describes the acceptance of the participant to the workgroup, team or environment gathered by various humans who are willing to be together. Once a participant is safe to include, he/she gains acceptance or admittance to the group and attention from others. Feeling included is the opposite of being ignored or rejected by others who are willing to be together in the same environment [5,8,16].



Psychological safety is feeling of being:

Included → learn → contribute → challenge the status quo
1 2 3 4

which leads to:

sharing opinion, feedback, information, doing mistake, raising a problem, asking question, disagree with other

→ 4 levels

Fig. 1. Psychological safety [5,8]

2. **Learn**: The second level is the feeling of being safe to engage and learn with others. Participants need to be heard and engaged by asking about some information, experimenting or making mistakes to discover something. This learning passage helps participants harness confidence, independence, and resilience [5,8].
3. **Contribute**: Compared to the previous level, participants are more active with others and observed as qualified contributors. They demonstrate competence in the environment and usually are free to contribute. Participants expected contribution is visible at this level [5,8].
4. **Challenge the status quo**: At the final level, a participant is confident enough to challenge the ongoing situation in the environment. It requires courage and proper time to speak the truth when something needs to change or alter the current situation. Participants at this level are confirmed about the facts and could rank themselves in a creative process of contribution [5,8].

2.3 Psychological Safety Factors

Four factors influence psychological safety: *trust*, *mutual respect*, *constructive response* and *confidence* [7,8].

- **Trust**: It is a situation when participants have faith in peers. It is the mental attitude of participants that provides a comfort zone for others [7,8]. A study by Duehr et al. [6] claims that *trust* among participants provides clarity and understanding of work objectives during an OAR. Also, *trust* leads to increasing the transparency and contribution of information.
- **Mutual respect**: This factor leads to caring for each other and encourages a psychologically safe environment. It might be that there are issues inside a team [3]. However, mutual respect provides being tolerant of dealing with each other's responses and behaviours [7,8].
- **Constructive response**: A response provided on the mistakes or errors that help a participant improve without feeling discouraged. Errors are typical but should not lead to rejection and discouragement in a team [7,8].

- **Confidence**: It is a clear state of mind believing someone or something is correct, even if the evidence is entirely lacking. It is the ability to assure that something is correct [7,8].

With the factors mentioned above, participants are willing to be open about the actions they intend to consider and have a feeling of invulnerability in the group. They can share their beliefs without being scared. As a result, information and knowledge are transparent and circulate in a group [5,7,8].

2.4 Online Tools

Online meetings are comfortable if participants know or have met each other in person previously. There is a feeling of being connected to other peers, as faces and characters exist behind the names displayed during online meetings. The trend of meeting with online participants is increasing after the pandemic [1,10], which increases the use of online tools [22]. Below is a list of tools embedded in online meeting platforms that may influence psychological safety in an online environment [9].

- **Video**: The video is one of the known tools used during an online conference [1,10]. It replaces the physical essence of face-to-face conversation and creates an environment that leads to enhanced interaction with the participants [9,28,29]. Some participants blur the background or adapt to a banner or theme behind the face because participants do not want to share the room or background [22]. Video dramatically relies on the internet bandwidth. Breaking down or slowing down the internet, meaning participants can see the held faces [29]. A speaker should be encouraged to turn on the video and, if the rest participants prefer, should be allowed to switch it off [9]. Video without audio could be challenging to decipher [25] in the case of silence during the meetings. Participants in the meeting could hold silent for a few seconds [15], and then someone brave enough to break it and present their thoughts. Otherwise, a facilitator should be present at the moderate session [9]. The facilitator could turn on/off the video and the audio to make it comfortable for other participants.
- **Audio-only**: Audio helps make the session interactive during the meetings [10]. Crucial is paying acute attention to the speaker to avoid misinterpretation of what the speaker wants to express [28]. For example, a raised question could clear doubts if there is a misunderstanding. Participant's must not misjudge the silence when there is no audio. A participant's silence could mean either Yes or No. To overcome, a checkmark is helpful during the meetings [9,12].
- **Checkmark (Yes/No)**: Participants who prefer to be silent could use the checkmark to present their opinion. Usually, the tick (✓) sign represents the *Yes*, and the cross (X) sign represents *No*. These checkmarks act as an agreement or disagreement with the presenter's voice. However, checkmark has a problem; fully agreeing or disagreeing with speakers' information. Checkmark

is not helpful to present partial agree or disagree opinions. To overcome this, polls or chat should be used [9].

- **Polls/Votes**: This online tool improves the shared feedback from participants [29]. However, some participants could be afraid of displaying their names in the poll. In order to make these psychologically safe, anonymous polls/votes provide a fair outcome and valid opinion [9,12].
- **Chat**: Chat is an excellent tool for interaction (such as the risk of asking a question, expressing an opinion or raising a problem). However, messages could also distract from the speaker's conversation if the text is too long. Messages could become spam if they are redundant and not precise. Hence, participants should be aware of the length and quality. Chat should be applied if it is needed to share the information [9].
- **Breakout rooms**: Breakout rooms inside the online meeting provoke natural and safe conversation [29]. These rooms are safe spaces where it is possible to take the risk of raising a problem or making mistakes with a small group and seek feedback. A structured breakout room [29] involves participants interacting about a specific task or topic [9]. Often peers are comfortable and feel included in breakout rooms. Also, in breakout rooms, participants might know peers and could test, validate, and re-build the concepts [9].
- **Emoji (e.g., raise hand ✋)**: Emoji functions to interact (raising a problem, opinion or seeking information) with peers. However, participants should think wisely before using them. At some point, an emoji could also create an insecure environment at some speaker's presentation [9,12].
- **Digital board**: Digital boards are online tools that let participants comment, chat, reflect and share opinions on the task [24]. For example, Parabol, Retrium [27], Atlassian, or Mural digital boards conduct agile retrospectives with remote participants supporting psychological safety.

As far as the authors are aware, no study focused on how the usage of online tools can influence the psychological safety of participants of online agile retrospectives. Our study aspires to address this knowledge gap.

3 The Research Approach

A case study is an appropriate methodology to answer "how" research questions [30]. To answer our RQ, we conducted a case study of a research and development team of a sub-branch of a major multi-national software company (company name omitted due to anonymity agreement). The software company offers a solution for cybersecurity, business intelligence, enterprise resource planning, customer relationship management, and system and service management. This sub-branch also helps other companies in the digitalization and innovation processes.

Table 1. The recorded online sessions

Session	Description	Duration	Participants involved
A	Online group interview	45 min	Product manager & 2 team leaders
B	One recorded OAR session	75 min	11 participants & 1 team leader
C	Online interview	25 min	Product manager

3.1 Data Collection

Due to the COVID-19 pandemic situation, we collected the data in an online settings. Table 1 presents the data collected in the case study and the data collection methods used.

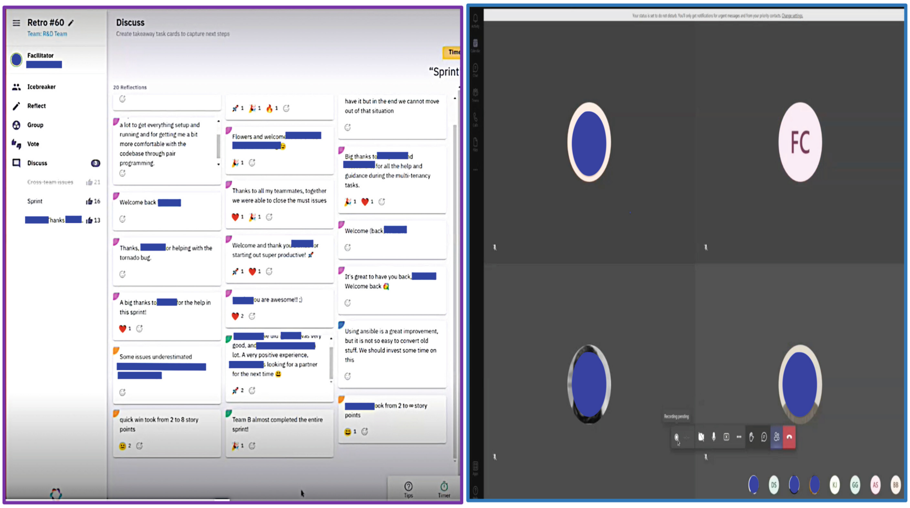
- **Session (A)**: It is an unstructured group interview session conducted to collect contextual information about the software company, the team studied and how they are doing agile software development. This recorded session displays various questions and answers with the product manager and two leaders.
- **Session (B)**: It is a complete recording of an OAR session of the team at the end of one Sprint. The recording was done by the team and handed over to the researchers. The researchers were not present at this OAR session.
- **Session (C)**: It is an unstructured interview session with the product manager who directly manages the studied team. This session helped clarify the data gathered in the previous two sessions.

3.2 Data Analysis

We found various instances of interest from OAR showing the psychological safety of the bracketing technique as a research approach. It is a technique that has been applied increasingly in qualitative research studies [13]. It is the art of picking various episodes of interest from an event and probably, clustering later those instances into another event [13, 21]. It is helpful where key sections of importance exist in the entire event. They could be diverse and assorted but are topics of interest. The researcher should describe precise breakpoints for the different instances of the event. The instances found were time-stamped and coded into transcripts using NVivo12 software, a qualitative data analysis software.

- **Session (A)**: This session revealed insights about the work routines of the studied team and agile practices involved in the online settings. The company performs various agile practices; sprint planning, standup, retrospective and low-level design meetings. The research and development participants are involved in OAR. Often, the service support members also take part in the OAR. The retrospective lasts between 60 to 75 min. The software company uses the digital board Parabol and the Microsoft Teams for conducting OAR, as shown in Fig. 2.

- **Session (B)**: We found various thematic codes corresponding to OAR. It is a method of systematically identifying themes or patterns across qualitative data [2]. The thematic codes *trust, mutual respect, constructive response, confidence, opinion, information, facilitator, included and contributed* emerged under the psychological safety and *icebreaker, reflect, group, vote and discuss* under the stages of OAR. Under online tools, we found many thematic codes such as *video, screen-share, audio, text and emoji.*
- **Session (C)**: While analysing the OAR, various questions occurred about the participants, process and OAR. All notes questions, later in the interview, were asked *"Who was the person leading the retrospective meeting? If he is not a scrum master, then? Who all are involved in the software development?"* to the project manager.



[redacted]

a lot to get everything setup and running and for getting me a bit more comfortable with the codebase through pair programming.

Welcome back [redacted]

Thanks, [redacted] or helping with the tornado bug.

A big thanks to [redacted] for the help in this sprint!

Some issues underestimated [redacted]

quick win took from 2 to 8 story points

Flowers and welcome [redacted]

Thanks to all my teammates, together we were able to close the must issues

Welcome and thank you [redacted] or starting out super productive!

[redacted] ou are awesome!! ;)

[redacted] we did [redacted] was very good, and [redacted] lot. A very positive experience. [redacted] s looking for a partner for the next time

Team B almost completed the entire sprint!

have it but in the end we cannot move out of that situation

Big thanks to [redacted] nd [redacted] for all the help and guidance during the multi-tenancy tasks.

Welcome (back [redacted]

It's great to have you back, [redacted] Welcome back

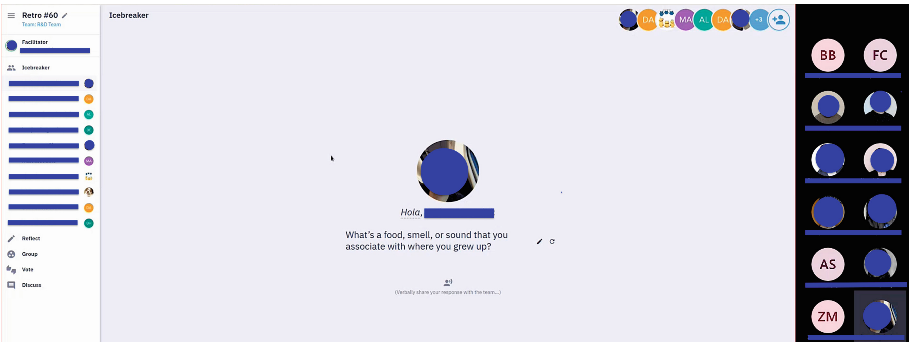
Using ansible is a great improvement, but it is not so easy to convert old stuff. We should invest some time on this

[redacted] ook from 2 to ∞ story points

Fig. 2. Online tools (*Left-side*: The shared screen of Parabol via Microsoft Teams, *Right-side*: Microsoft Teams)

4 Psychological Safety in OAR

Parabol is an agile meeting tool that provides a digital board helping remote participants to connect, reflect, and monitor the work progress. The board consists of five stages, in sequential order: Icebreaker, Reflect, Group, Vote, and Discuss, shown on the left side of the Parabol (see Fig. 2). The participants start with the Icebreaker stage and conclude the retrospective with the Discuss stage.



Hola, [redacted]

What's a food, smell, or sound that you associate with where you grew up?

(Verbally share your response with the team...)

Fig. 3. Icebreaker stage (The shared screen of Parabol via Microsoft Teams)

4.1 Icebreaker Stage

It is a warm-up stage. In this stage, all participants answer one from the 237 icebreaker questions provided by the digital board. The facilitator shared the screen using Parabol and Microsoft Teams (Fig. 3), where the question *"What is a food, smell, or sound that you associate with where you grew up"* was displayed. Each participant got a few minutes to answer this question, one by one. During this stage, the participants had the video off, their **avatars or photos with their names** were visible on the **shared screen**, and they used **audio** for verbal responses. We identified the following instances of interest in this stage.

Concerning psychological safety, first-level **included**. All participants had the feeling of being accepted to OAR. One participant verbally *raised a problem- "sorry, can anyone please share the Parabol link with me? My link is not working"*. The facilitator then shared an *opinion- "yes"* and used **text** to re-send the link. This behaviour gives the participant a safe feeling of being **included** at the OAR. Also, peers show *mutual respect* by waiting till everyone is on-board. After a few seconds, the same participant realises that a technical problem exists. The participant boldly explained the *information- "I have reset the password and laptop, but still have some technical issues"*. The online tool was not working. However, it was essential to respect the OAR schedule and other participants. Hence, the Facilitator gives a *constructive response* and shares the *opinion- "I think we can start the meeting, and once you join"* Parabol, *"you can be in the Icebreaker question list"*.

In some instances, psychological safety could be challenging. A participant should not **ignore** and must reply to the facilitator's question if asked. Regarding psychological safety level **included**. A participant during the OAR did not answer the question. The facilitator called a participant's name during his turn *"we cannot hear you if you are talking"*. The participant's *photo with the name* was visible on the *shared screen*, but no replies. It breaks the *trust* and *mutual respect* when peers want to contribute during OAR. To overcome if the participant cannot answer, should share *information*, and *raise the issue* by *chat or breakout room* to convey the problem. When there was **silence** for a short while, and participant did not responded. Another participant shared *information- "he is busy, he is in another meeting, but not attending this meeting"* using **audio**.

Concerning psychological safety, third-level **contribute** using available online tools. The facilitator takes a significant responsibility to run the OAR. Also, ensure that every participant is online connected to Parabol and **contributes**- *"Please let me know when you finish. Thank you"*.

A participant involved **self-referential humour** that created a joyful atmosphere during the OAR. Sharing a joke about oneself could make participants laugh. Concerning psychological safety, level **contribute**. A participant verbally shared *information- "I hope it is not a cliche, I still enjoy it"*. The facilitator shared the *opinion- "It is a bit of a cliche, I would say,"* and in return, the participant laughed at the *opinion- "Haha"*, and other participants also laughed *"Haha"*. Later, other participants shared similar *information- "The smell of fertilisers from the cow and the sound of (cows and cock) come at 4:00 am when you still have one more hour to sleep, but you cannot sleep. Haha"*. One participant used another online tool, which was a funny **image or picture**, to share an *opinion* in the *chat* window.

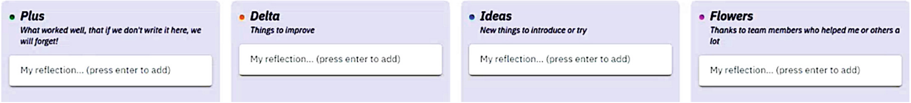
Concerning psychological safety level **contribute**. There was a voice break instance when a participant spoke and shared the *information* about the ice-breaker question. However, the other participants and facilitator could not able to hear. Hence, the facilitator *asked*, *"What?What?.."*. To reply the participant shared *information* via *text* in the *chat- "I am facing a sudden power cut and my laptop battery has only 30 min left"* and *sorry, restarting.* Peers showed *trust mutual respect* and gave a *constructive response* via **text**. Some used a **check-mark and emoji (Thumbs-up or like: 👍)** to give a *constructive response* to the participant's message.

4.2 Reflect Stage

Compared to the previous stage, this stage was challenging to analyse. Each participant must carry an individual reflection about the previous iteration cycle without interaction. Participants used the digital board and wrote down their thoughts on **small (post-it notes) cards**. Hence, silence existed during this stage. OAR was ongoing on Microsoft Teams, with **avatar/photo with name** visible on the facilitator **shared screen**. Concerning psychological safety level **included**. The facilitator shared the *opinion- "when you finish writing, please click on the button so that we can move on to mark the end of this stage and start the next one"*. It showed a sign of psychological safety where all participants were **included** and shared reflection.

4.3 Group Stage

The group stage is similar to the previous stage. Less **audio** interaction. The facilitator **shared a screen** with the **digital board**, which displayed all the inputs. The digital board displayed **text** inputs and the facilitator clustered



● Plus
What worked well, that if we don't write it here, we will forget!

● Delta
Things to improve

● Ideas
New things to introduce or try

● Flowers
Thanks to team members who helped me or others a lot

My reflection... (press enter to add)
My reflection... (press enter to add)
My reflection... (press enter to add)
My reflection... (press enter to add)

Fig. 4. Group stage (The shared screen of Parabol via Microsoft Teams)

them into four columns (Plus, Delta, Ideas, and Flowers) evident from Fig. 4. Each column had a question or topic (What worked well? Things to improve, New things to introduce and Thank the team members who helped) that participants addressed. With respect to psychological safety level **contribute**. The participant text was written on various cards and placed under the four columns. The facilitator sought the participants *feedback* by *asking the question- "Should we put the.."* **digital post-it** cards *"in the sprint? or.."*. Some participants **contributed** by giving their **consent**- *yes* and some replied by remaining **silent** and letting the facilitator continue to arrange the cards under the columns.

4.4 Vote Stage

Participants vote at this stage. The facilitator **shared the screen** with all the voting options and used **audio** as an online tool to explain the cluster of cards one by one. The participants used **emoji (thumbs-up or like: 👍)** on the **digital post-it** cards to vote. A negative factor is a **finger-pointing** or being accused, is not a good practice during OAR. It tampers psychological safety. If done, participants might feel unsafe and less motivated to continue the OAR. Regarding psychological safety, level **contribute**. A participant **finger pointed** and *asked- "who did not vote? It is exactly one person who did not vote? Maybe..?"* and the peer replied, *"I voted"*. Again the *question* was raised. *"OK, if you voted, who did not vote?"*. There might have been several reasons not to vote. Probably not aware of the functionalities of the online tool, or someone may be new to an online platform. Later a participant shared the *feedback- "maybe someone did not know how to vote.* Hence, this resulted in few votes. It is a *constructive response* that made OAR psychological safe.

4.5 Discuss Stage

In the final stage of OAR, shown in Fig. 2 left side, participants discussed the previous stage's context and the next iteration sprint. This instance existed in the "cross-team" issue cluster. Concerning psychological safety, level **challenge**. One participant **challenged** the current situation of the cross-team tasks. The facilitator had a **shared screen** where participant's **avatar or photo with their name** was visible with **Parabol**. With *confidence*, the participant raised the *problem* using **audio-** *"I really did not like"* and shared the *opinion- "Probably it is a controversial opinion"* about the situation, but *"it would better if it is done in the other way"*. The facilitator appreciated, *"I like your opinion, we*

Table 2. Online tools influencing psychological safety

Agile retrospective stages - ***Online tools***	Psychological safety factors	Psychological safety level: behaviour
4.1) ICEBREAKER STAGE - Screen share, avatar or photo with name, audio, text, image or picture, thumbs-up or like (👍)	Self-referential humor, ignoring, silence	Included, contribute
4.2) REFLECT STAGE - Screen share, avatar or photo with name, audio, digital post-it	Silence	Included
4.3) GROUP STAGE - Screen share, avatar or photo with name, audio, text, digital post-it	Silence	Contribute: consent
4.4) VOTE STAGE - Screen share, avatar or photo with name, audio, text, digital post-it, (Thumbs-up or like: 👍)	Finger pointing	Contribute
4.5) DISCUSS STAGE - Screen share, avatar or photo with name, audio, text, heart (❤️), smile face (😀), neutral face (😐), sad face (☹️), flowers bouquet (💐), fire (🔥) and rocket (🚀)		Challenge: consent, contribute

could try to handle it in this way". While another participant joined the conversation and, with *confidence*, showed the **consent** and shared the *opinion- "In the previous sprints, we handled the situation in this way. The wrong part was that we did it all in the same sprint. However, many jobs were there to do. We were forced to work across the team"*. Finally, to finish the conversation, the first participant ended up with *constructive response* and shared the *opinion- "OK, in this context. I agree"* to you.

Regarding psychological safety, level **contribute**. The facilitator presented three clusters of digital cards. First, a discussion with 21 cards about the "cross-team" cluster. Then the "sprint" cluster with 16 cards and finally "thanks (miscellaneous)" was clustered with 13 cards. The facilitator read **aloud** each card's content and participants shared their *opinions* through **text** and **emojis**.

As evident from Fig. 2, different emojis **heart** ❤️, **smiley face** 😀, **neutral face** 😐, **sad face** ☹️, **flowers bouquet** 💐, **fire** 🔥, **rocket** 🚀 peers responded to the facilitator's *question*, *"Do you want to add something to the cluster of* *cards? If you think something is underestimated"*. One participant shared an *opinion* using **audio**, *"I think maybe on..card, where i wrote..I work a lot using.."* and another participant with a *constructive response*, shared *feedback- "I think it is good idea to add"*. Finally, the facilitator shared the *opinion- "OK, I will add a task card"*.

4.6 Summary

Psychological safety is essential for every workplace. We obtained several instances of interest by bracketing technique as a research approach. The finding answers the rq: how does the usage of online tools influence psychological safety in online agile retrospectives? Table 2 presents online tools which influence psychological safety during OAR. The team preferred video (screen share, avatar or photo with name), audio, chat (text, image or picture) and emoji as online tools to moderate the OAR. Instead of video, participants were interested in keeping the camera off and putting the avatar or photo with the name. The table also presents self-referential humour, ignoring, silence, and finger-pointing are the psychological factors and agree as consent or psychological behaviour that participants practised during OAR.

- It is vital to intermingle with the participants and invest some social time cultivating psychological safety. Since participants are online, it is crucial to start the online retrospective by revealing some fun facts for a team to know peers' emotional context. The use of an online tool provides a list of icebreaker questions where each participant can intermingle by responding to one question and knowing the team and their emotions.
- The team that conducted OAR did not have a scrum master. One person among the participants took the role of the facilitator and hosted the OAR. We found out that the digital retrospective board, an online tool, allow a structured and efficient way to conduct a retrospective, although the team was missing the scrum master. If the scrum master is not present to moderate, other participants can become the facilitator using the digital board to run the OAR. The retrospective board was psychological safe and included different stages of agile retrospectives.
- We also found that the facilitator sharing screen enhances the team's willingness to share emotions and contributions during OAR. When the team members see anonymous input from peers on the digital cards, they are more motivated to share contributions without fear. Being anonymous gives the team freedom, confidence, and free to express themselves.
- We discovered it is more convenient for a facilitator to moderate and give more contributions without hassles using an online tool. The facilitator was concentrated and involved with the participants due to online functionalities. For example, using digital post-it notes and commenting on them with participants' emotions and moods saved the facilitator's time. Hence, the facilitator invested more time and had feelings of being included during OAR.

- An efficient time control watch is visible on the shared screen with online tools. In this way, each participant's input is given equal importance and considered. Hence, allowing participants a feeling of being included in OAR. This psychology helps the team to have a control discussion mechanism.

5 Discussion

Participant's interaction matters most when online with peers [10,28], which helps influence psychological safety during OAR. Interesting to discuss is the silence that might occur during the meetings. In terms of psychological safety, participants' audio and written text messages are easy to decipher, but silence being a participant online is challenging. Silence could be consent that is either yes or no. Short or long enough, silence online could mean differently [15]. Peers might psychologically feel ignored during OAR. A long silence could be awkward [15]. However, it could be that the participant is taking time to think during the reflecting stage 4.2. Whereas during the *icebreaker stage,*4.1, the participant was silent and, without informing, was busy in another meeting. To overcome if the participant is busy, should share *information* via online tools such as *chat or breakout room* to convey the problem to the facilitator.

On the other hand, interaction through audio or writing is crucial [28] to realise psychological safety. Misinterpretation about silence might occur during online meetings [15].

Also, if long enough silence exists, the facilitator could raise a proactive question, *what do you think about the situation?* [9] to encourage interaction. Online tools do give support to factors and raise the interaction among participants. Suppose participants are introverted and do not like to raise their *opinions* via audio as an online tool. The team repeated the pattern of using emojis as an online tool during the OAR. Emoji could be a powerful way to share the *contribution* and speak aloud to the participant's *opinion.* Participants were able to present their emotions during the OAR without interrupting the speaker.

Video is one of the most applied online tools [10] during meetings [1], influencing psychological safety [9]. Instead of video, participants with a photo can use audio or other online tools to lead an effectual interaction by sharing opinions and asking questions. An interesting thing to notice was that all the participant's video was off for the entire OAR. Still, the participants showed they are psychological safety by challenging the status quo during the OAR.

Threats: We analysed one OAR with a single case, the external threat to our study. To what extent does the proposed study apply to other participants involved in the online meetings. The internal threat to our study is the history of the participants. Previously, how much they were familiar or acquainted. Some might acquaint themselves as long time working colleagues who show trust and respect with peers. To overcome, pre-session gave us insights into the entire OAR process and its participants. The session involved the project manager and two teams leaders. Both of them have been working with the company for many years. Then we also did we did a post retrospective FAQ session 3.2 with the team leader, where we asked various OAR questions. We recorded and observed all three sessions thoroughly to know the in-depth phenomena of psychological behaviour of OAR participants. Further analysis of other company participants may be interesting, as switching to other agile software development practices and remote work might affect the various psychological behaviours.

6 Conclusion

OAR provides an opportunity for participants to learn, contribute, and discuss iteration cycles if the team feel psychologically safe. This study outlines how online tools influence psychological safety factors, corresponding levels and behaviours. Due to icebreaker questions, accessible digital inputs, anonymous emotion sharing, commenting, and online retrospective facilitation via structured five stages. For researchers, the study is helpful, as it serves as a base stone that guides psychological safety research focused on the online perspective. Further research could be considered the psychological safety levels, factors and online tools with other online meetings. For practitioners, participants could use the study during the online agile retrospective and other online meetings and see if they feel psychologically comfortable contributing and willing to share their learning.

References

1. Agusriadi, A., Elihami, E., Mutmainnah, M., Busa, Y.: Technical guidance for learning management in a video conference with the zoom and youtube application in the covid-19 pandemic era. J. Phys. Conf. Ser. **1783**, 012119 (2021)
2. Braun, V., Clarke, V.: Thematic analysis. (2012)
3. Buvik, M.P., Tkalich, A.: Psychological safety in agile software development teams: work design antecedents and performance consequences. arXiv preprint arXiv:2109.15034 (2021)
4. da Camara, R., Marinho, M., Sampaio, S., Cadete, S.: How do agile software startups deal with uncertainties by covid-19 pandemic? arXiv preprint arXiv:2006.13715 (2020)
5. Clark, T.R.: The 4 Stages of Psychological Safety: Defining the Path to Inclusion and Innovation. Berrett-Koehler Publishers, Oakland (2020)
6. Duehr, K., et al.: The positive impact of agile retrospectives on the collaboration of distributed development teams-a practical approach on the example of bosch engineering gmbh. Proc. Des. Soc. **1**, 3071–3080 (2021)
7. Edmondson, A., Mortensen, M.: What psychological safety looks like in a hybrid workplace. Harvard Bus. Rev. **3**, 109 (2021)
8. Edmondson, A.: Psychological safety and learning behavior in work teams. Adm. Sci. Q. **44**(2), 350–383 (1999)
9. Edmondson, A.C., Daley, G.: How to foster psychological safety in virtual meetings. Harvard Bus. Rev. **25** (2020)
10. Fauville, G., Luo, M., Queiroz, A.C., Bailenson, J.N., Hancock, J.: Zoom exhaustion & fatigue scale. Comput. Hum. Behav. Rep. **4**, 100119 (2021)

11. Feitosa, J., Salas, E.: Today's virtual teams: adapting lessons learned to the pandemic context. Organ. Dyn. **50**(1), 100777 (2020)
12. Frisch, B., Greene, C.: What it takes to run a great virtual meeting. In: Harvard Business Review (2020). https://hbr.org/2020/03/what-it-takes-to-run-a-great-virtual-meeting
13. Gearing, R.E.: Bracketing in research: a typology. Qual. Health Res. **14**(10), 1429–1452 (2004)
14. Hulshult, A.: Using eight agile practices in an online course. J. High. Educ. Theor. Pract. **19**(3), 55 (2019)
15. Isbister, K., Nakanishi, H., Ishida, T., Nass, C.: Helper agent: designing an assistant for human-human interaction in a virtual meeting space. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 57–64 (2000)
16. Kahn, W.A.: Psychological conditions of personal engagement and disengagement at work. Acad. Manage. J. **33**(4), 692–724 (1990)
17. Karl, K.A., Peluchette, J.V., Aghakhani, N.: Virtual work meetings during the covid-19 pandemic: the good, bad, and ugly. Small Group Res. 10464964211015286 (2021)
18. Khanna, D.: Experiential team learning in software startups. In: Proceedings of the 19th International Conference on Agile Software Development: Companion, pp. 1–3 (2018)
19. Khanna, D., Nguyen-Duc, A., Wang, X.: From MVPs to pivots: a hypothesis-driven journey of two software startups. In: Wnuk, K., Brinkkemper, S. (eds.) ICSOB 2018. LNBIP, vol. 336, pp. 172–186. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04840-2_12
20. Khanna, D., Wang, X.: The buried presence of entrepreneurial experience-based learning in software startups. In: SiBW, pp. 30–39 (2018)
21. Langley, A.: Strategies for theorizing from process data. Acad. Manage. Rev. **24**(4), 691–710 (1999)
22. Lechner, A., Mortlock, J.T.: How to create psychological safety in virtual teams. Organ. Dyn. 100849 (2021)
23. Lenberg, P., Feldt, R.: Psychological safety and norm clarity in software engineering teams. In: Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering, pp. 79–86 (2018)
24. Masood, Z., Hoda, R., Blincoe, K.: Adapting agile practices in university contexts. J. Syst. Softw. **144**, 501–510 (2018)
25. Raake, A., Fiedler, M., Schoenenberg, K., De Moor, K., Döring, N.: Technological factors influencing videoconferencing and zoom fatigue. arXiv preprint arXiv:2202.01740 (2022)
26. Sureshchandra, K., Shrinivasavadhani, J.: Adopting agile in distributed development. In: 2008 IEEE International Conference on Global Software Engineering, pp. 217–221. IEEE (2008)
27. Topp, J., Hille, J.H., Neumann, M., Mötefindt, D.: How a 4-day work week and remote work affect agile software development teams. In: Przybyłek, A., Jarzębowicz, A., Luković, I., Ng, Y.Y. (eds.) LASD 2022. LNBIP, vol. 438, pp. 61–77. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94238-0_4
28. Tree, J.E.F., Whittaker, S., Herring, S.C., Chowdhury, Y., Nguyen, A., Takayama, L.: Psychological distance in mobile telepresence. Int. J. Hum. Comput. Stud. **151**, 102629 (2021)
29. Vandenberg, S., Magnuson, M.: A comparison of student and faculty attitudes on the use of zoom, a video conferencing platform: a mixed-methods study. Nurse Educ. Pract. **54**, 103138 (2021)

30. Yin, R.K.: Case Study Research: Design and Methods, vol. 5. Sage, Thousand Oaks (2009)
31. Zhang, Y., Fang, Y., Wei, K.K., Chen, H.: Exploring the role of psychological safety in promoting the intention to continue sharing knowledge in virtual communities. Int. J. Inf. Manage. **30**(5), 425–436 (2010)

Coordination Strategies When Working from Anywhere: A Case Study of Two Agile Teams

Tor Sporsem(✉) and Nils Brede Moe

SINTEF Digital, 7034 Trondheim, Norway
tor.sporsem@sintef.no

Abstract. Effective coordination is the key to successful agile teams. They rely on frequent interactions and mutual adjustment to manage dependencies between activities, which traditionally has been solved by co-locating the team. As the world is adjusting to post-covid work-life, companies are moving towards a work-from-anywhere approach where workers can choose to what degree they want to work from home or office. However, little is known about coordination in such a context. We report findings on developers' emerging strategies when working-from-anywhere, from an exploratory case study in Norway, including eight interviews. Our study shows that new strategies for mutual adjustment emerged as teams experimented with different tools and approaches: developers chose tasks according to location, tasks with vague requirements are performed collocated while individual tasks requiring focus are best performed at home; large meetings are virtual, preserving co-located time for collaborative tasks; using virtual rooms to maintain unscheduled meetings as they communicate mental presence to teammates, lowering the threshold for intra-team unscheduled talks. The strategies can help organizations create a productive and effective environment for developers.

Keywords: WFX · Work from home · Large-scale agile coordination · Co-located · Mutual adjustment · Unscheduled meetings · Virtual rooms · Discord · Slack · Hybrid

1 Introduction

In March 2020, technology companies closed their offices and sent employees to work from home (WFH), due to the Covid-19 pandemic. While some reported a decrease in developer productivity a recent study [1] found that many software developers benefit from WFH, and argued that most developers do not want to fully return to the office, while at the same time teamwork suffers. Therefore, many companies will opt for a hybrid workplace – office days mixed with WFH days. Consequently, companies like Facebook, Twitter, Square, Shopify, and Slack have established policies of long-term or even permanent working from home [2]. Spotify announced the Work-from-Anywhere (WFX) policy that allows employees to choose how often they prefer to be in the office or at home, or somewhere else. At the same time, there is little knowledge about the

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 52–61, 2022.
https://doi.org/10.1007/978-3-031-08169-9_4

long term effects of WFX. We have little knowledge on consequences for learning, coordination and solving tasks [1].

In agile teams, work relies heavily on coordination by feedback and mutual adjustment, particularly in meetings and ad hoc conversations [3]. Therefore, distributed agile teams need an effective coordination structure, with both scheduled and unscheduled meetings and the right informal collaboration tools to support mutual adjustment [4]. However, mutual adjustment in its pure form requires everyone to communicate with everyone [5]. Coordination by mutual adjustment is challenging when part of the team is working full time from home or from the office, or the whole team is working from anywhere. Also, it is challenging to know what collaboration should occur when the team is co-located, which sometimes is only a few times per week, month, or year. Given that coordination by mutual adjustment is essential for agile teams, and that more and more organizations are implementing practices for working from anywhere, we identified the following research question: *What coordination strategies are used by agile teams when working from anywhere?*

To answer, we report empirical insight from a case study on two developer teams in the company Entur. Since the study is exploratory, we have included both inter- and intra-team coordination. Section II describes related work. Section III outlines our research method and case context, followed by our findings. Section V discusses the strategies found and compares them to related research, concludes our work, and points to future research.

2 Coordinating Work in Distributed Agile Teams

Agile practices have stretched from the intended ideal of small co-located teams and reached safety-critical, large-scale, and distributed software development programs. Effective coordination is the key to success for agile teams in all contexts. A key to coordination "is managing dependencies between activities" [6]. In agile teams, coordination is exercised through several mechanisms [7]. As agile software development relies on frequent interactions and mutual adjustment, and since physical distance makes people communicate less [8], virtual teams need tools that can mitigate the barriers of distance and reduced communication.

In their study of distributed teams, Stray and Moe [4] found the IM tool Slack to be one of the most important collaboration and coordination tools. While Slack supported coordination in the distributed teams, the research by Stray shows that some users were very active, while others posted very few messages. Further, experienced team members favored messages in open channels while less experienced people favored more direct messages (i.e.one-to- one communication). At the same time, Slack causes interruptions. In their study of a globally distributed project, Matthiesen et al. [9] found that interruptions on IM tools were perceived as normal or as negative disruptions, depending on the quality of the relationships between the distributed colleagues. While tools are important, Calefato et al. [10] argue that face-to-face meetings are essential for having more in-depth discussions. In line with this, Stray [4] found the importance of co-locating permanent distributed teams once or twice a year and that the most complex and challenging meetings be organized during the co-location periods. In global software development, the setup is planned and voluntarily. In March 2020 most had to go home. To understand how WFX can work, there is a need to understand what happened during the pandemic, and specially why some teams struggled.

During the pandemic, several explanations have been found for why developers and teams had problems managing dependencies between team members. Examples are connectivity problem and poor workspace equipment, lack of match of working hours in the team, and greater difficulty in interpersonal communication [11, 12]. Smite et al. [1] found a reduced speed of solving tasks resulting from an increased number of meetings, worse understanding of what is going on in the team, and exhaustion from running meetings virtually. Furthermore, brainstorming sessions and problem-solving sessions were reported to be more challenging and to require more time due to the lack of accustomed whiteboards, possibility to spontaneously connect to the needed people, and requiring considerably more time to prepare. Finally, developers have stopped pair programming practices because they lack tool support or are not aware of the status of other team members [13]. At the same time, many have reported more effective task solving and work coordination from the home office. Reasons include better focus time, fewer interruptions, more time to complete work, more efficient meetings, and a better/more comfortable work environment [11, 12]. Smite et al. [1] found fewer distractions and interruptions, increased flexibility to organize ones work hours, and easier access to developers a person depend on to complete the work. While tasks are solved more effectively, coordination suffers [1].

3 Method

To answer the research question, we conducted a case study, investigating practices in two developer teams at Entur; a public, mature large-scale agile development company. We chose this case because Entur is part of an established research program. Entur has twenty development teams, and each team is responsible for their part of the digital infrastructure they deliver to the Norwegian public transport system. Prior to Covid-19, the teams used tools such as Slack, Jira, and Confluence, and material artefacts such as task boards. The teams chose freely how they go about solving their tasks and rely on agile methods of choice. As such, there was no one unified agile approach across the teams. More details can be found in [14, 15].

We followed two teams. Team Alpha (12 members) is responsible for the app used by travelers. Team Beta (9 members) gathers data from travel companies and structure them into products that other teams use to build their features. We chose these teams because we wanted to explore if coordination strategies differed as Alpha hold lots of dependencies to other teams, while Beta is mostly independent (others are to a large degree dependent on them). We kept an exploratory approach as we did not set out to test any specific theory or hypothesis [16] further, we hold an interpretive view in this study, comprehending the world and its truths as subjective realities [17].

Data collection spanned over three months (November 2021 to January 2022), including eight semi-structured virtual interviews (86 transcribed pages) and notes from two virtual stand-ups. In addition, the first author accessed the virtual workspace of Team Alpha, to observe how members utilized virtual rooms. Analysis was conducted in parallel with data collection, with codes rising inductively from data and forming categories and phenomena. Nvivo was used for coding and building categories. In March 2022, we presented the preliminary findings both in text and in-person presentation to the two teams and facilitated discussions to verify and adjust our findings.

4 Results

According to the company guidelines, the teams decided how to execute work-from-anywhere as long as they followed national covid-restrictions. In the period of 24th of September to 30th of November 2021, there were no restrictions. "The offices were completely open, but many choose to use the home office as the main base [in our team]," (B1). Team Alpha came to the office 2–3 days per week, except for a few members that never came in. Team Beta were located in two cities, where three members came to the office most days in one city, while those in the other city rarely went to the office. Prior to the Covid-19 pandemic, all developers in both teams went to the office every day.

4.1 Choosing Tasks

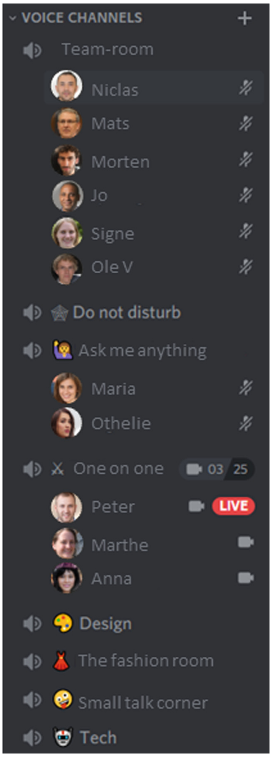
When choosing tasks from backlogs, developers take their location into consideration – whether they are at home or in the office. While co-located, the teams preferred tasks with an interpretive element, demanding frequent clarifications and discussions. "When developer and designer spend time together – that is the most valuable office-time. [...] These tasks have waited about a year, which we pick up now that we are hybrid and back in the office" (A2).

Two criteria are critical when choosing tasks for the home office: One criterion is that the task needs minor clarifications. "I pick simpler tasks [from the backlog] more often for the home office. [...] These are just-go-and-do-it tasks that we all agree on how to do," (A1). Informants in both teams tell a similar story of deliberately picking tasks with fewer dependencies with low coordination needs. This way, they "gain a feeling of progression" (B2). Examples of such tasks were bugfixes and small design adjustments.

The second criteria for home tasks is that the task requires uninterrupted focus. "We had this task where everyone worked alone on sub-tasks. We wouldn't gain the same degree of flow if we were at the office, even if we isolated ourselves in a meeting room. Some tasks are best suited when we can isolate at home" (A1). Despite setting up barriers to defend against interruptions, like putting up signs on the meeting room door, co-workers spotted them and found ways to squeeze in a quick talk. It is easier to hide away at home and stay uninterrupted". The team also avoids filling up their calendars with meetings during office days to enable collaborative work. This was a common opinion for all informants.

4.2 Use of Communication Tools

Team Alpha uses tools for mimicking their previous co-located work practices. When the teams were sent home when the pandemic started, an experienced gamer proposed using virtual rooms in Discord to sustain quick clarifications and short exchanges of information the same way online gamers do. They identified several rooms. A "Team-room" imitates their shared space at the office where they all sit together. A room called "One-on-One" imitates meeting rooms where developers can retreat for private discussions. "Do-not-disturb" is like a quiet room (Fig. 1).



VOICE CHANNELS
Team-room
Niclas
Mats
Morten
Jo
Signe
Ole V
Do not disturb
Ask me anything
Maria
Othelie
One on one 03 25
Peter LIVE
Marthe
Anna
Design
The fashion room
Small talk corner
Tech

Fig. 1. Shows the virtual rooms and their participants (pictures are generated by an AI for anonymity). In the 'Team-room', six members are present, all muted but with their speakers on, simulating their shared team space at the office. No one is present in 'Do not disturb'. While two are present in 'Open for questions', they are also muted. Three members have a live discussion in 'One-on-one' with their cameras on. The other rooms, 'Design', 'The Fashion Room', 'Small talk corner', and 'Tech' are empty.

Observing each other's presence in different rooms provides awareness of coworkers' state of mind. "I can see, for example, that Maria and Peter are sitting in another room and having a meeting. [...] you know where they are [mentally]" (A4). Awareness of what others are doing helps developers interpret if it is appropriate to approach them. "Discord matches how we work when we sit near each other in the office. We can get quick clarifications like 'can you have a brief look at this? Looks OK?'" (A1). Knowing when a person can be contacted lowers the threshold for contacting them, and helps progress in their tasks. All informants in Team Alpha told the same story, often using the same words to describe it.

In contrast, tools like Slack and Teams do not create the same awareness because there is a mistrust of status indicators (indicating i.e. *available* when green and *busy* when red). Unclear statuses make it hard to know when co-workers can be approached/contacted. "You don't know if you are interrupting people when you contact them on Slack. [...] you have no idea what they are doing. [...] I don't update it [my status indicator] much myself. Based on how I use it myself, I may not fully trust it" (B2). "Yellow or orange or red... I don't dare trust them" (B3). As we have seen, Team Alpha mitigated such challenges by using virtual rooms, while Team Beta relied on Slack.

Implementing tools like Discord requires experimentation. "In the beginning, everyone had their microphone unmuted to make it feel like you were in the office, but at home, you also have other sounds that come from the kitchen or children or cats and stuff, so it did not work well," (A3). Experimentation led Team Alpha to a practice where speakers are un-muted, combined with muted microphones when members are not speaking. In that way, they can unmute and ask questions or address someone while everyone hears it.

When asked if this is annoying for others in the same virtual room, all informants told us that the practice enabled transparency and opportunities to include oneself. “If you do not like it, you can always turn off your sound, it will be like putting on headphones in the office” (A3). “I thought maybe it would be a little tiring, but it’s not. People are very respectful and do not bother each other” (A4).

An important feature is moving members between rooms. “We are all administrators, so that we can move each other between rooms. It’s convenient if you want to talk to someone, just enter a room and stick him in there with you and we are off talking. This is the new way of tapping someone on the shoulder when they have their earphones on in the office” (A1).

Although it may be true that virtual rooms maintain unscheduled meetings in virtual settings, things look different on days when the majority of the team is co-located. When presenting preliminary findings to Team Alpha, discussions revealed that they downgraded their use of Discord when coming to the office because they physically observed each other’s mental presence. Those few who worked virtually on such days stopped relying on the virtual rooms to communicate teammates’ mental presence. However, they all agreed that on non-office days, Discord was still the “lifeline of operations.”

4.3 Meetings

Unscheduled meetings in the office have transformed into scheduled meetings virtually. Informants highlight this transition as one of the biggest challenges when working virtually. “In the office, it is easy just to say “hey, shall we do this?” and then you have sort of made a clarification in 15 s. While digitally, you often end up having to invite for another meeting” (A4). When virtual, people first ask for a talk, then agree if they should meet face to face or virtually, then find a time that suits both calendars. Discord is a way of shortening this process.

While Discord solved the problem of scheduling meetings on team level, the problem still persisted on the inter-team level.: “…each team is on its own Discord server. However, collaboration across teams takes place mainly via Slack or Teams. And there it is again – you have to arrange meetings in advance” (A3).

Even when teams are free to work at the office, inter-team meetings are still challenging. “On those days we were at the office, the other teams weren’t” (A3). Informants speculated on various reasons for this: it is more comfortable to go when there are fewer colleagues to share the space with; the best meeting-rooms are available; it is precious time for the teams to meet internally and build cohesion. On the other hand, managers tend to go in on the same days. “Those I need to meet in person [outside my team], I almost always meet them on Tuesdays and Thursdays [their common office days]. Once we have started talking in person, it’s easier to take it up again digitally on Slack” (A2).

Interestingly, Team Alpha has concluded that large meetings and retrospectives are exclusively for home-office. The combination of well-functioning virtual whiteboards, competition for the best equipped meeting rooms and that teams are seldom present simultaneously makes virtual meetings easier. “There is always someone with a cold or has a sick child, or an [private] appointment to run to. There are always at least two at home" (A2). Virtual meetings led to higher inclusion as everyone always gets to participate. Additionally, retrospectives are automatically documented in virtual whiteboards, whereas they have to convert the whiteboard in physical meetings into digital documents.

5 Discussion and Conclusion

We have seen how two software development teams over a period of 3 months used various tools and strategies to cope with working from anywhere. Entur offers a full flex solution where teams decide themselves where to work from and how many days at the office. Now, we turn to discuss our research question, *what coordination strategies are used when working from anywhere?* Three distinct strategies that emerge from our data, are summarized in Table 1.

Table 1. Strategies for mutual adjustment when working from anywhere

Strategy	Description/Rationale
Work location decides tasks	Tasks with vague requirements are performed collocated because they often require unscheduled discussions and clarifications, which are more effective in-person. Individual tasks requiring focus are best performed at home
Unscheduled meetings are maintained in virtual rooms	Virtual rooms reveal mental presence to teammates, lowering the threshold for intra-team unscheduled talks
Meeting type decides location	Meetings reporting status are reserved for virtual time to free up office time for unscheduled meetings. Those forced to stay at home, for various reasons, are still included and updated on important information

Tasks with vague requirements are chosen for office time because they often require continuous clarifications, joint decision-making, or discussions while working (mutual adjustment or frequent coordination). Our findings are in accordance with Calefato et al. [10] who argue that face-to-face meetings are essential for having in-depth discussions. Co-location seems especially important when tasks require multiple competencies or domains, for example when a developer and designer collaborate on a task. Being co-located makes it easier to adjust to each other's expectations and comprehensions by solving problems together. Further this practice reduced waiting time and blockages which is important for effective coordination [7], and reduced communication problems when solving complex tasks. Teams with communication problems are likely to experience problems coordinating their work [18]. To secure enough time for working co-located, large meetings (typically reporting status) and individual work are down-prioritized, and set aside for the home-office.

Unscheduled meetings are close to the core of mutual adjustment and upheld through virtual rooms. Being present in a room reveals hints about mental presence that help coworkers interpret when it is appropriate to approach them – making it easier to reach out for a quick clarification. For example, when a developer observes a coworker in a meeting room with their manager, he recognizes that this is not the right moment to interrupt. On the other hand, if the developer observes them together at the coffee machine, he can take this opportune moment to interrupt with a quick question. Smite et al. [13] found that a lack of tools showing status of the other teams members was a reason for not being able to mimic the old working practices like pair programming. Further, awareness of what is happening and who is doing what also seemed to initiate unscheduled meetings. Our findings suggest that virtual rooms through Discord facilitates constant informal communication, which improves communication in distributed agile projects [19]. Increased transparency also builds trust, which is vital for distributed teams' success [20].

To conclude, the three strategies affect mutual adjustment by maintaining unscheduled meetings and informal talks. This especially holds true in an intra-team setting, while these strategies seem to struggle in inter-team settings.

Our explorative findings show a need to further understand emerging strategies when WFX. Especially, investigating how these new strategies differ from those already known in the fields of Global Software Engineering and Computer-Supported Cooperative Work (CSCW). Future research should examine the three strategies in new contexts as they will change in the coming years. For example, virtual rooms have only been utilized for a few months in a hybrid setting and will most likely change as teams keep adapting. Also, what long term effects on processes like user involvement, knowledge transfer and onboarding new team members are worth investigating.

Acknowledgements. We wish to thank Entur and the informants for willingly sharing their experiences. Also, we thank Knowit AS and the Norwegian Research Council and for funding the research through the projects Transformit (grant number 321477) and A-Team (grant number 267704).

References

1. Smite, D., Tkalich, A., Moe, N.B., Papatheocharous, E., Klotins, E., Buvik, M.P.: Changes in perceived productivity of software engineers during COVID-19 pandemic: the voice of evidence. J. Syst. Softw. **186**, 111197 (2022). https://doi.org/10.1016/j.jss.2021.111197
2. Stoller, K.: Never want to go back to the office? Here's where you should work. https://www.forbes.com/sites/kristinstoller/2021/01/31/never-want-to-go-back-to-the-office-heres-where-you-should-work/. Accessed 18 Feb 2022
3. Moe, N.B., Dingsøyr, T., Rolland, K.: To schedule or not to schedule? An investigation of meetings as an inter-team coordination mechanism in large-scale agile software development (2018)
4. Stray, V., Moe, N.B.: Understanding coordination in global software engineering: a mixed-methods study on the use of meetings and Slack. J. Syst. Softw. **170**, 110717 (2020). https://doi.org/10.1016/j.jss.2020.110717

5. Groth, L.: Future organizational design: the scope for the IT-based enterprise (1999)
6. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. ACM Comput. Surv. **26**, 87–119 (1994). https://doi.org/10.1145/174666.174668
7. Strode, D.E., Huff, S.L., Hope, B., Link, S.: Coordination in co-located agile software development projects. J. Syst. Softw. **85**, 1222–1238 (2012)
8. Noll, J., Liu, W.-M.: Requirements elicitation in open source software development: a case study. In: Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, pp. 35–40 (2010)
9. Matthiesen, S., Bjørn, P., Petersen, L.M.: "Figure out how to code with the hands of others" recognizing cultural blind spots in global software development. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp. 1107–1119 (2014)
10. Calefato, F., Damian, D., Lanubile, F.: Computer-mediated communication to support distributed requirements elicitations and negotiations tasks. Empir. Softw. Eng. **17**, 640–674 (2012)
11. Ford, D., et al.: A tale of two cities: software developers working from home during the covid-19 pandemic. ACM Trans. Softw. Eng. Methodol. (TOSEM) **31**, 1–37 (2021)
12. Oliveira Jr., E., et al.: Surveying the impacts of COVID-19 on the perceived productivity of Brazilian software developers. In: Proceedings of the 34th Brazilian Symposium on Software Engineering, pp. 586–595 (2020)
13. Smite, D., Mikalsen, M., Moe, N., Stray, V., Klotins, E.: From collaboration to solitude and back: remote pair programming during Covid-19. In: Gregory, P., Lassenius, C., Wang, X., Kruchten, P. (eds.) XP 2021. LNBIP, vol. 419, pp. 3–18. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78098-2_1
14. Berntzen, M., Moe, N.B., Stray, V.: The product owner in large-scale agile: an empirical study through the lens of relational coordination theory. In: Kruchten, P., Fraser, S., Coallier, F. (eds.) XP 2019. LNBIP, vol. 355, pp. 121–136. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19034-7_8
15. Berntzen, M., Hoda, R., Moe, N.B., Stray, V.: A taxonomy of inter-team coordination mechanisms in large-scale agile. IEEE Trans. Softw. Eng., 1 (2022). https://doi.org/10.1109/TSE.2022.3160873
16. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**, 131 (2008)
17. Oates, B.J.: Researching Information Systems and Computing. Sage (2005)
18. Marks, M.A., Mathieu, J.E., Zaccaro, S.J.: A temporally based framework and taxonomy of team processes. Acad. Manag. Rev. **26**, 356–376 (2001)
19. Ramesh, B., Cao, L., Mohan, K., Xu, P.: Can distributed software development be agile? Commun. ACM **49**, 41–46 (2006)
20. Moe, N.B., Šmite, D.: Understanding a lack of trust in Global Software Teams: a multiple-case study. Softw. Process Improv. Pract. **13**, 217–231 (2008)

Agile Processes

Roles of Middle Managers in Agile Project Governance

Maduka Uwadi[1](✉), Peggy Gregory[1], Ian Allison[1], and Helen Sharp[2]

[1] University of Central Lancashire, Preston PR1 2HE, UK
{mcuwadi,ajgregory,iallison}@uclan.ac.uk
[2] The Open University, Milton Keynes MK7 6AA, UK
helen.sharp@open.ac.uk

Abstract. Project governance is an important activity in agile software development (ASD) projects for project success. Middle managers are part of the governance structure in ASD projects. Despite the efficacy of project governance and existence of middle managers in agile teams, project governance and middle management in ASD projects are under-researched. This multiple-case study investigates the roles of middle managers in agile project governance activities within two Nigerian ASD projects through the lens of activity theory. We collected data in semi-structured interviews, observations, questionnaires, and company documents. Our findings show that middle managers performed 25 roles related to planning and coordination for project alignment and execution, continuous improvement and organisational change, agile and technical leadership, monitoring, and capability building. We conclude that middle managers are pivotal to project governance practice and the effectual functioning of agile teams in ASD projects. The study will help agile practitioners to better understand the roles of middle managers in agile project governance. Results from this work contribute to the 'middle management in agile' debate and offer an alternative view that may change beliefs about middle managers in agile project settings.

Keywords: Agile project governance · Middle managers · Agile software development · Activity theory · Interpretive case study

1 Introduction

Project governance (PG) is an important but complex activity performed during agile software development (ASD) projects, and encompasses the necessary oversight, processes, tools, manpower, and support to accomplish projects [23]. Despite its importance, PG vis-à-vis ASD projects, is under-researched and not fully understood [13,23].

Middle managers (MMs) in ASD projects participate in project activities, relay senior management (top management) directives to lower-level personnel, ensure implementation of directives in projects, and communicate implementation progress reports back to senior management (SM). MMs in agile teams may

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 65–81, 2022.
https://doi.org/10.1007/978-3-031-08169-9_5

include Scrum masters as gatekeepers and product owners as stakeholder representatives [29], as well as line managers [1]. Although MMs exist in agile teams, there is a lack of clarity about the role of MMs in ASD projects [12,24], and Barroca et al. [6] show this is one of the top ranked challenges affecting agile teams. Agile projects are considered lightweight, self-organising, and flexible, hence practitioners question how 'management' and 'governance' fit in. Middle manaager (MM) role uncertainty may generate tensions within agile teams during task execution [12], thereby threatening team stability and project congruity.

To shed light on this topic, this study seeks to answer the question: *What are the roles of middle managers in agile project governance?* To answer, we conduct case studies of PG activities in ASD projects within two companies: HOLDCOY and BANKCOY, in order to determine the roles of MMs in agile PG.

This article is an extended version of [32], which presented preliminary findings from a single case study. In this extended article, we include further empirical data from additional interviews and observations conducted in the first case study and findings from a second case study to present a composite thematic model of middle management roles in agile project governance (PG).

2 Related Work

PG is the "framework, functions, and processes that guide project management activities in order to create a unique product, service, or result to meet organizational strategic and operational goals" [28, p. 4]. In project management, governance includes "the set of policies, regulations, functions, processes, procedures and responsibilities" that are involved in establishing, managing, and controlling projects, programmes, and portfolios [2, p. 8]. PG is an important project activity with the capacity to advance project performance and success. It provides SM with crucial information to make informed investment and risk decisions regarding projects, while allowing developers to build products iteratively and incrementally under conditions of uncertainty [16]. PG enables operation of governance mechanisms, roles, and metrics, which allow project personnel to monitor project performance and risks in order to realise business value [31].

Kujala et al. [21] derived a six-dimensional PG framework, which Lappi et al. [23] synthesised with findings from their review of 42 agile studies to develop a framework conceptualising agile PG in six PG dimensions, viz., goal setting, incentives, monitoring, coordination, roles and decision-making power, and capability building. This agile PG framework by [23] answered the question: "What is agile project governance?" in Lappi [22]. The six PG dimensions include activities, agile practices, and roles that are utilised and performed by various actors in agile PG [23]. For example, agile PG actors include the project manager: acts as coordinator or administrator of agile team; agile coach: supervises agile capabilities in agile team; and Scrum master: manages team performance and sprints. They did not discuss the actors in the context of organisational levels they belong to, hence middle management was not considered. However, the study calls for further research to better understand agile PG across organisational levels and its pervading effects in organisations; “from top management via projects to individuals” [23, p. 54]. The authors also highlight weak organisation-project strategic connections as an agile PG issue and the need for further research to examine how PG structures and practices can help strengthen such connections.

Middle managers (MMs) are the intermediary workforce that link SM with other teams that operate in the lower echelon of an organisation [5]. They occupy the middle-level position in an organisation’s governance structure, reporting to SM who provide strategic direction, and serving as nexus between SM and the workforce that executes core tasks at project-level [5]. In essence, MMs receive, consume and transmit strategic directives in top-down fashion, perform and oversee implementation activities, and communicate implementation reports to SM. According to Cheng et al. [8], MMs are subordinate to SM and supervise at least two layers of lower-ranking staff. Still, the positions “in the middle” may vary depending on organisation size and context [4]. For instance, several layers of people may be positioned “in the middle” in large organisations, and in the wider organisation they are all regarded as MMs. Smaller organisations may have fewer organisational levels and few people in the middle echelon.

Kalenda et al. [19] argues that agile teams are no longer expected to be managed by MMs. MMs are seen as liabilities to organisational agility because they tend to resist change and agile transformation initiatives [19]. Nevertheless, there is ‘management’ and ‘leadership’ in agile settings. Parker et al. [27] suggest when a manager embraces agile practices, the manager can become an adaptive leader while managing the agile team. Little is known about MM role in ASD projects [6,12,24]. Hoda et al. [17] examined self-organising roles in ASD teams and identified several self-organising roles that exist within agile teams, viz., mentor, coordinator, champion, promoter, translator, and terminator. They highlighted positive influences of SM in supporting self-organising agile teams, however, the role of MMs was not considered in the study. Shastri et al. [30] examined the “agile manager” role in agile project management in a generic context without specifying the managerial level. They identified four agile manager roles: coordinator, mentor, negotiator, and process adapter. Moe et al. [24, p. 16] mentions “Redefining the managers [*sic*] role” and “Right level of responsibility” as major barriers to effective functioning of self-organising teams, thus highlighting issues in ASD projects, which includes issues associated with middle management and governance. There is also a lack of understanding as to the decision-making power of MMs, and the legacy roles required in ASD projects [24].

Regarding impact of MMs in ASD projects, Russo [29] reports in an agile transformation study that MMs were taking the roles of Scrum masters and product owners. They were ranked above developers. The MMs were hands-on in mediating between SM software expectations and daily development issues to develop a desired system. SM valued the domain knowledge and adaptability of the Scrum masters, who also served as gatekeepers that focused on agile values in the project environment. Scrum master leadership skills were also vital in dealing with various day-to-day project issues. Product owners ensured alignment between stakeholder expectations and completed software features. Hermkens et al. [15] argue that MMs will remain instrumental to organisational agility, albeit this brings changes to the role of MMs. [15] therefore calls for research to ascertain the impact of the agile approach on the middle management role, as well as ascertain the roles of MMs that are most contributory to organisational agility.

3 Research Design and Case Description

This study adopts a qualitative and interpretive multiple-case study design. This is well-suited because it puts the researchers in the world of the study participants living the PG and middle management experience in the ASD project settings, thereby allowing them to interpret the views and experiences of the participants [33]. Case study design was selected because case studies are recommended when prior research is limited and under-researched [7]. In addition, case studies are particularly suited for practitioner-oriented studies aiming to address "practice-based problems where the experiences of the actors are important and the context of action is critical" [7, p. 369], which applies to this study. Multiple-case design provides broader picture of issues in different organisations, which strengthens evidence and generalisability of findings [7]. A case study protocol was used as the agenda for inquiry at each case organisation.

Agile PG is complex and multifaceted in nature given that it involves multiple actors, processes, tools, and socio-technical interactions aimed at achieving project success [23]. Consequently, our study demanded a flexible socio-technical theoretical framework with expansive analytical and interpretive power; activity theory lends itself to these demands [11,18,20]. Activity theory was used as the principal theory to develop an Activity-oriented Project Governance (APGov) conceptual framework (Fig. 1) to aid data collection, analysis, and results interpretation. In this present article, we only report on division of labour in relation to the roles of middle managers (MMs) in the agile PG activity. The unit of analysis for this study is the PG activity, which has ASD project as the main governance object, and middle management as one of the activity actors.

Data was collected from two companies between February and March 2020 and it involved 20 semi-structured interviews, three project team meeting observations, company documents, and questionnaires (which were only used to collect qualitative data about the companies and their ASD projects). The interviews, observations, and administering of questionnaires were performed by the first author. The use of semi-structured interviews facilitated information elicitation, interview question adaptation, and further probing, which helped to obtain first-level constructs (facts) and interesting insights from participants. Interviewees included three members of SM, ten MMs, and seven members of lower-level workforce (LOW) so as to obtain a variety of perspectives. Interviewees were asked to reflect on past project events. We used observations to complement other data sources and facilitate discovery of occurrences, subtleties, and actions in the cases [7]. For observations, we employed direct non-participant observation approach [9], and took 'outside observer' role [33]. Only one company was observed

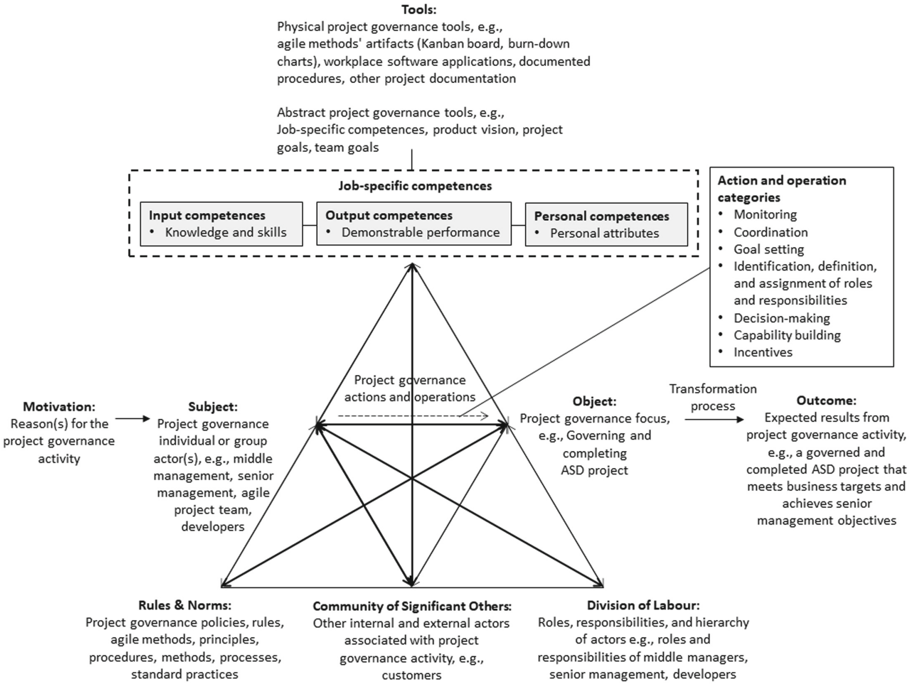

Tools:
Physical project governance tools, e.g., agile methods' artifacts (Kanban board, burn-down charts), workplace software applications, documented procedures, other project documentation
Abstract project governance tools, e.g., Job-specific competences, product vision, project goals, team goals
Job-specific competences
Input competences
• Knowledge and skills
Output competences
• Demonstrable performance
Personal competences
• Personal attributes
Action and operation categories
• Monitoring
• Coordination
• Goal setting
• Identification, definition, and assignment of roles and responsibilities
• Decision-making
• Capability building
• Incentives
Project governance actions and operations
Motivation: Reason(s) for the project governance activity
Subject: Project governance individual or group actor(s), e.g., middle management, senior management, agile project team, developers
Object: Project governance focus, e.g., Governing and completing ASD project
Transformation process
Outcome: Expected results from project governance activity, e.g., a governed and completed ASD project that meets business targets and achieves senior management objectives
Rules & Norms: Project governance policies, rules, agile methods, principles, procedures, methods, processes, standard practices
Community of Significant Others: Other internal and external actors associated with project governance activity, e.g., customers
Division of Labour: Roles, responsibilities, and hierarchy of actors e.g., roles and responsibilities of middle managers, senior management, developers

Fig. 1. APGov framework [32]

because the project in the second company was already completed at the time of data collection. Observations in the observed company were limited to three project team meetings due to the COVID-19 outbreak. Use of observations in one company did not affect overall results from both companies: observation data substantiated other collected data. For more sample population details, interview protocol, and other data sources details, visit https://bit.ly/3uL1Ryl.

Data analysis was performed using thematic network analysis [3]. A thematic network consists of (a) basic themes, which are the lowest-order premises found in the data, (b) organising themes, which are higher-order themes (categories of grouped basic themes) summarising main discoveries contained in the data [3], and (c) global theme, which is the superordinate theme that encapsulates "the principal metaphors in the data as a whole" [3, p. 389]. Interview transcripts and observation notes were read several times and coded by applying a coding framework comprised of components of the APGov framework, research interests, and emerging discoveries from data [3]. NVivo and Microsoft Word were used to organise text segments into codes, which later formed themes for the construction of a thematic network interpreting various roles of MMs in agile PG. All possible roles of MMs referenced in the raw data were coded. This process produced a total of 40 codes, which were reduced to 25 basic themes (MM roles). The basic themes were grouped into organising themes (role categories) by considering the MMs' contexts. As a quality check, collected data and analysis findings were shared with participants. Responses were noted and helped clear up misconceptions. Cross-case analysis was done to identify similarities and differences in the MM roles across the two cases. The steps in the analysis process were performed by the first author and checked by the other authors to ensure analysis and interpretations accorded with data and research standards.

Two Nigerian case studies involving a financial technology (fintech) company; HOLDCOY, and a bank; BANKCOY, were undertaken. Both companies were undergoing agile transformation. The Nigerian technology and finance industries were germane for this study due to the use of agile development to create and deploy software solutions for financial services in the region [26]. Brief descriptions of each case organisation will now be given.

HOLDCOY is a Nigerian fintech holding company that was established in 2008. It has five divisions and several functional areas (e.g., Operational Excellence (OpEx) team), which provide shared services to all the divisions. The company has used agile methods to implement and govern software projects for eight years. HOLDCOY's corporate customers include banks and other financial services providers. The research in HOLDCOY was limited to analysis of the PG activity and middle management in one of its divisions: the TECHCOY division, which was the agile project team executing the ASD project under examination. The project entailed development of a software to be used by financial services providers for inter-banking services to their customers and it had been ongoing for two and a half years. The project used Scrum, Kanban and Dynamic Systems Development Method (DSDM) in its delivery with modifications tailored to suit the company. The TECHCOY agile project team performed daily Scrum meetings in weekly/biweekly sprints, sprint planning, sprint reviews, monthly retrospectives, and Monthly Performance Review (MPR) sessions. MPR is used by SM to review, provide feedback, and grade the performance of TECHCOY agile project team as a whole, as well as the performance of the sub-teams. It is also used to set, plan, and continuously review monthly project goals in collaboration with the TECHCOY agile project team. The observed MPR session was attended by SM (led by the Group CEO), TECHCOY agile project team, and other internal stakeholders. The observed daily Scrum and sprint planning meetings were attended by the TECHCOY agile project team members only.

The TECHCOY agile project team was co-located and cross-functional, comprised of 13 persons (ten full-time employees and three interns), which included three MMs: Head of Operations (P1), Head of Technology and Scrum Master (also a senior software developer) (P6), and Head of Business Development (P7). It was led by a divisional CEO (P9), who is not a MM but a member of HOLDCOY's SM team. The agile project team comprised of several sub-teams. Developers in the agile project team were mostly junior-level developers who had limited competency and industry domain knowledge. This was a concern. The developers were not competent to the point where they could perform their tasks unsupervised, hence middle management closely monitored the project (using code reviews for example) to ensure the quality and integrity of software outputs were not flawed. The agile project team spent project time travelling between their office and customer offices to collaborate with customer teams.

BANKCOY is a Nigerian microfinance bank that has used agile methods for software project implementation and governance for three years. The bank was established in 2008. It implements projects to build software solutions for financial services to customers. The bank has an IT team of 40 staff which provide IT services, including in-house software development. The IT team is led by a Chief Information Officer (CIO) and supported by seven MMs.

The BANKCOY project was an ASD project to build a solution that allows customers transfer funds from other banks to their BANKCOY bank accounts. It was completed in nine weeks in 2019 through monthly sprints. The project used Scrum and Kanban. The agile project team was co-located and cross-functional. It comprised of 12 full-time employees, including six of the seven MMs: Project and Change Coordinator (P11), E-channels Manager (P12), DevOps Lead (also a software developer) (P13), IT Operations Manager (P14), Information Security and Assurance Lead (P16), and Head of Service Delivery (P18). The CIO (P21) is not a MM; he is part of the senior management (SM) team.

The MMs were part of the agile project team in each case. The three MMs in HOLDCOY and six MMs in BANKCOY—all SM direct reports—were the people officially recognised by SM in each company as the MMs in the respective agile project teams based on each company's organisational structure. For organisational structure diagrams of both cases, visit https://bit.ly/3uL1Ryl.

4 Results

Results show that the MMs performed 25 roles in the two cases during the governance of their ASD projects. Comparing and combining the identified themes in the two cases produced a composite thematic network comprised of 25 basic themes that represent the roles MMs performed within the agile PG activity's division of labour in the two companies (see Fig. 2). The roles were grouped into five organising themes (role categories): *Planning and coordination for project alignment and execution*, *Continuous improvement and organisational change*, *Agile and technical leadership*, *Monitoring*, and *Capability building*, and linked to a global theme - *Roles of middle managers in agile project governance*. Through these roles, the MMs supported their respective agile project teams and contributed towards agile PG practice in their respective ASD projects.

There were similarities and differences regarding the MMs roles we found. We found that of the 25 roles, 24 roles were performed by MMs in HOLDCOY, whereas in BANKCOY 21 roles were performed by the MMs. Four roles in HOLDCOY were not found in BANKCOY, i.e., *Pastoral Care Provider*, *Auxiliary Resource*, *Foreseer*, and *Auditor*. One role in BANKCOY was not found in HOLDCOY, i.e., *Mediator*. Results suggest there were no differences regarding the role categories under which the MM roles were performed in the respective agile PG activities of the two companies. The following subsections and tabular figures describe each role under the five role categories. Results show that a MM

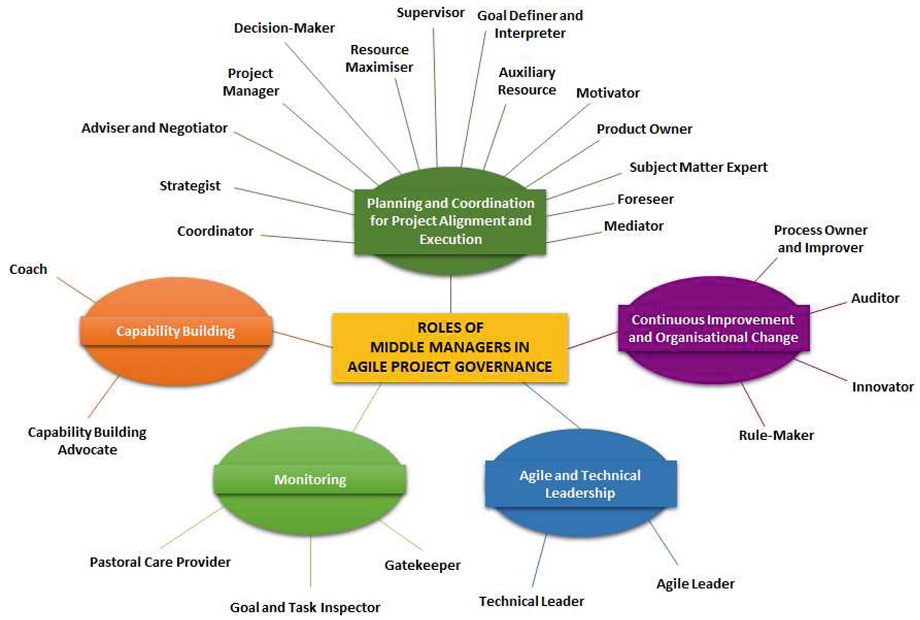


Fig. 2. Thematic network of MM roles in agile PG

can perform one or more of these roles in different instances as circumstances demand during project implementation. Also, more than one MM can take up the various MM roles regardless of job title.

4.1 Planning and Coordination for Project Alignment and Execution

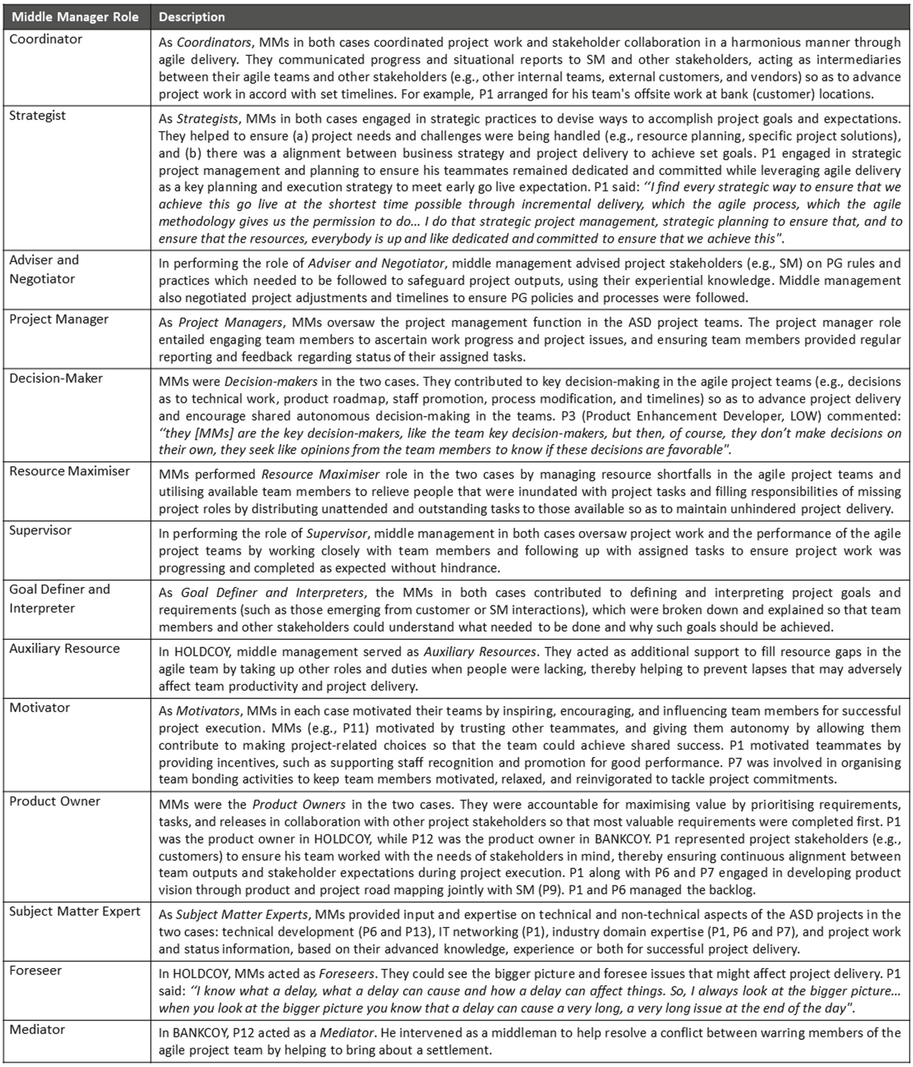
In ASD projects, stakeholders need to work together in order to be successful and accomplish project tasks and goals. Planning, coordination, and maintaining alignment between and with stakeholders, timelines, and business strategy throughout project delivery are important for project success. MMs supported these practices through several roles described in Fig. 3.

4.2 Continuous Improvement and Organisational Change

The MMs engaged in continuous improvement efforts to improve working processes and support team productivity. These efforts tended to result in organisational changes. They engaged in such efforts by performing *Process Owner and Improver*, *Auditor*, *Innovator*, and *Rule-maker* roles (see Fig. 4).

4.3 Agile and Technical Leadership

ASD projects involve developing software solutions following a set of work rules, principles, values, and technical activities to decompose and accomplish solution

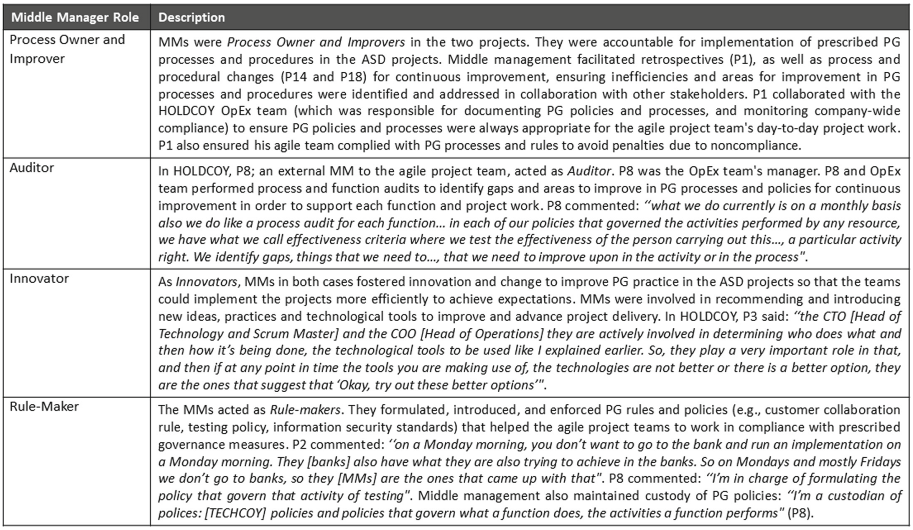


Middle Manager Role	Description
Coordinator	As *Coordinators*, MMs in both cases coordinated project work and stakeholder collaboration in a harmonious manner through agile delivery. They communicated progress and situational reports to SM and other stakeholders, acting as intermediaries between their agile teams and other stakeholders (e.g., other internal teams, external customers, and vendors) so as to advance project work in accord with set timelines. For example, P1 arranged for his team's offsite work at bank (customer) locations.
Strategist	As *Strategists*, MMs in both cases engaged in strategic practices to devise ways to accomplish project goals and expectations. They helped to ensure (a) project needs and challenges were being handled (e.g., resource planning, specific project solutions), and (b) there was a alignment between business strategy and project delivery to achieve set goals. P1 engaged in strategic project management and planning to ensure his teammates remained dedicated and committed while leveraging agile delivery as a key planning and execution strategy to meet early go live expectation. P1 said: *"I find every strategic way to ensure that we achieve this go live at the shortest time possible through incremental delivery, which the agile process, which the agile methodology gives us the permission to do... I do that strategic project management, strategic planning to ensure that, and to ensure that the resources, everybody is up and like dedicated and committed to ensure that we achieve this".*
Adviser and Negotiator	In performing the role of *Adviser and Negotiator*, middle management advised project stakeholders (e.g., SM) on PG rules and practices which needed to be followed to safeguard project outputs, using their experiential knowledge. Middle management also negotiated project adjustments and timelines to ensure PG policies and processes were followed.
Project Manager	As *Project Managers*, MMs oversaw the project management function in the ASD project teams. The project manager role entailed engaging team members to ascertain work progress and project issues, and ensuring team members provided regular reporting and feedback regarding status of their assigned tasks.
Decision-Maker	MMs were *Decision-makers* in the two cases. They contributed to key decision-making in the agile project teams (e.g., decisions as to technical work, product roadmap, staff promotion, process modification, and timelines) so as to advance project delivery and encourage shared autonomous decision-making in the teams. P3 (Product Enhancement Developer, LOW) commented: *"they [MMs] are the key decision-makers, like the team key decision-makers, but then, of course, they don't make decisions on their own, they seek like opinions from the team members to know if these decisions are favorable".*
Resource Maximiser	MMs performed *Resource Maximiser* role in the two cases by managing resource shortfalls in the agile project teams and utilising available team members to relieve people that were inundated with project tasks and filling responsibilities of missing project roles by distributing unattended and outstanding tasks to those available so as to maintain unhindered project delivery.
Supervisor	In performing the role of *Supervisor*, middle management in both cases oversaw project work and the performance of the agile project teams by working closely with team members and following up with assigned tasks to ensure project work was progressing and completed as expected without hindrance.
Goal Definer and Interpreter	As *Goal Definer and Interpreters*, the MMs in both cases contributed to defining and interpreting project goals and requirements (such as those emerging from customer or SM interactions), which were broken down and explained so that team members and other stakeholders could understand what needed to be done and why such goals should be achieved.
Auxiliary Resource	In HOLDCOY, middle management served as *Auxiliary Resources*. They acted as additional support to fill resource gaps in the agile team by taking up other roles and duties when people were lacking, thereby helping to prevent lapses that may adversely affect team productivity and project delivery.
Motivator	As *Motivators*, MMs in each case motivated their teams by inspiring, encouraging, and influencing team members for successful project execution. MMs (e.g., P11) motivated by trusting other teammates, and giving them autonomy by allowing them contribute to making project-related choices so that the team could achieve shared success. P1 motivated teammates by providing incentives, such as supporting staff recognition and promotion for good performance. P7 was involved in organising team bonding activities to keep team members motivated, relaxed, and reinvigorated to tackle project commitments.
Product Owner	MMs were the *Product Owners* in the two cases. They were accountable for maximising value by prioritising requirements, tasks, and releases in collaboration with other project stakeholders so that most valuable requirements were completed first. P1 was the product owner in HOLDCOY, while P12 was the product owner in BANKCOY. P1 represented project stakeholders (e.g., customers) to ensure his team worked with the needs of stakeholders in mind, thereby ensuring continuous alignment between team outputs and stakeholder expectations during project execution. P1 along with P6 and P7 engaged in developing product vision through product and project road mapping jointly with SM (P9). P1 and P6 managed the backlog.
Subject Matter Expert	As *Subject Matter Experts*, MMs provided input and expertise on technical and non-technical aspects of the ASD projects in the two cases: technical development (P6 and P13), IT networking (P1), industry domain expertise (P1, P6 and P7), and project work and status information, based on their advanced knowledge, experience or both for successful project delivery.
Foreseer	In HOLDCOY, MMs acted as *Foreseers*. They could see the bigger picture and foresee issues that might affect project delivery. P1 said: *"I know what a delay, what a delay can cause and how a delay can affect things. So, I always look at the bigger picture... when you look at the bigger picture you know that a delay can cause a very long, a very long issue at the end of the day".*
Mediator	In BANKCOY, P12 acted as a *Mediator*. He intervened as a middleman to help resolve a conflict between warring members of the agile project team by helping to bring about a settlement.

Fig. 3. Planning and coordination for project alignment and execution MM roles

requirements in iterations and increments so as to quickly release good-quality software that meet stakeholder expectations. In the two cases, middle management led the respective ASD teams as *Agile Leaders* and *Technical Leaders*.

As *Agile Leaders*, middle management ensured the agile project teams implemented their projects in accord with the agile approach (P1, P6, and P11). They helped to keep the agile project teams current regarding technologies they adopted for project delivery by showing interest in technology trends and keeping up to date with technologies being used in industry (P6 and P18). They encour-



Middle Manager Role	Description
Process Owner and Improver	MMs were *Process Owner and Improvers* in the two projects. They were accountable for implementation of prescribed PG processes and procedures in the ASD projects. Middle management facilitated retrospectives (P1), as well as process and procedural changes (P14 and P18) for continuous improvement, ensuring inefficiencies and areas for improvement in PG processes and procedures were identified and addressed in collaboration with other stakeholders. P1 collaborated with the HOLDCOY OpEx team (which was responsible for documenting PG policies and processes, and monitoring company-wide compliance) to ensure PG policies and processes were always appropriate for the agile project team's day-to-day project work. P1 also ensured his agile team complied with PG processes and rules to avoid penalties due to noncompliance.
Auditor	In HOLDCOY, P8; an external MM to the agile project team, acted as *Auditor*. P8 was the OpEx team's manager. P8 and OpEx team performed process and function audits to identify gaps and areas to improve in PG processes and policies for continuous improvement in order to support each function and project work. P8 commented: *"what we do currently is on a monthly basis also we do like a process audit for each function… in each of our policies that governed the activities performed by any resource, we have what we call effectiveness criteria where we test the effectiveness of the person carrying out this…, a particular activity right. We identify gaps, things that we need to…, that we need to improve upon in the activity or in the process"*.
Innovator	As *Innovators*, MMs in both cases fostered innovation and change to improve PG practice in the ASD projects so that the teams could implement the projects more efficiently to achieve expectations. MMs were involved in recommending and introducing new ideas, practices and technological tools to improve and advance project delivery. In HOLDCOY, P3 said: *"the CTO [Head of Technology and Scrum Master] and the COO [Head of Operations] they are actively involved in determining who does what and then how it's being done, the technological tools to be used like I explained earlier. So, they play a very important role in that, and then if at any point in time the tools you are making use of, the technologies are not better or there is a better option, they are the ones that suggest that 'Okay, try out these better options'"*.
Rule-Maker	The MMs acted as *Rule-makers*. They formulated, introduced, and enforced PG rules and policies (e.g., customer collaboration rule, testing policy, information security standards) that helped the agile project teams to work in compliance with prescribed governance measures. P2 commented: *"on a Monday morning, you don't want to go to the bank and run an implementation on a Monday morning. They [banks] also have what they are also trying to achieve in the banks. So on Mondays and mostly Fridays we don't go to banks, so they [MMs] are the ones that came up with that"*. P8 commented: *"I'm in charge of formulating the policy that govern that activity of testing"*. Middle management also maintained custody of PG policies: *"I'm a custodian of polices: [TECHCOY] policies and policies that govern what a function does, the activities a function performs"* (P8).

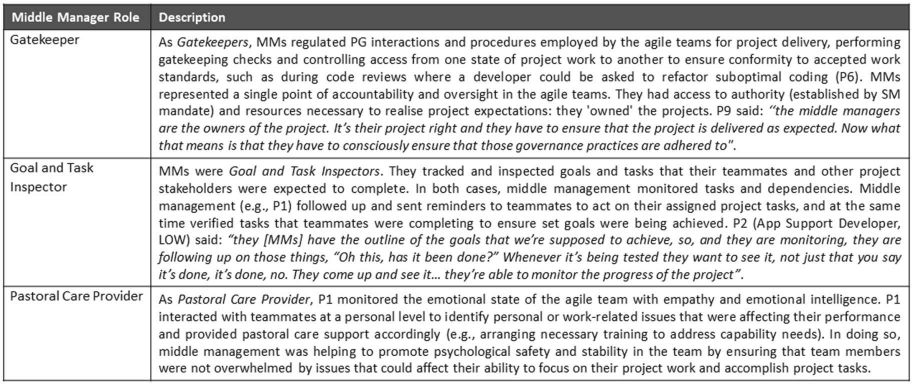
Fig. 4. Continuous improvement and organisational change MM roles

aged shared decision-making (P6 and P11). P6 exercised business sense through his appreciation and understanding of the business opportunities associated with the ASD project, thereby helping to bring clarity of such opportunities to the agile project team—opportunities for the company to quickly introduce a new product to customers through agile delivery and gain advantage over competitors. P1 helped his team to maintain agility by adapting weekly work approaches when necessary to ensure the team achieved project goals. The MMs engaged team members with a listening ear and emotional intelligence to ascertain work situations and personal issues that might affect project delivery (P1 and P6).

As *Technical Leaders*, MMs (P6 and P13) provided technical leadership by leading software development in the projects, supporting the agile teams with advanced technical expertise and hands-on support. P6 ensured work completed by developers were within project scope and aligned with project expectations. He ensured technology requirements to accomplish the project were identified and provisioned, ensuring that all necessary technical considerations for development were made in order to achieve expected results. P13 ensured alignment between BANKCOY and external vendor technical specifications for their project.

4.4 Monitoring

The MMs monitored project work and team members' performance in the PG activity as *Gatekeeper*s, *Goal and Task Inspectors*, and *Pastoral Care Providers* to ensure the agile project team members accomplished assigned project tasks and goals as required with healthy state of mind (see Fig. 5).

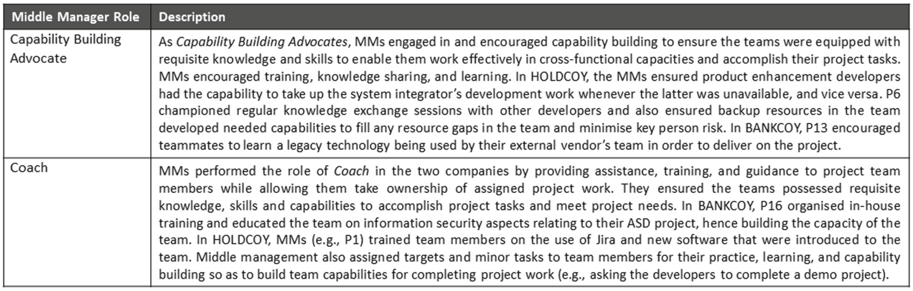


Middle Manager Role	Description
Gatekeeper	As *Gatekeepers*, MMs regulated PG interactions and procedures employed by the agile teams for project delivery, performing gatekeeping checks and controlling access from one state of project work to another to ensure conformity to accepted work standards, such as during code reviews where a developer could be asked to refactor suboptimal coding (P6). MMs represented a single point of accountability and oversight in the agile teams. They had access to authority (established by SM mandate) and resources necessary to realise project expectations: they 'owned' the projects. P9 said: *"the middle managers are the owners of the project. It's their project right and they have to ensure that the project is delivered as expected. Now what that means is that they have to consciously ensure that those governance practices are adhered to"*.
Goal and Task Inspector	MMs were *Goal and Task Inspectors*. They tracked and inspected goals and tasks that their teammates and other project stakeholders were expected to complete. In both cases, middle management monitored tasks and dependencies. Middle management (e.g., P1) followed up and sent reminders to teammates to act on their assigned project tasks, and at the same time verified tasks that teammates were completing to ensure set goals were being achieved. P2 (App Support Developer, LOW) said: *"they [MMs] have the outline of the goals that we're supposed to achieve, so, and they are monitoring, they are following up on those things, "Oh this, has it been done?" Whenever it's being tested they want to see it, not just that you say it's done, it's done, no. They come up and see it… they're able to monitor the progress of the project"*.
Pastoral Care Provider	As *Pastoral Care Provider*, P1 monitored the emotional state of the agile team with empathy and emotional intelligence. P1 interacted with teammates at a personal level to identify personal or work-related issues that were affecting their performance and provided pastoral care support accordingly (e.g., arranging necessary training to address capability needs). In doing so, middle management was helping to promote psychological safety and stability in the team by ensuring that team members were not overwhelmed by issues that could affect their ability to focus on their project work and accomplish project tasks.

Fig. 5. Monitoring MM roles

4.5 Capability Building

MMs were found to contribute towards the capability building and competence development of members of the agile project teams in the two cases. They did so by assuming the *Capability Building Advocate* and *Coach* roles (see Fig. 6).



Middle Manager Role	Description
Capability Building Advocate	As *Capability Building Advocates*, MMs engaged in and encouraged capability building to ensure the teams were equipped with requisite knowledge and skills to enable them work effectively in cross-functional capacities and accomplish their project tasks. MMs encouraged training, knowledge sharing, and learning. In HOLDCOY, the MMs ensured product enhancement developers had the capability to take up the system integrator's development work whenever the latter was unavailable, and vice versa. P6 championed regular knowledge exchange sessions with other developers and also ensured backup resources in the team developed needed capabilities to fill any resource gaps in the team and minimise key person risk. In BANKCOY, P13 encouraged teammates to learn a legacy technology being used by their external vendor's team in order to deliver on the project.
Coach	MMs performed the role of *Coach* in the two companies by providing assistance, training, and guidance to project team members while allowing them take ownership of assigned project work. They ensured the teams possessed requisite knowledge, skills and capabilities to accomplish project tasks and meet project needs. In BANKCOY, P16 organised in-house training and educated the team on information security aspects relating to their ASD project, hence building the capacity of the team. In HOLDCOY, MMs (e.g., P1) trained team members on the use of Jira and new software that were introduced to the team. Middle management also assigned targets and minor tasks to team members for their practice, learning, and capability building so as to build team capabilities for completing project work (e.g., asking the developers to complete a demo project).

Fig. 6. Capability building MM roles

5 Discussion

We have undertaken a multiple-case study to answer the question - *What are the roles of middle managers in agile project governance?* The previous section described results from two cases, which suggest that MMs performed 25 pivotal roles in agile PG. This section will discuss findings in light of related work.

Comparing our model with the agile PG framework in Lappi et al. [23], the MM roles and categories are represented in the six dimensions, albeit not in the same grouping; for instance, coordination (e.g., coordinator), capability building (e.g., coach), monitoring (e.g., goal and task inspector), goal setting (e.g., goal definer and interpreter), roles and decision-making power (e.g., decision-maker), and incentives (e.g., motivator). Our agile and technical leadership category fits into the roles and decision-making power dimension, in which Lappi et al. [23] highlights the adaptive nature of leadership provided by an agile project manager which is needed to handle seemingly increasing workload due to risks and greater coordination needs in autonomous teams. As an adaptive leader, the project manager also serves as coordinator or administrator for the agile project team [23]. This role interchange behaviour is similar to that of MMs in our study.

Regarding continuous improvement and organisational change in our cases, MMs facilitate innovation, rule-making, auditing, process and procedural changes, and retrospectives. These mechanisms allow the project teams to review and reflect on how they operate and devise and implement improvements and strategies to address inefficiencies in their work processes, thus affecting not only their projects, but also PG practice in the organisations as a whole. Our MMs roles highlight the pertinence of continuous improvement and organisational change to agile PG. While Lappi et al. [23] categorises retrospectives as a mechanism within the coordination dimension, our study posits continuous improvement and organisational change as a possible dimension of agile PG warranting further research. A hallmark of agility is the continuous affinity for and responsiveness to change [10]. This should also reflect in the way agile PG is exercised. From our study, MMs facilitate continuous improvement [15] and change [1,5], hence contributing to a culture of PG in ASD projects that is not rigid and static, but one that is dynamic and mutative: constantly evolving so as to remain effective.

From our study, middle managers (MMs) tend to switch between roles to cater for project needs that are occasioned by project events. There can be one or more MMs performing the same middle management role regardless of their job titles, which is how agile managers tend to operate in agile projects [30]. This dynamic, instantaneous, and transitory nature of the MM roles in agile teams during agile PG is characteristic of roles found in self-organised teams [17].

Gatekeepers, such as the MMs in our cases, are viewed as "organizational actors that sit at the junction of a number of communication channels in such a way that they can regulate the flow of demands and potentially control decision outcomes" [14, p. 11]. Hence, a gatekeeper is essentially an entity that controls 'who' or 'what' is given access to something, or one that controls the advancement of a thing from a particular state or condition to another. In Russo [29, p. 30], the MMs (Scrum masters and product owners) were collectively designated the "gatekeepers between the top management directions and the implementation efforts". The Scrum masters in particular "acted as gatekeepers, focusing on Agile values" [29, p. 29], which is related to the *Agile Leader* MM role in our study and the agile manager mentor role in Shastri et al. [30] in that the three roles ensure project delivery follows the agile approach. The Scrum masters were also domain experts [29], similar to our *Subject Matter Expert* role. The product owners represented stakeholders and ensured software outputs matched user expectations [29]. This is similar to our *Product Owner* role.

In our study, middle management as a collective 'owned' the projects and acted as single point of accountability and oversight, ensuring tasks were completed by the right people to achieve stakeholder expectations and best project outcomes. This is closely related to the 'single point of accountability' PG function in agile settings [25]. Moran [25] argues that ultimately, any agile undertaking (e.g., project) must be traced back to a single person who has access to the necessary resources and authority to direct activities and can be held accountable for performance and outcomes. Despite being project owners by SM mandate, the MMs worked alongside their teammates with a shared project ownership and team autonomy mindset. For example, P1 believed that for their agile project to succeed, each person in the agile team had to own the project, as well as own their respective project tasks: *"the only way an agile project can succeed is if your team members actually own this project and own each task"* (P1).

As *Strategists*, the MMs contributed to strategy making and implementation efforts within the two companies, as in Balogun [5], which argues that MMs are enabling and influential in defining and implementing strategy in organisations due to their intermediary position. This also links with the *Coordinator* role in our work in that MMs are intermediaries. As *Coordinators*, the MMs in our study coordinated the agile teams' interactions with internal and external stakeholders for optimal collaboration to achieve shared project goals. This is similar to an aspect of the agile manager coordinator role in Shastri et al. [30], where the agile manager coordinates team collaboration with customers and specialists, as well as collaboration within and between teams. The boundary spanning position of the MMs in our study gives them access to knowledge from across intra- and inter-organisational boundaries, thus providing substantial intelligence for generating and implementing useful ideas. Projects are apparatus in organisations that enable transformation of business ideas and strategies into achieved goals. In agile settings, weak strategic connections between organisations and their projects is a PG issue [23]. Our study suggests the strategic and coordination agency of MMs may potentially help strengthen organisation-project strategic connections in agile settings considering middle management's frequent participation in strategic and technical-operational multistakeholder exchanges.

A few other MM roles we found match other findings in Shastri et al. [30]. For example, in our *Coach* role, MMs train teammates on new software tools for project work. They provide guidance and assistance while allowing teammates to own their project tasks. The MMs also assign minor tasks to teammates to build their know-how and aid their growth. This is on par with the coaching aspect of the mentor role in [30], which entails guiding and assisting teammates to complete tasks, and aiding their growth by giving them minor tasks to complete. The mentor role also builds team relations using different means, including organising team bonding activities. This is close to our *Motivator* role whereby MMs support and organise team bonding activities to inspirit teammates. It is, therefore, noteworthy that as multirole actors, MMs are vital to ASD projects and teams. Our study and other recent studies [1,15,29] call attention to the relevance and evident potential of MMs in present-day agility landscape.

As for limitations, we acknowledge our study involved a short period of fieldwork. This was due to COVID-19 pandemic. Still, useful data was collected leading to the discovery of 25 roles of MMs in agile PG. The nature of qualitative studies is subjective, however, our use of multiple data sources for corroboration strengthens validity of findings. The two case studies are limited to companies in Nigeria and the finance industry. The finance industry is an intensely regulated industry. The sensitive nature of business activities in such industry may demand a certain degree of oversight and control, which may influence how governance is performed and how MMs operate in ASD projects within such contexts. The small number of companies involved may limit generalisability of findings to our two cases. Nonetheless, the companies we studied are representative of companies that use agile approaches, hence companies with like contexts, structures, and projects may derive instructive insights from our research.

6 Conclusion and Future Work

Our study suggests that MMs are important to agile PG. As conspicuous and influential actors in agile teams, MMs perform a variety of pivotal roles through which they contribute to agile PG practice and support the effectual functioning of agile teams, thereby helping to accomplish mandated ASD projects.

This study has developed a thematic model of MMs' roles in agile PG that describes multiple roles, which MMs can perform when working alongside agile teams and governing ASD projects. It contributes to the 'middle management in agile' debate in hopes of prompting scholarly discussions on the topic. It contributes to filling a gap in knowledge as to the spectrum of middle management involvement and impact in agile PG and agile teams by offering alternate, clarifying, and optimistic views about the middle management role. It adds to studies on agile PG and MMs in ASD projects, which are limited. The study exemplifies the use of activity theory in agile PG research through its application of the APGov framework, and advances the use of activity theory in ASD research.

Organisations that use agile methods and have MMs may use the model of MMs' roles as a tool for (a) creating job descriptions and person specifications for recruitment of MMs, (b) education and training for continuing professional development of MMs and aspiring MMs, and (c) ensuring MMs maintain acceptable levels of job performance in the governance of ASD projects. The model should help MMs, SM teams, aspiring MMs, agile teams, and researchers to better understand the roles of MMs in agile PG practice, which may lead to stronger organisation-project strategic connections and project success, as well as foster organisational agility, better working relationships between MMs and their teammates in agile project teams, and further research. We encourage SM teams to involve agile MMs in strategic exchanges as they may possess unique technical-operational knowledge and insights regarding project work and complexities on the ground. Participation of MMs in strategic exchanges with SM can reinforce project teams' commitment, dedication, and ownership of ASD projects to ensure mission-critical initiatives are realised with short time to value.

Future work should further explore continuous improvement and organisational change as a PG dimension in ASD projects. Also, the roles of MMs in PG within additional ASD projects in finance, other industries, and other countries should be examined—with larger sample size—to validate, generalise, or build upon our findings. To further validate our findings, quantitative research is also suggested (e.g., determine the relative importance of the MM roles in agile PG).

Acknowledgements. We thank the Agile Research Network for funding this study.

References

1. Annosi, M.C., Foss, N., Martini, A.: When agile harms learning and innovation: (and what can be done about it). Calif. Manage. Rev. **63**(1), 61–80 (2020)
2. APM: APM Body of Knowledge, 6th edn. Association for Project Management (2012)
3. Attride-Stirling, J.: Thematic networks: an analytic tool for qualitative research. Qual. Res. **1**(3), 385–405 (2001)
4. Aucoin, P.: Middle managers-the crucial link: discussion summary. Can. Public Adm. **32**(2), 187–209 (1989)
5. Balogun, J.: From blaming the middle to harnessing its potential: Creating change intermediaries. Br. J. Manag. **14**(1), 69–83 (2003)
6. Barroca, L., Dingsøyr, T., Mikalsen, M.: Agile transformation: a summary and research agenda from the first international workshop. In: Hoda, R. (ed.) XP 2019. LNBIP, vol. 364, pp. 3–9. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30126-2_1
7. Benbasat, I., Goldstein, D.K., Mead, M.: The case research strategy in studies of Information Systems. MIS Q. **11**(3), 369–386 (1987)
8. Cheng, H., Song, F., Li, D.: How middle managers' participation in decision-making influences firm innovation performance: evidence from china employer-employee survey data. Chin. Manage. Stud. **11**(1), 72–89 (2017)
9. Ciesielska, M., Boström, K.W., Öhlander, M.: Observation Methods. In: Ciesielska, M., Jemielniak, D. (eds.) Qualitative Methodologies in Organization Studies, pp. 33–52. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-65442-3_2
10. Conboy, K.: Agility from first principles: reconstructing the concept of agility in information systems development. Inf. Syst. Res. **20**(3), 329–354 (2009)
11. Crawford, K., Hasan, H.: Demonstrations of the activity theory framework for research in information systems. Australas. J. Inf. Syst. **13**(2), 49–68 (2006)
12. 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)
13. Gregory, P., Barroca, L., Sharp, H., Deshpande, A., Taylor, K.: The challenges that challenge: engaging with agile practitioners' concerns. Inf. Softw. Technol. **77**, 92–104 (2016)
14. Heiskanen, A., Similä, J.: Gatekeepers in the action structure of software contracting: a case study of the evolution of user-developer relationships. ACM SIGCPR Comput. Pers. **14**(1–2), 30–44 (1992)

15. Hermkens, F.J., Romme, A.G.L., Dolmans, S.A., et al.: An exploratory study of middle manager's roles in continuous improvement. Int. Bus. Res. **13**(5), 9–30 (2020)
16. Highsmith, J.: Agile project governance: the evolution of phase-gate 1. INSIGHT **17**(2), 23–26 (2014)
17. Hoda, R., Noble, J., Marshall, S.: Self-organizing roles on agile software development teams. IEEE Trans. Softw. Eng. **39**(3), 422–444 (2013)
18. Iyamu, T.: A case for applying activity theory in IS research. Inf. Resour. Manage. J. (IRMJ) **33**(1), 1–15 (2020)
19. Kalenda, M., Hyna, P., Rossi, B.: Scaling agile in large organizations: practices, challenges, and success factors. J. Softw. Evol. Process **30**(10), e1954 (2018)
20. Karanasios, S.: Framing ICT4D research using activity theory: a match between the ICT4D field and theory? Inf. Technol. Int. Dev. **10**(2), 1–17 (2014)
21. Kujala, J., Aaltonen, K., Gotcheva, N., Pekuri, A.: Key dimensions of project network governance and implications to safety in nuclear industry projects. In: EURAM 2016: Manageable Cooperation? (2016)
22. Lappi, T.: Digitalizing Finland: Governance of government ICT projects. Ph.D. thesis, University of Oulu, Finland (2019). http://urn.fi/urn:isbn:9789526222899
23. Lappi, T., Karvonen, T., Lwakatare, L.E., Aaltonen, K., Kuvaja, P.: Toward an improved understanding of agile project governance: a systematic literature review. Proj. Manage. J. **49**(6), 39–63 (2018)
24. Moe, N.B., Stray, V., Hoda, R.: Trends and updated research agenda for autonomous agile teams: a summary of the second international workshop at XP2019. In: Hoda, R. (ed.) XP 2019. LNBIP, vol. 364, pp. 13–19. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30126-2_2
25. Moran, A.: Managing Agile. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16262-1
26. Nwohiri, A., Sonubi, F.: Deploying data mining techniques to gain deeper insight into Nigerian customers' financial activities. Niger. J. Technol. **39**(2), 553–561 (2020)
27. Parker, D.W., Holesgrove, M., Pathak, R.: Improving productivity with self-organised teams and agile leadership. Int. J. Prod. Perform. Manage. **64**(1), 112–128 (2015)
28. PMI: Governance of Portfolios, Programs, and Projects: A Practice Guide. Project Management Institute (2016)
29. Russo, D.: The agile success model: a mixed-methods study of a large-scale agile transformation. Trans. Softw. Eng. Methodol. **30**(4) (2021). https://doi.org/10.1145/3464938
30. Shastri, Y., Hoda, R., Amor, R.: Understanding the roles of the manager in agile project management. In: Proceedings of the 10th Innovations in Software Engineering Conference, pp. 45–55 (2017)
31. Talby, D., Dubinsky, Y.: Governance of an agile software project. In: 2009 ICSE Workshop on Software Development Governance, pp. 40–45. IEEE (2009)
32. Uwadi, M.C.: Multiple roles of middle managers in agile project governance: an activity theory perspective. In: Gregory, P., Kruchten, P. (eds.) XP 2021. LNBIP, vol. 426, pp. 195–200. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88583-0_19
33. Walsham, G.: Interpretive case studies in IS research: nature and method. Eur. J. Inf. Syst. **4**(2), 74–81 (1995)

Building a Toolbox for Working with Psychological Safety in Agile Software Teams

Mikkel Agerlin Christensen(✉) and Paolo Tell

IT University of Copenhagen, Copenhagen, Denmark
magc@itu.dk, pate@itu.dk

Abstract. This paper presents the design of eight tools for working with psychological safety in agile software teams, which were designed in collaboration with industry practitioners using design science. The tools were adopted over a two-week period by four Danish industry software teams and evaluated through team interviews and surveys. Results show that the designed tools can be successfully adopted and integrated in the practices of a software team. Participating teams found the tool format valuable, as it allowed them (i) to engage in discussions they were not always capable of having, (ii) to find the right shared vocabulary to frame these discussions, and (iii) to provide them with needed prompts to let such discussion surface. Finally, teams unanimously reported interest in the continued use of the designed tools.

Keywords: Psychological safety · Agile · Teams · Design science

1 Introduction

In 1999, Edmondson published her seminal work on psychological safety [6], defining the term as “a shared belief held by members of a team that the team is safe for interpersonal risk taking” and laying the foundation for future research on the subject. Edmondson found that psychological safety existed in most interpersonal interactions, and that psychological safety was a key component in team learning and innovation. Fifteen years later Google found psychological safety to be the most important predictor of team effectiveness [5]. Though psychological safety is important to understand and attend to, it is hard to measure, and even harder to improve. Research on measurements of psychological safety have been published in the medical domain [16], but research on affecting change of psychological safety is sparse, especially in the domain of software engineering [12].

Due to this sparsity, a previous 6-month case study in a Danish software company was conducted [1], replicating survey and observation methods used to measure levels of psychological safety in the medical domain [16]. “Triggered by an industry [...] need that can be addressed by developing an artifact” [17], the

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 82–98, 2022.
https://doi.org/10.1007/978-3-031-08169-9_6

study included an initial exploration focused on the design of an intervention through a workshop to affect change in levels of psychological safety.

To reduce the gap identified in [1] and [5], among others, and improve the understanding of viable practices for working with psychological safety in software teams, this paper aims to further explore this topic through the use of the methodology presented by Peffers et al. [17] for conducting design science research, hence, by creating, evolving, and evaluating artifacts (*tools*) to assist and enable teams to work with psychological safety. In this paper, the term *tool* refers to the tangible, descriptive representation of an intervention activity designed to affect change (intervene) on levels of psychological safety. When referring to such *tools*, the following italicized format will be used: *tool*.

This paper presents the design and production of a toolbox comprising eight such *tools*, which can be selected and adopted by teams wishing to incorporate working with psychological safety into their practice. Four software teams participated in evaluations after implementing a selection of *tools* over a two weeks period. Through the design and evaluation of these *tools*, this paper aims to answer the following research questions:

RQ1: How can interventions on psychological safety be designed as actionable *tools* which enable agile software teams to work with psychological safety as part of their practices?

RQ2: To what extent can *tools* aid agile software teams in working with psychological safety?

The remainder of this paper is structured as follows: Sect. 2 presents related work, while Sect. 3 presents the method used to develop and evaluate the *tools*. Section 4 presents the design and evolution of the *tools*. Finally, results are presented in Sect. 5, discussed in Sect. 6 and concluded on in Sect. 7.

2 Related Work

Google's study "Aristotle" found psychological safety to be the number one predictor of team effectiveness across 180 international teams [5]. Additionally, Google's 2019 "State of Dev-Ops" named a "Culture of Psychological Safety" a major contributor to "organizational performance, and productivity, showing that growing and fostering a healthy culture reaps benefits for organizations and individuals" [10], a result found independently through the application of two separate research models. Of the five key dynamics found to be significant (psychological safety, dependability, structure and clarity, meaning of work, impact of work) they found that "Psychological safety was far and away the most important of the five dynamics we found – it's the underpinning of the other four" [5]. This indicates that, despite the lack of research on the application in the domain of software, its importance is well-established.

Measuring Levels of Psychological Safety. Several attempts have been made at quantifying psychological safety in the medical domain. In particular, research has been done by O'Donovan et al. on both measuring [16] and intervening [14] on psychological safety. The work of O'Donovan et al. in [16] developed a method to measure levels of psychological safety in teams, which was replicated in the pre-study [1]. This method of data gathering was designed specifically to inform interventions on psychological safety. This method was replicated in the pre-study [1], in which explorative work on measuring and affecting change on levels of psychological safety within software teams was conducted in a 6-month project with two teams from a Danish software company. In [1], the survey and observation methods for measuring psychological safety from [14,16] were applied within the software domain, in order to measure the effects of intervening on psychological safety.

Intervening on Psychological Safety. In the pre-study, the measurements of O'Donovan et al. [16] were used to measure levels of psychological safety before and after an intervention workshop aiming to heighten the awareness of psychological safety within the participating teams. While the project's explorative (and short) nature was only an initial step towards the improvement of psychological safety, several of the lessons learned motivated this paper. Specifically, the workshop showed that awareness alone could act as an intervention on psychological safety, something which became an early inspiration for the *tool* concept. The measurement techniques used, while applied successfully to the domain of software, were deemed more appropriate for continued measurements over longer periods of time, and as such will not be re-used in this paper, given its short and exploratory nature. O'Donovan et al. also analysed outcomes of interventions to improve psychological safety in [15]. Herein they concluded that the reviewed attempts on improving psychological safety had mixed results, in part identifying that "multifaceted interventions may allow future studies to further investigate the efficacy or effectiveness of these interventions." [15]. The *tools* designed in this paper explore such multifaceted intervention, with the intent of investigating their effectiveness within software teams.

3 Method

This paper expands on preliminary work [1] and, by following design science research guidelines (i.e., Hevner et al. [11], Peffers et al. [17], and Wieringa [20]), aims to answer the research questions providing knowledge supporting the design of solutions in the form of artifacts to real-world either *construction* or *improvement* problems [3].

Figure 1 depicts the four cycles followed for artifact (*tool*) design and the mapping of steps of the design science process proposed by Peffers et al. [17]. These cycles and the related process steps (depicted as squares) are detailed in chronological order in the following. Process steps involving external input, such a workshop with participants, are marked with a triangle corner. Artifact versions from each Cycle are depicted by the rounded squares at the top of each Cycle, with the details of their evolution being presented in Sect. 4.

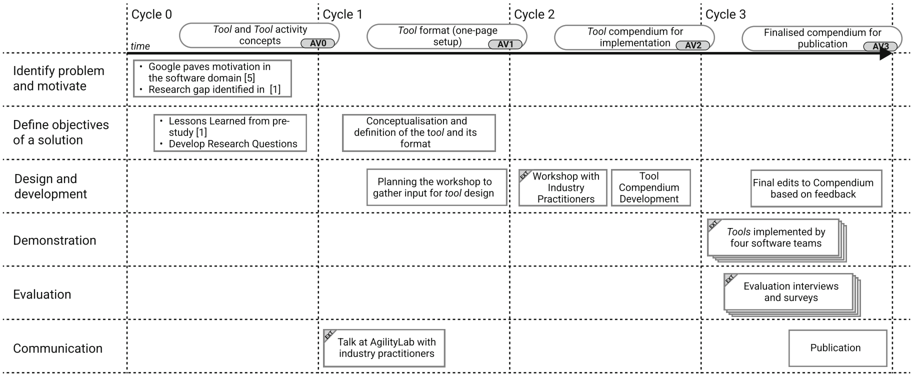

Cycle 0
Cycle 1
Cycle 2
Cycle 3
time
Tool and Tool activity concepts
AV0
Tool format (one-page setup)
AV1
Tool compendium for implementation
AV2
Finalised compendium for publication
AV3
Identify problem and motivate
• Google paves motivation in the software domain [5]
• Research gap identified in [1]
Define objectives of a solution
• Lessons Learned from pre-study [1]
• Develop Research Questions
Conceptualisation and definition of the tool and its format
Design and development
Planning the workshop to gather input for tool design
Workshop with Industry Practitioners
Tool Compendium Development
Final edits to Compendium based on feedback
Demonstration
Tools implemented by four software teams
Evaluation
Evaluation interviews and surveys
Communication
Talk at AgilityLab with industry practitioners
Publication

Fig. 1. Project activities based on the model proposed in [17]

Cycle 0. Initiated in [1], and leading to an objective-centered entry point (i.e., “triggered by an industry or research need that can be addressed by developing an artifact“[17]), the cycle led to: the identification of the main challenges, the analysis of the motivation to solve these, and the objectives of a potential solution. This cycle founded the research questions and the insight of using a designed artifact to solve the challenges, with motivation drawn from the identified gap in research, and the insights provided by Google in [5].

Cycle 1. Cycle 1 initiated early industry engagement through a talk held at a virtual Danish meetup designed to raise interest among local practitioners. The core concepts of psychological safety and the results from [1] were presented to 70 attendants. Input was gathered through collaborative discussion activities held as part of the talk, as well as a following Q&A session. The meetup contained a call to sign up for Cycle 2’s workshop, which used the gathered input.

Cycle 2. Industry input for *tool* design continued through a digital workshop held on April 6th 2021 with 5 participants (14 sign-ups) comprising a mix of attendants from Cycle 1 and participants from industry. The goal of this workshop was to collect concrete experiences of psychological safety to inform *tool* design. It was conducted digitally using Zoom and Miro – an online collaborative white-board solution. Herein participants explored how psychological safety was experienced in their workplace, and proposed action points for making those experiences more psychologically safe. These points, and the discussions that emerged, became central to the design of *tools*. Concluding industry input collection, Cycle 2 resulted in the design of a *tool* compendium containing eight *tools* for working with psychological safety.

Cycle 3. The designed *tools* were evaluated to answer the research questions. Importantly, the subject of evaluation was the *tool* concept itself and the degree to which it aided the teams in working with psychological safety, not the success of each individual *tool* or their comparison.

Participating teams were recruited via calls to action distributed in several agile communities (e.g., AgilityLab: the host of the meetup from Cycle 1). These communities were primarily targeted for practical reasons: a large concentration of technical teams interested in processes and open to trying new ways of working. Four software teams from three different companies volunteered. Each team received a copy of the *tool* compendium, and chose their *tools* in an initial meeting with the researchers. Before implementation, teams were asked to conduct a shared viewing and open floor discussion of Edmondson's Ted Talk on Psychological safety [7], in order to establish a baseline understanding of the subject. Teams then implemented their selected *tools* autonomously over a two-week period, immediately followed by two forms of evaluation: A) anonymous individual surveys distributed to all members of participating teams (see Table 1), and B) one-hour semi-structured group interviews held virtually with all members of each team. Both types of evaluation aimed to evaluate the degree to which the *tools* had worked as successful intervention activities, as well as their success of aiding the teams in working with psychological safety. The group interview focused on collecting this evaluation in the same group construct in which the psychological safety of the participating teams existed. The individual surveys allowed individual team members to voice their feedback through a safe medium wherein candid feedback could be given, even if their experience was negative or differed from that of the team. Following the conclusion of the two types of evaluation, results were gathered and analysed. Group interview responses were grouped using thematic clustering and analyzed alongside survey responses. The results of this analysis are presented in Sect. 5.

Table 1. Tool evaluation survey questions

Code	Question
Q1	How psychologically safe do you feel your team was prior to using the tools?
Q2	I felt psychologically safe while using the tools
TQ1	I enjoyed using the tool
TQ2	While using the tool, I reflected on things that my team does not normally discuss
TQ3	Using the tool made it easier for my team to work with psychological safety
TQ4	I could see the tool fit in with the way we normally work

Note: All questions but Q1 were answered using a 5-point Likert scale; Q1 used a 7-point numerical scale for higher granularity. Questions TQ1-TQ4 were repeated for each tool.

4 Building the Toolbox: Input and Design

This section will present the evolution of the artifacts designed in this paper, namely the *tools* for working with psychological safety. As mentioned in the introduction, this paper uses the term *tool* to refer to the tangible, descriptive representation of an intervention activity designed to affect change (intervene) on levels of psychological safety within a team. Concretely, a *tool* describes: A) an intervention activity, B) How and when this activity should be carried out, C) Meta-data about the activity, such as its duration or setting, D) Prerequisites of the activity, E) The purpose of the activity, and finally F) The expected outcome of the activity. Importantly, as the *tools* were designed during the Corona pandemic, all *tool* activities were designed to function within the boundaries of distributed and virtual work environments. The set of *tools* designed in this paper are collected and described in a "toolbox", namely the *tool* compendium, which is publicly available at [2]. In this compendium, each *tool* is presented alongside a short example of the *tool* in use. This format allows for a tangible representation of the interventions on psychological safety to exist in an accessible, shareable format, designed to enable any team (participants in this paper or otherwise) to implement the *tools* autonomously without the researchers' involvement. This Section will not go into detail about the contents of each individual *tool*, but will rather aim to describe the four stages of artifact evolution through the four design science phases outlined in Sect. 3, by presenting the four resulting artifact versions shown in Fig. 1.

AV0 – The Tool Concept. *Tool* design was initialized by A) identifying the problem to be solved, namely the research questions put forward in Sect. 1, and B) defining the objectives of a solution to said problem; the designed artifacts (*tools* for working with psychological safety). Importantly, improving psychological safety was not a direct objective of the *tool* design process, which rather aimed to produce *tools* that enabled teams to work with psychological safety, potentially (hopefully) with the outcome of improving it. This distinction is important, as the improvement of psychological safety—a cultural change—is most likely to result from a team paying continuous attention to it over a longer period of time [9, Chapter 8]. It is therefore rather the aim that a *tool* successfully enables the team to work with psychological safety by creating a useful frame for this change process. Based on this objective, *tool* design began an iterative journey that continued throughout the following cycles. The initial inspiration began in the explorative work of the pre-study [1], wherein an early attempt at intervention on psychological safety was conducted. The learnings from implementing this intervention with industry software teams inspired both the problem to solve, and the artifacts designed to solve it. The goal of the design process was to synthesize research and industry experiences of psychological safety into an accessible but powerful set of intervention activities which teams could utilize to work with psychological safety, and to present these in a digestible format as *tools*. The word "tool" was chosen to present the activities as practical and tangible items as accessible as picking up a hammer to hammer in a nail. This was a core goal of *tool* design; using the *tools* should be as simple as possible, and should be compatible and useful regardless of a team's existing practices. This phase resulted in the first, early artifact version; the definition of a *tool* based on the objectives identified. As described earlier, *tools* were defined as tangible, descriptive representations of an intervention activity designed to affect change (intervene) on levels of psychological safety, allowing teams to pick up and implement them in their practice. The following three phases took this idea through iterative artifact design to realize this goal.

AV1 – Tool Definition and Format. To initiate *tool* design, the concept of psychological safety was broken down into several factors. Due to its complex nature, this would allow different *tools* to cover smaller subsets of the many aspects of psychological safety. This list of factors was synthesized by the researchers based on descriptions of psychological safety in Amy Edmondson's seminal work [6]. An additional factor of "awareness" (i.e., the awareness of the concept of psychological safety itself) was also added to this list, based on findings from the pre-study [1], in which an awareness workshop was conducted with positive results. The list of factors is presented in Table 2.

Table 2. Factors of psychological safety

Code	Factor	Description
F1	Awareness	The awareness of the concept itself
F2	Identification	Identifying instances in your own particular environment
F3	Asking questions	Model curiosity, moving towards a culture of questions and interaction
F4	Acknowledging mistakes	Being able to detect, speak openly, and reflect on the mistakes of the team or the individual
F5	Learning	Framing work as a learning problem, not an execution problem. Fostering a culture of experimentation
F6	Challenging the status quo	Challenging the "way things are", accepting constructive criticism, reflecting on existing process
F7	Voicing concerns or Ideas	Enabling team members to voice their concerns and ideas instead of holding back

These factors would stay prominent throughout the further design evolution of the artifacts. They would come to influence the design of *tools* in phase 2 (see AV2 below), but for AV1 the factors were used to design the next step of artifact evolution: the *tool* one-page format, containing fields for different metadata about the activity, such as when and why a team might use it, in addition to a description of the activity itself. This format was inspired by the "structure" concept of Liberating Structures, a collection of structures that provide "an alternative way to approach and design how people work together" [13]. The format was designed for use in an ideation workshop with industry participants, in which participants related the factors of psychological safety to their existing practice, and shared early ideas of intervention activities that were later used in *tool* design. The format used in this workshop additionally became the foundation for the presentation of *tools* in the final *tool* compendium.

AV2 – Tool Design & Tool Compendium. The second artifact version consisted of the design of the *tools* and their activities, based on the synthesis of collected input from industry and the research background of psychological safety. Industry input was gathered through the pre-study [1], a talk given at AgilityLab, and an ideation workshop with industry practitioners (see Sect. 3). Research input was drawn from literature on both psychological safety [6,8], as well as agile practices and methods [4,18]. Several *tools* were designed to be integrable with Scrum due to its popularity among agile practitioners. Eight *tools* were designed with the aim of covering the several aspects of psychological safety (see Table 2). Table 3 presents each of these *tools*, which factors of psychological safety they target, and where the inspiration for each *tool* was drawn from. For *tools* inspired directly by activities discussed in the *tool* workshop held with industry practitioners, the indicators WA (workshop activity) 1 through 5 are used. For *tools* wherein the inspiration was drawn directly from Edmondson's descriptions [8] of how to work with that particular factor of psychological safety, the codes from the psychological safety factor table (Table 2) are used, pre-fixed with an E (i.e. EF1 for Edmondson's descriptions of how to work with factor 1).

Table 3. The designed tools - Factors and inspiration

Tool	Factors	Inspiration sources
What's it to me?	F1, F2	Learning diary methods; Pre-study awareness workshop [1]
Checking In	F1, F3	Daily Scrum [18]; WA4; Edmondson's 7-step survey for psychological safety [9]
Celebrating mistakes	F2, F4, F5	EF4; WA5
Presenting the journey	F2, F4, F5	EF4, EF5; Scrum sprint demo [18]; WA1
Three questions	F3, F6, F7	EF3, EF7; Edmondson, impression management, risking appearing incompetent, intrusive and disruptive [8]
Acting on concerns	F2, F6, F7	EF2, EF6, EF7; Scrum retrospective [18]
The way things Are	F2, F6, F7	EF6; Scrum retrospective [18]
Meeting from hell	F2-F7	EF2-EF7; Liberating structures (Triz) [13]; Scrum from hell [19]

WA: Workshop Activity, F: factor of psychological safety (Table 2), EF: Edmondson's Description of working with these factors

Tools were designed to differ along several axes of a design space in order to improve understanding of how teams could work with *tools* for psychological safety, as well as to provide a rich toolbox of viable options for the many different practices of different teams. Each *tool*'s placement within the design space axes was communicated in the *tool* compendium using an iconography, allowing teams to choose the *tools* they saw fit. Four axes were chosen for the design space:

Setting. The setting axis had two options: team activity or individual activity. Outside of practical differences, team activity *tools* could be more confronting, but allow for group reflection within teams finding such a setting useful, whereas individual activities could be a safer starting point other teams, or provide more time for individual reflection. Importantly, a team activity does not imply a physical meeting.

Duration. A linear scale of expected time needed to carry out a given *tool*'s intervention activity. Durations listed in the tool compendium were estimates made during *tool* design, and existed mostly to provide teams some expectation of time investment required. Letting *tools* vary across the duration axis allowed for the design of significantly different types of *tools*, ranging from short-and-sweet questions for a team to discuss, to longer activity formats.

Frequency. The frequency axis indicated the frequency with which a *tool* was expected to be carried out, and had the following values along its axis: once, iteratively (i.e., with a cadence of e.g. a week or a Scrum sprint), daily, and any. A value of "any" meant that the *tool* was used in an ad-hoc fashion, such as the *tool* "Celebrating Mistakes", which involved addressing mistakes as they happened. Distributing *tools* along the frequency axis allowed for the design of *tools* that were either meant as incremental continuous improvement *tools*, to *tools* that were designed to be one-off conversation starters.

Table 4. Overview of tools including selections from the evaluating teams

Tool	Setting	Duration	Frequency	Required Comfort	Selected by teams
What's it to me?	Individual	10 min	Iteratively	1	1,2
Checking In	Group	15 min	Iteratively	2	2,3
Celebrating mistakes	Individual	10 min	Any	2	None
Presenting the journey	Group	15 min	Iteratively	1	None
Three questions	Group	5 min + 5/person	Iteratively	1	None
Acting on concerns	Group	20 min	Iteratively	1	3,4
The way things Are	Group	10 min + 3/person	Iteratively	1	1,2
Meeting from hell	Group	45 min	Once	3	1,3,4

Required Level of Comfort with Dissent. The "Required Level of Comfort with Dissent" axis (numerical, 1–3) indicated how high a team's comfort with dissent should be to achieve a constructive outcome from using the *tool*. While neither the scale nor a team's self-assessment are well-defined values, distributing *tools* along this axis allowed *tool* design to challenge different teams at different levels, with self-assessment and *tool* selection being at the discretion of the teams. Some *tools* were designed to be introductory and safe, while others were more challenging. Importantly, comfort with dissent is a separate concept from psychological safety, though the two are related. A team could struggle with some factors of psychological safety, such as voicing concerns or challenging the status quo, but still have a strong comfort with dissent whenever dissent occurs. Such a team might have mediocre psychological safety, but might still be in a position to get a constructive outcome from *tools* with a higher requirement of comfort with dissent.

An aim of this design process was to spread *tools* across the design space, providing both safe and challenging options that could fit different practices. The only area of the design space for which no *tools* were designed, was the combination of short duration and a high requirement for level of comfort with dissent. This design decision was made to avoid exposing teams to challenging activities without being given the proper time to engage and reflect. For the purposes of sharing the designed *tools* for implementation, they were collected in a single document; the *tool* compendium. This compendium contained all of the designed *tools*, as well as introductions to the concept of psychological safety and using the *tools*. The compendium was designed with the aim that any team could pick up the compendium and use the *tools* autonomously, without any interaction with the researchers. This version of the designed artifact—the *tool* compendium—was the final artifact version used in evaluation.

AV3 – Finalised Tool Compendium. During the evaluation of AV2, several points were brought up resulting in minor changes being made for future users of the *tool* compendium. Upon the conclusion of evaluation, it was also decided that the introductory activity of watching Edmondson's Talk on Psychological Safety [7] would be added as the ninth *tool*, giving future *tool* compendium users a similar introduction to the subject, as the one given to the participating teams in this paper. This is also supported by Google's similar recommendation of the talk in [5]. This final version (AV3) of the *tool* compendium can be found in [2].

5 Results

This section presents the results from *tool* evaluation. *Tools* were evaluated with four software teams of 9, 6, 4, and 3 members from three different SaaS companies working with variations of Scrum. Table 4 presents the characteristics of the designed *tools* and details which team selected them for implementation.

5.1 Survey Results

Table 5 presents survey responses, grouped as positive (agree + strongly agree), neutral, and negative (disagree + strongly disagree) responses, for ease of presentation. All teams reported a high level of psychological safety prior to using the *tools* (Q1 between 5.8 to 6.75, 7-point scale). Overall, teams expressed enjoyment (TQ1), positive reflection (TQ2), and engagement with psychological safety (TQ3) across all tested *tools* and were mostly positive regarding the likelihood of fitting the *tools* in their process (TQ4). A notable pattern in the results was the exposure to the *Meeting from Hell tool*. While for Team 1 the use of this *tool* was still generally positive, for Team 3 and 4, the use of *Meeting from Hell* was a negative experience and the majority of the negative responses received in the survey are related to these pairing of team and *tool*. Table 5 accounts for this pattern by presenting two versions of response data: TQx for the overall and TQx* disregarding answers of Team 3 and 4 in relation to Meeting from Hell.

Table 5. Survey answers

Answer	TQ1	TQ1*	TQ2	TQ2*	TQ3	TQ3*	TQ4	TQ4*
Positive	*58%*	*67%*	*71%*	*81%*	*64%*	*69%*	*56%*	*66%*
Neutral	*27%*	*31%*	*18%*	*15%*	*24%*	*25%*	*20%*	*23%*
Negative	*15%*	*2%*	*11%*	*4%*	*13%*	*6%*	*24%*	*11%*

Note 1: TQx* columns are presenting results disregarding the answers from Team 3 and 4 in relation to Meeting from Hell.
Note 2: N(TQx) = 55; N(TQx*) = 48.

5.2 Evaluation Interview Results

The evaluation group interviews were held with each participating team. Each session was annotated and recorded. Thematic clustering was used to analyse annotations and recordings, which led to the six themes presented below.

Aiding Teams in Working Towards Better Psychological Safety. Teams were extremely positive on this topic. Participants stated that the *tools* (with the exception of some experiences with the *tool Meeting from Hell* discussed later) they used enabled constructive discussions about psychological safety, which they might not have had otherwise. *"I think that it was good for the team. It made us discuss stuff that we don't usually discuss."* says team 2. While team 4 highlights how *"Acting on Concerns made us have a lot of good discussions [...] I feel like we talked about it in a new way. Hopefully it would have come up anyways, but it was good to get it out in the beginning of the project."*

Multiple teams also experienced process improvements during their participation. While this was not a direct goal of this paper, the ultimate purpose of improving psychological safety is that of team excellence, not just pleasant culture [8]. Team 1 reports that *"it has provided some efficiency to our meetings, and some afterthought to one self."* Team 2 continues: *"The result of our The Way Things Are was really good. It actually already feels like it's made a bit of a change in how we do our stand-ups. [. . .] I am actually confident now, that no one is sitting and struggling with something, because we actually mention it."*

Finally, participants indicated that the *tools* were engaging and functional team activities. According to team 2, *"[. . .] it's quite often that our discussion go more to one domain than the other [...] But actually, for all of our tries with the tools, I noticed that everyone participated, all the way through."*

Putting a Label on It. Several participants spoke to the concept of psychological safety being a label to several things they had either worked with or otherwise experienced in the past and that having a name for this concept was almost as helpful as the *tools* themselves. This finding is in line with the experiences of the awareness workshop conducted in [1], in which some participants experienced higher levels of psychological safety after awareness of the concept was spread within the participating teams. Team 2 confirms that they were *"really good conversation starters in the sense that it's not necessarily things that are easy to bring up normally, but putting it within a frame made it very easy to go about."* And also: *"have it named within a team, right. We talked about this, we talked that it's okay to bring it up"*. Interestingly, for team 3 *"it is clear that the idea of speaking about psychological safety is something we might want to do"*, and team 1 explains how while *"we are free to challenge things already, [. . .] I still think that [using tools] can be a good jump start for some people."*

Prompted with a Purpose. Another re-occurring theme among participants was that of simply taking the *tools* as a prompt to have a discussion, which they might already have been able to have, but were not having. One reason as to why the teams did not have these discussions, was described as trying to avoid appearing a certain way to your co-workers, something that Edmondson identifies as key reason why people hold back, namely because of impression management [9]. When prompted to purposefully engage in this kind of behaviour, participants expressed that this worry was easier to let go, especially when seeing other team members engage in similar behaviour. *"Sometimes"* – says team 4 – *"if you are speaking about concerns, you might seem like a grumpy old man that is only seeing issues and road blocks, but actually [pause] making this room where you map out all the different concerns, and see that other people have the* *same concerns, or talk about some of the things that you believe are concerns which is not a concern for others. I think it's just a great tool."*

Others simply had not found a space for these discussions, or did not know where to start. Team 4 says that *[the tool] is just great at facilitating and getting those questions asked.*, which is confirmed by Team 3 that states: *"Acting on Concerns is a great way to kind of create a space, where [psychological safety, concerns] is what you are speaking about. And that just provides insane amounts of value. That is at least how i experienced it with everyone."* Additionally, acknowledging that working with psychological safety was worth allocating time for was identified as another enabling factor. Team 1 reports how *"it was great to see that we take it seriously, that we look into psychological safety, that we put it on our agenda, and that we want to spend time on it."*

Does it Matter what Tools we Use? During interviews, several participants pondered whether the overall outcome of implementation could differ depending on the *tools* selected. While the concrete experiences with each *tool* differed, and some *tools* were preferred over others, several teams, like Team 1, expressed that *"it almost does not matter what tool you use"*, alluding to the strength of simply addressing the topic of psychological safety. This could indicate that, when the *tool* activity goes well, a successful *tool* leaves the focus to the team's self-reflection rather than the *tool* itself. However, as mentioned earlier, some teams (i.e., Team 3 and 4) did have negative experiences with one of their *tools*, *Meeting from Hell*. Team 3 described the *tool* as *"decidedly awkward"*, struggling with getting the discussion started as *"it requires a lot from the person hosting it"*, who needs to *"assume control for it to go well"*. Team 4 also reported that their negative experience might have been due to a *"wrong mix of personas"*. Given that Team 1 had a very different (positive) experience with *Meeting from Hell*, a poor fit between a team and the *tool* could explain a negative experience. Additionally, Team 3 and 4 being from the same company might have been related to their similar experience. Team 3 and 4 successfully implemented their other *tools* explicitly voicing their preference: *"I don't think that Meeting from Hell is a particularly bad exercise [...] but it didn't create a lot of value considering the time we spent on it, whereas Acting on Concerns created a lot of value and a great discussion and dialogue with less effort"* (Team 4).

The Impact of Existing Levels of Psychological Safety. The question of how a team's existing level of psychological safety might impact *tool* outcomes was discussed by several teams. Participants reflected on whether a team with a lower existing level of psychological safety would have benefited more than one with a very high level, and whether a team with a "high enough" level of psychological safety would benefit from using the *tools* in the first place. These discussions resulted in similar assessments across teams: *"discussion about [psychological safety] is never bad, even if [the level of psychological safety] might still be good beforehand"* (Team 2); Team 1 *"did not think that [psychological safety] was a big issue [but] it was great to see that we take it seriously, that we look into psychological safety, that we put it on our agenda, and that we want to spend time on it"*; and, Team 2 highlights that an individual might think *"'oh* *yeah, this place is super psychologically safe', when in reality my team members are just shitting themselves if they have to say anything."*

Future use of Tools. As the final step of the evaluation interviews, teams were asked if they could see themselves using their *tools* again in the future. All teams responded positively with at least one *tool* they would like to continue to use, while some teams identified wanting to use multiple. Team 1 describes how *"The Way Things Are was super. It is a good tool. [...] we could definitely [use it again]. And also Meeting from Hell. [...] I think I could see Meeting from Hell in a [company name] version, wherein you take it up once in a while."* Team 2 thinks that *"we should do another The Way Things Are. Not necessarily the next, like, week or month or anything, but eventually. I think that was a really fun experience. [...] I definitely think it could be interesting to try it again."* And, even more decisively regarding *Acting on Concerns*: Team 3 *"I am convinced that we will be using it again"*; and Team 4, *"I think it is just a great tool. It is definitely something we will use again, I believe, in all our big projects, actually."*

6 Discussion

This section will discuss the results presented in Sect. 5. Results are discussed per research question in the subsections below, followed by future work. Where survey results are referenced, two results are presented using the following format: 25% (35%), parallel to the format of the results presented in Table 5, showing results from TQx, and TQx* respectively.

6.1 RQ1: Designing Tools to Enable Agile Software Teams to Work with Psychological Safety as Part of Their Practice

This paper saw *tools* for working with psychological safety designed as the synthesis of research and industry input through an iterative process using design science (see Sect. 3). These *tools* were implemented and evaluated with four industry software teams. In evaluation surveys, 64% (69%) agreed that using the *tools* made it easier for their teams to work with psychological safety, and 58% (67%) enjoyed using the *tools*. For a potentially sensitive subject such as psychological safety, the teams enjoying using the *tools* is an important aspect of whether those *tools* can aid the teams in working towards better psychological safety, especially for continuous use. Evaluation interviews saw overwhelmingly positive responses, with participants identifying the *tools* as enabling them to have discussions they did not normally have, and finding it easier to speak up. Additionally, 56% (66%) reported that they could see the *tools* they used fit their existing practice. These results indicate that the designed *tools* were largely successful, answering the research question of how such *tools* can be designed; namely through the synthesis of research on psychological safety, and the experiences of industry practitioners, into bite-sized intervention activities, shared through one-page descriptions, using the *tool* format (see Sect. 4).

The *tool* concept itself seemed to provide a useful frame for working with psychological safety for the teams. The presented format and the design space created for the *tools* appeared to make the different *tools* understandable and easy to pick up and implement for the teams, with none of the teams having any facilitation being conducted by the researchers. This indicates that the *tool* concept was successful, and could be re-used for the design of future *tools*.

6.2 RQ2: Aiding Agile Software Teams in Working Towards Better Psychological Safety Through Tools

The primary aim of the designed *tools* was to aid software teams in working towards better psychological safety. While the *tools* themselves could not guarantee the improvement of psychological safety within the teams directly, *tools* were designed to make it easier for teams to achieve this goal by providing an enabling frame for the team to work within. Most participants 72% (81%) reported that using the *tools* caused them to reflect on things which their team did not normally discuss. From interviews, participants reported that they in some instances found it easier to speak up and voice their concerns during or after using the *tools*, and recounted experiences in which they had spoken up as a direct result of using a *tool*. Even teams that viewed themselves as having high psychological safety prior to using the *tools* reported that they felt more confident in their psychological safety after using the *tools* within their team. Several participants mentioned that thinking that your team has a high level of psychological safety is different to openly discussing and aligning individual perceptions with the team. Additionally, participants identified that the *tools* gave their teams a needed prompt to address unspoken subjects. Allocating the time to discuss these things as a team was deemed an important part of the successful experience, with some participants stating that they found the prompt and the time allocation even more impactful than the activities of the *tools* themselves.

All participating teams reported that they wanted to continue using one or more of their selected *tools* going forward, in order to continue working with psychological safety. This both indicates a positive experience using the designed *tools*, as well as an expressed interest in continuous attention being paid to psychological safety over time, using these *tools*. This outcome falls in line with Edmondson's descriptions of psychological safety requiring continuous renewal over time [9, Chapter 8], further indicating that the designed *tools* could continuously aid software teams on their journey of working with psychological safety.

6.3 Threats to Validity

Team Levels of Psychological Safety. In the evaluation surveys, all teams unanimously reported high existing levels of psychological safety. Given the strategy of recruiting from an agile community, this is not surprising. However, it raises the question of whether the success of the designed *tools* depends on the existing levels of psychological safety of the implementing teams. Objective measurements of psychological safety have had limited success [1,16], which renders existing levels of psychological safety an undefined metric for most teams. Even though the designed *tools* were distributed across a varied design space to accommodate for this uncertainty, allowing different options for different teams, the question of how teams with little to no psychological safety could initiate their journey with psychological safety was considered out of scope, as it was deemed likely to require a specific focus on such environments.

The Tool or The Toolbox? As mentioned in Sect. 3, the center point of both design and evaluation was the *tool* concept itself, its design space, and the degree to which *tools* implementing the concept could be integrated into the practice of software teams. As such, an active choice was made not to focus on the success or differences of the intervention activities of individual *tools*. This choice had several implications: *tool* selection was conducted with team/*tool* fit being prioritized over aiming for all *tools* to be evaluated. Additionally, the implementation of different *tools* among the individual teams likely resulted in differing experiences of individual *tools*. This is, however, a direct goal of the *tool* design; namely that of finding a way for software teams to work with psychological safety, regardless of *tool* selection, practice or implementation details. The *tools* were by design not prescriptive, aiming rather to provide guidelines for teams to engage with the concept of psychological safety, than exact rules of implementation or discussion. To this end, the evaluation shows that the designed *tool* concept is one useful way for software teams to work with psychological safety as part of their practice, potentially being a step towards bridging the gap identified in Sect. 1. Whether more successful *tools* can be designed within the design space, or indeed the design space itself can be improved, is a topic for future research.

Tool Implementation. The designed *tools* were implemented over a two-week period by the participating teams. While it is possible that a longer duration could provide richer data, the intent of this paper was to experiment with integrating working with psychological safety into the practice of agile software teams. For this reason, many of the *tools* were designed around common foundations of agile practices, such as iterative structures, and had their frequency of use in part defined by such iterations. As such, it was the aim to explore the insertion of the designed *tools* into an existing iterative structure, which aligned with the two-week implementation period for the participating teams. Given the results of this paper, continuous implementation and evaluation could provide further insights.

6.4 Future Work

Research on psychological safety is still very new to the domain of software. The work conducted in this paper is an initial step into a broader subject of how software teams can adopt, work with, and improve their psychological safety. The continuous implementation and evaluation of the *tool* concept is a natural continuation of this paper. For continuous evaluation of the effect of *tool* usage on psychological safety over time, repeated quantitative measurement techniques akin to those designed by O'Donovan et al. [16] (as was utilized in the pre-study [1]) could be useful.

7 Conclusion

Using design science research, this paper presents the design of actionable *tools* to aid and enable software teams in working with psychological safety. Eight such *tools* were designed and implemented autonomously by 4 software teams over a two-week period, followed by survey and group interview evaluations. Evaluation showed that teams found the *tools* both enjoyable and helpful as both conversation starters and frames within which to work with psychological safety. Teams additionally found the *tools* to fit within their existing practice, and universally planned to use one or more of their *tools* in the future.

References

1. Agerlin Christensen, M.: Tr-2021-207: Measuring and affecting psychological safety in software teams: A case study. IT University of Copenhagen, Technical Report (2020)
2. Agerlin Christensen, M.: Tool compendium: tools for working with psychological safety in agile teams (2022). https://doi.org/10.13140/RG.2.2.26823.16800
3. Aken, J.E.V.: Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. J. Manage. Stud. **41**(2), 219–246 (2004)
4. Beck, K., et al.: Manifesto for agile software development (2001)
5. Dubey, A.: Project aristotle - understanding team effectiveness (2014). https://rework.withgoogle.com/print/guides/5721312655835136/
6. Edmondson, A.: Psychological safety and learning behavior in work teams. Adm. Sci. Q. **44**(2), 350–383 (1999)
7. Edmondson, A.: Building a psychologically safe workplace (2014). https://www.youtube.com/watch?v=LhoLuui9gX8
8. Edmondson, A.C.: Managing the risk of learning: psychological safety in work teams. Citeseer (2002)
9. Edmondson, A.C.: The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth. Wiley, Hoboken (2018)
10. Forsgren, N., Smith, D., Humble, J., Frazelle, J.: 2019 accelerate state of devops report (2019)
11. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. 75–105 (2004)
12. Lenberg, P., Feldt, R.: Psychological safety and norm clarity in software engineering teams. In: Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering, pp. 79–86 (2018)
13. Lipmanowicz, H., McCandless, K.: The surprising power of liberating structures: Simple rules to unleash a culture of innovation. Liberating Structures Press Seattle, WA (2013)
14. O'Donovan, R., McAuliffe, E.: Exploring psychological safety in healthcare teams to inform the development of interventions: combining observational, survey and interview data. BMC Health Serv. Res. **20**(1), 1–16 (2020)
15. O'Donovan, R., McAuliffe, E.: A systematic review exploring the content and outcomes of interventions to improve psychological safety, speaking up and voice behaviour. BMC Health Serv. Res. **20**(1), 1–11 (2020)

16. O'Donovan, R., Van Dun, D., McAuliffe, E.: Measuring psychological safety in healthcare teams: developing an observational measure to complement survey methods. BMC Med. Res. Methodol. **20**(1), 1–17 (2020)
17. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manage. Inf. Syst. **24**(3), 45–77 (2007)
18. Schwaber, K., Sutherland, J.: The scrum guide. Scrum Alliance (2020)
19. Wake, B.: Scrum from hell - xp123 (2004). https://xp123.com/articles/scrum-from-hell/
20. Wieringa, R.: Design science as nested problem solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, pp. 1–12 (2009)

Understanding Leadership in Agile Software Development Teams: Who and How?

Johann Weichbrodt[1](✉), Martin Kropp[2], Robert Biddle[3], Peggy Gregory[4], Craig Anslow[5], Ursina Maria Bühler[1], Magdalena Mateescu[1], and Andreas Meier[6]

[1] University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland
johann.weichbrodt@fhnw.ch
[2] University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland
[3] Carleton University, Ottawa, Canada
[4] University of Central Lancashire, Preston, UK
[5] Victoria University of Wellington, Wellington, New Zealand
[6] Zurich University of Applied Sciences, Winterthur, Switzerland

Abstract. The principles in the Agile Manifesto, the Scrum Guide and most other approaches to agile software development emphasize self-organizing teams, but rarely address issues of leadership. In this paper we report on a study of the nature of different aspects of leadership in agile teams. We used an established model of leadership, distinguishing transactional and transformational styles, and asked IT professionals a set of questions about the leadership they experience, both from direct supervisors (hierarchical leadership) and from the team itself (shared leadership). We determined correlation measures of these four types of leadership with the extent of agility in the whole organization. Our results show that agility is indeed related to the transformational style, but that the transactional style also plays a part, especially as shared leadership. Furthermore, even in highly agile software development, leadership by direct supervisors still plays an important role. We propose that, as software development becomes more agile, the transactional aspects of leadership may shift away from the leadership dyad between supervisor and employee into the agile team, while transformational leadership is important for both the team and supervisors. We discuss our results in light of applications for both research and practice.

Keywords: Leadership · Agile software development · Shared leadership · Transactional leadership · Transformational leadership

1 Introduction

When compared with classic hierarchical and Tayloristic management, agile software development is a radically different way of organization. While early agile methods like XP and Scrum aimed at the team level only and more or less ignored the organizational context, nowadays whole organizations "go agile". Such a transformation requires taking into account more than just core teams: management processes and responsibilities, the underlying organizational culture, and leadership will be affected, the more an organization implements agile software development [1–3].

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 99–113, 2022.
https://doi.org/10.1007/978-3-031-08169-9_7

Early approaches to agile software development did not explicitly address leadership. In fact, leadership or the leader's role are not even mentioned in the original Agile Manifesto and its twelve principles [4], or in the latest version of the official "Scrum Guide" [5]. On the other hand, self-organization and autonomy are at the core of agile teams. It is striking that these approaches seem to ignore the wider organizational context, and especially the role and responsibilities of "classic" hierarchical leaders or line managers. While the classical leadership roles might have changed, the tasks of leadership have not disappeared. But how are they executed in agile teams and organizations? How are they adapted in order for agile methods to work in an organizational context?

Recently, industry has become more aware of this new challenge. The "Agile 2" movement postulates that the "largest defect in agile thinking regards the role of leadership" [6]. They propose a new set of values and principles, many of which directly concern leadership and its role in agile organizations. In the Harvard Business Review article "The Agile C-Suite", the authors state the need for a new leadership approach [7]. Such practitioner-led endeavors manifest the change in the leadership role and maybe the need for a better understanding of it. On the academic side, while some studies have investigated questions around "agile leadership", the overall body of research is still rather thin [9]. In this paper, we present our findings from an online survey about agile software development and leadership in IT companies. We show how leadership styles and practices change in more agile contexts. We address the following research questions:

Q1: Do organizations implementing agile software development show less hierarchical leadership and more shared leadership than less-agile contexts?

Q2: How does transactional and transformational leadership differ in agile vs. less-agile software development?

Our results show that while there are, broadly speaking, shifts from hierarchical to shared leadership and from transactional to transformational leadership, reality seems to be more complex.

The rest of the paper is structured as follows: In the next section, we present our theoretical framework. Section 3 explains our research design and measurement of constructs. In Sect. 4 we present the results of our study, followed by a thorough discussion and final conclusions.

2 Related Work

Leadership is a mature area of organizational research underpinned by numerous theories and approaches [8]. However, in the agile software practice literature, leadership is rarely addressed explicitly. Guidelines such as the Scrum Guide [5] only briefly mention servant-leadership and self-managing teams. In academic literature, a few studies have been conducted on the role of leaders and leadership in agile software development. A recent systematic literature review [9] categorizes studies into three groups: a) studies based on leadership theories, b) tangential theories and models where leadership is included, and c) leadership styles. Leadership theories used include full range leadership theory (transactional, transformational, and laissez-faire leadership), a leadership taxonomy, complexity leadership theory, and role theory. Leadership styles explored include adaptive, shared, transformational, ad-hoc, mentor, servant, situational, expert, and super leadership. They conclude that while research on agile leadership has grown since 2005, it is still a nascent research area in which more empirical research studies are needed. They did not find a common view, but indicate that the focus moves away from hierarchical and bureaucratic leadership, and that leadership needs to change as agile teams change and mature. Yang et al. [10] asked traditional and agile project managers whether a transformational, transactional, or laissez-faire leadership approach best suited their projects. They found more need for transformational leadership in agile projects than in traditional ones. A paper by Gren and Ralph [11] reports on a small qualitative study with self-described leaders in agile development projects, finding that leadership is shared with teams, builds a sense of common purpose, and adapts to organization culture. Spiegler et al. [12] undertake a grounded theory study of Scrum Master leadership and identify nine roles that are transferred from the Scrum Master to the development team as it matures.

For this paper, we focused on two dimensions of leadership, namely *leadership style* (transactional or transformational) and *leadership locus* (hierarchical or shared) as they are well-researched, classical concepts that encapsulate some of the key differences between traditional and agile organization.

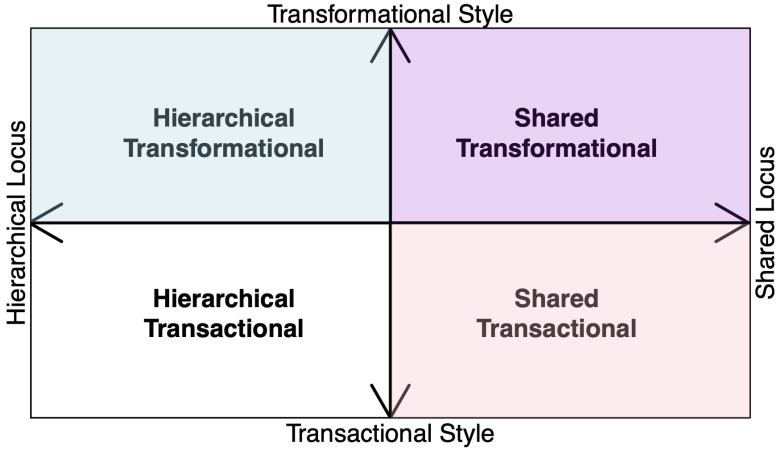


Fig. 1. Leadership locus/style matrix. Vertical axis is leadership style (transactional/transformational). Horizontal axis is leadership locus (hierarchical/shared)

First, a long-established body of leadership theory pertains to the *style* with which leadership is executed. Classic concepts distinguish *transactional* and *transformational* leadership styles [13]. Transactional leadership is, in essence, the idea of leading people by designing and adjusting an economic contract between leader and follower. Labor and its output are traded for a salary or for opportunities for promotion. The function of transactional leadership is to set, monitor and adjust goals, expectations and incentives. In contrast, transformational leadership describes a relational contract rather than an economic one. Avolio et al. [14] define transformational leadership as "leader behaviors that transform and inspire followers to perform beyond expectations while transcending self-interest for the good of the organization." The function of transformational leadership is therefore to create a sense of mission and purpose within those being led.

Second, it has long been recognized that leadership is not just situated in an individual with formal authority, but can rather manifest in different loci like context, team, dyads, etc. [15]. In our paper, we focus on the leader (individual with formal authority) and on the team (group of people interacting with little or no regard to formal hierarchy) and call these loci *hierarchical leadership* and *shared leadership*, respectively. Shared leadership has been defined as "a dynamic, interactive influence process among individuals in groups for which the objective is to lead one another to the achievement of group or organizational goals or both" [16]. In contrast, we define hierarchical leadership as influence processes occurring in a relationship characterized by formal authority (e.g., a line manager and their respective employee). Leadership can thus be found in at least two places, or loci: in the hierarchical relationship between formal leader and follower, and shared (distributed) among team members. This structure of two leadership loci and two leadership styles is illustrated in Fig. 1, with locus on the horizontal axis, and style on the vertical axis.

It should be noted that both transactional and transformational leadership were originally thought of as personal styles, existing purely on the individual level of the formal leader. Following Schein [17] we argue, however, that both these leadership styles can also be seen as important *leadership functions* in the organization, which can be served by different loci. The goal-setting and -adjusting of transactional leadership can therefore (theoretically) also be accomplished on a team level, as can the inspiration, creation and affirmation of a sense of mission typically attributed to transformational leadership. Using these two distinctions – hierarchical vs. shared leadership and transactional vs. transformational leadership – we can now theorize and derive questions about changes in leadership in less agile vs. more agile contexts of software development.

Reading many agile concepts and methods could lead one to assume that only transformational and shared leadership is important in agile software development. Most agile methods still presume the existence of a formal leader (sometimes called "line manager"), but their importance is reduced and many leadership tasks are distributed among the development team, using specified roles, as well as principles of self-organization. Because of this, and because of a presumed general occurrence of agile methods in "flatter" organizations, one would assume that agile software development is correlated with shared leadership. But does this also mean that hierarchical leadership decreases or do both exist simultaneously? Regarding leadership style, does the importance of short-term-iterated planning and adjusting of goals, inherent in agile principles, relate to a decrease or increase of transactional leadership? Does the relevance of transformational leadership increase in more agile software development, because creating and maintaining a sense of purpose becomes more important in self-managed organizations, as some have argued [18]?

We found that using our theoretical lense of leadership style and locus produced a number of interesting issues, all worthwhile pursuing, which led us to apply a more explorative approach. We do not aim to provide definitive answers to any of these questions, but rather want to open up avenues for further debate and research. We therefore decided against testing specific and focused hypotheses and formulated the following research questions instead:

Q1: Do organizations implementing agile software development show less hierarchical leadership and more shared leadership than less-agile contexts?

Q2: How does transactional and transformational leadership differ in agile vs. less-agile software development?

3 Research Methods

3.1 Data Collection and Sample

This study is based on the online survey "International Agile Study 2018/2019" conducted in Switzerland, the United Kingdom, and New Zealand in 2018 and 2019 regarding the usage of development methods and practices in the IT industry, and about the impacts of applying agile methods. For a detailed description of the survey instrument see Kropp et al. [19]. The survey addressed both agile and plan-driven companies, as well as both agile and plan-driven IT professionals, or any hybrids. There were in fact two independent surveys: one for companies, and one for individual IT professionals. In the company survey we targeted representatives of the company or the development department of a company, i.e., typically upper management level. The addresses of the companies were collated from participating IT associations from all involved countries as well as from our own institutional databases. To ensure a company was represented only once in the company survey, we sent personalized links to one management representative of each company. The IT professional survey was anonymous, and we invited wider participation. We sent invitations with a link to the survey via email and through professional social media like LinkedIn and XING (a career-oriented social networking site popular in German-speaking markets). Participants were typically directly involved in software development, and we describe the demographics in the section below. The survey was a general survey about the state of agile software development, either in IT companies or in companies with significant IT activities (e.g., banks, insurance, chemistry). The questions covered a broad range of aspects in agile software development and were the same for both surveys[1]. In this paper we focus on the analysis of the leadership questions.

3.2 Participants

The survey was answered by 199 professionals and by 88 company representatives. Since we wanted to study shared leadership, we removed high-level leaders (because they most likely are not part of a real team), and we excluded all those who did not answer any of the leadership questions (missings). The final sample was N = 200 (20.5% of which from the company and 79.5% from the professionals' survey). The average age of the participants was 42 years with an average IT experience of 18 years. The participants were IT professionals working in various sectors like retail, medical and health, finance, transportations and shipping. Of the 200, 75% were male, 12% female, 3% explicitly

[1] The complete questionnaire is available at https://tinyurl.com/5n749v6y.

preferred not to say and 10% did not indicate gender. The participants mainly came from the organizing countries, but we also had answers from Austria, Germany, and the United States.

Table 1 shows the roles of the participants in their company.

Table 1. Roles of participants.

Role	%
Project Manager	22.5%
Software Developer	16.5%
Team Leader	13.2%
Development Manager	9.9%
Product Manager	7.1%
Designer/Architect	4.9%
Coach/Scrum Master	3.8%
QA Tester	2.7%
Researcher	2.2%
UX Expert	1.6%
Other	15.4%

3.3 Questions, Constructs and Analysis

Extent of Organizational Agility. In order to measure the extent of agility of an organization, we used the single-item question: "*Is your organization currently practicing plan-driven or agile software development?*" with a 5-point Likert-scale with the following anchors: (1) all plan-driven, (2) mostly plan-driven, (3) both plan-driven and agile, (4), mostly agile and (5) all agile. Note that the question specifically referred to software development rather than other aspects of the organization. To gain further insight, we also asked which agile methods were used, if any, the number of years of the organization's experience with agile methods, and to what extent participants were satisfied with the organization's methodology.

Leadership Loci: Hierarchical vs. Shared Leadership. In order to measure hierarchical leadership, we used the questionnaire from Ismail et al. on transactional and transformational leadership styles [20], which is an adaptation of Bass and Avolio's Multi-Factor Leadership questionnaire [21]. To assess shared leadership, we re-formulated the items by replacing "my direct supervisor" with "my team." This means that each participant saw 20 leadership questions, 10 for hierarchical locus and 10 for shared locus, each with 5 for transactional style and 5 for transformational style, as shown in Table 2. Each question was answered using a 5-point Likert scale from 1 (Strongly Disagree) to 5 (Strongly Agree). The responses were combined, resulting in an aggregate score from 1 to 5. The internal consistency of the answers was good to very good: for the four combinations of locus and style, we report Cronbach's Alpha in Table 3.

Table 2. List of items used to measure leadership: answered using Likert scale from 1 (strongly disagree) to 5 (strongly agree).

Style	**Locus**	
	Hierarchical Leadership: *“My direct supervisor/line manager...*	**Shared Leadership:** *“In my team/My team members ...*
Transactional Leadership	makes clear expectations.”	expectations are made clear.”
	tells us standards to carry out work.”	we set standards to carry out work.”
	will take action before problems are chronic.”	will take action before problems are chronic.”
	works out agreements with me.”	works out agreements with each other.”
	monitors my performance and keeps track of mistakes.”	monitor each other’s performance and keep track of mistakes.”
Transformational Leadership	instills pride in me.”	instills me with pride.”
	encourages me to perform.”	encourages me to perform.”
	spends time teaching and coaching.”	we spend time teaching and coaching.”
	listens to my concerns.”	we listen to each other’s concerns.”
	encourages me to rethink never-questioned ideas.”	encourages me to rethink never-questioned ideas.”

Table 3. Four sets of responses for locus and style, with Cronbach’s Alpha showing good internal consistency.

Locus and Style	Cronbach’s Alpha
Hierarchical Transactional	0.76
Hierarchical Transformational	0.83
Shared Transformational	0.77
Shared Transactional	0.84

Analysis. Our approach in this study emphasizes understanding and is principally exploratory. While we do address our research questions, we therefore refrained from proposing and testing specific hypotheses. Our analysis consists mainly of inspecting descriptive results, correlations, and graphical comparisons of distributions. We hope this approach serves to inform future work that is then able to frame and test hypotheses.

4 Results

The participants worked in companies which are experienced in agile software development, with a large majority practicing Scrum alone or in combination with other methodologies. Most companies (74.8%) have been practicing agile software development for at least three years. The vast majority of the participants (81%) worked in organizations which are at least slightly experienced in agile software development, with 28% very experienced, 31% moderately experienced, 28.5% slightly experienced. Only 5% stated that the company had no experience with agile software development (7% did not rate the experience of the company).

The extent of agility in software development varied across the organizations: 13 participants (6.5%) reported all plan-driven software development, 25 participants (12.5%) mostly agile, 78 participants (39%) work in organizations where they practice both plan-driven and agile software development, 65 (32.5%) participants report mostly agile, and 19 (9.5%) participants report all agile software development. Elsewhere in our survey we asked questions about use of a range of agile practices, and we found strong correlations between that data and the level of agility reported.

The companies used a broad range of agile methodologies (Scrum, XP, SAFe). Most companies claim to follow the Scrum methodology (47%), followed by Kanban (8.5%), combined Scrum and eXtreme Programming (6.5%) and DSDM/AgilePM (6.0%). 12.5% used the free text option and most of them stated that they use a mix of different methodologies; 0.5% did not state the methodology of the company. The majority (59%) of the participants were satisfied with the company's current methodology. Only 11.5% of the participants were unsatisfied about their company's current methodology.

In Table 4, we display descriptive statistics for the extent of agility and leadership by locus and style. On the right of the table, we display the correlation between extent of agility and leadership, showing Spearman's rho and the p value (uncorrected for multiple tests). Although the intent of our study is principally exploratory, rather than hypothesis testing, we report p values as an indication of the rarity of the results in order to inform future work.

We can see some general differences in the data for both leadership loci and styles. In every case where we distinguish loci, shared leadership is consistently rated higher than hierarchical leadership. In every case where we distinguish styles, transformational leadership is rated higher than transactional leadership. For the four specific cases (last four rows), ANOVA and Tukey HSD tests show all differences to be significant.

Table 4. Descriptive statistics for Extent of Agility, for leadership by locus and style, and correlation between Extent of Agility and leadership (for measures combining loci or styles, we only include cases where we had responses for each).

Metric (1–5)		**n**	**M**	**SD**	**Correlation**	
Extent of Agility		200	3.26	1.01	—	—
Locus	*Style*	**n**	**M**	**SD**	**rho**	**p**
Both Loci	Both Styles	194	3.54	0.57	0.285	< 0.001
Both Loci	Transactional Style	198	3.35	0.60	0.183	0.010
Both Loci	Transformational Style	198	3.73	0.63	0.325	< 0.001
Hierarchical Locus	Both Styles	197	3.35	0.69	0.110	0.124
Shared Locus	Both Styles	196	3.73	0.64	0.370	< 0.001
Hierarchical Locus	Transactional Style	197	3.15	0.74	0.008	0.914
Hierarchical Locus	Transformational Style	197	3.55	0.67	0.179	0.012
Shared Locus	Transactional Style	198	3.55	0.77	0.311	< 0.001
Shared Locus	Transformational Style	197	3.90	0.68	0.370	< 0.001

Examining the relationship between the extent of agility and leadership, we can see that, in general, over both loci and both styles, leadership is related to the extent of agility (rho = 0.277, p < .001). At a finer level, however, we can discern several differences. The strongest relationships are with a shared locus (overall rho = 0.370, p < .001) and with transformational style (overall rho = 0.321, p < .001). The hierarchical locus does not show a correlation overall (rho = 0.111, p = 0.117), and in particular no correlation is seen for a hierarchical locus and a transactional style (rho = 0.008, p = 0.914). To examine the patterns, we created the series of graphs shown in Fig. 2. Each of the four graphs corresponds to one of the four combinations of locus and style, arranged as described earlier in Fig. 1. Each graph shows five boxplots, one for each of the extents of agility (All Plan-Driven to All Agile), showing the rating for the leadership locus and style specified.

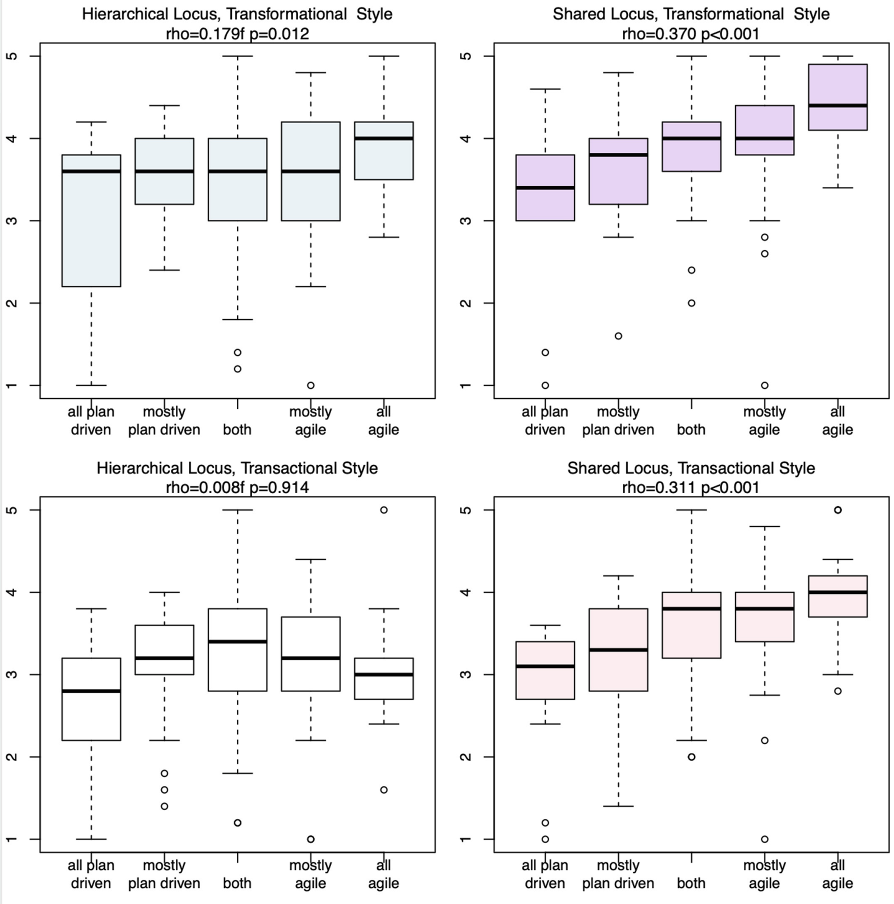


Fig. 2. Plots showing relationships for each of the four pairings of locus (hierarchical and shared) and style (transactional and transformational). The boxplots show the relationship between the Extent of Agility (horizontal axis), and level of Leadership (vertical axis). [Each boxplot shows the median (dark horizontal line, the inner quartiles (colored box), the outer quartiles (whiskers) and outliers (circles).]

The pattern for hierarchical locus & transactional style (bottom left) shows an initial rise from all plan-driven, but then a fall for mostly and all agile, corresponding to the lack of correlation (rho = 0.008, p = 0.914). However, it may be important to note that while there is no correlation: the measure is fairly consistent, and even for all agile, hierarchical-transactional leadership is rated midway on the scale. Hierarchical locus with transformational style (top left) shows a modest rise (rho = 0.179, p = 0.012). Shared locus with transformational style (top right) shows a consistent and strong rise (rho = 0.370, $p < 0.001$). Shared locus and transactional style, interestingly, also shows a strong and consistent rise (rho = 0.311, $p < 0.001$).

5 Discussion

5.1 Interpretations of Our Findings

We set out to study the relationship between leadership style and locus and the extent of agility in agile software development, and we found strong correlations between some aspects of leadership, but not all of them.

Our first research question concerned hierarchical and shared leadership and their connection to agility. Our data show that while shared leadership is (somewhat unsurprisingly) strongly related to more agile contexts, scoring very high in all-agile software development, the results are a bit more nuanced regarding hierarchical leadership. Overall, the intensity with which people experience hierarchical leadership does not change much as software development becomes more agile. Differentiating between the transactional and transformational style within the hierarchical leadership locus showed us that transformational hierarchical leadership increases slightly, showing a weak correlation, whereas the relationship between the transactional leadership style and agility resembles an inverted U-shaped curve. In essence, it is fair to say that in agile software development, hierarchical leadership is still present – especially in combination with the transformational style. Our data do not tell us whether this generally is positive – it could very well be that agile software development with less hierarchical leadership outperforms other practices. Nevertheless, it is still surprising to see that hierarchical leadership does not wane much.

With our second research question, we looked specifically at changes in leadership style as software development becomes more agile. We found distinct evidence that transformational leadership is related to the extent of agility in software development. This effect is very strong for shared transformational leadership and weak (but still present) for hierarchical transformational leadership. We also found that shared transactional leadership markedly increases in more agile contexts, while for hierarchical transactional leadership the above-mentioned inverted U-shaped relationship applies.

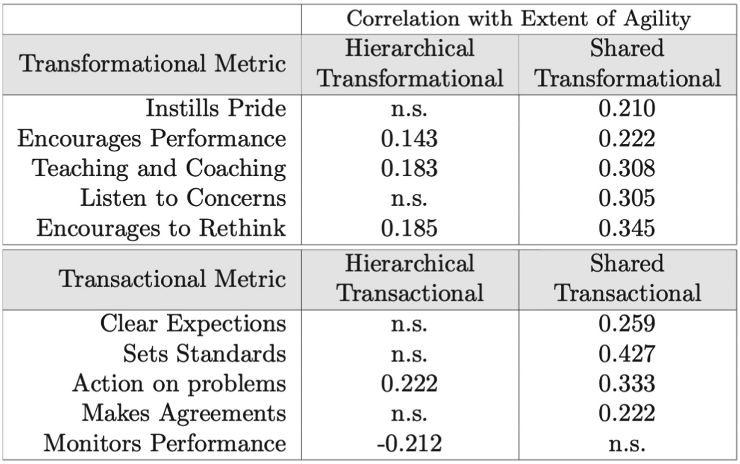
In our view, the two most interesting results of our study are:

(1) Hierarchical leadership does not become irrelevant in agile software development. People experience both transactional and transformational hierarchical leadership quite strongly, even in mostly or all-agile contexts. While leadership does become more distributed, leadership executed by direct supervisors and/or line managers still holds relevance.

(2) Transactional leadership does not become irrelevant in agile software development, either. Goal-setting, accountability and other more "directive" aspects of leadership are still very present in agile contexts, but their locus seems to shift from the line manager to being shared in the team.

As we described earlier, our questions on leadership were based on Ismail et al.'s questionnaire [20], with five each for transformational and transactional styles, and we adapted these to distinguish a hierarchical and a shared locus. To further investigate our results post-hoc, we explored correlations between extent of agility and the responses to individual questions. In Table 5 we show these correlations. One overall pattern is that almost all the correlations for the shared locus (rightmost column) are greater than the equivalent correlations for the hierarchical locus (column to the left). The only exception involves the question about monitoring performance, where the correlation is not significant for shared, but negative for hierarchical. Also, while this is the only non-significant correlation for the shared locus, there are many for the hierarchical. Moreover, with an alpha of .001, *none* of the correlations are significant for the hierarchical, whereas six remain significant for shared locus. Looking at the three strongest single correlations could give us some idea of what differentiates agile from non-agile leadership the most: "Setting standards to carry out work", "encouraging to rethink never-questioned ideas", and "taking action before problems are chronic" within the team (shared locus) seem to be good indicators for agile leadership. Notably, two of these regard the transactional style.

Table 5. Correlations between Extent of Agility and responses to individual leadership questions, by locus and style; columns at right show Spearman's rho, where below p = 0.05 (uncorrected for multiple tests).



	Correlation with Extent of Agility	
Transformational Metric	Hierarchical Transformational	Shared Transformational
Instills Pride	n.s.	0.210
Encourages Performance	0.143	0.222
Teaching and Coaching	0.183	0.308
Listen to Concerns	n.s.	0.305
Encourages to Rethink	0.185	0.345
Transactional Metric	Hierarchical Transactional	Shared Transactional
Clear Expections	n.s.	0.259
Sets Standards	n.s.	0.427
Action on problems	0.222	0.333
Makes Agreements	n.s.	0.222
Monitors Performance	-0.212	n.s.

Another question that arises from this in-depth analysis is the role of performance monitoring, which notably does not increase with a shared locus and seems to become even *less* relevant with a hierarchical locus. At least in part, the phrasing of the question as "monitoring performance and keeping track of mistakes" might be the cause of this result, as that could have a rather negative connotation for people. However, the drastically different result for this single item still raises the question: Who monitors performance in agile software development?

Looking at our results more broadly, it is also noteworthy that, overall, people experience more, or more intense, leadership (as measured with our items) in agile software development. One could have assumed that overall leadership is equally "strong" in plan-driven contexts, just more hierarchical and/or more transactional. This would have shown as a sort of x-shaped relationship in our data (as one aspect of leadership goes down, another one goes up). Instead, it seems that leadership *in general* is more prevalent in agile than in plan-driven software development (with the exception of hierarchical-transactional). The positive interpretation of such a finding might be that agile software development allows more people to participate in leadership processes as part of an empowerment or even emancipation process. On the other hand, one could argue that "more leadership" is not without cost, as it also means more complexity in decision-making and navigating relationships. Handling such increased complexity requires more psychological and social resources from people.

5.2 Implications for Research

The qualitative study of Gren and Ralph [11] found that self-described agile leaders emphasized the importance of shared leadership and fostering a sense of common purpose. Our results are consistent with those findings. Yang et al. [9] found that transformational leadership was more highly rated by agile managers than by traditional managers whereas transactional leadership was equally rated. We also find that transformational leadership becomes more important as organizations become more agile, but additionally that shared transactional leadership is important, and that hierarchical leadership still appears to play a part. Another consideration is the role of individual people. Gren and Ralph's participants all claimed to be leaders, and some of their job titles appeared to possibly suggest some hierarchical authority. The interplay between a hierarchical and a shared locus of leadership for agile development may be complex and subtle.

The nature of the transactional style within agile development also needs further study [9, 10]. The issue of hierarchical-transactional leadership relates to the role of a hierarchical locus within Agile, and while this is seldom acknowledged in articulation of agile processes, it is still commonplace in practice. Another issue relates to shared-transactional leadership. Our results suggest this is stronger in mostly or all-agile teams. This might relate to some well known practices, such as XP's "planning game" or "planning poker", where the whole team is involved in planning, and then commits to that plan. However, especially in an organizational context, this raises issues of stress and overwork, and overall responsibility. Even in a positive context, the effects of social pressure can be serious.

In future work, it would be interesting to look at different results based on individual roles. For example, do Scrum Masters perceive shared leadership in the agile software development teams differently than developers or product owners? Such detailed analyses could reveal insights about the distribution of leadership responsibilities and its effects on software development.

In summary, we suggest there is a need for further research into the role of transactional and hierarchical leadership in agile software development. While this study has identified their continued use, without contextual research that seeks to uncover the potentially complex stories underlying their use, we can only speculate about their role and relevance.

5.3 Implications for Practitioners

Members of agile software development teams could, firstly, use our results to clear out some myths that might exist around agile leadership: that hierarchical leadership is no longer present, or that encouragement, emotional support and other ideas around transformational leadership are the only important aspects in leading an agile team. We can show quite clearly that direct supervisors still play an important role and that transactional leadership on the team level is even more relevant in agile software development. This leads to our second implication, namely that teams should understand and take to heart the nature of shared-transactional leadership: Aspects such as goal-setting, making expectations clear, and taking action before problems become chronic are key for agile shared leadership. This requires actually a very disciplined work style of agile teams. Especially Scrum Masters should not only make sure there is commitment (in the emotionally invested sense), but also that all members are aware of exactly what they have committed to. This point is also noted by Spiegler et al. [12], who identify a leadership role called "disciplinizer on equal terms" for Scrum Masters which involves them helping the team to understand for themselves the importance of discipline and focus in their work.

5.4 Limitations

We need to recognize issues relating to our sample. We invited many people to participate in our survey on agile software development, but only some chose to participate, so our sample is self-selected. In our analysis we look for relationships between the extent of agility and attributes of leadership. We need to be cautious about several aspects of this issue. We determined the extent of agility on a scale from 1 to 5 by asking participants about software development in their organization. We acknowledge this is a complex issue which cannot easily be represented as a simple ordinality. The questions from which we derive our measure of leadership are based on established instruments, but there may have been different interpretations of the wording. For example, we discuss above how "monitoring performance" might be interpreted negatively. Perhaps most importantly, our analysis uses correlation. While this allows us to determine, for example, that more agility is associated with more shared leadership, we cannot assume that more agility is the cause of more shared leadership, or vice-versa. Establishing causality would require more detailed study.

6 Conclusions

Our study was to explore leadership in agile software development, in particular the *style* of leadership, transactional and transformational, and the *locus* of leadership, hierarchical or shared. We adapted an established questionnaire instrument and examined the responses from professionals actually involved in development. Our results suggest a strong relationship between the level of agility and the impact of a shared locus, including both a transformational style and also a transactional style. The extent of agility was also (more weakly) related to a hierarchical locus transformational style, but not with a transactional style.

For future work, we would like to address the limitations and probe the key findings. We especially wish to further examine how a shared locus of leadership appears to involve both transformational *and* transactional leadership at the same time. Furthermore, looking at outcome measures (e.g., productivity measures, satisfaction, or perceived success of agile transformation) and their relationship to the different aspects of leadership in agile software development should prove particularly valuable.

References

1. 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). https://doi.org/10.1016/j.jss.2016.06.013
2. Uhl-Bien, M., Arena, M.: Leadership for organizational adaptability: a theoretical synthesis and integrative framework. Leadersh. Q. **29**, 89–104 (2018). https://doi.org/10.1016/j.leaqua.2017.12.009
3. Kalenda, M., Hyna, P., Rossi, B.: Scaling agile in large organizations: practices, challenges, and success factors. J. Softw. Evol. Proc. **30**, 1–24 (2018). https://doi.org/10.1002/smr.1954
4. Agile Manifesto. http://www.agilemanifesto.org
5. Schwaber, K., Sutherland, J.: Scrum Guide (2020). https://scrumguides.org/
6. Berg, C., et al.: Agile 2: The Next Iteration of Agile. Wiley, Indianapolis (2021)
7. Rigby, D., Elk, S., Berez, S.: The agile C-suite. Harvard Bus. Rev. (2020). https://hbr.org/2020/05/the-agile-c-suite
8. Lord, R.G., Day, D.V., Zaccaro, S.J., Avolio, B.J., Eagly, A.H.: Leadership in applied psychology: three waves of theory and research. J. Appl. Psych. **102**, 434–451 (2017). https://doi.org/10.1037/apl0000089
9. Modi, S., Strode, D.: Leadership in agile software development: a systematic literature review. In: ACIS 2020 Proceedings, vol. 55 (2020). https://aisel.aisnet.org/acis2020/55/
10. Yang, H., Huff, S., Strode, D.: Leadership in software development: comparing perceptions of agile and traditional project managers. In: AMCIS 2009 Proceedings, vol. 184 (2009). https://aisel.aisnet.org/amcis2009/184
11. Gren, L., Ralph, P.: What makes effective leadership in agile software development teams? In: 44th International Conference on Software Engineering (2022). https://doi.org/10.48550/arXiv.2201.08058
12. Spiegler, S.V., Heinecke, C., Wagner, S.: An empirical study on changing leadership in agile teams. Empir. Softw. Eng. **26**(3), 1–35 (2021). https://doi.org/10.1007/s10664-021-09949-5
13. Judge, T.A., Piccolo, R.F.: Transformational and transactional leadership: a meta-analytic test of their relative validity. J. Appl. Psychol. **89**, 755–768 (2004). https://doi.org/10.1037/0021-9010.89.5.755

14. Avolio, B.J., Walumbwa, F.O., Weber, T.J.: Leadership: current theories, research, and future directions. Annu. Rev. Psychol. **60**, 421–449 (2009). https://doi.org/10.1146/annurev.psych.60.110707.163621
15. Hernandez, M., Eberly, M.B., Avolio, B.J., Johnson, M.D.: The loci and mechanisms of leadership: exploring a more comprehensive view of leadership theory. Leadersh. Q. **22**, 1165–1185 (2011). https://doi.org/10.1016/j.leaqua.2011.09.009
16. Pearce, C.L., Conger, J.A.: Shared Leadership: Reframing the Hows and Whys of Leadership. Sage, Thousands Oaks (2003)
17. Schein, E.H.: Organizational Psychology. Prentice Hall, Englewood Cliffs (1965)
18. Laloux, F.: Reinventing Organizations. Nelson Parker, Brussels (2014)
19. Kropp, M., Meier, A., Anslow, C., Biddle, R.: Satisfaction and its correlates in agile software development. J. Syst. Softw. **164**, 110544 (2020). https://doi.org/10.1016/j.jss.2020.110544
20. Ismail, A., Mohamad, M.H., Mohamed, H.A.B., Rafiuddin, N.M., Zhen, K.W.P.: Transformational and transactional leadership styles as a predictor of individual outcomes. Theor. Appl. Econ. **17**, 89–104 (2010)
21. Bass, B.M., Avolio, B.J.: Multifactor leadership questionnaire (MLQ) [Database record]. Am. Psychol. Assoc. PsyTest (1995). https://doi.org/10.1037/t03624-000

A Case for Data-Driven Agile Transformations: Can Longitudinal Backlog Data Help Guide Organizational Improvement Journeys?

Gijsbert C. Boon[1](✉) and Christoph Johann Stettina[2]

[1] FlowMeister, Amsterdam, The Netherlands
g.boon@flowmeister.net

[2] Leiden Institute of Advanced Computer Science, Leiden University,
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
c.j.stettina@liacs.leidenuniv.nl

Abstract. *Context:* Almost every organization with a strong digital capability has embarked on an agile transformation journey. But do these changes actually deliver on the envisioned transformation goals? What conclusions can we draw from measurements and observations?

Objective: The ambition of this report is to (1) assess whether tooling data can be used to guide a transformation towards improved organizational performance; (2) verify claimed benefits of agile transformations using tooling data in the presented case study.

Method: We measure productivity, time-to-market, and quality as transformation objectives by analyzing longitudinal Jira backlog tooling data within an embedded multiple-unit case study.

Results: By analyzing over 57,000 Jira issues from eight agile release trains over a period of three years, we (1) provide a proof of concept of how tooling data can be used to guide agile transformations; (2) provide empirical evidence on the assessment of transformation objectives over time and organizational layers at FinOrg; and (3) connect measurement results with available literature.

Conclusions: We may conclude that tooling data is a viable addition to guide transformations through identification of improvement opportunities on the set objectives. We connected the case study results to existing literature and identified similarities. We argue that there is a need for a measurement framework and better understanding of the dynamics between measurement and performance.

Keywords: Agile transformation · Backlog tooling data · Performance measurement framework · Metrics · Organizational performance

1 Introduction

Almost every organization with a strong digital capability is on an agile transformation journey [9]. However, whether this transformation benefits the organization, and whether goals are reached, are frequently heard concerns [33].

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 114–130, 2022.
https://doi.org/10.1007/978-3-031-08169-9_8

Measurement is fundamental to justifying change efforts and provides objective reference material from which to learn and improve (cf. [25,31]). While previous work (cf. [28]) displayed the feasibility of using data for individual organizations and metrics, the change across organizational layers over time has largely been unexplored to date. Can we find objective data to confirm whether these transformations were actually quantitatively measured and whether they improved organizational performance [33]? In this paper we report on a case study with multiple units, for the first time exploring the application of backlog data to measure and guide a large-scale agile transformation, based on eight Agile Release Trains in a large international financial services company.

2 Related Work

2.1 Large-Scale Agile Frameworks and Impact of Transformations

While agile techniques vary in practice, they share common characteristics, such as iterative development and the focus on people and their interactions, captured in the 2001 Agile Manifesto and its principles [2]. Current figures and surveys on scaled agile transformations [9,33] indicate that SAFe [19] is considered the most applied framework (35%), followed by Scrum of Scrums (16%), and others like Disciplined Agile Delivery (DAD), Large Scale Scrum (LeSS) [22], Enterprise Scrum, and Lean Management (4%).

Current literature documents multiple attempts to measure the impact of agile transformations [21,28,33]. Consolidating prior evidence, Stettina et al. [33] report the impact of agile transformations being significant along the dimensions of *Productivity, Responsiveness, Quality, Workflow health, and Employee satisfaction & engagement*. From a practitioner perspective, the Scaled Agile Framework (SAFe) proposes three dimensions of metrics, namely *Outcome, Flow*, and *Competency* [30]. *Outcome* metrics focus on whether solutions meet the needs of customers and business, *Flow* metrics focus on organizational efficiency, and *Competency* metrics focus on how proficient the organization is in its practices to enable business agility [30].

2.2 Research on Performance Measurement Frameworks

In general management literature, multiple performance measurement frameworks and models have been developed and applied, amongst others the (1) Balanced Scorecard (BSC); (2) Performance Pyramid [39]; and (3) Performance Prism [26]. A comparative overview is provided by Öztayşi and Uçal [42], based on the seven purposes formulated by Meyer [24] (i.e., (1) look back; (2) look forward; (3) roll up; (4) cascade down; (5) compare; (6) compensate; and (7) motivate) combined with two additional views: (8) alignment with company strategies; and (9) flexibility (dynamism) of the measurement model according to change. Öztayşi and Uçal [42] show that (only) BSC satisfies all purposes. The latter two purposes seem especially relevant in the context of agile transformations.

The BSC approach [16,17] has been introduced to capture strategic intent while linking it to the performance of an organization, and views strategy management as an integrated end-to-end process [16,27]. BSC is widely applied across different industries and describes four perspectives: (1) Learning & Growth (*can we continue to improve?*); (2) Customer (*doing the right things*); (3) Internal process (*doing things right*); and (4) Financial perspective. In the context of agile strategy, an elaborate description is provided using (Dynamic) Balance Scorecards by Wireaeus and Creelman [40], observing the absence of robust objective statements and not using tools such as driver-based models and so-called *Key Performance Questions* to bridge the gap between objectives and KPIs (p.15 [40]). We argue that the same challenge applies to the Objective & Key Results (OKR) approach and see a similar ambition in the Goal Question Metric (GQM) approach. In the software quality domain this approach has been proposed to define the *right* measures [1]; Goals need to be traced back to data that are intended to define those goals, and a framework needs to be provided for interpreting the data with respect to the stated goals.

2.3 Research on (Backlog) Data in Agile Software Development

Backlog tooling to support the application of agile frameworks is perceived by agile teams as highly important within their development toolchain [34]. Further, a combination of tool-driven quantitative reporting (e.g. based on backlog tooling) supplemented by cadence-driven qualitative insights (e.g., iteration reviews, demos as well as employee and customer surveys) is applied among more mature agile teams and organizations [35]. A literature study by Biesialska et al. [3] describes a multitude of tooling data sources available in agile software development and provides an overview of the use of backlog tools for monitoring the status and progress of projects, backlogs, and corporate initiatives. A substantial part focuses on estimation and predictability models [7,29] on diverse levels, ranging from team-level user stories [5,6], requirements [8], and Epics [4] to sprint projects and releases [23]. Using data from these sources raises reliability challenges such as (1) the need for automation (unobtrusiveness cf. [23]); (2) transforming the data; and (3) the assessment (of the maturity) of data quality [5]. Based on SWEBOK [15] knowledge areas, Biesialska et al. classify no research under *Software Engineering Economics*. By this we may conclude that areas such as efficiency, effectiveness, productivity, time-value, and business case are to this date not covered in the context of big data analytics, whereas these are crucial topics in the context of agile transformations. However, the case of Fannie Mae [31] describes the use of analytics to facilitate guidance during a Agile-DevOps transformation using automated function points for productivity and defects for quality measurements.

2.4 Summary of Literature and Research Question

Based on the current state of the literature, we can make the following observations: (1) there is no generally accepted view on success of agile transformation or of its impact on organizational performance; (2) although there are some measurement frameworks available for understanding the impact of agile transformations (cf. [21,28,33]), none of those have been used to act as common ground for reference or for the guidance of agile transformations; (3) the same applies to using backlog tooling data. These observations challenge us to pose the following research question: *How can we measure and guide the impact of agile transformations on organizational performance using backlog tooling data?*

3 Methodology

In order to address our research question, we conducted exploratory analyses (cf. Tukey [37]) on backlog data in an embedded multiple-unit single case study (Yin[41], Type 2). By analyzing a single organization and multiple units, we were able to compare results and observe the impact of interventions, maturity, and trends within the same context of the transformation. Units (i.e., value streams and shared services) have a 1:1 relation to Agile Release Trains (ARTs), and consist of multiple teams at FinOrg.

3.1 Our Case Study Subject: FinOrg

The subject of our case study is the agile transformation of a large Dutch financial services organization: 11 release trains, approximately 70 teams, ranging from development teams, DevOps teams, supporting staff departments (e.g., architecture, security, HR, procurement, marketing), and back-office business (non-IT) operations teams. All units are individually profit-and-loss responsible, have own product market propositions and are autonomous[1] in the implementation of the new agile way of working, which is driven by the following objectives:

1. Improve productivity (PROD)
2. Faster time to market (TTM)
3. Higher quality (QBD)
4. Higher customer satisfaction (CUST)
5. More engaged employees (EMPL)

No targets for these objectives have been communicated at FinOrg. FinOrg uses Jira as its backlog system, plugins Easy business intelligence for dashboards, and Structure to aggregate data across units and teams. For statistical analysis we used JASP and Jamovi for plots.[2]

[1] Transformation efforts are decentralized, supervised at c-level.

[2] Atlassian's Jira: www.atlassian.com; Jamovi: www.jamovi.org, JASP: jasp-stats.org; ALM Works Structure: almworks.com; Easy Business Intelligence: eazybi.com.

3.2 Mapping Literature to Transformation Objectives at FinOrg

In order to map the transformation objectives to categories in the literature, we will use the dimensions introduced by Stettina [33].

- `(a) Responsiveness`: maps to FinOrg's *faster time to market* driver (`TTM`). That, in turn, translates to the speed of epics and features on portfolio /program level and team issues of throughput time from creation up until the moment the item is resolved. Note that the notion of *delivered* or *to-market* is interpreted in different ways in the literature and tooling. In order to prevent confusion, we used the resolve time (i.e., the item reached its final state) denoted with TTR (*Time to Resolve*).
- `(b) Productivity`: maps to the productivity objective (`PROD`), which translates to *delivering more value to the customer.* The notions of `WSJF` and `Cost of Delay` are aligned with this objective. Items at program level and above are WSJF-estimated. WSJF is an abbreviation of *Weighted Shortest Job First* and relates the Cost of Delay (value) attributes as proposed by SAFe [19] to their Job Size (estimated effort).
- `(c) Workflow health`: For this dimension one must note that the measurements proposed in literature overlap with *responsiveness* and *time to market* [33], although with a different objective. An illustration: an increase in functionality per time unit (cf. [28]) can indicate higher productivity, but might be categorized as faster *time to market* as well. To dissolve this confusion, we assigned its measurements: (1) *Job Size*; and (2) the *number-of-items* resolved to both objectives: `PROD` and `TTM`.
- `(d) Employee satisfaction & Engagement`: maps to the `EMPL` objective. By focusing on backlog data, we were not able to address employee engagement. Instead, we used other employee-related measurements with regards to *number of people assigned* during the flow and a custom FinOrg indicator of *complexity* as proxy for collaboration/organized/planned measurement and classified these as part of `(c) Workflow health`.
- `(e) Quality`: maps to the *quality by design objective* (`QBD`). In order to keep track of compliance and quality aspects, FinOrg introduced a quality-by-design process, implying that all initiatives need to be checked against relevant compliance and quality policies (e.g., security, privacy, legal, ITSM) and are thus explicitly linked to quality backlog items. QBD items have to be resolved alongside the respective initiative (similar to acceptance criteria) assigned to, and executed by, the appropriate roles and colleagues. `Number-of-items` resolved and *TTR for QBD items* are used as measures.

4 Results

4.1 Case Background: FinOrg's Agile Transformation Journey

The framework implemented at FinOrg was based on SAFe [30] with a few additions, the most important being the introduction of the aforementioned quality-by-design process. Another addition was the integration of business operations, including non-IT teams, into the units. FinOrg implemented a workflow on program and portfolio level with funnel, review, analyze, backlog, and implementation stages, mandatory initiative statement registration, and multiple WSJF-estimation and *Quality by Design* sessions within a quarterly cadence.

With respect to the transformation timeline, we distinguish three phases in the transformation at FinOrg: *Wave 0*: agile at team level, using backlog system at team level with mixed maturity levels and agile models e.g. Kanban and Scrum variants (months 0–12). *Wave 1*: introduction at program and portfolio level of a new way of working based on the SAFe framework (months 13–24). *Wave 2*: maturing at program and portfolio level (months 25–36, most recent). The lead author helped guide the digital transformation at team level during *Wave 0* and helped design and implement the operating model at portfolio and program level. At *Wave 1*, the lead author was responsible for creating and introducing the solution *on top* of the existing Jira backlogs. This functionality was created with the use of the plugins and custom scripting to facilitate guidance on the program/portfolio and quality-by-design aspirations. An extra layer was introduced using two additional backlogs containing: (1) *functional* items (i.e., Epics, Features); (2) *non-functional* also known as quality-by-design items. Release trains and teams are responsible for documenting initiatives and quality aspects and linking activities to the overarching items. This functionality has been iteratively developed and introduced with a minimum viable product at the start of *Wave 1* at corporate level.

Table 1. Descriptive information on the Jira tooling data of FinOrg

Agile release train (unit):	U01	U02	U03	U04	U06	U07	U08	U12	**Total**
Jira projects #	3	8	3	3	2	***9***	2	6	**36**
Teams #	7	***15***	6	3	7	11	4	4	**57**
Portfolio & program level									
- Epics #	10	***55***	37	17	41	14	9	28	**211**
- Features #	231	256	***295***	90	291	79	193	26	**1,461**
Team level									
- Team issues #	3,211	13,461	5,201	2,683	3,460	*13,751*	3,185	10,248	**55,200**
Quality by design									
- QbD items #	48	*192*	58	99	51	132	78	71	**729**
Total issues	**3,500**	**13,964**	**5,591**	**2,889**	**3,843**	**13,976**	**3,465**	**10,373**	**57,601**

Bold, italic and green indicates the highest score compared to other release trains.
Team backlogs are only included if historical data of **≥36** months were available, backlog size **>1,000 issues** and **recent (i.e. <1 month)** updates were registered at that backlog, thus excluding dormant backlogs.
Units are included if the unit was not explicitly excluded from transformation efforts and data spanned over a period of **24** months.
Quality by Design teams are not counted in the total number of teams count. These teams cover approximately **12** disciplines; the backlog data for these disciplines also span a period of **24** months historically.
Note on epics, features and team-level issues: at portfolio level, **epics** are defined as >1 quarter of impact for one or more units. **Feature** definition: **≤1** quarter and can be resolved within a train. **Team-level issues** are smaller than features and can be solved within one team. Issues at team level are compressed to one layer and the lowest level of sub-task is discarded.
Overall **>2,000** colleagues have been affected by this transformation.
Other data sources are available at FinOrg used for incident/problem management and CI/CD tooling. Both domains were impacted with coinciding migrations and are not included, since their data maturity was significant lower and alignment not yet feasible.

Table 1 presents our case study data. We performed data cleansing, resulting in dropping three units and multiple Jira team projects based on our assessment that their activities were not substantial enough as a basis for detecting empirical trends and differences. In addition, we harmonized workflows, different uses of statuses, issue types, and custom fields by adding an abstraction model, exposing backlogs in only three basic layers (i.e., Epics, Features, Team issues) and a simplified workflow (i.e., only create/open, resolve statuses). By this, clarity in presentation was improved, while keeping the backlog system intact (refer to additional notes Table 1 for details).

4.2 Uncovering Trends in Backlog Data

Our exploration ambition is to determine whether desired trends are noticeable in order to guide the transformation. We first illustrate *productivity* `PROD`. Figure 1 plots resolved *Cost of Delay*, our proxy for value delivery, relative to its mean[3], making comparison of results over time possible and uncovering potential trends. We share two observations based on this AVP plot: *Observation 1*: the start of the portfolio/program-level wave, starting in month 12, is visible by the cadence of resolved items/dots starting just before month 14, two months after the *Wave 1* kick off. As envisioned at program/portfolio level, we observe a positive trend. *Observation 2*: At month 25 a global cost-saving program was introduced within all units, displaying a flattening and subsequent decrease of *Cost of Delay*, a plausible explanation for this negative trend, since the organization was not able to focus on value delivery. WSJF measurements, the next identified measure of *productivity*, show the exact same trend.

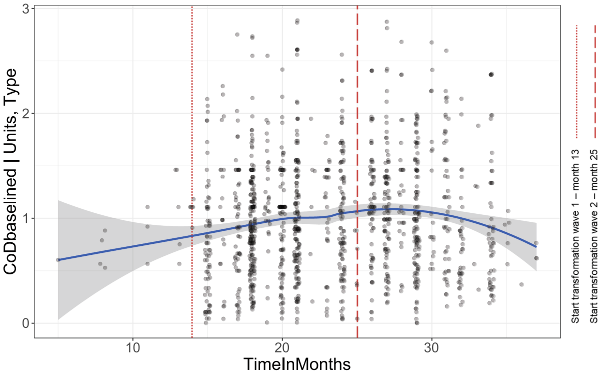


Fig. 1. Added Variable Plot (AVP) of Cost of Delay for units, baselined per issue type and unit over time (months). The value of 1 therefore represents the baseline. Outliers >3 have been discarded in the plot, to help improve the visualization quality. Dots represent resolved issues. Confidence bands and fitted line based on Loess.

[3] A *baseline* is essential since estimations are not standardized at FinOrg. Values are divided by its mean in the context of the unit and issue type.

4.3 Trends Across Organizational Layers, Focus on Responsiveness

Another sample demonstration looks at the trends in *responsiveness* (TTM) including diving into layers and units (Fig. 2). This proved helpful while deepening insights into the dynamics of flow. The impact of local interventions to improve refinement processes, creating better-sized and better-defined chunks of work, is visible over time. It reveals significant differences. One illustration: all trains started with the mandatory use of program/portfolio Epics at *Wave 1* (month 12), meaning that all initiatives had to be registered and estimated. Note that one ART (U02) already used features and greatly reduced the TTR for these items during the three years, mainly by defining smaller chunks of work. However, this downsizing of items at U02 did not lead to worsening TTR results at team level; rather the opposite seems true: more items were delivered and there were better rates of TTR for this level as well. Overall, we see decreasing TTR values, which is in line with the envisioned improvement on the TTM objective.

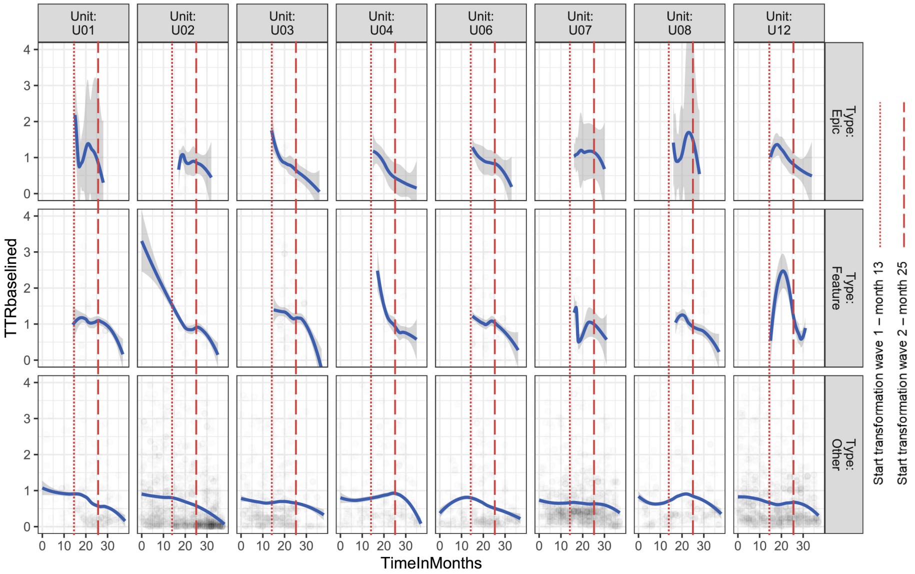


Fig. 2. Baselined Time-to-Resolve (TTR) measurements for ARTs U01-U12 across the three organizational layers and transformation *Waves* 0-2

4.4 Understanding Transformation Success

In this section we report on a way to provide evidence regarding the overall success of this transformation. For this purpose we compare data sets of the transformation on the program & portfolio level of *Wave 1* to *Wave 2* in Table 2. A summary of our findings:

Table 2. Impact (%) *Wave* 1 versus 2 transformation program level across ARTs on objectives

Agile release train (unit):		**U01**	**U02**	**U03**	**U04**	**U06**	**U07**	**U08**	**U12**	**AVG**
(a) Responsiveness										
Time-to-resolve Epics (ii)	`TTM`		−24%	−50%	−66%	−37%			−33%	**−34%**
Time-to-resolve Feat. (ii)	`TTM`	−20%	−22%	−25%	−42%	−24%	−30%	25%	−24%	**−27%**
Time-to-resolve Team (ii)	`TTM`	−52%	−68%	−22%	−35%	−45%	−48%	−33%	−35%	**−47%**
(b) Productivity										
Cost of Delay Epics (i)	`PROD`		−16%	21%	179%	116%			24%	**30%**
Cost of Delay Feat. (i)	`PROD`	−1%	−7%	−2%	315%	69%	51%	54%	−76%	**32%**
WSJF Epics (iv)	`PROD`		163%	4%	297%	181%			90%	**73%**
WSJF Features (iv)	`PROD`	13%	26%	253%	88%	99%	303%	64%	−92%	**114%**
(c) Workflow health										
JobSize Epics (i)	`PROD-TTM`		−62%	74%	−39%	−28%			−58%	**−22%**
JobSize Features (i)	`PROD-TTM`	−14%	−25%	−78%	149%	31%	−26%	−11%	200%	**−24%**
# items Epics (iii)	`PROD-TTM`		−59%	−68%	−45%	−76%			−44%	**−70%**
# items Features (iii)	`PROD-TTM`	48%	−39%	−55%	73%	−63%	−90%	−32%	667%	**−36%**
# Team Issues (iii)	`PROD-TTM`	−6%	47%	−70%	18%	−47%	−30%	−29%	−7%	**−15%**
Autonomy (iv)	`EMPL`	11%	13%	47%	−23%	−3%	36%	35%	−132%	**21%**
Complexity (iv)	`EMPL`	173%	−21%	−3%	21%	9%	−100%	−24%	48%	**−7%**
(e) Quality										
QbD # items (iii)	`QBD`	−63%	−66%	−29%	−66%	−35%	−69%	−41%	−58%	**−59%**
QbD time-to-resolve (i)	`QBD`	−45%	−55%	−47%	−47%	7%	−56%	−35%	−51%	**−41%**

Scales: **(i)** *pseudo*, Fibonacci 1–250, **(ii)** hours, **(iii)**, items, **(iv)** custom calculated measure at FinOrg: $complexity = \frac{assignees*handovers}{channels}$, $channels = \frac{assignees*(assignees-1)}{2}$ (i.e., rule of thumb number of communication lines), $autonomy = \Delta\ dependecies.$

☑ `(a)` (`TTM`): *Transformation improves (a) responsiveness.* At all issue layers a substantial improvement (i.e., decrease in resolve time) was observed, ranging from team-level improvements of over 47% to epic-level ones of 34%.

☑ (b) (`PROD`): *Transformation improves (b) productivity.* We are able to report that more value has been delivered (Cost of Delay) (30% epic level, 32% feature level) and *better* priorities (WSJF) have been set (73% feature level, 114% epic level). Notice the (large) improvements within some units which can be explained by interventions improving the WSJF estimation and prioritization events and redefining epics and features. Note that some units (U01, U08) did not report any resolved epic items.

□ (c) (`PROD`,`TTM`): *Transformation improves (c) workflow health.* Data from our case study displays an ambivalent picture. Averages of resolved Job Sizes decreased on both levels (22% features, 24% epics), which should be evaluated in the context of *number of items.* In Fig. 2 one can deduct that indeed the number of Epics and Features are decreasing since less data point are visible in more recent months. A reasonable explanation for this is creating less overarching items like epics and shift to smaller (right-sized) items facilitating a better flow. It is therefore interesting to observe the dynamics between priority setting (WSJF) and TTR to this dimension. Furthermore, note the differences in unit results: U04, for example, shows workflow decreases on Epic level, but improvements on other levels, which indicates that focus shifted to the delivery of smaller-sized items. Note: the lagging performance of U03 can be explained by specific challenges.

☑ (d) (QBD): *Transformation improves quality.* Data from our case study display interesting results. The number of QbD items decreased, while the resolve time improved. A plausible explanation is that since the number of resolved epics (initiatives) decreased, a decrease of associated QbD items is to be expected as well. The overall QbD numbers are positive: the ratio of quality aspects is congruent with the initiatives and the handling of the QbD aspects improved over time (TTR 41%).

□ (e) (EMPL): *Transformation improves employee engagement.* By not using subjective survey data on engagement, we fail to report on this measurement. However, we are able to report on *autonomy* and a (custom) *complexity* measurement as part the *(c) workflow health* category and report an increase in autonomy (21%) and a decrease in complexity (7%).

5 Discussion

5.1 Using Backlog Data to Guide Transformations Based on Trends

We will now continue to discuss how Jira data contributes to the understanding of the transformation impact and trends in relation to the five dimensions of impact established in agile literature and subsequently to the Balanced Scorecard (BSC). Figure 3 provides insights in how measures, objectives and perspectives are linked by establishing a connection between the Balanced Scorecard, the impact dimensions, and the measurements conducted during the transformation at FinOrg. The perspectives of the BSC as presented by Kaplan and Norton [16,17], offer a holistic view on the dimensions of organizational performance in contrast to the empirically, bottom-up understanding of impact of agile transformations as presented by Stettina et al. [33]. Plotting Jira backlog data over time and projecting data in multiple layers, as discussed in this paper, allows for zooming into organizational layers and trend analyses provide valuable augmentation.

Firstly, one can observe that the *Time-to-Resolve* and *Items-time (resolved)* on Epic, Feature and Team level augment the *Responsiveness* dimension. This dimension contributes to *Learning & Growth* through the opportunity of providing faster feedback through faster delivery. Based on the baselined Time-to-Resolve plots in Fig. 2, one can confirm the envisioned trend of decreasing resolve time. In Sect. 4.3 we discuss how smaller slices of Features contribute to lowering TTR using the example of U02. A further general observation that can be made when looking at Fig. 2, is that the impact differs significantly per organizational layer, as previously suggested by Stettina et al. [33].

Secondly, one can observe how the measures of *Cost of Delay* and *WSJF* contribute to the dimension of Productivity as they represent how implemented Epics, Features and Stories link to prioritization given by the customer. Here one assumes that a better adherence to previously defined customer issue priorities

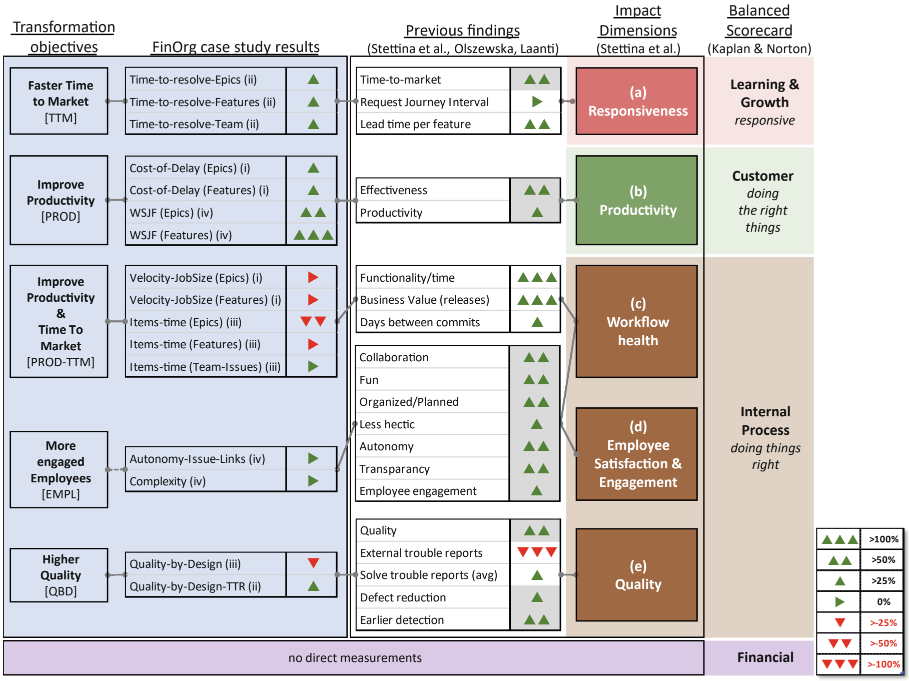

Transformation objectives
FinOrg case study results
Previous findings (Stettina et al., Olszewska, Laanti)
Impact Dimensions (Stettina et al.)
Balanced Scorecard (Kaplan & Norton)
Faster Time to Market [TTM]
Time-to-resolve-Epics (ii) ▲
Time-to-resolve-Features (ii) ▲
Time-to-resolve-Team (ii) ▲
Time-to-market ▲▲
Request Journey Interval ▶
Lead time per feature ▲▲
(a) Responsiveness
Learning & Growth
responsive
Improve Productivity [PROD]
Cost-of-Delay (Epics) (i) ▲
Cost-of-Delay (Features) (i) ▲
WSJF (Epics) (iv) ▲▲
WSJF (Features) (iv) ▲▲▲
Effectiveness ▲▲
Productivity ▲
(b) Productivity
Customer
doing the right things
Improve Productivity & Time To Market [PROD-TTM]
Velocity-JobSize (Epics) (i) ▶
Velocity-JobSize (Features) (i) ▶
Items-time (Epics) (iii) ▼▼
Items-time (Features) (iii) ▶
Items-time (Team-Issues) (iii) ▶
Functionality/time ▲▲▲
Business Value (releases) ▲▲▲
Days between commits ▲
(c) Workflow health
Collaboration ▲▲
Fun ▲▲
Organized/Planned ▲▲
Less hectic ▲
Autonomy ▲▲
Transparancy ▲▲
Employee engagement ▲
(d) Employee Satisfaction & Engagement
Internal Process
doing things right
More engaged Employees [EMPL]
Autonomy-Issue-Links (iv) ▶
Complexity (iv) ▶
Higher Quality [QBD]
Quality-by-Design (iii) ▼
Quality-by-Design-TTR (ii) ▲
Quality ▲▲
External trouble reports ▼▼▼
Solve trouble reports (avg) ▲
Defect reduction ▲
Earlier detection ▲▲
(e) Quality
no direct measurements
Financial
▲▲▲ >100%
▲▲ >50%
▲ >25%
▶ 0%
▼ >-25%
▼▼ >-50%
▼▼▼ >-100%

Fig. 3. Overview of case study results and objectives (blue, 1st block), literature (2nd block). Last column connection to BSC perspectives. Shaded gray results: converted Likert scales of qualitative survey results. (Color figure online)

leads to better performance as previously described in literature [10,11]. Figure 1 plots aggregated *Cost of Delay* values for the delivered issues over all units delivered to the customer. Based on the plot one can recognize positive as well as negative trends. Specifically, the implementation of the program & portfolio layer transformation of *Wave 1* indicates a positive impact on *Cost of Delay* values. The negative effect of a cost-saving program on performance due to loss in focus on value delivery can be visually identified to be starting in month 25.

Thirdly, the measures of *Autonomy*, represented by the number of dependencies linked in Jira across the implemented issues, as well as *Complexity*, represented by communication complexity (refer to notes Table 2), provide an indication for *Employee Satisfaction & Engagement*.

Fourthly, one can observe how the *Quality by Design* issues, can serve as an indicator for *Quality* improved. The perspective taken here is that the quality of design requirements and lower TTR values lead to better quality of the product. We point out that quality aspects are executed with improved speed and with fewer items. This indicates an improvement in quality by design, especially in the context of firmly enforced protocols and a rigorous (internal and external) audit process. In that respect we may exclude possible manipulation of measurements.

In line with previous findings of Lin et al. [23] we argue that unobtrusiveness and transparency are key success factors to using backlog data. To address this the measurements at FinOrg have been automated and made available in real time. The system is an integral part of the way of working, in other words, no extra effort is needed and, since the system provides relevant insights for users, they are motivated to maintain (1) high data quality and, (2) the inherent openness reduces the risk of *gaming* (cf. [18]). In addition, understanding how measures are interconnected and using more than one measure per objective strengthen the (3) reliability of the results. As an example over- or underestimating *Job Size* will show up in relation to the *Time-to-Resolve* and *number of items* measurements denoted by the connecting lines.

5.2 Transformation Success at FinOrg Compared to Prior Evidence

We will now continue to elaborate on the main question: *Can we declare transformation success based on FinOrg's objectives and what if we compare these to prior findings?* Fig. 3 presents the results of our case study (Sect. 2.3) and connect these to the (most) conservative findings from the literature from Stettina et al. [33]. Both categorized into seven levels (refer to legend). Based on the backlog data we were able to identify improvements on three of FinOrg's five transformation objectives.

(1) Improve Productivity (`PROD`) by >30%. Note: existing literature reports on effectiveness values >60%. However, we cannot confirm the significantly higher results reported in existing literature with regard to the *workflow health* dimension; linked to both *productivity* and *time to market* (`PROD-TTM`) as described in Sect. 3.2: *Functionality/time* (483%), *Business Value* (400%), *Days between commits* (38%). Moreover, the results of *number of delivered items* and *velocity* decreased and we postulated as explanation the shift from epics and features to better defined and smaller sized (team level) items.

(2) Faster Time To Market (`TTM`) by >27%; Note: existing literature reports higher numbers: *time-to-market* survey results (67%), and the *request journal interval* measurements (24%) and *lead time* (64%).

(3) Higher Quality (`QBD`) by >41%. We used the (leading) indicator of FinOrg: the quality-by-design measurement. The prior literature focuses on defects and incident/problem data, and in that respect focuses on lagging indicators and therefore comparison is problematic.

With the use of backlog data we were not able to look at (4) *Customer Satisfaction*, (5) *Employee Satisfaction & Engagement* as well as the *Financial* Balanced Scorecard perspective (not part of the FinOrg transformation objectives). Lacking measurements on customer feedback (i.e., customer satisfaction `CUST`) and employee satisfaction we argue that the perspectives can be improved using additional surveys or direct user experience data.

5.3 The Need for a Performance Management Framework

Our challenges with regard to the comparison and interpretation of measures and results in the literature indicate a need for more research on performance measurement, a topic often discussed but rarely defined (cf. [27]). It is important to learn how measurement (systems) can support, facilitate, and impact the change process and performance of an organization, especially in the context of agile transformations. There is sufficient motivation to suggest that the use of performance management systems can lead to improved capabilities, which then impact performance (cf. [13,20]). Advantages reported in the literature are higher results orientation, better strategic clarity, higher employee engagement, and quality. Reasons for use are improved focus on control and strategy [38]. An interesting area to pursue would be to verify these findings in the context of agile transformations. A way forward is to improve our understanding of measures (e.g., performance, productivity, effectiveness, and efficiency cf. [12,14,32,36]) and enhance the exploratory mapping we introduced with Balanced Scorecard perspectives in the context of agile transformations. Combining multiple sources of quantitative measurement of backlog with qualitative data such as surveys, customer experience data and (inter)subjective estimation data (e.g., Job Size and Cost of Delay estimations) need to be researched further.

5.4 Limitations and Threats to Validity

This report describes an exploratory data analysis of a case study demonstrating a proof of concept of using backlog data to measure agile transformations. An exploratory analysis imposes requirements on traceability on how data has been collected and used. We documented and automated all steps in gathering and transformation of the data, alongside our decisions not to use specific data (e.g., exclude dormant backlogs, exclude units and document outliers). In addition, since the data was transparently available, presented, and used throughout the whole organization, potential errors, deficiencies, or lack of quality in registering and maintaining data are largely eliminated. Finally, we were able to use an extensive data set ranging over a long period of time (36 months), which mitigates data-maturity issues. We therefore claim high *reliability.* With respect to *construct validity*, we used Jira software as a single data source. As noted, we paid considerable attention to the care, depth, and quality of data. In addition to this, we reviewed data and findings with relevant stakeholders at FinOrg. Finally, for all categories of measurements we used multiple measurements in order to substantiate the outcomes. Construct validity can be further improved by extending the research to other data sources and tool-providers and by doing so provide additional insights and knowledge on how to combine different data sources. Using substantial time series data, validating results and trends over multiple units, and providing plausible explanations on differences between units all strengthen the *internal validity* of our research. We suggest that further research on objective measurement attributes is a productive avenue to pursue, e.g., financial measures, experience and usage data on services, and problem and incident data. With respect to *external validity*, we used a case study with release trains as embedded units. These units are clearly defined, act within the same transformation context, and are therefore suitable for comparison in an exploratory case study. Finally, we projected our findings in the context of current literature. These efforts strengthen the external validity of our research. However, we recognize that broadening the scope to other organizations and branches, repeating our analysis, will improve generalization evidence.

6 Conclusions

The objective of this report was to discuss if, and how, backlog data can be used to help guide agile transformation journeys towards improved organizational performance. We conducted an exploratory embedded multiple-unit case study to identify trends and measure their development against FinOrg's five transformation objectives. We used Jira backlog data from eight Agile Release Trains and their teams over a period of three years, with a total of over 57,000 issues, supplemented by engagement of the first author in the transformation.

Our contribution is threefold: Firstly, we provide a proof of concept of how backlog data can be used to identify trends and provide guidance by creating a mapping of Jira data sources to impact dimensions proposed by Stettina et al. [33] as well as the Balanced Scorecard. Secondly, we provide empirical evidence on the assessment of transformation objectives over time at FinOrg. And thirdly, we compare our measurements to previously available literature.

We find evidence pointing towards improvements on three of FinOrg's five transformation objectives: (1) *improved productivity*, (2) *faster time to market*, and (3) *higher quality*. Backlog data did not enable us to report on *customer satisfaction* and *engaged employees*. We observe that results are in line with the current literature, although in trends rather than in absolute numbers. It is important to consider the point of departure of the transformation as context for the measurement of success or comparison.

We may conclude that backlog data can help guide agile transformations. By mapping Jira data to the impact dimensions as discussed in available literature, this report describes how backlog data provides a viable source of information to recognize trends and guide agile transformations and allows organizations to act upon them. Authors suggest to complement measurements with other data sources and apply a measurement framework as proposed here.

References

1. Basili, V.R., Caldiera, G., Rombach, H.D.: The goal question metric approach (1994)
2. van Bennekum, A., Beck, K., Schwaber, K., Fowler, M., Sutherland, J., et al.: Agile manifesto (2001). https://agilemanifesto.org/
3. Biesialska, K., Franch, X., Muntés-Mulero, V.: Big data analytics in agile software development: a systematic mapping study. Inf. Softw. Technol. **132**, 106448 (2021)
4. Blincoe, K., Dehghan, A., Salaou, A.-D., Neal, A., Linaker, J., Damian, D.: High-level software requirements and iteration changes: a predictive model. Empir. Softw. Eng. **24**(3), 1610–1648 (2018). https://doi.org/10.1007/s10664-018-9656-z
5. Boerman, M.P., Lubsen, Z., Tamburri, D.A., Visser, J.: Measuring and monitoring agile development status. In: 2015 IEEE/ACM 6th International Workshop on Emerging Trends in Software Metrics, pp. 54–62. IEEE (2015)
6. Conoscenti, M., Besner, V., Vetrò, A., Fernández, D.M.: Combining data analytics and developers feedback for identifying reasons of inaccurate estimations in agile software development. J. Syst. Softw. **156**, 126–135 (2019)
7. Czekster, R.M., Fernandes, P., Lopes, L., Sales, A., Santos, A.R., Webber, T.: Stochastic performance analysis of global software development teams. ACM TOSEM **25**(3), 1–32 (2016)
8. Dehghan, A., Neal, A., Blincoe, K., Linaker, J., Damian, D.: Predicting likelihood of requirement implementation within the planned iteration: an empirical study at IBM. In: 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), pp. 124–134. IEEE (2017)
9. Digital.ai: The 15th state of agile report (2021). https://stateofagile.com
10. 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)
11. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: a systematic review. Inf. Softw. Technol. **50**(9–10), 833–859 (2008)
12. Forsgren, N., Hering, M., DeGrandis, D., Guckenheimer, S.: Measure efficiency, effectiveness, and culture to optimize DevOps transformations - metrics for DevOps initiatives. DevOps Enterp. Forum (2015). IT Revolution
13. Franceschini, F., Galetto, M., Turina, E.: Techniques for impact evaluation of performance measurement systems. Int. J. Qual. (2013)
14. Grafton, J., Lillis, A.M., Widener, S.K.: The role of performance measurement and evaluation in building organizational capabilities and performance. Acc. Organ. Soc. **35**(7), 689–706 (2010)
15. IEEE Computer Society: SWEBOK Guide v3.0 (2004). http://www.swebok.org/
16. Kaplan, R.S., Norton, D.P.: The balanced scorecard-measures that drive performance (1992)
17. Kaplan, R., Norton, D.: Using the balanced scorecard as a strategic management system (1996). jackson.com.np
18. Kennerley, M., Neely, A.: Measuring performance in a changing business environment. Int. J. Oper. Prod. Manag. **23**(2), 213–229 (2003)
19. Knaster, R., Leffingwell, D.: SAFe 4.5 Distilled: Applying the Scaled Agile Framework for Lean Enterprises. Addison-Wesley Professional (2018)
20. Koufteros, X., Verghese, A.J., Lucianetti, L.: The effect of performance measurement systems on firm performance: a cross-sectional and a longitudinal study. J. Oper. Manag. **32**(6), 313–336 (2014)

21. Laanti, M., Salo, O., Abrahamsson, P.: Agile methods rapidly replacing traditional methods at Nokia: a survey of opinions on agile transformation. Inf. Softw. Technol. **53**(3), 276–290 (2011)
22. LeSS: Large-Scale Scrum. https://less.works/
23. Lin, J., Yu, H., Shen, Z., Miao, C.: Studying task allocation decisions of novice agile teams with data from agile project management tools. In: Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering, pp. 689–694 (2014)
24. Meyer, M.W.: Rethinking Performance Measurement: Beyond the Balanced Scorecard. Cambridge University Press, Cambridge (2003)
25. Moran, A.: Managing Agile. Strategy, Implementation, Organisation and People (2015)
26. Neely, A., Adams, C., Crowe, P.: The performance prism in practice. Measuring Bus. Excellence (2001)
27. Neely, A., Gregory, M., Platts, K.: Performance measurement system design: a literature review and research agenda. Int. J. Oper. Prod. Manag. **25**, 1228–1263 (2005)
28. Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., Porres, I.: Quantitatively measuring a large-scale agile transformation. J. Syst. Softw. **117**, 258–273 (2016)
29. Pospieszny, P., Czarnacka-Chrobot, B., Kobylinski, A.: An effective approach for software project effort and duration estimation with machine learning algorithms. J. Syst. Softw. **137**, 184–196 (2018)
30. SAFe®: Scaled Agile Framework. https://scaledagileframework.com/metrics/
31. Snyder, B., Curtis, B.: Using analytics to guide improvement during an agile-DevOps transformation. IEEE Softw. **35**, 78–83 (2017)
32. Sproles, N.: Identifying success through measures. Phalanx **30**(4), 16–31 (1997)
33. Stettina, C.J., van Els, V., Croonenberg, J., Visser, J.: The impact of agile transformations on organizational performance: a survey of teams, programs and portfolios. In: Gregory, P., Lassenius, C., Wang, X., Kruchten, P. (eds.) XP 2021. LNBIP, vol. 419, pp. 86–102. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78098-2_6
34. Stettina, C.J., Heijstek, W.: Necessary and neglected? an empirical study of internal documentation in agile software development teams. In: Proceedings of the 29th ACM International Conference on Design of Communication, pp. 159–166 (2011)
35. Stettina, C.J., Schoemaker, L.: Reporting in agile portfolio management: routines, metrics and artefacts to maintain an effective oversight. In: Garbajosa, J., Wang, X., Aguiar, A. (eds.) XP 2018. LNBIP, vol. 314, pp. 199–215. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91602-6_14
36. Tangen, S.: Demystifying productivity and performance. Int. J. Product. Perform. Manag. **54**(1), 34–46 (2005)
37. Tukey, J.W., et al.: Exploratory Data Analysis, vol. 2, Reading (1977)
38. de Waal, A., Kourtit, K.: Performance measurement and management in practice. Int. J. Product. Perform. Manag. **62**(5), 446–473 (2013)
39. Wedman, J., Graham, S.W.: Introducing the concept of performance support using the performance pyramid. J. Continuing High. Educ. **46**(3), 8–20 (1998)

40. Wiraeus, D., Creelman, J.: Agile Strategy Management in the Digital Age. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-76309-5
41. Yin, R.K.: Case Study Research. Sage, Thousand Oaks (2003)
42. Öztayşi, B., Uçal Sari, I.: Comparing MADM techniques for use in performance measurement (2009)

Work Engagement in Agile Teams: The Missing Link Between Team Autonomy, Trust, and Performance?

Marte Pettersen Buvik(✉) and Anastasiia Tkalich

SINTEF Digital, Trondheim, Norway
{marte.p.buvik,anastasiia.tkalich}@sintef.no

Abstract. To have engaged and high-performing agile teams are what most organizations strive for. At the same time, there is little research on the drivers of team work engagement in the software context. Team autonomy and trust are crucial for agile teams and are suggested as potential boosters of team work engagement and performance. In this study, we apply the Job Demands-Resources model to examine the role of autonomy and trust and their impact on work engagement and team performance in agile teams. We analyze quantitative survey data from 236 team members in 43 agile teams to examine how team autonomy and trust relate to team work engagement and how engagement mediates the relationship between these factors and performance. Our results show that while both autonomy and trust are positively related to team work engagement, team trust plays a more critical role than team autonomy. Teams with high team trust showed higher engagement, which enhanced team performance. Our results highlight the importance of social factors such as trust in creating conditions for high performance in agile teams through its effect on team work engagement.

Keywords: Agile teams · Team performance · Trust · Team autonomy · Work engagement · Job demands-resource model

1 Introduction

Having high-performing agile software development teams is what most organizations operating in the field strive for. Among the numerous determinants of team performance, *autonomy* and *trust* deserve special attention when it comes to agile teams. Team autonomy is considered crucial for team performance because it allows teams to self-organize and make better decisions without needing to wait for approval [1, 2]. When it comes to team trust, it has been found to be one of the fundamentals of agile teams [3] as it creates favorable conditions for cooperation by strengthening the interactions between team members and improves problem-solving and overall software quality because team members that trust each other are more likely to share knowledge and report problems.

Although both team autonomy and trust are acknowledged as crucial for agile teams, there is a lack of theoretical explanation for how these factors impact performance. One possible explanation may be found in the Job Demands-resource model (JD-R), which

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 131–147, 2022.
https://doi.org/10.1007/978-3-031-08169-9_9

depicts *work engagement* as a mediator of the relationships between job resources (e.g., team autonomy and trust) and performance [4]. In other words, factors such as team autonomy and trust may relate to work engagement, while work engagement, in its turn, relates to team performance.

Work engagement is in itself important for agile teams because it is closely related to the concept of motivation. According to the 5th principle in the agile manifesto, agile projects should be built around motivated individuals that have support and trust to get the job done. Motivation has been described as an important issue in software engineering [5], and job enthusiasm has been highlighted as the strongest predictor of developers' productivity [6]. Motivated teams are also highly engaged, which means they are full of energy, enthusiastic about their work, and persist when facing drawbacks. Research shows that engaged teams outperform teams with low levels of engagement [7].

Recently the interest in work engagement is starting to emerge in the field of agile. For example, Huck-Fries et al. [8] demonstrate that work engagement in agile teams is indeed influenced by job resources and that agile practices are positively related to these resources. However, there is still insufficient insight into the effects of job resources and work engagement on the performance of agile software development teams. Against this background, we are suggesting the following research question: *What are the effects of team autonomy and team trust on team work engagement and team performance in agile software development teams?*

To answer this question, we develop and test a statistical research model that investigates how team autonomy and trust relate to team work engagement and how team work engagement mediates the relationship between these resources and team performance. We use survey data from 236 team members in 43 software development teams in Norway. Our results have important theoretical and practical implications for the field of agile development and contribute to the existing literature in several ways. First, we show how a well-established psychological theory (JD-R) can be successfully applied to examine agile teams. Second, we expand the research on JD-R theory by including the team level of analysis. And third, we provide valuable theoretical as well as practical insights by showing how team autonomy and trust relate to work engagement and performance of agile teams.

2 Related Work and Hypothesis Development

2.1 Team Work Engagement in Agile Software Development Teams

Software development teams now commonly adopt agile methods, which emphasize the importance of a collaborative, people-oriented approach with the use of self-organizing teams with high levels of autonomy [1, 9]. With the increased use of teams in software development, there is a growing recognition of factors that influence the performance of teams in this context. While the Agile Manifesto is based on the idea of highly motivated team members [10], empirical research on work engagement in the agile development literature is still limited.

Work engagement can be defined as a positive, fulfilling, work-related state of mind characterized by vigor, dedication, and absorption [11]. Vigor is described by high levels of energy while working and persistence in the face of difficulties. Dedication refers to being strongly involved in one's work and experiencing a sense of significance, enthusiasm, and strong identification with the work. Absorption means being fully concentrated and immersed in one's work and difficulties with detaching oneself from work. In sum, engaged employees feel full of energy, are enthusiastic about their work, and often lose track of time when working. Based on an abundant amount of research, work engagement has been found to have numerous benefits, such as organizational commitment, job satisfaction, extra-role behavior, and superior work performance, as well as increased well-being and general health [12]. Although most studies on work engagement focus on the individual level of measurement, the concept also exists at a team level [7, 13]. *Team work engagement* (TWE) describes a shared perception of work engagement of the team as a whole and can be defined as "a positive, fulfilling, and shared emergent motivational state that is characterized by team vigor, team dedication, and team absorption, which emerges from the interactions and shared experiences of members of a team" [13].

2.2 Work Engagement and the Job Demands-Resource Model

The JD-R model has frequently been used as a framework to explain the antecedents and consequences of work engagement [14]. According to the JD-R model, working conditions can be broadly classified into two categories; job demands and job resources. Job demands are the aspects of the job that require sustained physical and/or psychological effort and are therefore associated with certain costs. Examples are high work pressure, role conflict, and emotionally demanding interactions. Job resources refer to the job-related aspects that are functional in achieving work goals that allow employees to cope with the demanding aspects of their job and stimulate their learning and development [14]. Job resources may exist at different levels: the task level (e.g., job autonomy), the social level (e.g., team climate), and the larger organizational level (e.g., organizational justice). The JD-R model further suggests that job demands and job resources trigger two distinct psychological processes: health impairment and the motivational process. The health-impairment process posits that poorly designed jobs or constant job demands exhaust employees' resources resulting in stress and health problems [15]. The motivational process, on the other hand, proposes that job resources both have intrinsic and extrinsic motivational potential and lead to high work engagement. Resources are intrinsically motivating because of their capacity to fulfill basic human needs such as autonomy, belongingness, and competence [16], and may also be extrinsically motivating because they translate into instrumental help that allows employees to successfully achieve work goals [14]. Research has consistently shown that job resources are the strongest predictors of work engagement due to their potential to enable employees to cope with demanding aspects of their job and, at the same time, stimulate personal growth, learning, and development [12, 17].

Some recent research indicates that agile work practices have a positive effect on work engagement through job resources [8, 18, 19]. Huck-Fries et al. [8] found, for instance, that agile practices significantly influenced the job resources of job autonomy and perceived meaningfulness, which again positively predicted team members' work engagement. Similarly, Rietze and Zacher [19] demonstrated that agile work practices were positively related to job resources such as autonomy, peer support, and feedback and indirectly influenced work engagement via these job resources. Neither of these studies, however, studied job resources and work engagement at the team level. Further, the mediating effect of work engagement on the relationship between job resources and team performance is lacking in the previous studies on work engagement in agile software development teams. In the software engineering literature, team autonomy and team trust have continuously been identified as central to the effectiveness of agile software teams [3] and have also been recognized as important resources in the work engagement literature [20].

2.3 Team Work Engagement, Team Autonomy, and Trust

While many types of job resources may boost work engagement [14], previous meta-analyses and reviews suggest that resources at the task level, such as autonomy, are strong drivers for work engagement [17, 21]. Indeed, recent findings indicate that team autonomy is positively related to work engagement, suggesting that team members with a voice in allocating tasks, managing time, and defining leadership roles express greater vigor, dedication, and absorption at work [22]. Team autonomy is a key principle of agile practices and is recognized as an important condition for the responsiveness and effectiveness of agile software development teams [1]. Team autonomy can be defined as the extent to which the team has considerable discretion and freedom in deciding how to carry out tasks [23]. The increased levels of autonomy in the team bring decision-making authority directly to the operational level resulting in increased speed and accuracy of problem-solving [1]. The self-determination theory (SDT) also suggests that autonomy triggers the motivation of team members and may thus increase the level of engagement. Muecke and Greenwald [24] suggest that autonomy influences work engagement through both motivational and cognitive mechanisms, leading to job enrichment. The motivational perspective suggests that autonomy affects work engagement by influencing employees' feelings of personal responsibility for work outcomes, feelings of mastery, and increased chances for learning and growth, all leading to higher motivation [25, 26]. The cognitive perspective focuses on the cognitive demands caused by job autonomy, such as increased problem-solving and information processing. As autonomy increases, employees are allowed to choose suitable strategies to deal with situations, resulting in more cognitive activities and higher cognitive demands that promote work engagement. Based on this review, we, therefore, hypothesize that: *H1: Team autonomy is positively related to team work engagement.*

Trust in the team represents a potentially vital job resource for agile teams because trust constitutes a central determinant of effective teamwork [27, 28] and has been found to play a crucial role in the functioning of teams in this context [3]. Trusting one's teammates implies positive expectations about their actions and motivation grounded in the belief of their competence, integrity, and benevolence [29]. It is proposed that a high level of trust within the team can positively boost the team's work engagement in several ways. For example, if team members trust their fellow teammates, they are confident that they have the competence to do their job and would not intentionally do anything to compromise them or the team. This could influence the motivation of team members and the collective engagement in the team. The confidence in their fellow team members may also increase their willingness to commit themselves to the goals [27] and increase their level of work engagement. By contrast, if team members lack confidence in their fellow team members and feel that they are not competent to do their tasks, they may not exert the effort and energy necessary for the team to succeed. In addition, if team members believe that their co-workers are consistent and would do what they say they will do, this could contribute to higher work engagement because they would be able to focus on achieving their tasks and goals as opposed to expending their energy and focus on monitoring and controlling actions of their fellow team members. Also, the support, mutual respect, and encouragement of fellow teammates provide team members with feelings of being accepted and cared for, satisfying their need for belonging and relatedness [16], thus increasing their work engagement. In addition, trust within the team has been found to facilitate the open sharing of knowledge and ideas in teams [28]. The increased sharing of knowledge and the presence of shared information may boost the team's engagement [30]. Trust as a resource at the team level has not been extensively studied in the work engagement literature. However, related factors such as social support have frequently been included in the work engagement and JD-R studies. At the team level, Torrente et al. [7] found that social resources such as supportive team climate, collaboration, and teamwork were positively related to team work engagement. Based on this review, we hypothesize that: *H2: Team trust is positively related to team work engagement.*

2.4 Team Work Engagement as a Mediator Between Job Resources and Team Performance?

Both the JD-R model and the SDT propose that engagement leads to a higher level of performance because of the fulfillment of psychological needs, which enhances intrinsic motivation. Indeed, work engagement at the individual level has been found to predict task performance and extra-role performance [17]. Christian et al. [17] suggest that engaged employees are more persistent and pursue their tasks with more intensity, making them more focused on their work tasks and thus promoting higher task performance. While the empirical studies on team work engagement so far are relatively limited, some findings show a positive relationship between team work engagement team performance, with engaged teams outperforming teams with lower levels of engagement [7]. Explanations for this might be that engaged teams are able to maintain high motivational levels, resulting in greater commitment to collective goals and focused action on goal achievement [31]. Furthermore, engaged team members consider their work meaningful and relevant [32]. Also, engaged teams create a positive and activated affective climate that is characterized by high levels of energy and feelings of pleasure while working. This positive affective climate is beneficial for the performance of teams. Based on this, we hypothesize that: *H3: Team work engagement is positively related to team performance.*

The JD-R model proposes that work engagement mediates the impact of job resources on organizational outcomes [33]. Previous research has lent support for the mediating role of engagement, indicating that resources at the team level will have an indirect effect on team performance. Indeed, Torrente et al. [7] reported evidence for at mediation role of team work engagement between social resources and team performance in their sample of 63 teams. And Costa et al. [32] also showed that team members job resources positively affected work engagement and, consequently, team performance. In line with this, we propose H4*: Team work engagement mediates the relationships between team autonomy and team performance. H5: Team work engagement mediates the relationships between team trust and team performance.*

Taken together, we hypothesize that the job resources, team autonomy, and team trust will both be positively related to team work engagement (H1 and H2). Team work engagement again will positively influence team performance (H3) and will mediate the relationship between team autonomy and team performance (H4) and team trust and team performance (H5). Figure 1 illustrates our research model and hypotheses.

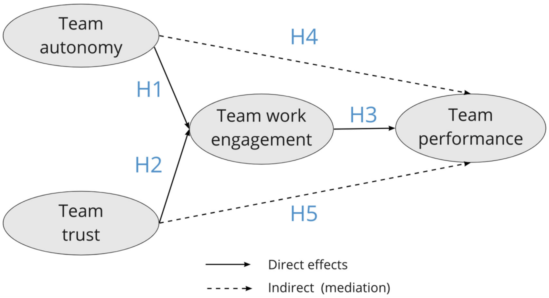


Fig. 1. The research model and the hypothesis

3 Method

In this section, we outline the sample and its context, the data collection process, the measures employed, and the statistical procedures used.

To test the proposed hypotheses, we conducted a quantitative study with survey data from software development teams in four companies in Norway, representing IT consultancy within software development and fintech. The teams included in the survey employ various agile practices, which are summarized in Table 1, along with information about the industry, number of employees, and number of teams included in the study.

Table 1. Description of the sample and its context

Company	A	B	C	D
Industry	IT consultancy	IT consultancy	FinTech	FinTech
No. of employees	150	750	2000	300
No. teams	7	12	14	10
Agile practices	Customer-centered teams with high autonomy using agile practices influenced by both Scrum and Kanban, including daily standups, backlog grooming and iterative planning	Self-organizing teams with common practices from Scrum and Kanban, such as standups, retrospectives, sprints, product backlogs, and visual task boards	Cross-functional teams with a combination of Scrum and Kanban with daily standups, sprint planning, backlog grooming, retrospectives, and lean startup principles	Cross-functional teams with Kanban-inspired ways of working, including daily standup, retrospective, and iterative planning

Email addresses from team members working in software teams were provided to the researchers, and the questionnaire was distributed and collected electronically via an online survey platform. All participants were given information about the purpose, data protection, and confidentiality before accepting the invitation to participate. In total, 239 team members from 45 teams responded. Two teams were excluded from the sample because they had fewer than three participants, leaving us with a final sample consisting of 236 team members from 43 teams, providing an overall response rate of 78 percent. The distribution of teams across the four organizations was 14, 10, 7, and 12. The team size ranged from 3 to 10 members, with an average of 5.5 members per team. A total of 72.7% of the participants were male, and the age distribution was as follows: 2.8% aged 18–24, 38.9% were 25–34, 34.1% were 35–44, 19% were 45–54, and 5.2% were 55 or older. All variables were measured with pre-existing validated measures. They were assessed on a Likert scale, ranging from 1 to 5 or 1 to 7.

Team autonomy was measured with six out of the eight original items from Langfred's [23] team autonomy scale. This is a modified version of a well-validated scale for individual job autonomy, adapted to the team level. An example of an item from the scale is *"The team is free to decide how to go about getting work done."* Team members were asked to assess how much they agreed with the statements concerning the team on a scale ranging from 1 ("completely disagree") to 5 ("completely agree").

Team trust was measured using a shortened version of the perceived trustworthiness in teams scale developed by Costa and Anderson [34]. An example item is: *"In this team, people can rely on one another."* Responses ranged from 1 (completely disagree) to 5 (completely agree).

Team work engagement was measured using the 3-item scale from the ultra-short version of the Utrecht work engagement scale [35], adapted to the team level by following

Costa et al. [13] using a reference shift from “I/me” to “we/our” to achieve the team focus. The items are: “*In our team, we feel bursting with energy at our work*,” “*In our team, we are enthusiastic about our job,*” and “*In our team, we are immersed in our work*.” The response alternatives ranged from 1 (“never the last year”) to 7 (“every day”).

Team performance was measured by three items based on scales developed by Jehn, Northcraft, and Neale [36]. Team members were asked to rate their team performance in terms of efficiency, quality, and overall performance. A sample item is: *“How would you assess your team performance in terms of efficiency?”* where the responses ranged from 1 (“very poor”) to 5 (“very good”).

Control variables included in the analysis were *team size* and *time spent in the team*, as these variables could potentially account for variance in the output variables. *Team size* was calculated based on how many team members from the team participated in the survey. We chose to proceed in this way because the average response rate per team was quite high (78%). The item for *time spent in the team* was *“How much of your time do you work on this team?*” (1 = less than 25%; 5 = around 90% or full-time). This measure was aggregated based on the scores provided by individual team members so that the scores represented the average for each team.

Data Aggregation. As all hypotheses in the present study refer to the team level of analysis, we aggregated the initially individual-level data to the team level. All the variables, except team performance, assumed a referent-shift consensus model [37]. In a referent-shift model, the referent is directed towards the team because these constructs are collective in nature. Rather than asking team members about their own individual perceptions, referent shift incorporates the team as a whole. In contrast, role clarity and team performance assumed a consensus model [37] with the referent items directed at the individual team members because the construct resides in the individual’s own perception of how well the team performed. Both forms of models assume that team members share a common perception, and therefore, the interrater agreement is necessary to justify aggregation. To do this, we assessed the within-group agreement index $r_{wg(j)}$ [38] for all measures.

Data Analyses. Data analyses were performed using Stata/MP version 16.1, which is a commonly applied software tool for statistical analyses. To test the hypothesis in the research model, we used partial least squares structural equation modeling (PLS-SEM) as the data analysis procedure. This procedure is recommended for data with relatively small sample sizes, and it allows for avoiding issues with non-normally distributed data [39]. The reliability and validity of the model were assessed by evaluating the measurement model (how well the latent variables reflect the variance in the measured items) [39]. This was done based on indicator reliability (item loadings’ size), composite reliability, convergent validity (average variance extracted (AVE), and discriminant validity [39]. Composite reliability was examined by evaluating Dillon-Goldstein’s rho (DG rho), which is an alternative to Cronbach’s alpha, in which the recommended level should be above 0.7. Discriminant validity (whether latent variables are sufficiently independent of each other) was assessed by comparing AVE values to the squared correlations among the latent variables in the model.

We tested the hypothesis by assessing the structural part of the model. To evaluate mediating relationships, one must compare the indirect paths suggested by the mediators to the direct paths [40]. Variables may have no mediating effect (the indirect effect is insignificant), a partial mediating effect (if the direct effect is significant), or a full mediating effect (if the direct effect is insignificant) [39]. The significance of the indirect effects was assessed based on bootstrap tests with 10 000 repetitions which is the procedure recommended by Hair et al. [39]. Finally, we tested potential common method bias (CMB) in the model through variance inflation factor (VIF), which is argued to be a reliable indicator of CMB in PLS-SEM [41]. Researchers argue that CMB can lead to results that are not due to the constructs of interest but rather to the measurement method, especially when it comes to behavioral research [42]. As a remedy, the assessment of VIF allows for uncovering possible multicollinearity in a PLS-SEM model [41].

4 Results

Since our study focuses on the team level, we first report the results of the within-group interrater agreement test that is recommended to justify the aggregation. As shown in Table 2, all average $r_{wg(j)}$ values are at about the threshold of 0.7, which, according to Le Breton and Senter [38], indicates acceptable interrater agreement within teams. This justifies us in aggregating the data collected at an individual level to a team level. Table 2 also shows average values and standard deviations of the aggregated variables.

Table 2. Summary of the aggregated variables for all teams

Aggregated variable	*M*	*SD*	$r_{wg(j)}$	
			M	*SD*
Team autonomy	3.93	0.45	0.87	0.16
Team trust	4.54	0.29	0.88	0.16
Team work engagement (TWE)	5.61	0.54	0.78	0.24
Team performance	4.10	0.31	0.90	0.06
Control var: Time in teamteam	3.64	0.34		
Control var: Team size	5.49	1.54		

As shown in Table 3, all the standardized loadings are close to or above the recommended threshold of 0.7, AVE exceeds the recommended level of 0.5, and all D.G. Rho values are above the level of 0.7. These findings indicate acceptable indicator reliability, composite reliability, and convergent validity.

Table 3. The measurement model (step 3)

Latent variable	Items	Loadings	D.G. Rho	AVE
Team autonomy	6	0.730–0.927	0.937	0.710
Team trust	4	0.835–0.933	0.931	0.773
Team work engagement (TWE)	3	0.884–0.967	0.909	0.863
Team performance	3	0.794–0.954	0.909	0.770

All AVE values (Table 3) are larger than the squared correlations among the latent variables in the model, which suggests acceptable discriminant validity of the measurement model.

Table 4. Discriminant validity (Squared correlations < AVE)

	Trust	TWE	Performance	Autonomy	Team size	Time in the team
Trust	1.000	0.348	0.249	0.162	0.005	0.011
TWE	0.348	1.000	0.367	0.205	0.017	0.002
Performance	0.249	0.367	1.000	0.069	0.041	0.021
Autonomy	0.162	0.205	0.069	1.000	0.001	0.025
Team size	0.005	0.017	0.041	0.001	1.000	0.008
Time in the team	0.011	0.002	0.021	0.025	0.008	1.000
AVE	0.773	0.863	0.770	0.710	1.000	1.000

Table 4 summarizes both direct and indirect effects in the model with "team work engagement" (TWE) and "team performance" as outcomes. Taking into account the potential relationship between "team autonomy" and "team trust" as job resources, we present the coefficients in a stepwise fashion. In Step 1, we entered "team autonomy" as a predictor, whereas "team trust" was entered in Step 2 and the control variables in Step 3. All the significant effects are highlighted in bold (Table 4).

In Step 1 we see that "team autonomy" has a positive direct effect on "team work engagement" (β = .453, p < .01), whereas "TWE" in turn has a positive effect on "team performance" (β = .613, p < .001). This means that teams with higher autonomy could be expected to also have a higher level of work engagement; and that the teams where the members were highly engaged also showed increased performance. There was no significant direct effect of "team autonomy on "team performance", whereas the indirect effect was significant (β = .277, p < .05). The combined findings at this step show an *indirect-only mediation* (according to Zhao et al. [40]) between "team autonomy" and "team performance" (β = .277, p < .05), meaning that "TWE" fully mediated the relationship between the two variables. For this step, we could conclude that "team autonomy" functions as a job resource, thus strengthening teams' engagement which then leads to subsequent increased performance.

Table 5. Summary (stepwise) of the effects with standardized path coefficients

	Step 1		Step 2		Step 3	
	Direct	Indirect	Direct	Indirect	Direct	Indirect
Autonomy → TWE	**0.453****		0.257		0.251	
TWE → Performance	**0.613*****		**0.494****		**0.443****	
Autonomy → Performance	−0.014	**0.277[1]***	−0.056	0.127[2]	−0.024	0.111[4]
Trust → TWE			**0.487****		**0.503****	
Trust → Performance			0.232	**0.241[3]****	0.276	**0.223[5]***
Time in the team – TWE					−0.037	
Team size → TWE					0.159	
Time in the team → Performance					−0.179	-0.012
Team size → Performance					0.148	0.070

Note. For the indirect effects the p-value is linked to the bootstrap test (10000 repetitions). 95% CI [1](0.112, 0.571); [2](−0.017, 0.256); [3](0.085, 0.454); [4](−0.024, 0.375); [5](0.036, 0.467). $*p < 0.05$, $**p < 0.01$, $***p < 0.001$.

In Step 2, we entered “team trust” as the second independent variable in the model. As shown in Table 5, “team autonomy” had neither direct nor indirect effect on “team performance” when controlled for “team trust”. At the same time, “team trust” showed a strong direct effect on “team work engagement” ($\beta = .487$, $p < .01$), which indicates that teams with a high level of trust were often highly engaged in their work. We also observed a significant indirect effect of “team trust” on “team performance” mediated by “TWE” ($\beta = .241$, $p < .01$). Since “team trust” did not have any direct effect on “team performance”, we concluded an *indirect-only mediation* (full mediation) between these two variables. We concluded that in Step 2 “TWE” fully mediated the relationship between “team trust” and “team performance” when controlled for “team autonomy”. In other words, “TWE” functioned as a mediator between “team trust” and “team performance”, but not between “team autonomy” and “team performance”, as it was in Step 1 when we did not control for “team trust”. In Step 3, the same results were validated by entering the control variables. Again, we saw that “team trust” had a significant indirect effect on “team performance” mediated by “TWE” ($\beta = .223$, $p < .01$), but no such effect was observed for “team autonomy”. As no control variable had either a significant direct or indirect effect on the dependent variables and the effects from Step 2 stayed significant (Table 4), we concluded that the findings could not be attributed to the properties of the particular teams. The overall conclusion from the analysis is that both “team autonomy” and “team trust” may function as team work resources, affecting "team work engagement" and eventually "team performance". However, "team autonomy" as a work resource seems to have a weaker effect than "team trust". Finally, all VIF values in the model ranged between 1.017 and 1.754, which is lower than the threshold of 3 recommended by Hair et al. [39] for PLS-SEM. This, in combination with other reliability diagnostics, indicates that the findings are not solely due to the measurement method.

5 Discussion

Team autonomy and team trust have long been acknowledged as fundamentals of agile teams [1, 3]. Our study indicates that these factors do not directly affect the performance of such teams but instead may affect team work engagement. Further, team work engagement seems to have a strong effect on team performance, thus indirectly linking it back to trust and - to a smaller extent – to team autonomy. In this way, our results confirm that work engagement is significant for the performance of agile teams [5, 6]. The results are summarized in Table 6.

Table 6. Summary of the results and implications

Hypothesis	Findings
H1: Team autonomy → Team work engagement	Partially supported. High autonomy can lead to engagement in teams, but this effect is weakened when trust is considered
H2: Team trust → Team work engagement (TWE)	Supported. Teams with high trust are likely to be highly engaged in their work. Trust shows a stronger relationship with TWE than team autonomy
H3: TWE → Team performance	Supported. TWE is positively related to team performance. Highly engaged teams perceive their performance higher than the teams with lower engagement
H4: TWE mediates Team autonomy → Performance	Partially supported. Team work engagement mediates the relationship between team autonomy and team performance, but the effect is eliminated when trust is controlled for
H5: TWE mediates Team trust → Performance	Supported. Team work engagement mediates the relationship between team trust and team performance, indicating an indirect effect of trust on perceived team performance

The absence of the direct effect of autonomy on performance and its weakened effect on team work engagement may sound surprising as autonomy consistently has been described as one of the fundamental needs of agile teams [1] and also one of the key characteristics in many work-stress models and theories (e.g., [26]). However, the strength of the relationship between autonomy and work engagement has been found to vary across studies [20]. This can partially be explained by the so-called autonomy paradox, meaning that greater autonomy can have both positive (e.g., increased control over tasks) and negative effects (increased stress due to increased job demands and expectations of their contribution to organizational performance) [22]. We follow Hakanen et al. [20], suggesting that the engaging power of autonomy is not so straightforward in the context of agile teams with complex tasks and organizational contexts.

Our findings indicate that team trust plays an important role in fostering work engagement and further enhancing team performance in agile teams. This is in line with the proposition of Moe et al. that mutual trust is of fundamental importance for agile teams and that teams that had not established mutual trust use more time on discovering and acknowledging issues [3]. Another explanation for our findings is the possible interaction between autonomy and trust. Our results indicate that the level of trust may impact the effect of team autonomy on engagement and performance. This corresponds to the findings in our recent study [43], showing that team autonomy positively affects psychological safety, a distinct but related construct of trust. Other studies also highlight lack of trust as one of the potential barriers to team autonomy [44]. We, therefore, invite researchers to further investigate whether and how team trust and team autonomy interact to affect the level of engagement in agile teams.

6 Limitations and Future Work

While providing valuable contributions to the literature, this study also has some limitations. First, the research model in our study is confined to a limited number of team-level factors influencing team work engagement and team performance. The reality for teams in organizations is obviously much more complex, with a daunting number of other factors, both on the individual, team, and organizational level, that impact the work and performance. The present study examines how job resources (trust, autonomy) and work engagement relate to the performance of agile software development teams and is a first step in understanding the factors impacting teams' engagement and performance in this setting. We acknowledge that there are several organizational and technical factors that could impact the engagement and performance of software development teams. Forsgren, Humble, and Kim [45], for instance, identified 24 capabilities that drive software delivery performance, including organizational culture, leadership, and architectural aspects. We thus encourage researchers to test more complex research models to further explore the effect of job resources at different levels that are relevant for the engagement and performance of agile software development teams. Second, the cross-sectional nature of our data does not allow us to conclude causality between the variables (for example, that work engagement leads to better team performance or vice versa). We are thus left with only indications of causality derived from theory and previous research. Future research should be conducted using a time-lagged design in order to examine the causal relationships between team autonomy and team work engagement; and team work engagement and performance. Further, self-reported data was the only foundation of the study. For example, we did not apply external actors' evaluation of the teams' performance, which may have biased the performance scores. Still, we believe that a strong relationship between teams' trust and work engagement; and between work engagement and their own perception of performance is a valuable result that deserves further investigation. We invite researchers to validate whether this result holds when additional measures of performance are also applied. Finally, the self-reported data may have inflated the correlations among the variables and thus potentially suffer from Common Method Bias (CMB). However, pre-existing measures were used, and statistical procedures for PLS-SEM were undertaken to reduce the risk of CMB.

7 Implications and Conclusion

Our results provide valuable theoretical insights and also have important practical implications for agile teams. The study demonstrates the theoretical value that the JD-R model and the work engagement literature can provide for agile research. Work engagement is a meaningful construct at the team level that mediates the impact of job resources on performance in teams. The overall results indicate that highly engaged teams are also likely to perform their tasks more efficiently and effectively, thus generating a competitive advantage. Agile practitioners should therefore promote team-based resources that contribute to engagement in their teams. Our findings suggest that both increasing the level of autonomy and, more importantly, building trust in the teams can foster team engagement, which in its turn has the potential to enhance the performance of agile teams. The "social fabric" of the teams plays an important role for team engagement and performance probably because succeeding in agile software development teams requires honest feedback, communication and collective problem-solving. We, therefore, urge practitioners to provide opportunities for teams to build trusting relationships where team members can demonstrate their competence, integrity, and benevolence.

Acknowledgments. The study was supported by the A-teams project and the Research Council of Norway (grant 267704).

References

1. Moe, N.B., Dingsøyr, T., Dybå, T.: Understanding self-organizing teams in agile software development. In: 19th Australian Conference on Software Engineering (ASWEC 2008), pp. 76–85 (2008). https://doi.org/10.1109/ASWEC.2008.4483195
2. Mohagheghi, P., Lassenius, C., Bakken, I.O.: Enabling team autonomy in a large public organization. In: Paasivaara, M., Kruchten, P. (eds.) XP 2020. LNBIP, vol. 396, pp. 245–252. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58858-8_25
3. Moe, N.B., Dingsøyr, T., Dybå, T.: A teamwork model for understanding an agile team: a case study of a Scrum project. Inf. Softw. Technol. **52**, 480–491 (2010)
4. Bakker, A.B., Demerouti, E.: Towards a model of work engagement. Career development international (2008)
5. Sharp, H., Baddoo, N., Beecham, S., Hall, T., Robinson, H.: Models of motivation in software engineering. Inf. Softw. Technol. **51**, 219–233 (2009). https://doi.org/10.1016/j.infsof.2008.05.009

6. Murphy-Hill, E., et al.: What predicts software developers' productivity? IEEE Trans. Software Eng. **47**, 582–594 (2021). https://doi.org/10.1109/TSE.2019.2900308
7. Torrente, P., Salanova, M., Llorens, S., Schaufeli, W.B.: Teams make it work: how team work engagement mediates between social resources and performance in teams. Psicothema **24**, 7 (2012)
8. Huck-Fries, V., Prommegger, B., Wiesche, M., Krcmar, H.: The role of work engagement in agile software development: investigating job demands and job resources. In: Hawaii International Conference on System Sciences 2019 (HICSS-52) (2019)
9. Hoda, R., Murugesan, L.K.: Multi-level agile project management challenges: a self-organizing team perspective. J. Syst. Softw. **117**, 245–257 (2016). https://doi.org/10.1016/j.jss.2016.02.049
10. Fowler, M., et al.: Principles behind the Agile Manifesto (2001). http://agilemanifesto.org/principles.html
11. Schaufeli, W.B., Salanova, M., Lez-Roma, V.G., Bakker, A.B.: The measurement of engagement and burnout: a two sample confirmatory factor analytic approach. J. Happiness Stud. **3**, 24 (2002)
12. Mazzetti, G., Robledo, E., Vignoli, M., Topa, G., Guglielmi, D., Schaufeli, W.B.: Work engagement: a meta-analysis using the job demands-resources model. Psychol. Rep. (2021). https://doi.org/10.1177/00332941211051988
13. Costa, P., Passos, A.M., Bakker, A.: Empirical validation of the team work engagement construct. J. Pers. Psychol. **13**, 34–45 (2014). https://doi.org/10.1027/1866-5888/a000102
14. Bakker, A.B., Demerouti, E.: Job demands–resources theory: taking stock and looking forward. J. Occup. Health Psychol. **22**, 273–285 (2017). https://doi.org/10.1037/ocp0000056
15. Demerouti, E., Bakker, A.B., Nachreiner, F., Schaufeli, W.B.: The job demands-resources model of burnout. J. Appl. Psychol. **86**, 499–512 (2001). https://doi.org/10.1037/0021-9010.86.3.499
16. Ryan, R.M., Deci, E.L.: Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Am. Psychologist. **11** (2000)
17. Christian, M.S., Garza, A.S., Slaughter, J.E.: Work engagement: a quantitative review and test of its relations with task and contextual performance. Pers. Psychol. **64**, 89–136 (2011). https://doi.org/10.1111/j.1744-6570.2010.01203.x
18. Muller, D., Kropp, M., Anslow, C., Meier, A.: The effects on social support and work engagement with scrum events. In: 2021 IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), pp. 101–104. IEEE, Madrid (2021). https://doi.org/10.1109/CHASE52884.2021.00019
19. Rietze, S., Zacher, H.: Relationships between agile work practices and occupational well-being: the role of job demands and resources. IJERPH **19**, 1258 (2022). https://doi.org/10.3390/ijerph19031258
20. Hakanen, J.J., Bakker, A.B., Turunen, J.: The relative importance of various job resources for work engagement: a concurrent and follow-up dominance analysis. BRQ Bus. Res. Q. (2021). https://doi.org/10.1177/23409444211012419
21. Halbesleben, J.R.: A meta-analysis of work engagement: relationships with burnout, demands, resources, and consequences. In: Bakker, A.B., Leiter, M.P. (eds.) Work Engagement: A Handbook of Essential Theory and Research, pp. 102–117 (2010)
22. Palumbo, R.: Engaging by releasing: an investigation of the consequences of team autonomy on work engagement. TPM **27**, 425–445 (2021). https://doi.org/10.1108/TPM-03-2021-0021
23. Langfred, C.W.: Autonomy and performance in teams: the multilevel moderating effect of task interdependence. J. Manag. **31**, 513–529 (2005). https://doi.org/10.1177/0149206304272190

24. Muecke, S., Linderman-Hill, K., Greenwald, J.M.: Linking job autonomy to work engagement: the mediating role of challenge demands. Proceedings **2020**, 13553 (2020). https://doi.org/10.5465/AMBPP.2020.13553abstract
25. Hackman, J.R., Oldham, G.R.: Motivation through the design of work: test of a theory. Organ. Behav. Hum. Perform. **16**, 250–279 (1976). https://doi.org/10.1016/0030-5073(76)90016-7
26. Theorell, T., Karasek, R.A.: Current issues relating to psychosocial job strain and cardiovascular disease research. J. Occup. Health Psychol. **1**, 9 (1996)
27. Buvik, M.P., Tvedt, S.D.: The impact of commitment and climate strength on the relationship between trust and performance in cross-functional project teams: a moderated mediation analysis. TPM **22**, 114–138 (2016). https://doi.org/10.1108/TPM-02-2015-0011
28. Buvik, M.P., Tvedt, S.D.: The influence of project commitment and team commitment on the relationship between trust and knowledge sharing in project teams. Proj. Manag. J. **48**, 5–21 (2017). https://doi.org/10.1177/875697281704800202
29. Mayer, R.C., Davis, J.H., Schoorman, F.D.: An Integrative model of organizational trust. Acad. Manag. Rev. **20**, 709 (1995). https://doi.org/10.2307/258792
30. Hakanen, J.J., Bakker, A.B., Schaufeli, W.B.: Burnout and work engagement among teachers. J. Sch. Psychol. **43**, 495–513 (2006). https://doi.org/10.1016/j.jsp.2005.11.001
31. Chen, G., Kanfer, R.: Toward a systems theory of motivated behavior in work teams. Res. Organ. Behav. **27**, 223–267 (2006). https://doi.org/10.1016/S0191-3085(06)27006-0
32. Costa, P., Passos, A.M., Bakker, A.B.: Direct and contextual influence of team conflict on team resources, team work engagement, and team performance. Negot. Confl. Manage Res. **8**, 211–227 (2015). https://doi.org/10.1111/ncmr.12061
33. Schaufeli, W.B., Bakker, A.B.: Defining and measuring work engagement: bringing clarity to the concept. In: Work Engagement: A Handbook of Essential Theory and Research, pp. 10–24. Psychology Press, New York (2010)
34. Costa, A.C., Anderson, N.: Measuring trust in teams: development and validation of a multifaceted measure of formative and reflective indicators of team trust. Eur. J. Work Organ. Psy. **20**, 119–154 (2011). https://doi.org/10.1080/13594320903272083
35. Schaufeli, W.B., Shimazu, A., Hakanen, J., Salanova, M., De Witte, H.: An ultra-short measure for work engagement. Eur. J. Psychol. Assess. **35**, 1–15 (2017)
36. Jehn, K.A., Northcraft, G.B., Neale, M.A.: Why differences make a difference: a field study of diversity, conflict and performance in workgroups. Adm. Sci. Q. **44**, 741–763 (1999)
37. Chan, D.: Functional relations among constructs in the same content domain at different levels of analysis: a typology of composition models. J. Appl. Psychol. **83**, 234 (1998)
38. LeBreton, J.M., Senter, J.L.: Answers to 20 questions about interrater reliability and interrater agreement. Organ. Res. Methods **11**, 815–852 (2008). https://doi.org/10.1177/1094428106296642
39. Hair, J.F., Hult, G.T.M., Ringle, C.M., Sarstedt, M., Danks, N.P., Ray, S.: Partial Least Squares Structural Equation Modeling (PLS-SEM) Using R: A Workbook. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80519-7
40. Zhao, X., Lynch, J.G., Jr., Chen, Q.: Reconsidering Baron and Kenny: Myths and truths about mediation analysis. J. Consum. Res. **37**, 197–206 (2010)
41. Kock, N.: Common method bias in PLS-SEM: a full collinearity assessment approach. Int. J. e-Collab. (IJEC) **11**, 1–10 (2015)
42. Podsakoff, P.M., MacKenzie, S.B., Lee, J.-Y., Podsakoff, N.P.: Common method biases in behavioral research: a critical review of the literature and recommended remedies. J. Appl. Psychol. **88**, 879–903 (2003). https://doi.org/10.1037/0021-9010.88.5.879
43. Buvik, M., Tkalich, A.: psychological safety in agile software development teams: work design antecedents and performance consequences. Presented at the Hawaii International Conference on System Sciences, 4 January 2022 (2022). https://doi.org/10.24251/HICSS.2022.880

44. Moe, N.B., Stray, V., Hoda, R.: Trends and updated research agenda for autonomous agile teams: a summary of the second international workshop at XP2019. In: Hoda, R. (ed.) XP 2019. LNBIP, vol. 364, pp. 13–19. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30126-2_2
45. Forsgren, N., Humble, J., Kim, G.: Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution (2018)

Agile in the Large

Design and Validation of a Capability Measurement Instrument for DevOps Teams

A Participatory Action Research Approach

Olivia H. Plant[1,2](✉), Jos van Hillegersberg[1,3], and Adina Aldea[1,4]

[1] Industrial Engineering and Business Information Systems, University of Twente, Enschede, The Netherlands
{o.h.plant,j.vanhillegersberg,a.i.aldea}@utwente.nl
[2] Quint Nederland B.V., De Oude Molen 1, Amstelveen, The Netherlands
[3] Tilburg University/JADS, Sint Janssingel 92, Den Bosch, The Netherlands
[4] LeanIX, Prins Bernhardplein 200, Amsterdam, The Netherlands

Abstract. This paper reports on the design and validation of a capability measurement instrument for software delivery teams that make use of the DevOps approach. The instrument is based on the results of a systematic literature review and was developed and validated by involving a total of five domain experts and conducting a field study among six DevOps team members. To this end, we used qualitative and survey-based data collection methods from participatory action research as well as design science. The resulting instrument encompasses five dimensions, covering seventeen capabilities and thirty-eight associated practices. The practices are evaluated on five capability levels. The results of the validation process indicate clear agreement of the domain experts and team members with all aspects of the instrument. As a contribution to practice, this research offers a pragmatic tool for IS practitioners which provides insight into the status of their DevOps transformation and offers directions for improving DevOps team performance. Furthermore, this research contributes to the ongoing research stream on DevOps by providing novel insights into the nature of DevOps capabilities and their potential configurations.

Keywords: DevOps · IS capabilities · Measurement instrument development · DevOps teams · Agile

1 Introduction

A growing amount of organizations is reorganizing their IT functions according to the DevOps paradigm. This calls for the establishment of cross-functional, agile teams that are responsible for development and operations of their systems and automate substantial parts of their processes [6,32]. While DevOps is becoming increasingly popular in practice, the approach has also attracted

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 151–167, 2022.
https://doi.org/10.1007/978-3-031-08169-9_10

growing attention from the IS research community over the past years. Multiple studies have attempted to create standardized definitions of DevOps [26] and identify its core elements [16] in order to foster a shared understanding of the paradigm. However, there is still no uniform definition of DevOps available [6,17]. Furthermore, there is little research-based guidance available to practitioners on how to implement DevOps and assess the current status of their transformation.

Prior research has related the implementation of IT capabilities to an increase in performance, both at team-level as well as on an organizational level [22,30]. We therefore propose to adopt a capability-based perspective when addressing the implementation of DevOps in organizations. Consequently, we argue that a standardized measurement instrument which evaluates the capabilities of DevOps teams will enable IT professionals to identify potential shortcomings or points for improvements in their transformation and will ultimately lead to an increase in team performance if the results of the measurement are addressed successfully.

While there have been efforts to create both industrial and scientific DevOps maturity models [34], to the best of our knowledge there is no instrument available which assesses the state of DevOps capabilities themselves. We therefore aim to develop a capability measurement instrument for DevOps teams which is based in extant academic literature but built in close collaboration with industry professionals in order to ensure its validity and practical use. Such a measurement instrument is expected to contribute to both the lack of a shared definition of DevOps and its practices as pointed out by Lwakatare, Kuvaja & Oivo [17] as well as provide a more structured approach for practitioners in how to implement DevOps and improve the performance of their DevOps teams.

This research makes use of the definition of a capability as proposed by Iacob, Quartel & Jonkers: *"A capability is the ability of an organization to employ resources to achieve some goal"* [14]. We furthermore build on the resource-based view and more specifically on the theory of dynamic capabilities [28] which argues that the competitive advantage of organizations lies within their resource base as well as in their ability to reconfigure their assets to address rapidly changing circumstances. According to Teece, Pisano and Shuen [28], these firm capabilities need to be understood in terms of managerial processes and organizational structures. Dynamic capabilities are idiosyncratic which makes them difficult to imitate for competitors [28]. However, Eisenhardt & Martin [5] suggest that while dynamic capabilities may be idiosyncratic in their details, they constitute a set of specific and clearly identifiable processes at a higher level. We therefore argue that it is possible to define a specific set of capabilities that are relevant to DevOps teams but that any measurement instrument of capabilities will need to capture various configurations of the same capability in order to account for their idiosyncratic implementation. Subsequently, our research is guided by the following main research question and sub-questions:

How to design a capability measurement instrument for DevOps teams?

(a) Which capabilities and practices are relevant for DevOps teams?

(b) How to assess varying configurations of capabilities with a measurement instrument?

2 Research Methodology

In order to develop the envisioned measurement instrument, we followed the procedural model proposed by Aldea & Sarkar [1] which is meant for developing valid and reliable measurement instruments for theoretical constructs. According to the aforementioned authors, the procedural model is suitable for researches in which the theory on which the instrument is based already exists and is sought to be empirically tested. The first stage of the model involves identifying theoretical constructs and candidate items which represent these constructs. The candidate items are then sorted into separate domain categories (substrata identification) from which a revised set of items is identified. These items are then further revised and improved. Finally, the instrument is validated in order to obtain evidence on the validity and reliability of the instrument.

An overview of all steps of the procedural model and the respective methodology applied in this research can be found in Table 1.

Table 1. Development of the DevOps capability measurement instrument

Instrument development stage [1]	Application to this research
1. Item creation	Systematic literature review
2. Substrata identification	Open and axial coding
3. Item identification	Domain expert workshops
4. Item revision	Domain expert interviews
5. Instrument validation	Domain expert evaluation survey & field study

2.1 Systematic Literature Review

The capabilities and practices that are part of the measurement instrument are based on the results of a systematic literature review (SLR) which we have conducted prior to this research and which we have detailed in a separate publication [21]. The review spanned 37 empirical research papers on DevOps capabilities and concepts. Data was gathered and synthesized by applying open and axial coding techniques in the qualitative data analysis tool Atlas.ti. To this end, we defined and applied codes to paragraphs of the papers which addressed capabilities and practices that were important for DevOps teams. The codes were continuously compared, merged or redefined and relationships between codes were established [33]. We then grouped the single codes into a more comprehensible set of code categories which resulted in an overview of DevOps practices and higher-level DevOps capabilities respectively. The core results of the review are summarized in Sect. 3.

2.2 Instrument Design

The capability measurement instrument was designed in close collaboration with industry practitioners by applying methods from Participatory Action Research (PAR). PAR seeks to combine theory and practice with the pursuit of designing practical solutions to pressing concerns of people [2]. This approach provides an opportunity for mutual learning and enriching dialogue between researchers and practitioners and is especially suitable when the nature of the artifact aligns with the participatory philosophy of PAR [24], as it is the case with our theory-based yet practically applicable measurement instrument.

Domain Expert Workshops. A first draft of the measurement instrument was created by conducting two workshops with a domain expert that served as a senior consultant at a Dutch consulting firm focused on digital transformations. This expert had vast experience with DevOps transformations and automation technologies.

Workshops are frequently used as qualitative data collection methods in PAR designs [3]. During the workshops, all candidate items were discussed in detail. Based on the suggestions made by the domain expert, items that displayed too much similarity to other items were eliminated in order to increase convergent and discriminant validity. Furthermore, one additional practice was added to the reference model based on the expert's suggestion. Additionally, all questions and answer options pertaining to the revised items were discussed and were clarified or supplemented with industry examples where applicable.

Domain Expert Interviews. The measurement items were further revised by interviewing four additional domain experts who also served as senior or principal consultants at a Dutch consulting firm. All of them had vast experience with Agile, DevOps or Lean methodologies and digital transformation projects in general. The capability measurement instrument was shared with the subjects before the interviews via e-mail.

The interviews had a semi-structured nature and were prepared beforehand through means of an interview guide [19]. The interviews lasted between 30 and 45 min. We started the conversation by introducing our research rationale and explaining our interpretation and definition of the concept of capabilities. We then discussed the capability levels with the interviewees and asked for their opinion on whether the scales and their definitions were understandable and covered all possible configurations of a DevOps capability sufficiently. This phase led to some minor adjustments in the capability level definitions. We then discussed the instrument taxonomy with the experts and asked whether the identified capabilities were indeed relevant for DevOps teams, whether there were any capabilities missing or redundant and whether the definitions of the capabilities were clear. The interviews led to the inclusion of another practice in the taxonomy and some minor adjustments regarding the names of some capabilities, the practices assigned to them and in the definitions of the capabilities and their measurement scales.

2.3 Instrument Validation

Maturity models can be evaluated through three different methodologies [23]: The first method is the evaluation of the instrument by the authors themselves. Another technique is the evaluation by domain experts which is performed through interviews, surveys or assignments. The last method is evaluation in a practical setting. The capability measurement instrument at hand was validated by applying a combination of domain expert evaluation and a field study. In doing so, we follow the suggestions of Venable, Pries-Heje and Baskerville [29] who propose to first evaluate design artifacts in an artificial setting, for example by using theoretical arguments, before moving towards a naturalistic evaluation in the real environment of the artifact.

Domain Expert Evaluation Survey. After the interviews, the four domain experts who were involved in the item revision stage were requested to fill in an online survey. They were asked to rate a number of statements regarding the instrument based on a five-point Likert scale, ranging from *strongly disagree* to *strongly agree*. The remaining domain expert who participated in the item identification workshops was not engaged in the validation of the measurement instrument due to their high involvement during the creation of the instrument.

The statements in the evaluation survey were based on the evaluation template for domain expert reviews of maturity models by Salah, Paige and Cairns [23]. The template was slightly adjusted to suit the nature of our capability measurement instrument better. The results of the survey indicate clear agreement of the domain experts with the validated aspects of the instrument. An overview of all statements and the mean agreement scores given by the four respondents as well as the standard deviations of these scores can be found in Table 2. [1].

Next to these statements, the experts were also asked a number of open questions focused on whether there were any questions, answers or descriptions which the respondents would add, remove or update and whether the model could be improved to make it more useful.

Field Study. Simultaneous to the expert validation, the instrument was presented to six DevOps team members from three different organizations. After taking the assessment, the team members were asked to rate a number of statements which were modified from the domain expert evaluation survey. The participants were solely asked to rate statements related to the understandability and ease of use of the instrument, as well as whether they thought that the capabilities covered all aspects relevant to DevOps teams. The evaluation of the underlying design of the instrument such as the sufficiency and accuracy of the capability levels or the general use in the industry were left to the domain experts and were not part of the field study evaluation. An overview of the validation statements, mean agreement scores and their standard deviations can be found in Table 2, along with the results of the domain expert validation survey.

[1] The individual scores given by the respondents will be provided upon request

Table 2. Validation survey statements and mean agreement scores from domain experts (n = 4) and field study participants (n = 6), based on a five-point Likert scale

Validation statements adapted from [23]	Domain experts	*Stand. Dev.*	Field study	*Stand. Dev.*
Capabilities & practices	**4.3**		**4.3**	
1. The capabilities and practices are relevant to DevOps teams (Relevance)	4.5	*0.5*	4.5	*0.5*
2. The capabilities and practices cover all aspects impacting/ involved in DevOps teams (Comprehensiveness)	4.0	*0.0*	4.0	*0.0*
3. Capabilities and practices are clearly distinct (Mutual Exclusion)	4.3	*0.4*	–	–
4. The answer options are clearly distinct (Mutual exclusion)	–	–	4.2	*0.4*
5. There are no questions asked more than once in the assessment (Mutual Exclusion)	–	–	4.3	*0.5*
Capability levels	**3.9***			
6. The five capability levels are sufficient to represent all states of a team capability (Sufficiency)	4.3	*1.3*	–	–
7. There is no overlap detected between descriptions of capability levels (Accuracy)	3.8	*0.4*	–	–
8. The question answers are correctly assigned to their respective capability level (Accuracy)	3.8	*0.4*	–	–
Capability assessment	**4.5***		**4.1**	
9. The capability descriptions are understandable (Understandability)	4.5	*0.5*	–	–
10. The capability levels are understandable (Understandability)	4.5	*0.5*	–	–
11. The questions and answers are understandable (Understandability)	4.3	*0.4*	4.3	*0.7*
12. The capability assessment is easy to use (Ease of use)	4.5	*0.5*	4.2	*0.9*
13. The capability assessment is easy to evaluate (Ease of use)	4.3	*0.4*	–	–
14. The capability assessment has the right length (Ease of use)	–	–	3.8	*1.2*
15. The capability assessment is useful for conducting assessments (Usefulness and Practicality)	5.0	*0.0*	–	–
16. The capability assessment is practical for use in industry (Usefulness and Practicality)	4.8	*0.4*	–	–

*Deviation from averages of values displayed in the table due to rounding errors.

3 Theoretical Framework

In a previous publication [21], we have extracted DevOps capabilities from extant literature and analyzed these in the light of the dynamic capabilities theory [27]. We then put forward the argument that DevOps teams can contribute to the competitive advantage of organizations by building capabilities that allow them to sense opportunities and threats, seize opportunities and rapidly transform their assets. The success of these capabilities however is dependent on the presence of a set of organizational enabler capabilities that allow the teams to perform their work independently and autonomously and work towards supporting the organizational strategy and vision. If these two sets of capabilities are implemented successfully, organizations can expect to achieve a third set of beneficial outcome capabilities. The identified DevOps team capabilities were divided into the classes *sensing*, *seizing* and *transforming* which is in line with the classification of dynamic capabilities by Teece [27]. An overview of the results of the literature review is given in Fig. 1.

DevOps teams need to develop capabilities on two levels: First, business-related capabilities concern structures, processes and habits in their way of working which the DevOps teams develop. Second, the teams need to develop technology-related capabilities which allow them to automate processes and perform monitoring activities.

In order to sense opportunities and act upon these, DevOps teams should design customer-centric processes [13,20] and have frequent information exchange with stakeholders [12]. Furthermore, they should have a clear process for translating customer wishes into requirements and manage the backlog [9]. At the same time, teams need to be venturous [31] and self-empowered by assuming responsibility and ownership of their system [10,25] so they can operate autonomously and take appropriate decisions quickly. This can be facilitated by building an open team culture which is focused on continuous improvement [20], sharing opinions [6] and in which team members trust and respect each other [26]. In order to shorten decision-making and authorization processes, teams should also be skilled at lean-process management [6] and collaborate well within the team as well as with other teams [7]. Once teams have decided to take action based on an identified opportunity or threat, they need to deal with changes

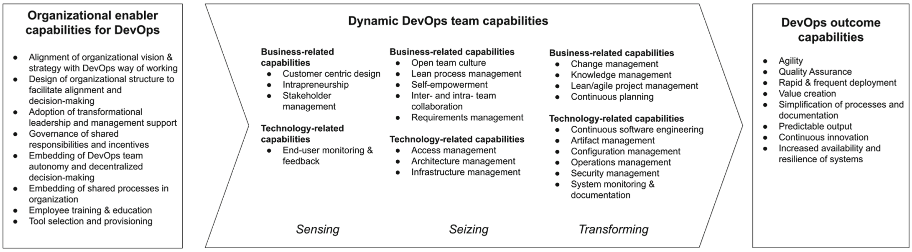

Organizational enabler capabilities for DevOps
• Alignment of organizational vision & strategy with DevOps way of working
• Design of organizational structure to facilitate alignment and decision-making
• Adoption of transformational leadership and management support
• Governance of shared responsibilities and incentives
• Embedding of DevOps team autonomy and decentralized decision-making
• Embedding of shared processes in organization
• Employee training & education
• Tool selection and provisioning
Dynamic DevOps team capabilities
Business-related capabilities
• Customer centric design
• Intrapreneurship
• Stakeholder management
Technology-related capabilities
• End-user monitoring & feedback
Business-related capabilities
• Open team culture
• Lean process management
• Self-empowerment
• Inter- and intra- team collaboration
• Requirements management
Technology-related capabilities
• Access management
• Architecture management
• Infrastructure management
Business-related capabilities
• Change management
• Knowledge management
• Lean/agile project management
• Continuous planning
Technology-related capabilities
• Continuous software engineering
• Artifact management
• Configuration management
• Operations management
• Security management
• System monitoring & documentation
Sensing
Seizing
Transforming
DevOps outcome capabilities
• Agility
• Quality Assurance
• Rapid & frequent deployment
• Value creation
• Simplification of processes and documentation
• Predictable output
• Continuous innovation
• Increased availability and resilience of systems

Fig. 1. Conceptual model of DevOps capabilities resulting from SLR [21]

effectively and timely [20]. This requires a flexible yet up-to-date planning process [26] as well as continuous exchange of knowledge and information [10] so team-members can assume multiple roles and responsibilities in this process.

On a technology-level, the automation of software delivery and provisioning processes enables DevOps teams to bring changes into production quickly. Most dominantly, many DevOps teams develop continuous engineering capabilities [9] in which they automate their entire delivery process including code testing and deployment activities. This process can be further supported by automation of infrastructure provisioning [15] and configurations [12]. Furthermore, DevOps teams should develop strong monitoring and logging capabilities [6] in order to secure their systems and act quickly in case of irregularities.

4 Results

4.1 Instrument Taxonomy

As an answer to the first sub-research question, we have defined a taxonomy of the capability measurement instrument, which is composed of *dimensions*, *capabilities* and *practices*. An overview of all capabilities, definitions and practices of the instrument is shown in Table 3.

The *dimensions* of the instrument serve as broad categories which enable easy communication of the results to stakeholders. They are represented by the CALMS acronym which was coined by Humble & Molesky [11] and is widely used to address the core components of the DevOps paradigm [8]. The CALMS acronym originally represents the dimensions of *culture*, *automation*, *lean*, *measurement* and *sharing*. However, in consultation with one domain expert it was decided to replace the measurement section in our instrument with the category *monitoring*, since the requirement to measure the progress of any capability is already integrated into the capability measurement scales of our model and is thus an inherent part of every capability which is performed at level four or higher (refer to Subsect. 4.2 for a detailed explanation of the capability levels). Adding this category to the taxonomy is in line with previous research which has defined monitoring to be another integral part of DevOps [16,17].

Every instrument dimension contains a set of *capabilities* which are in turn composed of between one to three *practices*. Each practice is represented by a single question in the assessment. In order to facilitate communication and understanding of the capabilities, we added a definition to each capability which was validated by the domain experts.

4.2 Capability Measurement Scales

The second research sub-research question is based on the argument that dynamic capabilities are idiosyncratic in their details [28], which suggests that the identified DevOps team capabilities may be exhibited in distinct ways by different teams. It was therefore decided to design the instrument in such a

Table 3. Final taxonomy of the capability measurement instrument

Capability	Description	Associated practices
Culture		
Intrapreneurship	The team has processes and structures in place to ensure it fulfills all necessary business activities in order to remain relevant to customers	Opportunity scouting Experimentation Problem recognition & solving
Continuous improvement	The team has processes and structures in place to ensure a team culture focused on communication and innovation	Accepting & providing feedback Continuous improvement Sharing goals & values
Self-empowerment	The team has processes and structures in place to ensure it can function independently without intervention from management	Change readiness Decision-making Self-organization
Automation		
Continuous software engineering	The team has processes and structures in place to ensure the continuous release of high quality software	Automated build Automated testing (Continuous) Integration (Continuous) Deployment
Infrastructure & configuration management	The team has processes and structures in place to ensure the necessary infrastructure is available and configured correctly	Infrastructure provisioning & containerization Managing configurations
Artifact management	The team has processes and structures in place to ensure artifacts are stored and versioned in a repository	Use of artifacts
Architecture management	The team has processes and structures in place to ensure the architecture is and remains flexible	Use of microservices or a modular architecture
Security & access management	The team has processes and structures in place to ensure their applications are secure, in line with compliance requirements and may only be accessed by authorized users	Performing risk analysis, risk evaluation, compliance requirements & security testing Using access policies
Lean		
Lean process management	The team has a process or framework in place to ensure optimum flow of work	Lean/Agile way of working Lean/Agile project management
Change & operations management	The team has processes and structures in place to manage change requests and systems operations	Resolving incidents Automated recovery Managing changes
Continuous planning	The team has processes and structures in place to ensure a flexible planning	Planning
Customer-centric design	The team has processes and structures in place to ensure their services are targeted at involving and meeting customer needs	Stakeholder management Product-oriented team setup Cross-functional team setup
Requirement management	The team has processes and structures in place to manage and prioritize system/service requirements	Requirement specification & prioritization Use of NFRs
Monitoring		
End-user monitoring & feedback	The team has processes and structures in place to ensure it is aware of how their system is used and improve it based on end-user behaviour and feedback	Monitor customer systems & receive feedback from end-users
System monitoring & documentation	The team has processes and structures in place to monitor the performance and behavior of their internal systems	Monitoring & logging of internal systems
Sharing		
Knowledge sharing	The team has processes and structures in place to ensure that information, knowledge and skills are equally distributed and disseminated throughout the team	Information sharing Continuous learning Sharing knowledge & skills
Team collaboration	The team has processes and structures in place to ensure regular alignment between team-members and with other teams in the organization	Intra-team alignment Inter-team alignment Sharing priorities

way that it captures numerous possible configurations of a capability instead of merely assessing whether a capability is performed at a sufficient level or not. The capability measurement instrument subsequently uses a continuous representation in which the separate capabilities are assessed on five different capability levels. This is opposed to many maturity models that make use of a staged representation in which the capabilities are assigned to maturity levels.

Table 4. Measurement scales and final definitions used for capability levels per instrument dimension

Dimension	Culture, Sharing *(Definition adapted from Magdaleno CollabMM* [18]*)*	Automation, Lean, Monitoring *(Definition adapted from CMMI capability levels* [4]*)*
Level 1	**Initial*** - The capability is not performed	**Incomplete** - The capability is not performed
Level 2	**Ad-hoc** - The team decides on the spot how to carry out collaboration activities	**Performed -** The capability is carried out and works in practice. However, the team has no agreed way of working for doing this
Level 3	**Planned** - The team has an agreed way of working in which it collaborates, e.g. through regular meetings or platforms	**Managed** - The team has agreed on a specific way of working such as a process or policy to ensure that the capability is performed
Level 4	**Aware** - Team members are aware of their tasks and of the agreed process, no central coordination is necessary for the members to collaborate. The team might include monitoring activities to ensure the way of working is leading to the desired capability	**Defined** - The team has worked out their way of working in detail, e.g. through process descriptions, monitoring performance measurement metrics or adapting the process from the organizational policy to their own needs
Level 5	**Reflexive** - Team members are aware of the agreed way of working and are self-organizing. They can identify which results are relevant and are continuously collaborating, interacting and sharing knowledge among each other. They use double-loop learning to recognize whether the desired goal state is still applicable.	**Optimizing*** - The team does not only have an elaborate way of working but also continuously reflects on the process and improves this to perform the capability even better

*Levels added by researchers to equalize scales.

Given the diverging nature of capabilities in the relationship-oriented dimensions of culture and sharing and the more traditional, process-oriented dimensions of automation, lean and monitoring, it was decided to use two different, yet comparable measurement scales to define the capability levels in our instrument.

The answer options to questions related to the culture and sharing dimensions were adapted from the Collaboration Maturity Model (CollabMM) by Magdaleno, Araujo and Werner [18]. This scale was chosen due to its explicit focus on team collaboration, as opposed to the more process-oriented focus of many other models. Although the CollabMM scale is originally used in a staged representation, we found the scale to also be useful for assessing the separate capabilities and have developed descriptions which suit this aim.

The capability levels of the dimensions automation, lean and monitoring were adapted from the CMMI continuous representation capability levels [4]. This measurement scale was chosen due to its wide recognition and use in both academia and practice, as well as the continuous nature of the scale.

In order to equalize the scales, we added a capability level to the lower end of the CollabMM and to the upper end of the CMMI capability level descriptions. The descriptions of each capability level were validated and adjusted based on feedback given by the domain experts. The final definitions can be found in Table 4.

4.3 Assessment Items

The practices and capability levels which we previously discussed were translated to fitting questions and answer options and were supplemented with industry examples with the help of a domain expert during the item identification stage. The final version of the instrument contains 38 assessment items which represent the practices in Table 3. Two example questions and answer options are displayed in Table 5.

5 Discussion and Conclusion

The research at hand describes the design and validation of a capability measurement instrument for DevOps teams. To arrive at this artifact, we have investigated the sub-research questions *"Which capabilities and practices are relevant to DevOps teams?"* and *"How to assess varying configurations of capabilities with a measurement instrument?"*. As an answer to these questions, we offer a comprehensive taxonomy of DevOps capabilities and practices and describe two measurement scales on which the varying configurations of a capability can be measured. Due to the taxonomy being based on the results of a SLR, the capabilities and practices in our measurement instrument are supported by existing literature on DevOps capabilities [17,25,26] but extend the aforementioned works. The resulting instrument was developed and validated in close collaboration with industry practitioners, using qualitative research approaches from PAR as well by collecting data via surveys. The results of the validation phase

Table 5. Measurement instrument

Dimension:	Culture	*Capability*:	Intrapreneurship
Practice:	Experimentation	*Measurement Scale*:	CollabMM [18]
Does your team experiment with new ideas or techniques regarding your way of working or product/service?			
Level 1:	We never experiment with new techniques. We rather stick to what we know and what works for us		
Level 2:	We sometimes experiment with new ideas but not in a coordinated way		
Level 3:	Experimentation is a planned and coordinated part of our work, e.g. we free up time during our sprints to try new things		
Level 4:	We regularly experiment with new techniques to improve our product and way of working as part of our daily work. This happens inside and outside of planned events		
Level 5:	We regularly experiment with new techniques to improve our product and way of working, inside and outside of planned events. These insights often lead to improvements in our product or way or working		
Dimension:	Lean	*Capability*:	Change & operations mngmt.
Practice:	Resolving incidents	*Measurement Scale*:	CMMI capability levels [4]
How do you deal with incidents?			
Level 1:	We do not have a procedure for this. We deal with incidents when they arise		
Level 2:	When an incidents arises we decide on a case-to-case basis based on our own judgement if we deal with it directly or later		
Level 3:	We have a standardized procedure for classifying and dealing with incidents, e.g. based on ITIL		
Level 4:	Dealing with incidents is part of our way of working, e.g. incidents are prioritized and placed on the backlog or we have reserved time every day to deal with important incidents		
Level 5:	Dealing with incidents is part of our way of working, e.g. incidents are prioritized and placed on the backlog or we have reserved time every day to deal with important incidents. We regularly reflect on our incident handling process and improve it, e.g. by performing a blameless post-mortem analysis		

indicate clear agreement of the experts and the DevOps team members with all aspects of the measurement instrument, resulting in high mean agreement scores as shown in Table 2.

Nevertheless, participants had varying opinions regarding the appropriateness of the length of the instrument and the associated number of questions which resulted in a high standard deviation of validation item number 14 (Table 2). When asked about the amount of time it took them to complete the survey, participants reported values between 10 and 30 min. Furthermore, the domain experts disagreed on the sufficiency of the five capability levels to represent all possible states of a team capability. Three respondents strongly agreed (score of 5) with this statement whereas one respondent disagreed (score of 2). One of the interviewed domain experts pointed out that a five-point scale is the industry standard on which many assessments and maturity models are based and that the scale should therefore be kept this way.

During the interview phase, multiple domain experts pointed out that they would like to include behavioural or intangible aspects such as trust and respect between the team members in the assessment. This is supported by the results of our literature review which has revealed the above mentioned factors to be essential to the performance of DevOps teams [26]. However, while we find these traits to be invaluable for DevOps teams, they did not fit our definition of a capability and could not be measured using one of our proposed measurement scales. We have therefore decided to not include these aspects in the assessment.

The proposed measurement instrument is designed to be used as a self-assessment. This is different to traditional capability maturity models, in which the researcher is often required to evaluate the organization in question based on pre-defined guidelines and templates [23]. One of the interviewed domain experts pointed out that a strong aspect of the proposed type of self-assessment is its ability to measure the capabilities over a large amount of teams. Furthermore, the standardized measurement instrument may help to compare the capabilities of different teams. However, the same interviewee indicated their preference for a more qualitative, in-depth approach when dealing with a smaller sample size of teams. This approach ensures that the neutral opinion and observations of the assessor are taken into account when conducting the assessment whereas our proposed approach is entirely dependent on the judgement of the team members using the measurement instrument.

5.1 Contributions to Theory and Practice

The research at hand provides novel contributions to both theory and practice. On the practical side, we contribute a tool that may be used by IT professionals to measure the capability configuration of DevOps teams. The results of the measurement provide valuable information into the status of the transformation process of DevOps teams and offer directions for further improving their team performance. The tool may also contribute to fostering a shared understanding of a DevOps definition and associated capabilities.

On the theory side, we provide insights into the nature of DevOps capabilities, the different configurations which they may take on as well as propose suitable scales to measure their maturity. Different to extant models and research on DevOps capabilities, our measurement instrument accounts for the idiosyncrasy of capabilities. Present DevOps maturity models are primarily focused on mapping capabilities to maturity levels [34] but did not investigate the potential ways in which a capability may be implemented. We therefore adopted a continuous representation in which we measure the configuration of DevOps capabilities in themselves on a five-level scale, but do not imply any hierarchy of capabilities or succession regarding their implementation as it would be the case in a staged representation maturity model.

5.2 Limitations and Further Research

Our research and the accompanying DevOps team capability assessment are limited by a number of factors. Primarily, our research was predominantly based on qualitative research approaches which was done to support the design of theory behind the instrument. No statistical methods were used to judge the validity and internal consistency of the categories. Future research should therefore further validate and improve our taxonomy by using techniques such as factor analysis or Cronbach's alpha. Collecting a larger number of responses on the survey would also support an in-depth psychometric analysis. Furthermore, our research solely focuses on the implementation and configuration of capabilities, to be understood in terms of underlying processes and structures. Behavioural and intangible aspects such as trust or respect were therefore excluded from our model and warrant further investigation in terms of how to measure and include these in a measurement instrument.

5.3 Conclusion

The research at hand proposes a capability measurement instrument for DevOps teams. Based on a systematic literature review and in close collaboration with industry practitioners, we developed a taxonomy which encompasses seventeen capabilities and thirty-eight associated practices that are measured on five capability levels. The resulting instrument and its taxonomy provide insights into the nature and configuration of DevOps capabilities as well as a standardized approach to measuring these and improving DevOps team performance.

References

1. Aldea, A., Sarkar, A.: A measurement instrument for enterprise architecture resilience research: a pilot study on digital transformation. In: Proceedings of the 55th Hawaii International Conference on System Sciences, pp. 7182–7191 (1 2022)
2. Brydon-Miller, M., Greenwood, D., Maguire, P.: Why action research? Action Res. **1**(1), 9–28 (7 2003). https://doi.org/10.1177/14767503030011002
3. Caretta, M.A., Vacchelli, E.: Re-thinking the boundaries of the focus group: a reflexive analysis on the use and legitimacy of group methodologies in qualitative research. Sociolog. Res. Online **20**(4), 58–70 (2015). https://doi.org/10.5153/sro.3812
4. Carnegie Mellon University Software Engineering Institute: CMMI for Development, Version 1.3. Technical report, November 2010
5. Eisenhardt, K.M., Martin, J.A.: Dynamic capabilities: what are they? Strateg. Manag. J. **21**(10/11), 1105–1121 (2000). http://www.jstor.org/stable/3094429
6. Erich, F.M.A., Amrit, C., Daneva, M.: A qualitative study of DevOps usage in practice. J. Softw. Evol. Process **29**(6) (2017). https://doi.org/10.1002/smr.1885

7. de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., Brinkkemper, S.: DevOps competences and maturity for software producing organizations. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (eds.) BPMDS/EMMSAD -2018. LNBIP, vol. 318, pp. 244–259. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91704-7_16
8. Fitzgerald, B., Stol, K.J.: Continuous software engineering: a roadmap and agenda. J. Syst. Softw. **123**, 176–189 (2017)
9. Gruhn, V., Schäfer, C.: BizDevOps: because DevOps is not the end of the story. In: Fujita, H., Guizzi, G. (eds.) SoMeT 2015. CCIS, vol. 532, pp. 388–398. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22689-7_30
10. Hemon, A., Fitzgerald, B., Lyonnet, B., Rowe, F.: Innovative practices for knowledge sharing in large-scale DevOps. IEEE Softw. **37**(3), 30–37 (2020)
11. Humble, J., Molesky, J.: Why enterprises must adopt DevOps to enable continuous delivery. Cutter IT J. **24**(8), 6–12 (2011)
12. Hussain, W., Clear, T., MacDonell, S.: Emerging trends for global DevOps: a New Zealand perspective. In: Proceedings - 2017 IEEE 12th International Conference on Global Software Engineering, ICGSE 2017, pp. 21–30. Software Engineering Research Lab (SERL), School of Engineering, Computer and Mathematical Sciences (SECMS), Auckland University of Technology (AUT), Auckland, New Zealand (2017). https://doi.org/10.1109/ICGSE.2017.16
13. Hussaini, S.W.: Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 178–183 (2014). https://doi.org/10.1109/ITSC.2014.6957687
14. Iacob, M.E., Quartel, D., Jonkers, H.: Capturing business strategy and value in enterprise architecture to support portfolio valuation. In: Proceedings of the 2012 IEEE 16th International Enterprise Distributed Object Computing Conference, EDOC 2012, pp. 11–20 (2012). https://doi.org/10.1109/EDOC.2012.12
15. Luz, W.P., Pinto, G., Bonifácio, R.: Adopting DevOps in the real world: a theory, a model, and a case study. J. Syst. Softw. **157** (2019). https://doi.org/10.1016/j.jss.2019.07.083
16. Lwakatare, L.E., Kuvaja, P., Oivo, M.: Dimensions of DevOps. In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP, vol. 212, pp. 212–217. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18612-2_19
17. Lwakatare, L.E., Kuvaja, P., Oivo, M.: An exploratory study of DevOps: extending the dimensions of DevOps with practices. In: 11th International Conference on Software Engineering Advances, ICSEA 2016, pp. 91–99, Rome, Italy (2016)
18. Magdaleno, A.M., Araujo, R.M.D., Werner, C.M.L.: A roadmap to the Collaboration Maturity Model (CollabMM) evolution. In: Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2011, pp. 105–112 (2011). https://doi.org/10.1109/CSCWD.2011.5960062
19. Myers, M.D., Newman, M.: The qualitative interview in IS research: examining the craft. Inf. Organ. **17**(1), 2–26 (2007). https://doi.org/10.1016/j.infoandorg.2006.11.001
20. Nagarajan, A.D., Overbeek, S.J.: A DevOps implementation framework for large agile-based financial organizations. In: Panetto, H., Debruyne, C., Proper, H.A., Ardagna, C.A., Roman, D., Meersman, R. (eds.) OTM 2018. LNCS, vol. 11229, pp. 172–188. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_10

21. Plant, O.H., van Hillegersberg, J., Aldea, A.: How DevOps capabilities leverage firm competitive advantage: a systematic review of empirical evidence. In: 2021 IEEE 23rd Conference on Business Informatics (CBI) 2021, pp. 141–150. Institute of Electrical and Electronics Engineers (IEEE) (2021). https://doi.org/10.1109/cbi52690.2021.00025
22. Ravichandran, T., Lertwongsatien, C.: Effect of information systems resources and capabilities on firm performance: a resource-based perspective. J. Manag. Inf. Syst. **21**(4), 237–276 (2005). https://doi.org/10.1080/07421222.2005.11045820
23. Salah, D., Paige, R., Cairns, P.: An evaluation template for expert review of maturity models. In: Jedlitschka, A., Kuvaja, P., Kuhrmann, M., Männistö, T., Münch, J., Raatikainen, M. (eds.) PROFES 2014. LNCS, vol. 8892, pp. 318–321. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13835-0_31
24. Santini, C., Marinelli, E., Boden, M., Cavicchi, A., Haegeman, K.: Reducing the distance between thinkers and doers in the entrepreneurial discovery process: an exploratory study. J. Bus. Res. **69**(5), 1840–1844 (2016). https://doi.org/10.1016/j.jbusres.2015.10.066
25. Senapathi, M., Buchan, J., Osman, H.: DevOps capabilities, practices, and challenges: insights from a case study. In: ACM International Conference Proceeding Series, EASE 2018, vol. Part F1377, pp. 57–67. ACM, New York (2018)
26. Smeds, J., Nybom, K., Porres, I.: DevOps: a definition and perceived adoption impediments. In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP, vol. 212, pp. 166–177. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18612-2_14
27. Teece, D.J.: Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. Strateg. Manag. J. **28**(13), 1319–1350 (2007)
28. Teece, D.J., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management. Strateg. Manag. J. **18**(7), 509–533 (1997)
29. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: a framework for evaluation in design science research. Eur. J. Inf. Syst. **25**(1), 77–89 (2016). https://doi.org/10.1057/ejis.2014.36
30. Vishnubhotla, S.D., Mendes, E., Lundberg, L.: Understanding the perceived relevance of capability measures: a survey of agile software development practitioners. J. Syst. Softw. **180**, 111013 (2021). https://doi.org/10.1016/j.jss.2021.111013
31. Wiedemann, A., Schulz, T.: Key capabilities of DevOps teams and their influence on software process innovation: a resource-based view. In: Proceedings of the 23rd Americas Conference on Information Systems, AMCIS 2017. Neu-Ulm University of Applied Sciences, Germany (2017)
32. Wiedemann, A., Wiesche, M., Gewald, H., Krcmar, H.: Understanding how DevOps aligns development and operations: a tripartite model of intra-IT alignment. Eur. J. Inf. Syst., 1–16 (2020). https://doi.org/10.1080/0960085X.2020.1782277

33. Wolfswinkel, J.F., Furtmueller, E., Wilderom, C.P.: Using grounded theory as a method for rigorously reviewing literature (2013). https://doi.org/10.1057/ejis.2011.51

34. Zarour, M., Alhammad, N., Alenezi, M., Alsarayrah, K.: A research on DevOps maturity models. Int. J. Recent Technol. Eng. **8**(3), 4854–4862 (2019). https://doi.org/10.35940/ijrte.C6888.098319

Toward an Agile Product Management: What Do Product Managers Do in Agile Companies?

Anastasiia Tkalich(✉), Rasmus Ulfsnes, and Nils Brede Moe

SINTEF, 7034 Trondheim, Norway
Anastasiia.Tkalich@sintef.no

Abstract. The product manager (PM) role is well established in leading technological companies, such as Google, Amazon, Microsoft, and Facebook. PMs are responsible for integrating technical, design, and business perspectives when developing software products and product portfolios. In agile methods (e.g., Scrum), similar responsibilities are linked to the Product Owner (PO) role. In contrast, in large-scale agile, one can find both Product Owners and product managers who sometimes compete. Despite the widespread adoption of the product manager role, the attention toward it in the agile academic community has been surprisingly limited. In this multiple case study, we analyzed 17 interviews with 11 product managers from four agile companies. We found that the PMs facilitated continuous product experimentation and innovation, supported the product teams, and engaged in additional activities to achieve optimal product development. Our summary of the product management activities can guide product managers working in agile companies.

Keywords: Agile · Product manager · Innovation · Product discovery · Continuous improvement · Lean startup · Scaled agile framework · Product owner

1 Introduction

In an increasingly more complex environment, companies need a holistic product strategy to develop products. When products must adapt to constantly changing user needs and new products should be launched, there is an increasing need to establish an end-to-end flow between customer demand and the fast delivery of a product or service. In agile software development (e.g., Scrum), the Product Owners (PO) are responsible for this flow. POs translate business needs into practical software requirements, elicit and prioritize requirements, approve software produced for release to customers [1] and make sure the product is profitable [2]. In this way, POs represent the customer demand. However, as described and practiced in Scrum, the Product Owner role may not be sufficient. What is needed is a product analytics capability to constantly evaluate the current value of the products and adjust them. A dedicated manager should systematically discover features that maximize the product value and quickly experiment with the delivery of those features, the cost of their delivery, the usage by customers, and the actual return

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 168–184, 2022.
https://doi.org/10.1007/978-3-031-08169-9_11

on investment from these features, as argued by Fitzgerald and Stol [3]. This is similar to what the Lean Startup approach [4] and dual-track agile [5] aim to achieve through a continuous build-measure-learn loop and experimentation with users. To achieve this, one needs a continuous end-to-end flow between customer demand, business strategy, and software development [3].

Product management is a discipline that can achieve such flow. The role of a product manager is to continuously develop product portfolios and sustain their link to the customer demand [6]. The adoption of product management is now standard in companies like Google, Facebook, Amazon, and Microsoft and has been increasingly popular after the success of Marty Cagan's *Inspired* [6], which provides guidance on product management based on the experience of the most advanced technological companies. The academic literature has also been investigating product management for decades. We now have detailed descriptions of the activities that product managers are supposed to perform towards software development [7], the impact of these activities [8], and how they manifest in practice [9]. However, these descriptions seem to be based on a plan-driven approach, making it unclear whether they can guide product managers in agile companies.

Although we see that product managers clearly have an important role in today's agile companies, we do not fully understand how this role is practiced. While the role of the Product Owner has been extensively researched during the last two decades, the role of a *product manager* has been disregarded by the agile academic community, possibly for being too conservative and plan-driven [10]. Nevertheless, we know from the research on large-scale agile that the PM role is indeed being utilized in agile projects [10–12], but not which activities they perform, how, and why. The answer to these questions may be a way to guide product managers into more agile ways of working. We are therefore asking the following research question: *How is the product manager role practiced in agile companies?*

To answer the research question, we conducted a multiple case study based on data from four agile companies with the product manager role. The paper is structured in the following way. The next chapter gives an overview of the roles responsible for product development (product owner and product manager). Section 3 describes our research approach and the case contexts. We present the findings in Sect. 4 and discuss them in Sect. 5.

2 Background

There are no extensive studies on product management in the agile academic literature. However, a product manager role is similar to that of a Product Owner because both roles represent customer demand. We will thus begin our literature review by describing PO. Then we will draw on the management literature to describe what we mean by a *product manager*.

2.1 The Product Owner Role

The customer relationship is key in agile, where the customer should be on-site and co-located with the development teams [13]. In Scrum, the PO is defined as a person who gathers and prioritizes features, interacts with the customer [1] and communicates the customer's business needs to the development team [13]. The PO also decides on release dates and content and is responsible for the profitability of the product [2]. During the planning meeting (usually every second or fourth week), the product owner presents a prioritized product backlog. The highest priority items from the backlog are then detailed in a sprint backlog by the developers. The development team is responsible for designing, testing, and deploying systems. In Kanban and XP, the PO role is not defined but similar activities are performed. In addition to these practices, Sverrisdottir et al. [14] found in their survey that POs use several additional project management practices.

When there are several teams in an organization (e.g., large-scale agile), POs often form PO teams to gather and prioritize inter-team requirements to solve conflicting and competing business needs [13]. The POs on these teams can either share the responsibility or be responsible for a subset of product features [15]. Bass [13] described the PO role in large-scale as a complex one with a broad set of responsibilities and identified nine different functions: architectural coordination, assessing risk, and ensuring project compliance with corporate guidelines and policies. Further, Berntzen et al. [16] found that there are differences in coordination both amongst POs and between POs and their teams. This may be due to differences in coordination preferences among the POs, different routines in a team, and different understandings of goals. Berntzen et al. [16] also argue the POs need to invest in building good relationships for effective coordination among them. They suggest regular knowledge-sharing activities and retrospectives focusing on improving coordination, strengthening shared knowledge and goals, and reinforcing mutual respect and trust within the PO group.

2.2 The Product Manager Role

Product manager is a role uniting technical and business perspectives in developing software products to provide value to the customer [9]. Product management has been existing since its adoption at Procter &Gamble in the 1930s [17] but did not become popular in software organizations until the late 1990s (aka software product management) [7]. The academic and industrial knowledge on the topic is summarized in the software product management body of knowledge (SPMBoK) described, for example, in [7]. This framework encompasses 38 activities within seven functional areas that product managers are said to be involved in. PMs participate in strategic management, are responsible for product strategy and product planning, and orchestrate development, market, sales, distribution, and service and support. Based on this framework, Maglyas et al. [9] identified 12 activities that product managers are engaged in practice, where *vision creation*, *product lifecycle management*, *roadmapping, release planning,* and *product requirements engineering* are described as core activities (see Table 1 for the definitions). In addition to the core activities, the authors describe other activities that the product manager may be involved in but that can, in practice be delegated to other functions. These are portfolio management, product analysis, product launches, product support, and product (software) development. Finally, Springer and Miller [18] compared product managers across several companies and summarized responsibilities that were the same regardless of context: defining goals, proposing solutions, prioritizing projects or tasks, user research, analysis of requirements, market analysis, stakeholder management, cooperation with the development team.

Table 1. Core product management activities with definitions from [9]

PM activity	Definition
Vision creation	Defining and positioning the product based on the targeted markets, product use, and product scope. A product vision answers the questions of *Where do we want to go? How will we get there? Why do we think we will be successful?* [9]
Product lifecycle management	Planning the product lifecycle by collaborating with company functions (e.g., development, marketing, sales, etc.)
Roadmapping	Planning the evolution of the product showing how product features, technologies, and resources should evolve. Roadmaps are used to translate product strategy into long-term plans and obtain the respective company's commitment and support [7]
Release planning	Selection and assignment of requirements to projects for implementing sequences of product releases [7]
Product requirements engineering	Collecting stakeholder needs, expectations, and ideas for guiding the implementation of the software product. The product manager performs triage to reduce a large number of inputs and to ensure the inputs' relevance and feasibility. These inputs are translated into product features [7]

This review shows that the product manager's responsibilities described in the academic literature do not necessarily reflect how product managers act in agile companies (e.g. described in *Inspired* [6]). Specifically, it does not explain how to integrate product discovery and delivery, which is crucial for creating the desired products [5]. Further, the product manager responsibilities described by the academic literature so far are quite broad and may overlap with other roles. For example, some authors consider product manager and Product Owner to be similar role roles [19]. On the other hand, it is argued that while a Product Owner is a role within the Scrum framework, whereas product manager can be adopted regardless of the framework, and that covers a much broader range of responsibilities [20]. A product manager can sometimes assume a Product Owner role, but this may hinder her from fulfilling other obligations (such as market analysis and product strategy) because PO tasks require intensive interaction with the development team [20]. In large-scale agile, there is no agreement on how and whether a product manager role should be applied. On the one hand, SAFe recommends that PMs be responsible for vision, roadmap, and features to meet customer needs [12]. On the other hand, LeSS outlines the necessity of the product owner to look more outward than just managing the backlog, thus essentially recommending expanding the Product Owner to being a product manager [21]. Nevertheless, the research on large-scale agile has not looked into the particulars of the PM role but focused more on the challenges and success factors of the introduction of the frameworks [10, 11]. For example, it has been documented that product managers can overrun POs when it comes to prioritizing new features [15]. We have thus conducted our own study to examine how the PM role is practiced agile.

3 Methods and Case Description

Our overall research strategy is a multiple case study. We chose this strategy because we were following up the companies for several years and had access to various data sources that allowed for deep insight into the company contexts. We collected and analyzed data in four Norwegian companies that applied agile practices (see Table 2 for an overview of the practices). Company A is a large, globally distributed company focusing on maritime services. Company B is a technology and investment company in digital product development. Company C is a financial services company that offers pension, savings, insurance, and banking products to both the private and the business markets. Company D is a leading app developer and content platform focusing on mobile phone personalization and entertainment.

Table 2. Overview of agile practices and data collected in the case companies

Company	A	B	C	D
Industry	Maritime	Technology and investment	FinTech	Personalization and entertainment
No. employees	3700	80	2160	39
Market	B2B	B2C/B2B	B2B	B2C
Industry	Maritime	Technology and investment	FinTech	Personalization and entertainment
Agile practices	Lean startup principles, Scrum in the development teams, standups in the data scientist team	Lean startup principles, self-managing product teams with agile practices (e.g., standups, retrospectives)	Scrum in the product teams (daily standups, sprint planning, backlog grooming, retrospectives), lean startup principles	Scrum (daily standups, retrospectives, backlog, Scrum of Scrums, elements of Spotify model (tribes)
Data collected	Interviews with product managers, minutes from meetings, product slides	Interviews with product owners, pitching slides, notes from participatory observations	Interviews with product managers, notes from the feedback session, notes from a workshop on product management	Interviews with product managers, getting-started guides from product managers, description of the development process

The primary data sources for this study were interviews with product managers (Table 3) triangulated with other data sources (see Table 2). Interviews ranged between 40 and 90 min and were recorded and subsequently transcribed. All PMs were asked to describe their products, areas of responsibility, work routines, and challenges.

Table 3. Overview of the informants.

Firm	ID	Role	Product description	Interviews
A	I1	Idea owner	AI-enabled services monitoring the condition of ballast water	3
	I2	Idea owner	Data-based service providing insight into vessel's emissions	2
	I3	Idea owner	AI-enabled service facilitating maritime inspection of the hull	3
	I4	Idea owner	AI-enabled service that allows to automatically process Q&A requests	1
B	I5	Startup founder	Mobile application for patient rehabilitation	1
	I6	PM	Online second-hand store	2
C	I7	PM	A set of products that enables digital claims; plus internal automation solutions	1
	I8	PM	A set of products that enables business customers to perform purchases as a self-service	1
D	I9	PM	Interface towards the software users ("user journey")	1
	I10	Technical PM	Internal billing system	1
	I11	SVP* of Product	Content platform focusing on mobile phone personalization and entertainment	1
Total				17

*Note** Senior Vice President (SVP).

The data were analyzed by the first and the second authors using NVivo version 1.6.1. The analysis approach was thematic analysis. The authors first coded all the transcripts in searching for the instances that had to do with the activities of product managers, which resulted in 748 initial codes. The codes were subsequently grouped into higher-order sub-categories (e.g., leading product teams, product monitoring, and adjustment). Finally, we grouped the sub-categories to achieve a logical structure and formulate the overarching categories of activities.

4 Results

Our data analysis resulted in three overarching categories of PM activities: 1) those related to the *products,* 2) related to the *product teams,* and what we coined 3) *supporting* *activities.* We will now describe all the product management activities with the respective quotes in detail (see Table 4 for the total number of quotes per category).

4.1 Product-Related Activities

This category encompasses activities related to developing new products, improving the existing products, and formulating the product strategy.

Product Discovery. A significant aspect of product management was related to exploring new products and business models, something we labeled *product discovery.* Activities in this category could be further grouped into *product ideation* and *idea evaluation. Product ideation* concerned formulating and/or collecting new ideas for products (companies A, B) or features (D). In companies A and B, the product managers came up with new ideas and pitched them internally to receive feedback and potential opportunity for subsequent *idea evaluation.* In company B, the product manager shared his idea to attract internal support: "*I came up with an idea and presented it at an internal forum receive feedback*" (I5).

Table 4. Number of codes per product management activity in the case companies

Product manager activities		Case company			
Category	Sub-category	A	B	C	D
Product related	**A1:** Product discovery	24	20	1	14
	A1.1: Product ideation	1	3	0	2
	A1.2: Idea evaluation	23	17	1	12
	A2: Product monitoring and adjustment	4	0	10	15
	A3: Strategic vision creation	2	2	1	4
Related to product teams	**A4:** Supporting team delivery	2	3	7	9
	A5: Individual follow up	1	0	6	3
	A6: Process lead	1	2	4	4
Supporting activities	**A7:** Engaging internal stakeholders	14	0	4	2
	A8: Collaborating with other PMs	1	0	5	2
	A9: Acquiring resources	2	7	3	0

Idea evaluation comprised examining the market fit for new ideas typically through the use of minimal viable products (MVP) at different stages (e.g., mock-up, working prototype, pilot version of the product). Exemplar activities in this sub-category are gathering user feedback on the MVP with the purpose of evaluating whether there exists the need for the product and whether the product can create revenue. A PM from company A described: "*We build something that is cheap to build, which is fast to build, and then we will test our assumptions that it will give value for the customers and that they are* *really willing to pay for this*" (I4). *Idea evaluation* is happening hand in hand with the product team (e.g., UX designers and software engineers) that collaborates with a PM to develop a first working prototype and then incrementally adjust it according to user feedback.

Product Monitoring and Adjustment. If *product discovery* relates to exploring new products and features, *product monitoring and adjustment* encompasses activities directed to the existing products and product portfolios. These activities were characteristic of companies C and D, where PMs worked primarily on improving the existing products. Typical examples from this category were participating in joint planning and coordinating events to evaluate the current state of the products and set priorities and product goals for the subsequent period. KPIs were often used to track the product performance and report on the planning events. In terms of formulating and communicating priorities, roadmaps were often-applied. PMs in the large company C felt bounded by the roadmaps they committed to because they needed to "please" all internal stakeholders who influenced the prioritization. At the same time, PMs in a much smaller company D were more flexible in the ways they applied their roadmaps. Product manager I9 said: "*It's really unlikely to get through that roadmap exactly as it is, then you're doing something wrong. So, it's nice to have ideas and to have a plan on what you want to work with, and to be able to present that to the rest of the company. But it's on the premise that this will change*" (I9).

Strategic Vision Creation. For the product managers to guide their *product discovery* and *monitoring and adjustment* activities, they need to outline the strategic vision for their product. These activities were typical for companies A, C, and D as they had established business goals. In companies A and C, the product managers utilized higher order business goals to create the strategic vision for their products. In contrast, product managers in company D were part of the business goals for the whole company. The output of this activity is typically formulated using Objectives and key-results, and KPIs. The frequency of this activity varies from company to company (from annually in C and D to a 5-year horizon at A). A PM from company C outlined: *Our company has some fluffy overarching business goals. We are using those to formulate our objectives based on those goals. The objectives are set on a one to two-year basis. We then define measurable key results for the next quarter*" (I7).

4.2 Activities Related to the Product Teams

The work of the product managers did not stop when UX designers and developers worked on the product. The PMs took on multiple leadership responsibilities to ensure that the development of the product was on track. We identified four areas of the team leadership activities across the case companies: *Coordination activities, process lead, support teams delivery,* and *individual follow-up.*

Support Team Delivery. The PMs took an active role in supporting the teams throughout the various stages of product development. PMs in all companies were involved in discussions and dialogs with the UX designers and developers, ensuring that the business, design, and technology aspects were considered. Product managers also communicated goals by presenting them to the product teams. Some PMs were also collecting feedback on the goals and how to measure their success (e.g., key results) from the team members. They did not only monitor the development progress but worked together with the teams. In companies C and D, the PMs collaborated with the product owners on the backlog and formulating acceptance criteria. I9 described: "*I work really closely with the product owner and try to bring him in quite early to the discovery process. Once something goes into development, he's the one who's responsible for keeping on track*". However, one PM from company D was clearly acting like a Product Owner and took full responsibility for the product backlog. She explained: "*As a product manager, I am not doing anything different from when I was a PO. Because it's the same; your main goal is to ensure that your product goal as planned*" (I10).

Process Lead. PMs in all case companies took on a *process lead* role by structuring the work process of the product teams (e.g., running agile meetings (company C), arranging kick-offs for new products and features (company D), helping to find better ways to collaborate (company C), coaching (company C) and setting up and leading new teams (company B and C). PMs in company D had the team lead responsibilities ensuring the team was motivated and worked on improving the development process. She said: *"I work on the process and how we can improve it*" (I10). As shown in Table 4, activities in this sub-category did not occur equally frequently across the case companies. In companies C and D, the process lead aspect of the product management role was more prominent. For example, in C, the PMs took responsibility for finding the best way for teams to come back from the home office at the end of the COVID-19 pandemic. PM I8 said: "*We chose Wednesday as an office day based on a team survey. On Tuesday, we have a mix of work from home and office*" (I8).

Individual Follow-Up. In cross-functional product teams, members are highly specialized in their tasks, from designers to backend developers. PMs took responsibility for following up with individual team members by running one-on-ones (company C and A), supporting new team members (C), and even monitoring the members' emotional state (company D). For example, a product manager in D set up a tool for tracking the team's health. She explained: "*You have like battery pictures, and every team member should indicate if he feels fully charged or empty"* (I10).

4.3 Supporting Activities

Apart from being responsible for the products and product teams, the PMs engaged in activities that helped them successfully fulfill their other tasks, which we called *supporting activities.*

Acquiring Resources. Many PMs took responsibility for acquiring both financial and human resources to fulfill the product goals. In company B, product managers were actively acquiring external financing for their new products. They were also competing with other PMs and functions in the company to attract the software developers and convince them to work on their products. In big companies, PMs were also attracting resources, normally by contacting internal stakeholders (A7) to allocate additional budgets or software workforce. In A and C, which were large-scale organizations with independent software units, the competition for developers was even stronger. One product manager described how he had to “fight” for the software developers: *“I had just to threaten that I would go extremely high in the organization if they did not give me the software resources, so I received them at the end (laughs)*” (I4).

Collaborating with Other Product Managers. Such activities played an essential role for the product managers who relied on each other to coordinate, exchange knowledge and experience, and sometimes solve the problems together. PMs in companies C and D coordinated their collective effort through regular steering meetings. The product managers expressed several challenges regarding their roles and how to perform their tasks. They believed that discussing with other PMs could help. Companies A and C had formalized communities of practice for the product managers where topics related to the role were discussed. In contrast, D had an informal CoP that did not have a specific structure or agenda. A product manager from C said, “*Lean coffee is a place to discuss methods and how we work together. We nominate topics before the meetings and arrange it every other week*” (I7).

Engaging Internal Stakeholders. Product managers often link the product teams and other functions (e.g., finance, legal, sales, and marketing) or members of multiple product teams. In companies C and D, it was crucial to consider the stakeholders’ interests because they partly constituted an input to what the product teams were supposed to deliver. For example, I4 from company A took charge of multiple delivery teams to develop and integrate the new product into the existing ecosystem. He arranged coordination meetings twice a week where three different teams met for 15 min. Multiple products in company A were based on internal and external data for both the data scientist and the product managers to understand how the data should be contextualized. A product manager explained: “*The data scientist had competence on the things I did not know. And a fantastic ability to understand the business context, not only the technical data parts*” (I3).

4.4 Activities Across the Companies

We have observed that the frequency of product managers mentioning the activities varied from company to company which can partly validate our findings. As can be seen in Table 4, **Product discovery (A1)** was often mentioned in A, B, and D, but not in company C. This corresponded well to our observations and collected documents from the companies, as both A, B, and D were heavily focusing on *new* products and services, whereas company C mainly was concerned with the evolution of the *existing* services. This is also supported by the frequent mentioning of A2 (**Product monitoring and adjustment**) in company C. At the same time, A2 was not so frequent in companies A and B, indicating that the product managers were not involved in working with the existing products (because the company did not have a holistic product approach to the current portfolio). In company B, all products were relatively new since it was a venture builder; thus, there were no activities identified as **product monitoring and adjustment (A2)**. Another striking difference is that **A7 (engaging internal stakeholders)** was mentioned more often in company A than in all other companies. Although this can partly be explained by the high number of interviews in that company, it is also worth noting that company A is very large and has only a short history of product management, where the product managers were very new to their tasks. Therefore, PMs had to **engage internal stakeholders (A7)** for collaboration, validation, coaching, and, not the least, for **acquiring resources (A9)**. Finally, **acquiring resources (A9)** was described as a PM activity in all companies except D probably because, in that company, the product teams were fully dedicated to their respective products, which was not always the case for the other cases. In A and C, software developers could sometimes be moved from team to team because the software resources were insufficient. In B, PMs were competing for the interest of the developers, who could be involved in the products part-time.

The relationships between the three categories are visualized in Fig. 1. *Product discovery* is at the center of what a PM does. *Product discovery* iterates between *product ideation* and *idea evaluation*, which happens in close collaboration with the product team (arrows toward the *team activities*). *Product discovery* contributes to *strategic vision creation* and is also defined by it. Finally, *product discovery* creates the need for *supporting activities* (e.g., acquiring resources), which mobilize organizational resources to achieve optimal product outcomes.

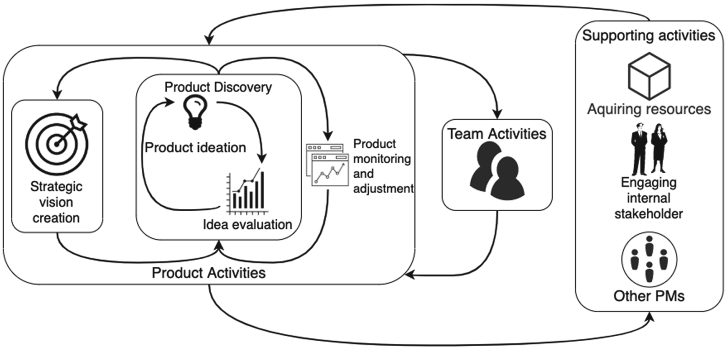


Fig. 1. Product manager activities

5 Discussion

The adoption of product management is growing, and the practice is used by companies like Google, Amazon, Facebook, and Microsoft. However, there is a lack of research on how the product manager role is practiced in agile organizations. Therefore, we have described what activities the product managers performed in four agile companies. We will now answer our research question, "*How is the product manager role practiced in agile companies?*" by discussing the three groups of activities described in the Results section: product-related, team-related, and supporting activities.

5.1 Product-Related Activities

The core activities of the product managers in our study were related to discovering and developing the products (activities **A1, A2** in Table 4). While the PO role in Scrum is about gathering and prioritizing features [1], the product managers in our cases focused primarily on formulating the hypotheses on which features should be developed and then testing these hypotheses to provide a further direction and insight for the product teams. Instead of believing that the customer requirements exist upfront and should only be gathered, as POs would do, our PM would first ask whether the features are needed in the first place **(Activity A1.2)**. Thus a lot of the effort of product managers in our study was dedicated to hypothesis formulation and testing in close collaboration with both the users and the product team, which is the essence of the Lean Startup [4] and dual-track agile [5]. These PM activities also remind what is described by the SAFe-framework, where the PMs are responsible for "defining and supporting the building of desirable, feasible, viable, and sustainable products that meet customer needs over the product-market lifecycle" [12] The PMs were working in this way both when it comes to new (**A1**) and existing products (**A2**) in the companies of different scale. This highlights that product discovery can be applied in most organizational contexts.

In addition to discovering and developing the products, product managers in most companies were involved in formulating the strategic vision for their products/product areas and even overall business strategies of their companies (**A3: Strategic vision creation**). This is in line with the concept of BizDev and the idea that integration between business strategy and software development is needed in the same way as between development and deployment [3]. A model for continuous experimentation [22] also suggests that product and business strategy should be informed by the results of systematically testing the product assumptions.

Earlier research on product management described roadmapping, release planning, and product requirement engineering as separated activities [9]. In contrast, we found that these activities are not always possible to differentiate in the agile context because they are all part of product discovery. This is in line with Fabijan et al. [23] who highlight the importance of evolving the continuous experimentation approach from the ad-hoc approach in new products to a more targeted experimentation for established products. In the same way, development of the existing products (**A2: Product monitoring and adjustment**) required both portfolio management, product lifecycle management, and roadmapping that are all part of the same activity (e.g., planning events and product steering forums in companies C and D). Therefore, we believe our description of product manager activities better fits the PM practice in agile firms compared to the earlier frameworks (e.g., described in [7]).

5.2 Team-Related Activities

We found that product managers had several responsibilities toward the product teams, including **supporting their delivery (A4)**, **following-up individual team members (A6),** and being **process leads (A7)**. These activities are somewhat similar to those of a Product Owner. POs are typically involved in steering the delivery of the teams by deciding on the release date and content and prioritizing the product backlog [2].

We found that in one case (informant I10) a PM who earlier was a PO, continued to identify herself as PO and was assuming PO responsibilities (e.g., managing the backlog, formulating the requirements). However, another PM, who had not had experience with agile before, was clearly distinguishing herself from a Product Owner. She talked about a PO as her partner whose responsibility was to make sure that the tasks were "on track." This shows that some PMs can confuse these two roles because many agile practitioners are less familiar with the PM-role. While some companies introduce the role due to its increasing popularity, the content of the PM and the PO-roles may overlap. The inconsistency of how much the PM role is similar to that of a PO can be explained by the size of the company. Earlier findings suggest that product managers can assume a PO-role in small companies because they have fewer responsibilities around steering the product [20]. However, other sources describe PO and PM as two distinct roles that sometimes even compete with each other [15]. We can conclude that the PM-role can partly overlap with that of a PO, but that the PM-role is much broader, as we have identified a plethora of PM activities that a typical PO does not cover.

An example of such activities of PM is functioning as **process leads (A7)** and even **following-up team members (A6).** We were surprised to find out that all PMs were so attentive to their teams given that earlier literature on product management argues that a PM should not have responsibilities toward teams [7, 20]. In contrast to this, PMs in our study reminded us of Scrum Masters or agile coaches [24] in that they took responsibility for the team goals, climate, and process structure.

We found that many product managers defined goals for the product teams (**A4**). These findings correspond well to what had earlier been described by both managements literature [7, 18] and agile practitioners (e.g., the SAFe-framework [12]). However, the fact that only certain PMs involved product teams in the goal-setting (e.g., by formulating Key results) is alarming. Moe et al. found that if a process lead does not involve teams in goal setting, team autonomy may be reduced [25], which jeopardizes the agile principles. Autonomy is also crucial when new products are developed inside established companies [26, 27]. We can thus recommend that product managers in agile companies collect the teams' feedback on the team goals.

5.3 Supporting Activities

While the PO collaborates with customers and other POs, in our case, the product managers acted more as negotiators that took into account the ideas and interests of various **internal stakeholders** (**A7**) and sometimes convinced them to **allocate additional resources** (**A9**). We found that engaging internal stakeholders was especially important in the large established company A. When developing new digital products in such companies, many functions can be involved in the product development (e.g., legal, marketing, sales, etc.) [28]. Therefore, the PMs need to collect their input on the new products. Our findings are in line with [18], who highlights a PM's responsibility as a stakeholder manager. This is also consistent with what was described by Mikalsen et al. [29] that an agile product team needs to negotiate with several other departments and stakeholders to reduce dependencies. We thus highlight the role of a product manager as a negotiator in agile organizations.

We also found that product managers worked together and that many of them saw value in communities of practice (**A8**). Just as product managers in our study, Product Owners also tend to team up when they need to solve competing business requirements (e.g., in large-scale agile) [13]. Communities of practice are often introduced in large-scale agile [11], and internal software startups [30] to improve learning and knowledge-sharing. However, neither academic nor practical literature on product management (such as Inspired [6]) described such interaction with other PMs as crucial to their job. Therefore, we suggest that product managers working in agile firms allocate sufficient time to collaborate with other product managers.

6 Practical Implications

Based on our results, we can summarize some recommendations for those working as product managers. First, a PM should serve as a continuous link between business needs and software development. Our results show that this is the essence of product management regardless of the company context. Thus, we believe that all PM practices should be chosen based on whether they contribute to achieving this goal. We also see that both internally and externally communities of practice across product managers is one way to learn and teach such practices. Our next advice is for the PMs working in large companies facing organizational inertia in product development. Most likely, such inertia is what you as a product manager will and should deal with. Our findings show that successful PMs actively involve internal stakeholders and collaborate with them to achieve a more seamless product development. This implies that PMs should have a good network within their company and have solid negotiator skills. Finally, we recommend that product managers in agile companies invite their product teams to set goals for themselves (e.g., by facilitating the formulation of the so-called "key results"). Many PMs were asking the teams to formulate their own goals, which had been shown to increase teams' autonomy and hence agility.

7 Conclusions, Limitations, and Further Work

Despite the increasing popularity of product managers in agile companies, little research exists on how this role is performed in practice. We have thus conducted a multiple case study of product managers to find out how the product manager role is practiced in agile companies. Given today's increasing rate of PM-role adoption, our findings can provide guidance for how product managers should work. The paper's contribution is a summary of the PM activities toward products, teams, and organizations, which is a step toward a theoretical understanding of agile product management. We found that the essence of the PM's role in agile is to make sure that the products are continuously linked with market demand, which is in line with the concept of BizDev. The main goal of a product manager is to set up experiments that will help the product teams decide which features are needed for the new or the existing products. Besides, PMs are highly dedicated to their teams, supporting their delivery, individual members, and their overall autonomy. In this sense, the role is sometimes practiced in a similar fashion as the Product Owner (e.g., managing backlog, keeping things on track). However, the PM role involves responsibilities that a typical PO does not cover (e.g., contributing to the organizational strategy and acquiring additional resources).

This was the first academic attempt to holistically describe the role of product managers in agile companies, which is not without limitations. First of all, the study relied on a specific sample (companies based in Norway), which may reduce the generalizability of the results. We thus encourage researchers to investigate agile product management in other countries further. Second, we provided only a preliminary description of how organizational context (large- vs. small-scale, B2B vs. B2C) influences PM practices. Further investigations are needed for more conclusive results. Finally, we need to know more about how exactly the product manager role differs from that of a Product Owner. We also need to understand how these two roles may collaborate to achieve optimal product outcomes.

Acknowledgments. This study was supported by the 10xTeams, A-teams, and Digital Class projects; and the Research Council of Norway (grants 267704, 309344, and 309631).

References

1. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall Upper Saddle River (2002)
2. Sutherland, J., Schwaber, K.: The scrum papers: nuts, bolts and origins of an agile process (2007)
3. Fitzgerald, B., Stol, K.-J.: Continuous software engineering: a roadmap and agenda. J. Syst. Softw. **123**, 176–189 (2017). https://doi.org/10.1016/j.jss.2015.06.063
4. Reis, E.: The Lean Startup, vol. 27. Crown Business, New York (2011)
5. Trieflinger, S., Münch, J., Heisler, B., Lang, D.: Essential approaches to dual-track agile: results from a grey literature review. In: Klotins, E., Wnuk, K. (eds.) ICSOB 2020. LNBIP, vol. 407, pp. 55–69. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67292-8_5
6. Cagan, M.: Inspired: How to Create Tech Products Customers Love. Wiley (2017)
7. Fricker, S.: Software product management. In: Maedche, A., Botzenhardt, A., Neer, L. (eds.) Software for People (2012). https://doi.org/10.1007/978-3-642-31371-4_4
8. Ebert, C.: The impacts of software product management. J. Syst. Softw. **80**, 850–861 (2007). https://doi.org/10.1016/j.jss.2006.09.017
9. Maglyas, A., Nikula, U., Smolander, K., Fricker, S.A.: Core software product management activities. J. Adv. Manag. Res. (2017)
10. Paasivaara, M., Behm, B., Lassenius, C., Hallikainen, M.: Large-scale agile transformation at Ericsson: a case study. Empir. Softw. Eng. **23**(5), 2550–2596 (2018). https://doi.org/10.1007/s10664-017-9555-8
11. Edison, H., Wang, X., Conboy, K.: Comparing methods for large-scale agile software development: a systematic literature review. IEEE Trans. Softw. Eng., 1 (2021). https://doi.org/10.1109/TSE.2021.3069039
12. Scaled Agile: SAFe 5.0 Framework. https://www.scaledagileframework.com/. Accessed 21 Feb 2022
13. Bass, J.M.: How product owner teams scale agile methods to large distributed enterprises. Empir. Softw. Eng. **20**(6), 1525–1557 (2014). https://doi.org/10.1007/s10664-014-9322-z

14. Sverrisdottir, H.S., Ingason, H.T., Jonasson, H.I.: The role of the product owner in scrum-comparison between theory and practices. Procedia Soc. Behav. Sci. **119**, 257–267 (2014)
15. Paasivaara, M., Lassenius, C., Heikkilä, V.T.: Inter-team coordination in large-scale globally distributed scrum: do scrum-of-scrums really work? In: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 235–238 (2012)
16. Berntzen, M., Moe, N., Stray, V.: The product owner in large-scale agile: an empirical study through the lens of relational coordination theory. In: Kruchten, P., Fraser, S., Coallier, F. (eds.) XP 2019. LNBIP, vol. 355, pp. 121–136. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19034-7_8
17. Kittlaus, H.-B., Fricker, S.A.: Software Product Management. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-642-55140-6
18. Springer, O., Miler, J.: The role of a software product manager in various business environments. In: 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 985–994 (2018)
19. Kelly, A.: The product manager role. In: The Art of Agile Product Ownership, pp. 97–101. Apress, Berkeley, CA (2019). https://doi.org/10.1007/978-1-4842-5168-3_14
20. Kittlaus, H.-B.: Software product management and agile software development: conflicts and solutions. In: Software for People, pp. 83–96. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31371-4_5
21. The LeSS Company: Product Owner. https://less.works/less/framework/product-owner. Accessed 21 Feb 2022
22. Fagerholm, F., Guinea, A.S., Mäenpää, H., Münch, J.: Building blocks for continuous experimentation. In: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering - RCoSE 2014, pp. 26–35. ACM Press, Hyderabad (2014). https://doi.org/10.1145/2593812.2593816
23. Fabijan, A., Dmitriev, P., Olsson, H.H., Bosch, J.: The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale. In: 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), pp. 770–780. IEEE (2017)
24. Stray, V., Tkalich, A., Moe, N.B.: The agile coach role: coaching for agile performance impact. In: Proceedings of the 54th Hawaii International Conference on System Sciences, p. 6816 (2021)
25. Moe, N.B., Dahl, B., Stray, V., Karlsen, L.S., Schjødt-Osmo, S.: Team autonomy in large-scale agile. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019). https://doi.org/10.24251/HICSS.2019.839
26. Tkalich, A., Moe, N.B., Sporsem, T.: Employee-driven innovation to fuel internal software startups: preliminary findings. In: Gregory, P., Kruchten, P. (eds.) XP 2021. LNBIP, vol. 426, pp. 145–154. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88583-0_14
27. Tkalich, A., Moe, N.B., Ulfsnes, R.: Making internal software startups work: how to innovate like a venture builder? In: Wang, X., Martini, A., Nguyen-Duc, A., Stray, V. (eds.) ICSOB 2021. LNBIP, vol. 434, pp. 152–167. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91983-2_12
28. Tkalich, A., Mikalsen, M., Moe, N.B., Sporsem, T.T.: Digital transformation of a traditional business model: a case Stury from the maritime industry. Presented at the European Conference on Information Systems (ECIS), Marrakech (2021)

29. Mikalsen, M., Moe, N.B., Stray, V., Nyrud, H.: Agile digital transformation: a case study of interdependencies. In: International Conference on Information Systems 2018, ICIS 2018 (2018)
30. Sporsem, T., Tkalich, A., Moe, N., Mikalsen, M., Rygh, N.: Using guilds to foster internal startups in large organizations: a case study. In: Gregory, P., Kruchten, P. (eds.) XP 2021. LNBIP, vol. 426, pp. 135–144. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88583-0_13

A Countrywide Descriptive Survey of Agile Software Development in Brazil

Rafaela Mantovani Fontana[1](✉), Jaime Wojciechowski[1],
Razer Rojas Montaño[1], Sabrina Marczak[2], Sheila Reinehr[3],
and Andreia Malucelli[3]

[1] Universidade Federal do Paraná, Curitiba, Brazil
{rafaela.fontana,jaimewo,razer}@ufpr.br
[2] Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brazil
sabrina.marczak@pucrs.br
[3] Pontifícia Universidade Católica do Paraná, Curitiba, Brazil
sheila.reinehr@pucpr.br, malu@ppgia.pucpr.br

Abstract. For years, industry institutions and academic researchers have been surveying software practitioners on agile software development methods adoption. These surveys have been useful in describing the characteristics, challenges, and impacts of agile adoption, mainly in Europe and North America. Latin American practitioners miss information on the state of agile adoption. This study aims to fill this gap by describing agile software development adoption in Brazil. We collected data from 897 countrywide-distributed practitioners. We used descriptive statistics and machine learning algorithms to understand our dataset. Results show the profile of companies and teams, characteristics of agile usage, perception of success, applied principles and practices, and reasons, challenges and impacts of agile adoption. We also explore the relevance of principles in software process improvements. We contribute by mapping the state-of-the-practice of agile adoption in Brazil and by contrasting our results to previous literature, which points out how we further current knowledge in academia.

Keywords: Agile software development · Brazil · State of agile · Challenges · Improvements

1 Introduction

Agile Software Development (ASD) arose in the early 2000s [4] and several studies [11,13,14] have aimed to understand challenges faced and mechanisms adopted by teams in the transformation to agile, which encompasses mainly changes in team culture, people skills, and mindset. Among the studies investigating agile adoption, opinion-based surveys contribute to bringing up a big picture of how practitioners have embraced these agile practices.

Industrial surveys to investigate ASD are common [35], and some of them have been conducted year after year for years now, as Version One's [30]. In

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 185–202, 2022.
https://doi.org/10.1007/978-3-031-08169-9_12

academia, we also find surveys that describe agile adoption, but they are usually less frequent. The last countrywide survey in Brazil took place in 2011 [26].

The goal of this study is thus to describe current agile usage in Brazil. We conducted a countrywide survey in 2018–2019, asking ASD practitioners which are the practices, principles, and methods they apply. We also investigated the perception of project success, reasons for agile adoption, challenges faced, and the impacts they perceive. Furthermore, we collected data to identify which principles influence improvement in different aspects of the software process. Data were analyzed using descriptive statistics and machine learning techniques.

The remainder of this paper is organized as follows: Sect. 2 briefly presents related work, from the perspective of academic surveys. Section 3 presents our research approach. Section 4 shows the results and Sect. 5 discusses our results by comparing them with other ASD surveys. Section 6 concludes the paper.

2 Related Work

Industrial surveys have long been performed in the context of ASD [30]. They have been serving as a benchmark for practitioners to understand other companies' characteristics and outcomes [33]. Besides methodological limitations pointed out by Stavru [35], these surveys are mainly represented by practitioners in North America and Europe. Academic studies, on the other hand, have a less frequent application but present more methodological rigor. They usually focus on specific contexts but are still important since they characterize the studied community and allow for future comparisons between contexts.

For example, Livermore [25] studied the Extreme Programming (XP) adoption among 112 practitioners associated with the Software Engineering Institute's Software Process Improvement Network (SPIN). In the European context, Salo and Abrahamsson [34] report on Extreme Programming and Scrum adoption in 13 industrial organizations and Kuhrmann et al. [23] showed how 69 European practitioners combined agile development with traditional approaches. In Finland, Rodriguez et al. [33] investigated the adoption of Lean Principles while also studying ASD adoption. Bustard et al. [9] studied how agile is adopted in 37 software companies from the Northern Ireland. There are also surveys characterizing the adoption of agile in India [28] and in North America [32].

From South America, Melo et al. [26] present the results that serve as a reference to our study. They conducted a large-scale survey in Brazil in 2011. They had 471 participants. Still focusing on Brazil, the study by Diel et al. [12] was conducted to describe the understanding that Brazilian practitioners have about agile methods. This study collected data from about 200 professionals mainly located in Brazil' South and Southeast regions. Two years later, Bolatti et al. [5] conducted a smaller survey in Argentina (79 participants) given the lack of studies focusing in the Argentinian market.

The above shows that academic agile surveys on how ASD has been adopted in different contexts and countries indeed have been conducted for several years. But it also demonstrates how infrequent they are. Our study aims to update the current state-of-practice in Brazil taking Melo et al.'s work [26] as reference.

3 Research Approach

The goal of this research is to describe agile usage in Brazil. The GQM model – Goal-Question-Metric [3] – that guided our study design, is shown in Table 1. Using GQM approach is a recommended practice in surveys to limit scope and stick analysis to the research objective [27]. We chose the opinion-based survey (or survey only from now on) as our research method. A survey is a "comprehensive system for collecting information to describe, compare, or explain knowledge, attitudes, and behavior" [31, p. 16]. Surveys must have a target population - the group of individuals to whom the survey applies [20]. In this survey, the population is information technology practitioners that work with agile methods in Brazil. We chose a non-probabilistic sampling, denoted convenience sampling, proper for when respondents are easily accessible [20]. Our sample is practitioners that attend agile industrial conferences in Brazil. We chose this approach to locate those who work with agile methods more likely. We collected data in 6 editions of 4 distinct agile software development industrial conferences during 2018 and 2019, namely: Agile Trends (2018 and 2019 editions), Agile Trends Gov (2018), Agile Brazil (2018 and 2019), and Agilidade Recife (2019)[1].

Kitchenham and Pfleeger [18] advocate that survey instruments are conceived in four steps, namely: 1) search the relevant literature; 2) construct the instrument; 3) evaluate the instrument, and 4) document the instrument. We based our instrument on existing studies for our research's first step. We started (Step 1) by searching for the relevant literature, carefully analyzing the questions published in the following reports: The 11th annual state of the agile report - Version One [29], Azizyan et al. [2], Rodriguez et al. [33], Melo et al. [26], Bustard et al. [9], Diel et al. [12], Bollati et al. [5], and Kuhrmann et al. [23]; and proposed our questionnaire from them (Step 2)[2]. Next, we evaluated our instrument (Step 3) [19] with six full-professors and researchers for readability and completion time. We also applied it with an experienced practitioner for content validation. In the fourth step, we documented the process in our research protocol.

We chose to make the questionnaire available for the first conference (Agile Trends Teams - São Paulo/SP, in 2018) using the Qualtrics tool, but the strategy was not effective, as we got only 16 responses out of 854 attendees. We changed then the approach to apply the questionnaire personally, in a printed form; this way we could approach people face-to-face and ask them for their attention [19].

Our new data collection strategy included personally approaching conference attendees during check-in and coffee breaks. Three or four people (depending on the conference size) were hired to aid data collection. In Agile Trends Gov, Brasília/DF (2018), there were 192 filled questionnaires out of 550 attendees; in Agile Brazil (Campinas/SP), 2018, we got 225 responses (we do not have the

[1] The 2020 editions were called off giving the Covid-19 pandemic. We had initially planned to collect data at this time too; thus we do have the most recent data that was possible to collect. Conferences in 2021 were shorter in days and in programme, and we judged it was best to not add extra work for people during a pandemic.

[2] Our questionnaire and the mapping of where the questions that compose it came from can be found at https://doi.org/10.5281/zenodo.5997108.

Table 1. Research goal, questions and metrics

Goal: to describe agile usage in Brazil	
Question	Metric
Which is the profile of companies that use agile methods?	Percentage of companies* in different sizes, locations in Brazil, team sizes, distributed teams, and industry
Which is the profile of practitioners that use agile methods?	Percentage of practitioners with different ranges of experience with software development and agile methods
What are the characteristics of agile software development usage?	Three metrics describe these characteristics: 1) the percentage of respondents that apply different methods; 2) the percentage of respondents that combine agile methods with traditional ones, and 3) the adoption range in their companies
What is the perception of success in agile projects?	Percentage of respondents that perceive projects are successful or not
What is the extent to which agile methods principles are applied?	Percentage of respondents that apply agile principles with different intensities and respondents who apply principles considering their project success perception
Which are the reasons for adopting agile methods?	Percentage of respondents that point out different reasons for adoption
Which are the practices applied?	Percentage of respondents that apply different practices
Which are the challenges faced?	Percentage of respondents that perceive different challenges
Which are the perceived impacts with the agile adoption?	Percentage of respondents that perceive different levels of impact in software process aspects
Which principles affect the perception of improvement in software processes?	True positive values for machine learning models to predict improvements as outcomes based on the agile application of principles

*We consider that the respondent represents the company in describing agile usage aspects.

number of attendees); in Agile Trends Teams (São Paulo/SP) in 2019 we got 226 responses (from 898 attendees); in Agile Brazil (Belo Horizonte/MG), 2019, there were 161 responses out of 771 participants in the conference; and, finally, in Agilidade Recife (Recife/PE), in 2019, there were 77 responses from a group of 350 attendees. We got 551 full-responses from 897 answered questionnaires.

We chose to consider also the partially responded questionnaires for data analysis given that questions can be individually analyzed. Kitchenham and

Pfleeger [21] recommend doing so when questions are independent of one another. All of our questions data were analyzed with descriptive statistics. Cronbach's Alpha was used to measure consistency when applicable.

To complement our analysis, we used machine learning (ML) to predict improvements in the software process's different aspects based on principles application. The application of ML techniques instead of statistical ones – such as Linear Discriminant Analysis (LDA) or Logistic Regression (LR) – has been shown to perform better in several application domains [1,10,17]. Furthermore, the application of statistical procedures require assumptions about data or about relationships among them, such as homoscedasticity, which are not necessary when using ML [7].

Three different techniques were applied: Artificial Neural Network – ANN [16], Support Vector Machine - SVM [36], and Random Forest – RF [6]. The first – ANN – was used to identify which *improvements* were predicted by applying specific sets of principles. Then, we used SVM and RF to determine *which principles* were more relevant to get to improvements. We trained the ML models in two rounds as follows.

First Round. We trained thirty different ANNs (using Weka software) for each of the evaluated impacts (each data set). The values we tested for the hidden layers parameter were 30, 40, 60, and 70, for the learning rate parameter were 0.2, 0.5, and 0.7, and for the momentum, parameters were 0.3, 0.5, and 0.7. A specific value for the predictor variable is called "class", and our interest was in class "Improved". For the class "Improved", the classification precision was used to identify which impacts could be predicted by applying certain principles. We chose the impacts that had accuracy for "Improved" class higher than 97%;

Second Round. After identifying which impacts were predicted by applying certain principles, we trained the model with SVM and RF algorithms (using R programming language) to determine precisely which principles mostly affected the perceived impact by extracting the most important attributes when training the models. Both models resulted in all prediction statistics and showed each attribute's relevance in the prediction. We executed 3328 SVMs with a radial base function kernel for each data set in the SVM execution. The C parameter was adjusted from 0.01 to 100. The sigma kernel parameter was tested from $10 \wedge -15$ to $10 \wedge 3$. The models' quality was measured using holdout with 80% for the training data set and 20% for the test data set. In the Random Forest execution, the parameter that sets the number of trees was tested from 100 to 1000. The parameter that sets the number of attributes used was tested from 1 to the total number of attributes in each data set. As a result, we considered the 75% more relevant attributes for each evaluated aspect. Using this information, we identified which attributes (principles) were more critical in predicting the resulting variable (improvement in each aspect).

3.1 Threats to Validity

Reliability and validity are relevant concerns in survey research [8]. We consider we addressed reliability by using questions already asked and analyzed in different contexts by other researchers, and using Cronbach's alpha test for our questions, which ranged from 0.51 to 0.92. Regarding validity, we addressed it by using the accuracy and precision metrics – outcomes from ML techniques, and using the measures already applied in numerous research in other contexts and other moments in time and comparing them. On respect to generalizability, ours was a convenience non-probabilistic sampling. Our data is representative only to the context where we have collected them, considering that conference attendees might represent a subset of agile practitioners profile. However, as for the reference studies we used here, serve as a benchmark for comparison to other surveys with the same objective. Thus, there is a practical relevance in replacing the statistical relevance [20].

4 Results

As previously mentioned, this survey aims to describe agile software development usage in Brazil. We collected 897 responses (men = 69.8%, women = 30.2%). Over 40% of the participants (41.6%, n = 897) have between 36 and 45 years old and the remaining are distributed as follows: 6.7% are under 25 years, 45% have between 26 and 35 years, and 6.7% are above 45. Our results are described next as per the research questions and metrics established in our GQM model.

Which Is the Profile of Companies that Use Agile Methods? Regarding the company size where the respondents' work (n = 897), more than half of the respondents (61.1%) reported to work in companies with 1000 people or more and a bit over one quarter of them (26.8%) in companies with 100 to 999 people. Other results include: 1.7% - less than 9 employees; 5.7% - 10 to 49 people; and 4.8% - 50 to 99 people. These companies are mainly distributed in the following Brazil's regions: 55.7% in Southeast, 15.5% in Midwest, and 11.8% in Northeast.

Regarding team sizes, of all valid responses (n = 876), 13.6% work in a team with less than 6 people; 22.8% in teams with 6 to 10 people; 20.8% with 11 to 20 people; 15.0% work in teams with 21 to 50 people; and 27.9% work with more than 50 people. When asked about the teams physical distribution (n = 888), 60.0% said the teams are not distributed; 28.0% said they are distributed within Brazil; 9.0% is globally distributed; and 2.9% in located in South America only.

The top-5 industries that the participants' companies belong to are: software (34.8%), financial services (27.5%), government (26.7%), education (9.9%), and internet services (9.3%). Due to the expressive amount of respondents related to the Brazilian government and public services, an excerpt from this specific 2018 dataset is reported in [15]. Moreover, regarding the length of time that the respondents' companies use agile methods (n = 880), 14.7% said it is less than a year; 28.1% have 1 to 2 years of agile usage; 34.3% have 3 to 5 years of use; 15.9% have 6 to 10 years; and 7.0% have more than ten years.

Which Is the Profile of Practitioners that Use Agile Methods? When asked about their software development experience, 6.8% stated to have less than a year of experience; 5.9% have 1 to 2 years; 12.2% have 2 to 5 years; 25.0% state to have 5 to 10 years of experience. An expressive amount, 39.9%, have 10 to 20 years of experience; and 10.2% have more than 20 years (n = 844). Concerning their experience with agile methods, 11.4% have very little knowledge, 61.5% are moderately experienced, 20.7% are very experienced, and 6.4% declared to be extremely experienced in agile methods (n = 886).

What Are the Characteristics of Agile Software Development Usage? Our questionnaire asked practitioners about the methods they use. We also asked whether they usually combine agile methods with more traditional ones, i.e., hybrid methods [23]. We found out that 79.3% use Scrum, 67.3% use Kanban and 21.6% point out to combine them with Scrumban. As relevant results we also see that 20.6% report to apply a hybrid customized method, 15.6% use Scrum/XP hybrid and 11.6% report to use Lean Development. Finally, 11% report to use XP (Cronbach's alpha = 0.51, n = 893). Respondents could choose more than one option.

When asked whether they combine these agile methods with traditional ones, 56.7% combine them, 35.6% do not combine, and 7.6% stated not to know (n = 894). Regarding the range of adoption, 3.3% of the respondents use agile methods in none of the teams, probably teams that are starting agile methods usage; 42.0% use in less than a half of the teams; 33.3% use in more than a half, and 21.4% of the respondents use agile methods in all teams (n = 886).

What Is the Perception of Success in Agile Projects? We asked about their general perception of success in projects that use ASD. They could answer yes, no, sometimes, or that they did not know. As a result, we got 41.9% of respondents saying that yes, projects are successful; 6.7% said they are not successful. The majority – 48.1% – said that they are successful sometimes, and 3.3% stated that they do not know about projects' success.

What Is the Extent to Which Agile Methods Principles Are Applied? Based on results presented by [33], we investigate agile principles together with lean principles. Respondents could point out the intensity of application for the principles that applied to them (this is why n differs for each principle). Table 2 shows that the most frequently applied principles are working together with business people (62.4%), valuing continuous improvement (60.8%), and valuing working software more than comprehensive documentation (58.1%). The least applied principles are limiting work in progress (22.1%), inspecting team members' work (31.9%), and measuring progress with working software (38.4%).

When contrasting the agile principles for the success perceptions, we could see that, depending on project success perception, the intensity of agile principles' application differs. We clustered respondents into three groups: one that reported successful projects, one that reported sometimes-successful projects, and those that reported unsuccessful projects. We then calculated the mean percentage of principles application. Successful projects apply principles more frequently: when practitioners reported that their projects were successful, 58.8% reported

Table 2. Intensity of the application of agile principles in practitioners' companies (percentage of respondents). Cronbach's alpha = 0,91

Agile and lean principles	n	Frequently	Rarely	Never	Do not know
Working together with business people	840	62.4	34.4	2.3	1.0
Valuing continuous improvement	859	60.8	35.2	2.4	11.6
Valuing working software more than comprehensive documentation	856	58.1	35.5	3.9	2.6
Transparency in work and communication with team members	840	57.3	39.0	2.3	1.4
Attention to technical excellence	855	53.3	40.4	3.6	2.7
Prioritizing face-to-face communication between stakeholders	857	53.0	41.1	3.7	2.2
Solving problems with simple solutions	840	52.5	42.7	2.5	2.3
Valuing individuals and interactions over processes and tools	842	51.0	42.4	5.0	1.7
Responding to changes more than following a plan	837	50.3	43.8	3.3	2.5
Easily adapting to changes	860	49.4	47.8	1.5	1.3
Valuing customer collaboration over contract negotiation	854	45.8	45.0	4.6	4.7
Getting to know product value based on customer perception	860	44.8	50.0	2.6	2.7
Building projects around motivated individuals	855	43.6	51.8	2.5	2.1
Valuing self-organizing teams	855	42.5	49.4	6.2	2.0
Avoid work that does not add value to the customer	849	39.2	51.8	5.8	3.2
Measuring progress with working software	842	38.4	46.7	10.2	4.8
Inspecting team members work	850	31.9	52.1	11.9	4.1
Limiting Work in Progress (WIP)	845	22.1	54.0	14.3	9.6

applying principles frequently, 29.9% reported to rarely apply, and 20% reported to never apply them. Conversely, when projects are not successful, 3.2% reported that they frequently apply principles, 9.8% that rarely apply, and 18.7% that never apply them. Figure 1 shows the mean percentage for each intensity of applying principles for each project success perception group.

Which Are the Reasons for Adopting Agile Methods? Table 3 shows the reasons for agile adoption. The main reported reasons are accelerating software delivery (70.4%), increasing productivity (62.5%), and enhancing the ability to manage

Table 3. Reasons for agile development adoption. Cronbach's alpha = 0,67

Reason	n	Percentage
Accelerate software delivery	620	70.4
Increase productivity	551	62.5
Enhance the ability to manage changing priorities	368	41.8
Enhance software quality	339	38.5
Improve business/IT alignment	296	33.6
Enhance delivery predictability	286	32.5
Reduce project risk	261	29.7
Improve project visibility	223	25.3
Reduce project cost	162	18.4
Better manage distributed teams	160	18.2
Increase software maintainability	149	16.9
Improve team morale	147	16.7
Improve engineering discipline	133	15.1
Do not know	27	3.1
Other	40	4.5

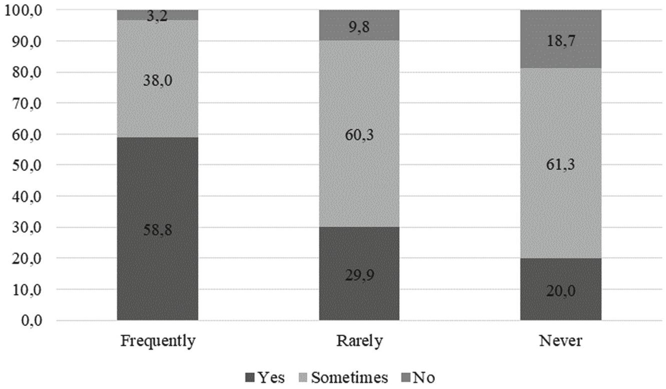


Fig. 1. Mean percentage of principles' application by projects' success perception

changing priorities (41.8%). The less reported reasons are improving engineering discipline (15.1%), improving team morale (16.7%), and increasing software maintainability (16.9%). Respondents could chose multiple reasons.

Which Are the Practices Applied? When asked about the respondents' practices in their daily routine, the most used practices are daily standup meetings (78.4%), kanban boards (76.7%), and retrospectives (67.4%). Among the least-used practices are emergent design (7.1%), agile portfolio planning (14.1%), and behavior-driven development (16.4%).

Table 4. Challenges faced when using agile methods. Cronbach's alpha = 0.54

Challenges	n	Percentage
Cultural change	546	62.8
Resistance to change	466	53.6
Agile practices customizing	425	48.9
Top management commitment	397	45.7
Customer collaboration	363	41.8
Defining business value	277	31.9
Measuring agile success	269	31.0
Troubles with self-management	264	30.4
Translating agile principles from development to business	258	29.7
Fixed-price contracts	229	26.4
Agile methods scaling	153	17.6
Inadequate documentation	141	16.2
Inadequate training	130	15.0
Steep learning curve	121	13.9
Lack of formal guidance	115	13.2
Decreasing predictability	110	12.7
Activities synchronization	100	11.5
Loss of management control	92	10.6
Inadequacy of existing technologies and tools	68	7.8
Need for special skills	51	5.9
Other	23	2.6

Which Are the Challenges Faced? Table 4 shows the challenges that the practitioners perceive in the use of agile methods. The most cited challenges are cultural change (62.8%), resistance to change (53.6%), and agile practices customizing (48.9%). The least-mentioned were the need for special skills (5.9%), the inadequacy of existing technologies and tools (7.8%), and loss of management control (10.6%). Respondents could also select all options that apply.

Which Are the Impacts Felt with Agile Methods Adoption? We asked respondents to rate the perceived impact of listed aspects between Improved, No effect, Got worse, and Do not know. Table 5 shows that team collaboration (87.9%) was the aspect perceived as improved by most of the respondents, along with team communication (83.6%), and learning and creating knowledge (82.2%). The aspects least perceived as improved are project cost reduction (37.3%), engineering discipline (37.4%), and managing distributed teams (37.9%). Table 5 also shows that the aspect most mentioned as getting worse due to agile adoption is project predictability, indicated by 6.0% of the respondents.

Table 5. Percentage of respondents that report each impact level for different aspects of agile adoption. Cronbach's alpha = 0.92

Aspect	n	Improved	No effect	Got worse	Do not know
Team collaboration	800	87.9	6.4	1.5	4.3
Team communication	794	83.6	9.8	1.6	4.9
Learning and creating knowledge	836	82.2	10.0	0.6	7.2
Ability to adapt to changes	813	78.4	14.1	1.0	6.5
Business/IT alignment	825	77.2	11.3	1.2	10.3
Productivity	792	74.2	15.7	1.8	8.3
Ability to manage changing priorities	789	72.9	16.1	2.9	8.1
Time to market	789	71.7	17.9	1.3	9.1
Self-management skills	825	71.5	17.9	1.8	8.7
Stakeholders satisfaction	786	71.1	14.8	1.1	13.0
Customer collaboration	794	69.6	19.8	1.8	8.8
Value creation	785	69.3	18.3	0.9	11.5
Project visibility	771	69.3	17.8	2.5	10.5
Team morale	762	66.8	21.0	2.0	10.2
Software quality	785	61.0	25.0	3.1	11.0
Customer comprehension	792	59.0	28.4	1.0	11.6
Project predictability	762	56.3	6.5	6.0	11.2
Software maintainability	759	51.8	31.2	2.5	14.5
Project risks reduction	786	50.8	27.4	2.8	19.1
Waste and excessive activities	784	48.2	33.5	4.6	13.6
Managing distributed teams	760	37.9	34.3	2.8	25.0
Engineering discipline	751	37.4	35.8	4.0	22.8
Project cost reduction	782	37.3	32.5	2.9	27.2

Which Principles Affect the Perception of Improvement in the Software Process? We applied two rounds of machine learning algorithms to verify whether principles adoption could predict improvements. In the first round, the Artificial Neural Network (ANN), we were interested in models that are good at predicting improvements. We identified the impacts that presented best measurements in precision values. The impacts which have best precision (>97%) for the class "Improved" are: learning and creating knowledge, business/IT alignment, team collaboration, team communication, self-management skills, time to market, ability to adapt to changes, and ability to manage changing priorities. It means that different combinations of applying principles might define improvements in these aspects. In our second round of analysis, we ran Support Vector Machine (SVM)

Table 6. Improvements predicted by the application of agile principles

Principle	Improvements predicted
Attention to technical excellence	Time to market and Ability to adapt to changes
Easily adapting to changes	Business/IT alignment, Self-mngt skills, Time to market and Ability to adapt to changes
Prioritizing face-to-face communication between stakeholders	Learn and create knowledge, Business/IT alignment, and Team communication
Reducing work that does not add value to the customer	Time to market
Responding to changes more than following a plan	Learn and create knowledge, Business/IT alignment, Self-mgmt skills, Time to market, Ability to adapt to changes and Ability to manage changing priorities
Solving problems with simple solutions	Learn and create knowledge, Team collaboration, Time to market, Ability to adapt to changes and Ability to manage changing priorities
Transparency in work and communication with team members	Learn and create knowledge, Business/IT alignment, Team collaboration, Team communication, Time to market, Ability to adapt to changes, and Ability to manage changing priorities
Valuing continuous improvement	Team collaboration, Self-mgmt skills, Time to market, Ability to adapt to changes, and Ability to manage changing priorities
Valuing customer collaboration over contract negotiation	Self-mgmt skills, Time to market, Ability to adapt to changes, and Ability to manage changing priorities
Valuing individuals and interactions over processes and tools	Learn and create knowledge, Business/IT alignment, Self-mgmt skills, Time to market, and Ability to adapt to changes
Valuing self-organizing teams	Self-mgmt skills, Time to market, and Ability to adapt to changes
Valuing working software more than comprehensive documentation	Learn and create knowledge, Team collaboration, Self-mgmt skills, Time to market, Ability to adapt to changes, and Ability to manage changing priorities
Working together with business people	Learn and create knowledge, Business/IT alignment, Team collaboration, Self-mgmt skills, Time to market, Ability to adapt to changes, and Ability to manage changing priorities

and Random Forest (RF) algorithms to identify which specific principles positively affect the perception of improvement in these aspects. Using the confusion matrix results for the execution of SVM and RF techniques for each evaluated aspect – considering the 75% more critical attributes – we identified the True Positive resulting values, as they reflect the percentage of prediction in which the models correctly predicted improvement in the evaluated aspects.

The execution of these models resulted is a list of principles that are more relevant for the predictions. Table 6 shows the principles that contribute to improvements in agile software development. For instance, when the principle "Attention to technical excellence" was applied, machine learning models could predict improvements in "Time to market" and "Ability to adapt to changes". The same interpretation applies to the other principles in Table 6.

5 Discussion

The goal of this study was to describe current agile usage in Brazil. We conducted a survey in 6 editions of 4 industry-based agile conferences in 2018–2019, resulting in 897 responses. Descriptive statistics and machine learning models were used to analyze data. We learned that most Brazilian practitioners that participated in our research work in teams with up to 20 people and that most of these teams are not geographically distributed. Most of the respondents were from the software, financial services, and government industries. The majority have been using agile between 3 to 5 years, although there is also a significant percentage of companies that are young in agile usage (1 to 2 years of adoption).

Scrum and Kanban are the most used methods, albeit we could see that more than half of practitioners state to mix agile methods with traditional ones. This combination of traditional with agile methods seems to be an established trend, as [24] also observed. In their research, a purely agile or traditional application was seldom evident. In our results, we also saw that about 20% of companies use agile methods in all teams, and 41.9% say that agile projects are indeed successful.

Practitioners showed us that the most frequently applied agile principles are working together with business people, valuing continuous improvement, and valuing working software more than comprehensive documentation. When relating the application of agile methods principles to the perception of project success, we could show that, when respondents pointed out that agile projects were mostly successful, the intensity of the application of ASD principles was frequent for most of them. The main reasons for adopting ASD are accelerating software delivery, increasing productivity, and enhancing the ability to manage changing priorities.

Regarding agile practices, we see that the most applied are daily standup meetings, kanban boards, and retrospectives. Practices are a important part of application of agile methods, as they have been related to an increase in the degree of agility [24]. Moreover, the study by [22] identified a relation of practices with team satisfaction, as they enable team cohesion and support tracking of the progress.

Table 7. Comparison of our results with Melo et al. (2013)'s.

Length of time using Agile			
	Melo et al. (2013)	Our results	p-value
<1 year	29.1	14.7	<0,001
1–2 years	28.5	28.1	
3–5 years	29.5	34.3	
>5 years	7.0	22.9	
Never	5.9	0*	
Reasons for adopting Agile			
	Melo et al. (2013)	Our results	p-value
Reduce risk	69	29.7	<0,001
Reduce cost	47	18.4	<0,001
Increase productivity	91	62.5	<0,001
Improve team morale	64	16.7	<0,001
Improve project visibility	65	25.3	<0,001
Improve engineering discipline	59	15.1	<0,001
Improve alignment between IT and business	72	33.6	<0,001
Enhance software quality	83	38.5	<0,001
Enhance software maintainability extensibility	66	16.9	<0,001
Enhance ability to manage changing priorities	86	41.8	<0,001
Accelerate time to market	73	70.4	0.359
Benefits of Agile adoption			
	Melo et al. (2013)	Our results	p-value
Time to market	55.42	71.7	<0,001
Team morale	66.87	66.8	0.937
Software maintainability	50.11	51.8	0.553
Risk reduction	51.59	50.8	0.850
Quality	60.29	61	0.743
Project visibility	62.85	69.3	0.018
Productivity	69.21	74.2	0.044
Manage distributed teams	24.84	37.9	<0,001
Manage changing priorities	67.94	72.9	0.047
Engineering discipline	45.86	37.4	0.003
Cost reduction	38.22	37.3	0.756
Alignment IT - Business	55.63	77.2	<0,001

*This option was not available in our study.

The main challenges teams face are related to personal issues, such as cultural change and resistance to change (also presented as hindrances in [22]), and process issues as practices customizing. Moreover, our data shows that improvements could be perceived mainly on team collaboration, team communication, and learning and creating knowledge. We also uncovered that improvements in the areas of learning and creating knowledge, business/IT alignment, team collaboration, team communication, self-management skills, time to market, ability to adapt to changes, and ability to manage changing priorities could be predicted by the application of certain principles (uncovered by machine learning models).

Part of our results can be directly compared with other studies. We did so with the Brazilian study by Melo et al. (2013) [26] and with the international commercial survey by Version One (2019) [30]. Chi-square tests were used to identify differences in frequency distributions, in which p values lower than 0.05 mean statistically significant difference. Not all items that we asked in our study were available to compare to the others. When contrasting our study results to Version One's (2019)'s [30], we could apply comparisons to the length of time using agile, reasons for adopting agile, benefits and agile methods. We see that companies in Brazil seem to be younger on the use of agile methods. Regarding the reasons for adopting ASD, a similar number of Brazilian practitioners state reasons for accelerating software delivery, reducing project risk, and better managing distributed teams. We see a similar perception of benefit in team morale, project risk reduction, and better managed distributed teams. Concerning the adopted agile methods, we see that our results differ from Version One's (2019) [30]; that is, the percentage of practitioners who adopt each method is different in Brazil, mainly expressed by significantly larger Kanban adoption in Brazil.

By comparing our results to Melo et al. (2013)'s [26], it is possible to identify the evolution of Brazilian community (see Table 7). Regarding the time using agile, it is interesting to notice how, in our study, seven years later, the aging of the teams appeared. We have more teams that use agile for more than five years, although we still have young companies with regards agile adoption. Reasons for adopting agile has also evolved over the years. The only reason that remains with the same distribution is accelerating time to market.

Last but not least, we also compared the perceived benefits upon agile adoption. The perception of benefit remains similar to the Melo et al.'s study for team morale, software maintainability, risk reduction, quality, and cost reduction. Our dataset shows that more people perceives benefits on time to market, project visibility, productivity, manage distributed teams, manage changing priorities, and alignment between IT and business.

6 Conclusions

This study aimed to report how ASD has been applied in Brazil. Based on responses from 897 practitioners, we showed the profile of companies and teams, characteristics of agile usage, perception of success, principles and practices applied, reasons, challenges, and impacts of ASD adoption. We also explored the relevance of principles in practitioners' improvements.

Although results are limited to a non-probabilistic sample, the information we presented here might help practitioners understand the state-of-the-practice of ASD adoption in the country and compare their own practices and maturity in contrast to a previous portrait. Although there are no ground-breaking insights, our results should motivate people to improve and seek for better alternatives to software development in their own ecosystem. Results should also shed some light to researchers with themes that might be of attention for further investigation.

Acknowledgments. This project was supported by the Brazilian National Council for the Technological and Scientific Development (CNPq) through the research grant no. 408976/2016-0. We thank the conference chairs and organization members that allowed us to collect our data during the events. Special thanks to the 2018 and 2019 Agile Brazil committee, Dairton Bassi, and Rodrigo Cursino. Sabrina Marczak would like to thank CNPq (grant no. 307177/2018-1).

References

1. Ashwood, A.J.: Portfolio selection using artificial intelligence. Ph.D. thesis, Queensland University of Technology, Queensland, Australia (2013)
2. Azizyan, G., Magarian, M.K., Kajko-Mattson, M.: Survey of Agile tool usage and needs. In: Proceedings of the Agile Conference (2011)
3. Basili, V.: Software modeling and measurement: the goal/question/metric paradigm. Technical report, University of Maryland at College Park, USA (1992)
4. Beck, K.: Agile manifesto, April 2001. http://agilemanifesto.org
5. Bolatti, V.A., Gaona, G., Pletsch, L.C., Gonnet, S., Leone, H.: The state of agile development adoption in argentine software companies. In: Proceedings of the Latin American Computer Conference (2017)
6. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
7. Breiman, L.: Statistical modeling: the two cultures. Stat. Sci. **16**, 199–231 (2001). https://doi.org/10.1214/ss/1009213726
8. Bryman, A.: Social Research Methods. Oxford, New York (2012)
9. Bustard, D., Wilkie, G., Greer, D.: The maturation of agile software development principles and practice: observations on successive industrial studies in 2010 and 2012. In: Proceedings of the International Conference and Workshops on the Engineering of Computer-Based Systems, pp. 139–146 (2013)
10. Comrie, A.C.: Comparing neural networks and regression models for ozone forecasting. J. Air Waste Manag. Assoc. **47**, 653–663 (1997). https://doi.org/10.1080/10473289.1997.10463925
11. Conboy, K., Coyle, S., Wang, X., Pikkarainen, M.: People over process: key challenges in agile development. IEEE Softw. **28**(4), 48–57 (2011)
12. Diel, E., Bergmann, M., Marczak, S., Luciano, E.: What is agile. Which practices are used and which skills are necessary according to Brazilian professionals: findings of an initial survey. In: Proceedings of the Brazilian Workshop on Agile Methods (2016)
13. Dikert, K., Paasivara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: a systematic literature review. J. Syst. Softw. **119**, 87–108 (2016)

14. Fontana, R.M., Fontana, I.M., Garbuio, P.A.R., Reinehr, S., Malucelli, A.: Processes versus people: how should agile software development maturity be defined? J. Syst. Softw. **97**, 140–155 (2014)
15. Fontana, R.M., Marczak, S.: Characteristics and challenges of agile software development adoption in Brazilian government. J. Technol. Manag. Innov. **15**, 3–10 (2020)
16. Haykin, S.: Neural Networks and Learning Machines. Pearson, New Jersey (2009)
17. Issitt, R.W., Cortina-Borja, M., Bryant, W., Bowyer, S., Taylor, A.M., Sebire, N.: Classification performance of neural networks versus logistic regression models: Evidence from healthcare practice. Cureus **14** (2022). https://doi.org/10.7759/cureus.22443
18. Kitchenham, A.B., Pfleeger, S.L.: Principles of survey research - part 3: constructing a survey instrument. ACM SIGSOFT Softw. Eng. Notes **27**(2), 20–24 (2002)
19. Kitchenham, A.B., Pfleeger, S.L.: Principles of survey research - part 4: questionnaire evaluation. ACM SIGSOFT Softw. Eng. Notes **27**(3), 20–23 (2002)
20. Kitchenham, A.B., Pfleeger, S.L.: Principles of survey research - part 5: populations and samples. ACM SIGSOFT Softw. Eng. Notes **27**(5), 17–20 (2002)
21. Kitchenham, A.B., Pfleeger, S.L.: Principles of survey research - part 6: data analysis. ACM SIGSOFT Softw. Eng. Notes **28**(2), 24–27 (2003)
22. Kropp, M., Meier, A., Anslow, C., Biddle, R.: Satisfaction and its correlates in agile software development. J. Syst. Softw. **164** (2020). https://doi.org/10.1016/j.jss.2020.110544
23. Kuhrmann, M., et al.: Hybrid software development approaches in practice: A European perspective. IEEE Softw. **36**(4), 20–31 (2019)
24. Kuhrmann, M., et al.: What makes agile software development agile? IEEE Trans. Softw. Eng. (2021). https://doi.org/10.1109/TSE.2021.3099532
25. Livermore, J.: What elements of XP are being adopted by industry practitioners? In: Proceedings of the IEEE SoutheastCon (2006)
26. Melo, C., et al.: The evolution of agile software development in brazil. J. Braz. Comput. Soc. **19**, 523–552 (2013)
27. Molléri, J.S., Peterson, K., Mendes, E.: An empirically evaluated checklist for surveys in software engineering. Inf. Softw. Technol. **119** (2020)
28. Nazir, N., Hasteer, N., Bansal, A.: A survey on agile practices in the Indian it industry. In: Proceedings of the International Conference Cloud System and Big Data Engineering (2016)
29. VersionOne: 11th annual state of agile survey (2017). https://stateofagile.com/#ufh-i-613554036-11th-annual-state-of-agile-report/7027494
30. VersionOne: 13th annual state of agile survey (2019). https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508
31. Pfleeger, S.L., Kitchenham, B.: Principles of survey research - part 1: turning lemons into lemonade. ACM SIGSOFT Softw. Eng. Notes **26**(6), 16–18 (2001)
32. Raunak, M., Binkley, D.: Agile and other trends in software engineering. In: Proceedings of the International Conference on Cloud System and Big Data Engineering (2016)
33. Rodriguez, P., Markkula, J., Oivo, M., Turula, K.: Survey on agile and lean usage in finnish software industry. In: Proceedings of the International Symposium on Empirical Software Engineering and Measurement (2012)
34. Salo, O., Abrahamsson, P.: Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum. IET Softw. **2**(1) (2008)

35. Stavru, S.: A critical examination of recent industrial surveys on agile method usage. J. Syst. Softw. **94**, 87–97 (2014)
36. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995). https://doi.org/10.1007/978-1-4757-2440-0

Investigating the Current State of Security in Large-Scale Agile Development

Sascha Nägele(✉), Jan-Philipp Watzelt, and Florian Matthes

Technical University of Munich, Munich, Germany
{sascha.naegele,jan-philipp.watzelt,matthes}@tum.de

Abstract. Agile methods have become the established way to successfully handle changing requirements and time-to-market pressure, even in large-scale environments. Simultaneously, security has become an increasingly important concern due to more frequent and impactful incidents, stricter regulations with growing fines, and reputational damages. Despite its importance, research on how to address security in large-scale agile development is scarce. Therefore, this paper provides an empirical investigation on tackling software product security in large-scale agile environments. Based on a literature review and preliminary interviews, we identified four essential categories that impact how to handle security: (i) the structure of the agile program, (ii) security governance, (iii) adaptions of security activities to agile processes, and (iv) tool-support and automation. We conducted semi-structured interviews with nine experts from nine companies in five industries based on these categories. We performed a content-structuring qualitative analysis to reveal recurring patterns of best practices and challenges in those categories and identify differences between organizations. Among the key findings is that the analyzed organizations introduce cross-team security-focused roles collaborating with agile teams and use automation where possible. Moreover, security governance is still driven top-down, which conflicts with team autonomy in agile settings.

Keywords: Large-scale agile · Security · Software development

1 Introduction

The use of agile methods is omnipresent. According to the most recent "State of Agile Report", agile adoption within software development teams has surged from 37% in 2020 to 86% in 2021 [11]. Agile development methods are also increasingly applied to large projects and companies with numerous software development teams working together [12]. Companies thereby aim to benefit from the advantages of these methods, such as enhanced adaptability to fast-evolving environments and accelerated time-to-market [37].

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 203–219, 2022.
https://doi.org/10.1007/978-3-031-08169-9_13

At the same time, software security is becoming an increasingly important concern due to stricter legislation and growing fines [9]. In addition, there is a growing intrinsic motivation for companies to pay more attention to security. As a global risk management survey with thousands of participating companies shows, cyberattacks, data breaches, and reputational damage are the most significant perceived risks to business success [4]. The global Covid-19 pandemic further exacerbates the complexity and growing number of cyberattacks as changing work conditions and consumer behavior further increase the dependence on Information Technology (IT) [14]. Despite the importance of software security in scaled agile environments, there are only few empirical studies, and more empirical research is needed [1,31,48].

This study contributes to the empirical evidence on how organizations tackle software security in large-scale agile development (LSAD). The primary research question we strive to answer is: **How is security approached in LSAD, and what are recurring best practices and challenges?** We provide a cross-industry overview based on literature and interviews with nine experts from nine companies in five industries. The remainder of this paper is organized as follows: Section 2 presents the theoretical background and related work. Section 3 explains the research methodology. Section 4 summarizes the results, which are discussed in Sect. 5. Section 6 presents the conclusion and outlook.

2 Background and Related Work

We follow Dikert et al. in defining LSAD, who speak of a minimum of 50 people or at least six teams [12].

One of the earlier related works is by Bartsch [45], who studied security in agile development by interviewing ten practitioners but does not explicitly address LSAD contexts yet. Relevant for our work is the more recent study by Amber et al. who identified three unique security challenges in LSAD: "(i) alignment of security objectives in a distributed setting; (ii) developing a common understanding of roles and responsibilities in security activities; and (iii) integration of low-overhead security testing tools" [48]. Our key findings discuss how our results relate to these challenges.

In addition, valuable related work includes widespread software security maturity frameworks, e.g., the Building Security in Maturity Model (BSIMM) [28] and the OWASP Software Assurance Maturity Model (SAMM) [35]. These are mainly driven by practical experience from the industry and provide a highly comprehensive insight into secure software development initiatives. Even if they do not explicitly address LSAD and describe themselves as agnostic of the development approach, many of the listed organizations working with these models fulfill the definition of LSAD. However, we base our study on a literature review to achieve unbiased research independent of these models.

In the following subsections, we present the theoretical background and related work using four categories that emerged from our literature review and can be mapped to Amber et al.'s [48] challenges. We also use these categories to structure our interviews and results.

Structure of the Agile Program. Poller et al. [38] emphasize that considering organizational structures, e.g., roles and their interaction, is vital for promoting security approaches and governing agile teams in LSAD. Alsaqaf et al. [1] found in a systematic literature review that additional roles are introduced in LSAD to address quality requirements, e.g., a security architect. The authors emphasize that further empirical research on such roles is needed. Newton et al. [33] discovered security-related communities of practices, while Rindell et al. [39] observed an internal software security group that, e.g., carries out security reviews. Steghöfer et al. and Dännart et al. note that LSAD frameworks do not provide security compliance out-of-the-box [10,46]. Moyon et al. [30] recommend further adaptions, e.g., by introducing security roles. Oyetoyan et al. [36] describe a group of security experts supporting with, e.g., adherence to security standards and organizing security audits. The proposal of Boström et al. [8] includes a team of security engineers, e.g., to support the definition of security stories and risk assessments together with product teams.

Also, publications from software companies such as SAP [42], Microsoft [27] and Google [15] show that dedicated security roles are being used in practice, although the exact range of tasks is not always explained in detail. We thereby derive that the *structure of the agile program* is vital for addressing security in LSAD.

Security Governance. Security governance can be seen as a subset of IT governance, often characterized by top-down control [17]. Despite limited empirical studies on IT governance in agile and lean environments, its importance has been recognized [47]. The literature recommends moving to agile and lean governance approaches to better align governance and agility. The term lean governance is more frequently used in industry publications such as white papers and large-scale agile frameworks [3,43]. Horlach et al. [16] found that traditional governance structures hinder autonomous agile teams in LSAD. Ambler [2] stated early on that a lean form of IT governance is required to achieve agility in software development at scale. Vejseli et al. [49] found that agile IT governance positively affects business-IT alignment and, thus, enterprise performance, similar to traditional governance. By fostering the necessary engagement of all parts of the business, agile governance helps increase business agility [23]. Agile governance focuses on enabling and motivating development teams through collaborative and supportive practices [2]. Instead of top-down control, it promotes bottom-up engagement, autonomy, and self-organization [3,24]. Because of this tension, we derive *security governance* as an essential category.

Security Activities. We understand security activities as a set of practices that directly or indirectly enhance software security. A typical example is threat modeling. It is a component of security risk analysis [25] and supports the identification of security risks and appropriate measures [44]. Other common examples are penetration testing [5] and code reviews [41]. Multiple researchers agree that incorporating security activities in agile development is feasible and necessary [31,33]. Beznosov and Kruchten [7] propose integration strategies depending on the match between security practices and agile principles. As stated by Keramati and Mirian-Hosseinabadi, security activities are integrated with agile software development based on balancing “the costs of decreased level of agility [...] and benefits from developing more secure systems” [19]. Hence, we derive the category of *security activities* for our interviews.

Tool-Support and Automation. In their case study, Barbosa and Sampaio note that the “demand to build software quickly and cost-effectively” impedes the integration of agile security approaches due to the associated cost and time effort [6]. Therefore, automating manual, work-intensive tasks is crucial to reduce the friction between security and iterative deployment practices. In recent years, the term DevSecOps matured from a buzzword to a well-established movement [32]. One of the primary goals is integrating security activities and practices into development pipelines facilitated by security automation tools [29]. Researchers emphasize automating repetitive manual tasks, like security code reviews, to ensure security while sustaining a high velocity in agile software development [18,34]. Examples of security automation include static and dynamic application security testing. Since reducing manual effort and a more frictionless integration of security activities is critical in scaled environments, we derive *tool-support and automation* as the fourth category for our interviews.

3 Research Methodology

We present the three stages of our methodological process below: study design, data collection, and analysis.

Study Design. To gain cross-case insights into our research question, we deemed an interview study the most suitable primary research method. We excluded a multiple case study because not enough cases provided multiple sources for data collection due to the topic’s sensitive nature. To allow for a better aggregation and comparability of results, we roughly structured the interview with the categorization described in the background and related work. Before conducting the actual interview study, we performed four preliminary expert interviews in two organizations to discuss and evaluate the categorization. In each category, we used semi-structured questions, which allow for enough freedom in the answers and the possibility for individual adjustments during the interview [13].

In contrast to expert-focused surveys, we also considered the experts’ current organizations, i.e., we did not select the experts solely based on their role, competency, and experience, but an important factor was the organization they currently work for. The organizations must fulfill the previously described definition of LSAD.

Data Collection. For the interview study data collection, experts from nine companies participated in our study.

We collected data across five industries to ensure better generalizability of results. The following sectors are represented based on the main product focus of the case company: IT and software development, software development consulting, media, insurance, and automotive. Two researchers interviewed six of the nine interviewees. Three were interviewed by one researcher. After obtaining explicit consent to record the interviews for transcription purposes, we used online video conferencing tools and recorded all interviews. On average, the respondents had about six years of experience with LSAD, with a minimum of three years and a maximum of fifteen years. The experts' roles included security leads of agile programs, security engineers and security champions, an IT (security) consultant, an IT (security) architect, and a product owner (PO). To protect the anonymity of our interviewees, we intentionally do not provide further details.

Data Analysis. There are several standardized methods for the analysis of qualitative material. We used the Kuckartz [20] model to analyze our interview study data because it offers a deductive-inductive possibility for coding classification formation. We conducted the content-structuring qualitative content analysis using the qualitative data analysis software *MAXQDA* [26]. The two researchers who performed the interviews also conducted the analysis.

4 Interview Results

In this section, we present the main findings from the data analysis of the expert interviews. We first overview our results, then summarize framework usage and challenges, followed by the findings in the four categories of our interviews. To ensure the anonymity of the participating organizations, we intentionally describe the results only in an aggregated format and not specific for each case, except for Table 1.

4.1 Overview

Table 1 contains a summary of the results. We identified and selected recurring best practices that emerged from the interview analysis. We classify and visualize them according to their usage in each organization through *harvey balls*. The table does not represent a complete summary, but we filtered our results for two main cases. First, the concepts with the highest recurrence, and second, concepts with the highest ratio of conflicting viewpoints among the experts. We thus prioritize displaying the most important findings based on these two criteria.

4.2 Frameworks and Challenges

Scaled Agile Framework Usage. In the beginning, we asked about the scaled agile frameworks used in the organizations. Two experts stated that their organizations adhere to the guidelines of a specific framework, in one case LeSS [22], in the other case SAFe [43]. A third and fourth expert described a more heterogeneous agile landscape where teams choose frameworks individually depending

Table 1. Overview of recurring best practices

	01	02	03	04	05	06	07	08	09
Integration of security activities									
Security self-assesment	●	●	◑	○	◑	●	●	●	◕
Bug bounty		◑		◑	●	○	●	○	●
Threat modeling	◑	●	●	●	◕	◑	◑	●	◕
Penetration testing	●	●	●	●	●	●	●	●	●
Security audits		●	●	●		●	●	◑	●
Security code review	◑	◑	●	○	●	●	○	●	◑
Tool-support and automation									
DevSecOps pipeline	◕	●	●	◕	●	◑	◑	●	●
Static code analysis	◑	●	●	●	◑	○	◑	●	●
Vulnerability scanning	◑		●	◕	●	●	◕	●	◑
Dependency checks	●	◑	●	◕		○	◑	○	◑
Security governance									
Bottom-up	◑	○	○	○	◑	○	◑	◔	◔
Top-down	●	●	●	●	●	●	●	●	●
Reusable components		●	●		◑	○	◕	◕	◑
Organizational structure									
Security champion	◑	●	○	○	●	○	●	◕	◑
Security engineers or architects	◕	●	◕	●	◑	●	●	●	●
Central security teams	●	●	◕	◕	●	◑	●	○	◕
Communities of practice	◕	●		◑	◕	●	●	●	●

none: ○ | rare or planned: ◔ | partial: ◑ | frequent: ◕ | complete: ●
no classification possible: *empty*.

on the requirements. Two experts stated that no "textbook framework" is being used for scaled agility. The remaining three experts indicated that their organizations built their own frameworks, including parts of established frameworks.

Security Challenges in LSAD. Initially, we also asked the participants about the main challenges related to security in their LSAD environment. However, we will only present challenges mentioned by at least three independent experts. The first challenge is the lack of personnel with sufficient experience in both security (governance) and agile software development. The scaled agile environment amplifies the problem because centralized security teams have frequent contact with agile teams due to short development cycles. Also, the expected response times of security experts to inquiries of agile teams are lower, resulting in a higher pressure on central security experts and possible frictions and delays in the development process.

The second challenge is the conflict between security governance and team autonomy when coordinating many teams. Teams should work as autonomously as possible, yet security policies and standards must be defined and managed. Scaling makes it challenging to monitor and control, as it is no longer possible to "look over the shoulders of the developers", as one expert stated.

4.3 Organizational Structure

All interviewed experts report that their organization is performing some sort of structural adaptations of their agile programs due to a higher relevance of security. Figure 1 shows a generalized summary of the results.

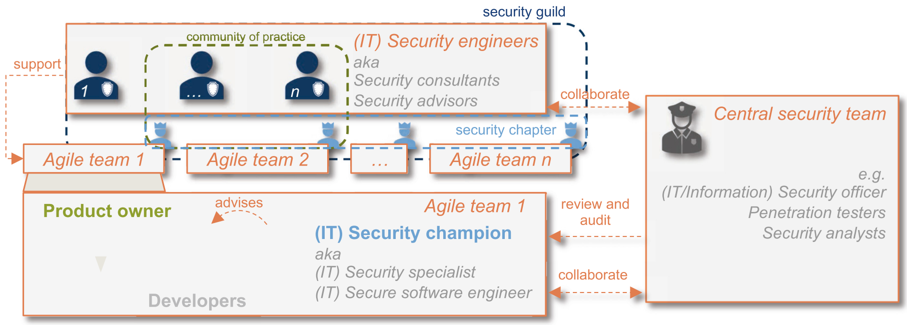


Fig. 1. Overview of organizational structure of agile programs

Centralized Security Teams. A common theme between the experts, with one exception, is that their organizations leverage existing central security teams to work with agile programs. These teams include individuals dedicated to security, e.g., penetration testers, security analysts, or information security officers. Centralized teams set overarching security quality criteria for deployments of software product increments and perform security verification. They also identify and handle compliance issues, perform risk analyses and security reviews (e.g., code review or penetration tests). Some activities such as threat modeling are performed collaboratively with individual development teams. This collaboration is beneficial for training purposes. The achieved knowledge transfer might enable agile teams to perform these activities by themselves in the future, reducing the workload of central teams. Depending on the criticality and security requirements of the software artifact, some of the analyzed organizations use central security teams for auditing and approving release-ready changes before deployments to production environments. Both threat modeling and reviews are discussed in more detail in Sect. 4.6. Members of central teams are often focused on a product area or specialized in a specific security topic or technology. As mentioned in the challenges in Sect. 4.2, central teams face scaling issues and become a bottleneck when collaborating with agile development teams.

This bottleneck motivates the introduction of new roles within the agile programs. The goal is to reduce the workload on central teams and, more importantly, increase the security capabilities and thereby the autonomy of agile teams. Based on the collected data, we distinguish between two types of security-focused roles, *team-internal* and *team-external*.

Team-Internal Roles. These agile team members continue to be developers but receive additional security training. The analyzed cases use designations such as security champion, security specialist or secure software engineer, hereafter referred to only as security champion (SC). They provide the benefit of increasing security awareness. As developers, they know their products and are also familiar with security standards and best practices. One interviewee stressed that it is essential to clarify that the whole team is still responsible for the security of their application. The SC takes the lead on security activities, serves as a fixed contact person to communicate with team-external parties, and advises other team members and the PO. Three cases do not use an SC and rely more on other measures such as automated security testing.

Team-External Roles. They are referred to as security engineers, security consultants or security advisors, hereafter referred to only as security engineer (SE). They support two to twenty teams with security expertise and are often placed between the development teams and a central security department, acting as facilitators. In some organizations, SEs conduct threat modeling workshops with development teams. In other cases, this is the responsibility of the SC, to prevent bottlenecks. SEs may also analyze laws, policies, and security best practices and ensure knowledge transfer to development teams. They specialize in a software stack or are assigned to specific development teams. Two of the analyzed cases currently have no plans to introduce a specialized security role. A solution architect is responsible instead.

Cross-Team Collaboration. Security knowledge sharing takes place through regular meetings and training. Some organizations use the concept of communities of practices. Others unite the previously described roles in so-called guilds or chapters. A difference is in the scope, frequency, and target audience for which these exchanges occur. Moreover, organizations use corporate social networks and wikis to share and document security knowledge and search for experts. However, knowledge sharing remains a challenge. Existing documentation is not always helpful due to its complexity or lack of specific details for certain combinations of platforms and software. According to one expert, providing code examples for security topics is most helpful for developers.

4.4 Security Governance

All analyzed companies mainly rely on a top-down governance approach. In most cases, centralized security governance teams create company-wide standards from applicable regulations, international standards, and best practices. The companies differ in how development teams can participate in shaping security governance. One interviewee explicitly stresses that individual teams should not influence security governance because they should prioritize the development of their product. Others grant development teams a limited say in the governing standards, allowing a partly bottom-up approach. In those cases, agile teams support shaping internal standards adjustments with sufficient justification. A promising approach for effective security governance in LSAD is providing standardized, security-focused components that teams can reuse. Interviewees mentioned that these components also simplify application security verification. Stated examples are identity and access management, validation of inputs, encryption of data, or secure communication. Challenges include outdated documentation, uncertainties about correct usage, and lack of awareness.

4.5 Tool Support and Automation

All interviewees stated that their companies use DevSecOps pipelines for their applications' build and deployment phases.

Static Application Security Testing. A common denominator is the use of static code analysis tools, which are mandatory to varying degrees. In some companies, the usage depends on project requirements and the development team's decisions. In others, it is compulsory for all applications. Depending on the criticality of the findings, teams have to meet different thresholds to deploy changes to production. False positives are a commonly reported challenge of static security testing. They are especially problematic because they may lead to developers ignoring analysis results. A particular form of static analysis is using automated dependency checks, e.g., to look for the usage of outdated open-source libraries that could introduce new vulnerabilities into the product.

Dynamic Application Security Testing. The use of dynamic application security testing is not yet as mature as static code analysis. The experts stated that there are initiatives to evaluate and establish dynamic application security testing tools. They aim to automate parts of manual penetration tests. Furthermore, the experts mentioned the use of regular vulnerability scans, e.g., to check the infrastructure of the development teams for unnecessary open ports, insecure TLS versions or cipher suites, insecure HTTP header, or other security misconfigurations. Usually, central teams provide these scanning tools. Reports are immediately made available to development teams or at regular intervals, depending on the criticality.

Metrics and Quality Gates. Automation tools that are part of a DevSecOps pipeline provide metrics, e.g., for automated deployment decisions. Those metrics might include the number of open findings, the average criticality, or a total score. For these metrics, the experts stress the importance of agreeing on thresholds for quality gates. These thresholds set the boundary of whether an application is likely to be secure enough to release to production. Due to the limited capabilities of automated tools, experts stressed not to rely exclusively on automation. As an outlook, one interviewee noted that the increasing use of machine learning might soon blur the line between the areas of security testing that can be automated and those that cannot.

4.6 Integration of Security Activities

Performing concrete activities to directly or indirectly increase the degree of security of a software product is crucial. The focus of the interviews was especially on which activities are most suitable in LSAD environments, and discussing their benefits and drawbacks. The following activities were the most discussed ones by our interviewed experts.

Code Reviews and Pair Programming. Most companies use code reviews as a form of manual intervention in developing secure applications. In two cases, pair programming is used instead as the primary quality assurance activity. A reported challenge in multiple analyzed cases is that code reviews usually deal with code quality in general (except for dedicated security code reviews), and security aspects may frequently fall short. One expert explained that they focus on automated static code analysis due to the high time consumption of code reviews. Also, other experts mentioned that code reviews are a trade-off between cost and the prospect of higher code quality. Nevertheless, one expert calls code reviews "the most pragmatic approach to developing secure software". The extent and frequency of code reviews vary. Some companies decide based on the criticality and required level of protection of the software product, while others leave it to the development teams. Especially when deploying critical code to production, organizations tend to mandate code reviews. Experts mentioned that it would be helpful to conduct security code reviews only if there was a security-relevant change. However, the crux lies in identifying those relevant changes, but automation may help in the future.

Penetration Tests and Bug Bounty Programs. All case companies regularly perform penetration tests. Both internal teams, as well as contractors, are used for this purpose. The frequency and scope vary depending on the product's criticality and size. The primary reported challenge of penetration testing is the lack of continuity because of the necessary preparation and follow-up work. Short penetration tests that only assess the changes of a smaller product increment are usually not seen as economically viable. Bug bounty programs are a valuable alternative to detect vulnerabilities continuously and provide the advantage of scaling through crowd-sourced security testers.

Security Reviews and Audits. Companies use security reviews to assess compliance with internal and external regulations. Depending on the criticality of the application, the audit frequency varies from quarterly to yearly. Reviews might include assessing system architecture or security documentation, code reviews, or penetration tests. A distinction can be made between pre-deployment and post-deployment audits. A hybrid approach is also possible, e.g., regularly using post-deployment audits and applying pre-deployment controls every few sprints, or only if a product recently failed security audits. For low-risk applications, code can be deployed before all checks have been performed. When assessing the compliance of an application with given standards, respondents pointed to the commitment to guidelines. Some are merely recommendations, while others are considered indispensable.

Threat Modeling. Because of its good fit for iterative software development, threat modeling has a high priority for the interviewees. It can be performed during the initial design phase. For continuous integration into short sprints, delta threat modeling is performed. Delta threat modeling focuses on changes of the increment. The results of threat modeling can be used to prioritize specific components for code reviews or penetration testing.

Security Self-assessments. There are two main usages for security self-assessments. First, to determine whether the product in development is compliant with policies and guidelines. Second, to determine the security relevance and criticality. Self-assessments can be an efficient tool at scale because they delegate responsibility to the teams. One interviewee stressed that the goal is to keep the number of validations by team-external stakeholders as low as possible. A benefit of self-assessments is the creation of security awareness. The concept of "comply or explain" was also mentioned. Developers may explain where they have made a conscious decision not to meet a requirement. Depending on the criticality, this might be considered during risk management. One organization deliberately avoids self-assessments because they are too time-consuming.

Security Risk Management. A recurring aspect in the interviews is the possibility to release or keep operating software with certain security risks or compliance issues, often referred to as "risk acceptances". A PO has to take responsibility for the risk and systematically document it. A SC or SE usually supports the PO to identify and report risks proactively. Furthermore, risks can also result from other activities, e.g., threat modeling, penetration testing, or security reviews. Some teams perform and document risk assessments themselves, e.g., as attributes or flags of their feature tickets or user stories.

Security Documentation. On the on hand, experts stated that extensive security documentation is often not feasible for frequent product iterations. Therefore, companies evaluate tools to automatically create documentation, e.g., risk reports generated from threat models. On the other hand, experts explained that incrementally adapting and extending existing documentation with every sprint is feasible. They suggested using existing tools to include security requirements, e.g., issue tracking software.

5 Discussion

We answer our research question by discussing the key findings and then critically describe the limitations.

5.1 Key Findings

We identified two current challenges specific to security in LSAD that at least three experts mentioned. The first challenge is the lack of qualified personnel with sufficient experience in both security (governance) and agile software development. This challenge amplifies in LSAD due to the larger number of teams. The second challenge is the conflict between security governance and team autonomy when coordinating many teams.

An essential aspect addressing the first identified challenge is the structure of the agile program. Our findings show that all analyzed cases introduce additional security roles, as recommended in the literature. We were able to identify the use of central security teams, roles within the development team, and roles outside of a team. Furthermore, we show that some organizations are not leveraging team-internal security roles, such as a SC. Nevertheless, these roles might be most effective long-term because they enable teams to perform more security activities independently, resulting in more autonomy. To support agile teams, a solid DevSecOps pipeline with static and dynamic application security testing tools is indispensable.

The second challenge fits well with our findings in the security governance category. In all of the analyzed cases, security governance is mainly driven top-down, in contrast to the recommendations from the literature. However, bottom-up approaches are beginning to establish, e.g., development team members gathering in dedicated security communities. In our opinion, leaving the definition of security standards up to individual teams results in substantial, economically unjustifiable efforts and might result in conflicts of interest. A certain level of top-down control is still necessary, e.g., to prepare for external audits. Nevertheless, agile teams should be able to influence the security governance decision-making, and top-down governance should partly shift to self-governance. The described security roles provide a good starting point for building the necessary competency in and around agile teams. This shift could be a way to find the right balance between autonomy and control, consequently bringing closer security governance and LSAD.

Finally, we would like to place our results in the context of the security challenges described by Amber et al. [48], and existing software security maturity models. Our findings regarding the structure of the agile program, security governance, and security activities provide more clarity on how to address the challenge of aligning security objectives in a distributed setting, and contribute to solving the challenge of a common understanding of roles and responsibilities. Our results in the tool-support and automation category relate to the third challenge described by Amber et al., which is "the integration of low-overhead security testing tools" [48].

We identified common patterns between our results and established software security maturity models. For example, the BSIMM [28] identifies so-called *software security groups* in the studied organizations, which are described very similarly to the observed centralized security teams in our study. Another example is the *satellite* role, whose description is largely consistent with the team-internal roles reported in our study. In this particular aspect, our study provides even more granularity by identifying and describing the team-external roles, which are even more widespread than the team-internal roles in the LSAD environments analyzed in this study. Further research on the similarities and differences between our results and software security maturity frameworks could lead to additional interesting findings.

5.2 Limitations

Even though we conducted an interview study, some of the common limitations of case studies described by Runeson and Höst [40] are also relevant for our study and help to structure our limitations. We addressed the threat of *construct validity* by clarifying any ambiguity directly during the conversation with the interviewees. To overcome the threat of *external validity*, which refers to a limited generalizability of results, we based our interviews on scientific literature and conducted the interviews in nine organizations from five industries. However, since we interviewed one expert at each company, we have only a limited picture of each organization. Companies are rarely homogeneous enough for one expert to grasp the entire situation. We countered this by designing our questions to identify overarching patterns within an organization. Additionally, we encouraged our interviewees to keep generalizability in mind. Moreover, the total number of interviewees might be considered relatively small. However, we had already reached a certain level of saturation in the sense that the data collected in the last few interviews became increasingly redundant compared to the data previously collected. To ensure *reliability*, we recorded, transcribed and coded the interviews. This analysis was documented, validated and discussed by the two researchers. Finally, typical problems arise when conducting interviews. That is why we followed the guidelines for good interviews by Kvale [21].

6 Conclusion and Future Work

Addressing security in LSAD is a significant challenge. Despite the importance, there is a paucity of research. Therefore, this paper provides insights into the research question of how security is addressed in LSAD by presenting the results of an interview study. We conducted a literature review to categorize the research topic and interview guide, resulting in four categories: agile program structure, security governance, security activities, and tool support and automation. Our interviews were conducted with nine experts from nine organizations in five industries. One of the key findings is that organizations use centralized security teams, team-internal and team-external security roles. In addition, organizations are using automation for security testing and integrating security activities such as threat modeling or code reviews. Security governance is mainly top-down, while our recommendation is to shift attention to bottom-up approaches. Our findings contribute to raising awareness of the areas to focus on when developing secure software at scale. Practitioners could leverage our results by discussing and applying the identified best practices in their organizations.

Our research could serve as the basis for further scientific investigation. The recurring best practices could be analyzed for their relative impact and effectiveness. Due to the complexity of the research topic, further research could also identify and explore other important aspects regarding security in LSAD, in addition to the four categories identified in our work. Moreover, as we suggest a shift toward more bottom-up security governance, a more in-depth study or evaluation of existing approaches could be conducted. For example, further research could focus on the impact of relevant secure software development maturity models to adapt security governance and compliance processes to agile at scale. More mature development teams may be more capable to self-govern their security posture, and their organizations may be able to afford less top-down control.

Funding. This work has been supported by the German Federal Ministry of Education and Research (BMBF) Software Campus grant 01IS17049.

References

1. Alsaqaf, W., Daneva, M., Wieringa, R.: Quality requirements in large-scale distributed agile projects – a systematic literature review. In: Grünbacher, P., Perini, A. (eds.) REFSQ 2017. LNCS, vol. 10153, pp. 219–234. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54045-0_17
2. Ambler, S.W.: Agile software development at scale. In: Meyer, B., Nawrocki, J.R., Walter, B. (eds.) CEE-SET 2007. LNCS, vol. 5082, pp. 1–12. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85279-7_1
3. Ambler, S.W., Lines, M.: Choose Your WoW. Boston (2020)
4. Aon PLC: 2019 global risk mangement survey - report — Aon (2019). https://www.aon.com/getmedia/8d5ad510-1ae5-4d2b-a3d0-e241181da882/2019-Aon-Global-Risk-Management-Survey-Report.aspx. Accessed 21 Feb 2021
5. Arkin, B., Stender, S., McGraw, G.: Software penetration testing. IEEE Secur. Priv. **3**, 84–87 (2005)
6. Barbosa, D.A., Sampaio, S.: Guide to the support for the enhancement of security measures in agile projects. In: 2015 6th Brazilian Workshop on Agile Methods (WBMA), pp. 25–31 (2015)
7. Beznosov, K., Kruchten, P.: Towards agile security assurance. In: Raskin, V. (ed.) Proceedings of the 2004 Workshop on New Security Paradigms, ACM Conferences, p. 47. ACM, New York (2004)
8. Boström, G., Wäyrynen, J., Bodén, M., Beznosov, K., Kruchten, P.: Extending XP practices to support security requirements engineering. In: Bruschi, D. (ed.) Proceedings of the 2006 International Workshop on Software Engineering for Secure Systems, ACM Conferences, p. 11. ACM, New York (2006)
9. Breaux, T.D., Anton, A.I.: Analyzing regulatory rules for privacy and security requirements. IEEE Trans. Software Eng. **34**, 5–20 (2008)
10. Dännart, S., Moyón, F., Beckers, K.: An assessment model for continuous security compliance in large scale agile environments: exploratory paper. In: Advanced Information Systems Engineering, pp. 529–544 (2019)
11. Digital.ai: 15th annual state of agile report (2021). https://digital.ai/resource-center/analyst-reports/state-of-agile-report. Accessed 21 Feb 2021

12. 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)
13. Döring, N., Bortz, J.: Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften. S, Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-642-41089-5
14. E. U. A. for Cyber Security.: Enisa threat landscape 2020 (2020). https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends. Accessed 21 Feb 2021
15. Google: Google security whitepaper - google cloud (2019). https://cloud.google.com/docs/security/overview/whitepaper. Accessed 21 Feb 2021
16. Horlach, B., Böhmann, T., Schirmer, I., Drews, P.: It governance in scaling agile frameworks. In: Tagungsband Multikonferenz Wirtschaftsinformatik 2018 (2018)
17. IT Governance Institute: Information Security Governance: Guidance for Boards of Directors and Executive Management. IT Pro, IT Governance Institute (2006)
18. Johnson, B., Song, Y., Murphy-Hill, E., Bowdidge, R.: Why don't software developers use static analysis tools to find bugs? In: Proceedings of the 2013 International Conference on Software Engineering, ICSE 2013, pp. 672–681. IEEE Press (2013)
19. Keramati, H., Mirian-Hosseinabadi, S.: Integrating software development security activities with agile methodologies. In: 2008 IEEE/ACS International Conference on Computer Systems and Applications, pp. 749–754 (2008)
20. Kuckartz, U.: Qualitative Inhaltsanalyse: Methoden, Praxis, Computerunterstützung. Grundlagentexte Methoden, Beltz Juventa, Weinheim and Basel, 4. auflage edn. (2018)
21. Kvale, S.: Doing Interviews, The Sage Qualitative Research Kit. Flick, U. (ed.), vol. Pt. 2. SAGE, Los Angeles (2007)
22. LeSS: Overview - large scale scrum (less) (2022). https://less.works/. Accessed 21 Feb 2021
23. Luna, A., Kruchten, P., E. Pedrosa, M.L.D., Almeida Neto, H.R., Moura, H.P.M.: State of the art of agile governance: a systematic review. Int. J. Comput. Sci. Inf. Technol. **6**, 121–141 (2014)
24. Luna, A., Kruchten, P., Riccio, E., Moura, H.: Foundations for an agile governance manifesto: a bridge for business agility. In: 13th International Conference on Management of Technology and Information Systems, São Paulo, SP, Brazil (2016)
25. Ma, Z., Schmittner, C.: Threat modeling for automotive security analysis. Security Technology 2016, pp. 333–339 (2016)
26. MAXQDA: Maxqda — all-in-one qualitative & mixed methods data analysis tool (2022). https://www.maxqda.com/. Accessed 21 Feb 2021
27. Microsoft IT: Security for modern engineering: information security & risk management (2016). https://www.microsoft.com/en-us/download/details.aspx?id=54092. Accessed 21 Feb 2021
28. Migues, S., Erlikhman, E., Ewers, J., Nassery, K.: Building Security in Maturity Model (BSIMM) Report - Version 12 (2021). https://www.bsimm.com/
29. Mohan, V., Othmane, L.B.: SecDevOps: is it a marketing buzzword? - Mapping research on security in DevOps. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 542–547. IEEE (2016)
30. Moyón, F., Méndez Fernández, D., Beckers, K., Klepper, S.: How to integrate security compliance requirements with agile software engineering at scale? In: Product-Focused Software Process Improvement, pp. 69–87 (2020)

31. Moyón, F., Almeida, P., Riofrío, D., Méndez Fernández, D., Kalinowski, M.: Security compliance in agile software development: a systematic mapping study. In: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 413–420 (2020)
32. Myrbakken, H., Colomo-Palacios, R.: DevSecOps: a multivocal literature review. In: Mas, A., Mesquida, A., O'Connor, R.V., Rout, T., Dorling, A. (eds.) SPICE 2017. CCIS, vol. 770, pp. 17–29. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67383-7_2
33. Newton, N., Anslow, C., Drechsler, A.: Information security in agile software development projects: a critical success factor perspective. In: Proceedings of the 27th European Conference on Information Systems (ECIS) (2019)
34. Nguyen Quang Do, L., Wright, J., Karim, A.: Why do software developers use static analysis tools? A user-centered study of developer needs and motivations. IEEE Trans. Softw. Eng. **48**, 835–847 (2020)
35. OWASP Foundation: OWASP Software Assurance Maturity Model - Version 2.0 (2020). https://owaspsamm.org/model/
36. Oyetoyan, T.D., Cruzes, D.S., Jaatun, M.G.: An empirical study on the relationship between software security skills, usage and training needs in agile settings. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 548–555 (2016)
37. Petersen, K., Wohlin, C.: The effect of moving from a plan-driven to an incremental software development approach with agile practices. Empir. Softw. Eng. **15**, 654–693 (2010)
38. Poller, A., Kocksch, L., Türpe, S., Epp, F.A., Kinder-Kurlanda, K.: Can security become a routine? In: Lee, C.P. (ed.) Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 2489–2503, ACM Digital Library. ACM, New York (2017)
39. Rindell, K., Ruohonen, J., Hyrynsalmi, S.: Surveying secure software development practices in Finland. In: Proceedings of the 13th International Conference on Availability, Reliability and Security, ACM Other Conferences, pp. 1–7. ACM, New York (2018)
40. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**, 131–164 (2009)
41. Sadowski, C., Söderberg, E., Church, L., Sipko, M., Bacchelli, A.: Modern code review. In: Paulisch, F., Bosch, J. (eds.) Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, pp. 181–190. ACM, New York (2018)
42. SAP: The secure software development lifecycle at sap (2020). https://www.sap.com/documents/2016/03/a248a699-627c-0010-82c7-eda71af511fa.html. Accessed 21 Feb 2021
43. Scaled Agile Framework: Safe 5.0 framework (2022). https://www.scaledagileframework.com/. Accessed 21 Feb 2021
44. Shostack, A.: Threat Modeling: Designing for Security. Wiley, Indianapolis (2014)
45. Bartsch, S.: Practitioners' perspectives on security in agile development. In: 2011 Sixth International Conference on Availability, Reliability and Security (ARES), pp. 479–484 (2011)
46. Steghöfer, J.-P., Knauss, E., Horkoff, J., Wohlrab, R.: Challenges of scaled agile for safety-critical systems. In: Franch, X., Männistö, T., Martínez-Fernández, S. (eds.) PROFES 2019. LNCS, vol. 11915, pp. 350–366. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35333-9_26

47. Sulejman, V.: It governance and its agile dimensions: exploratory research in the banking sector. In: Proceedings of the 52nd Hawaii International Conference on System Sciences 2019 (2018)
48. van der Heijden, A., Broasca, C., Serebrenik, A.: An empirical perspective on security challenges in large-scale agile software development. In: Oivo, M. (ed.) Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 1–4. ACM, New York (2018)
49. Vejseli, S., Rossmann, A., Connolly, T.: Agility matters! Agile mechanisms in it governance and their impact on firm performance. In: Bui, T. (ed.) Proceedings of the 53rd Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences (2020)

Agile Data Management in NAV: A Case Study

Kathrine Vestues[1](✉), Geir Kjetil Hanssen[1], Marius Mikalsen[1,2],
Thor Aleksander Buan[1], and Kieran Conboy[1,3]

[1] SINTEF, Trondheim, Norway
{kathrine.vestues,geir.k.hanssen,marius.mikalsen,
thor.buan}@sintef.no

[2] Norwegian University of Science and Technology, Trondheim, Norway
marius.mikalsen@ntnu.no

[3] Lero Centre for Software Research, NUI Galway, Galway, Ireland
kieran.conboy@nuigalway.ie

Abstract. To satisfy the need for analytical data in the development of digital services, many organizations use data warehouse, and, more recently, data lake architectures. These architectures have traditionally been accompanied by centralized organizational models, where a single team or department has been responsible for gathering, transforming, and giving access to analytical data. However, such centralized models presuppose stability and are incompatible with agile software development where applications and databases are continuously updated. To achieve more agile forms of data management, some organizations have therefore begun to experiment with distributed data management models such as "data meshes". Research on this topic is however limited. In this paper, we report findings from a case study of a public sector organization in Norway that has begun the transition from centralized to distributed data management, outlining both the benefits and challenges of a distributed approach.

Keywords: Agile software development · Distributed data management · Data mesh · Empirical · Case study

1 Introduction

Most software organizations are aiming to become data driven, where all business units take an active role in both the production and consumption of analytical data. However, this "democratization" of data [1] challenges traditional centralized data management architectures and organization models, such as the data warehouse [2]. Data warehouse models, where a single team or department is responsible for managing analytic data require predictability and stability, characteristics which are incompatible with agile development.

A common challenge within data management is that the logic and flow of data does not follow the structure of the organization [3]. For instance, centralized data management does not follow the logic of agile software development where autonomous, cross-functional teams have end-to-end responsibility for products. Mismatches between

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 220–235, 2022.
https://doi.org/10.1007/978-3-031-08169-9_14

organizational structures and data usage can lead to issues such as data silos and unclear responsibilities. This is especially problematic when developing analytical solutions that cross organizational boundaries and rely on data from different silos [3]. A proposed remedy gives teams increased ownership of data produced by their applications [4]. Such initiatives aim to improve the coordination of people, process, and technology to enable more agile and automated approaches to data analytics [4, 5]. The goal is to bring stakeholders such as data architects, data engineers, data scientists, application developers and data consumers together in building analytical solutions in an agile and collaborative manner [6].

One approach to agile data management is "data-mesh" [7]. Dehghani [7, 8] describes the data mesh in terms of four core principles: 1) domain-oriented decentralized data ownership and architecture, 2) data as a product, 3) self-serve data platform, and 4) federated computational governance. Unlike central management models, such as data warehouse or data lake, the data mesh sees data as context-dependent and best managed in a distributed manner [9]: Those who produce and know the data are best equipped for curating and distributing it.

While there is a rich body of literature on data management, focusing on areas such as the collection, curation, consumption and control of data, empirical papers describing distributed data management and data mesh are still scarce. This is problematic, considering the emphasis among researchers and practitioners on increasing the use of analytical data in improving the efficiency and quality of services. We therefore ask the following research question: *What are the challenges for agile software development organizations when introducing distributed data management?*

We seek to answer this question by reporting findings from an interpretive case study of the development unit in NAV, short for the Norwegian Labor and Welfare administration. NAV forms the backbone of the Norwegian welfare system and is responsible for redistributing one third of the national budget through schemes such as age pension, sick benefit, and unemployment benefit. To provide analytical insight both inside and outside of NAV, data has been collected and curated by a centralized unit and processed in a data warehouse consisting of many registers. Whereas the centralized model worked satisfactory in a system landscape with large systems that rarely changed, it has proven problematic as the organization transitions towards agile development teams and continuous development. To address these challenges, NAV has begun the implementation of a distributed data management model, inspired by the principles of data mesh [7].

Our study sheds light both on the potential benefits of distributed data management, as well as the challenges that such approaches cause. The findings are a first step towards a process model capturing the transition from centralized to decentralized data management. It will also assist practitioners who consider a similar change.

The rest of the article is organized as follows: Section 2 presents the case background and the methods, while Sect. 3 presents the findings. Section 4 discusses the findings and outlines some key challenges that must be solved. Section 5 concludes with a consideration of future possibilities for research.

2 Background

2.1 Analytics

In order to study analytics the first step is to provide a definition of what the term means in the context of this study. Analytics are frequently referred to as the techniques, technologies, systems, practices, methodologies, and applications that enable organisations to analyse critical business data [10]. Seddon and Currie [11] propose a definition that is concerned with evidence-based problem recognition and solving that occur within the context of business environments, namely *"the use of data to make sounder, more evidence-based business decisions"*. This is the definition adopted in this study. However, the extant conceptualisation and classification of business analytics is quite limited and what does exist [11–13] tends to vary greatly. In terms of getting to a more specific and operationalised definition of business analytics that can be used, this study draws on [13], which systematically reviewed and consolidated the extant conceptualisations of business analytics. Their literature review showed that, in terms of describing the data characteristics that underpin the notion of business analytics, many characteristics exist; however, the three key attributes include *volume*, *velocity* and *variety* of data [14, 15]. Given that this is an exploratory study, we chose to adopt a broader perspective regarding the data attributes that are relevant in business analytics.

2.2 Analytics in the Public Sector

The Norwegian public sector is highly digitalized and represents a data-rich domain with access to advancing technologies for analyzing and utilizing data. This underpins the idea of a "data-driven" public sector where data analytics are seen as a path to better policymaking and improved services [16]. However, business analytics can also be challenging. In a study of the Norwegian public sector, Broomfield and Reutter [16] identified several challenges. Among these were: 1) *Organizational culture*, 2) *Privacy and security concerns,* 3*) Outdated legal and regulatory frameworks*, 4) *Data quality* - where the use for data analytical purposes may put additional requirements relating to contextualization, biases, and the suitability of data, and 5) *Access to data* - where data needs to be accessible, both from a technical and an organizational standpoint.

Although analytics in the public sector has become an established research field [17], especially from the organizational and regulatory perspective, the technical and IT perspectives are in a nascent state with few empirical studies available [18–20].

2.3 Agile Analytics and Data Mesh

There are many debates emerging regarding the use of analytics in high speed, agile environments, e.g., the use of analytics in a democratized manner [1] or the use of analytics to enable dynamic capabilities [21]. From a practitioner literature, the concept of a "data mesh" [7, 8] has been proposed as a novel means of managing analytical data. Inspired by Eric Evans book on domain-driven design [22], Dehghani [7] argues that data should be built and managed around "domains", proposing 4 principles which will enable organizations to manage analytical data at scale: *1) Domain-oriented decentralized data* *ownership and architecture*: Data are owned, managed, and located according to their business or thematic domain, e.g., being the responsibility of domain teams that have deep insight in their domain, and then also the domain-oriented data. *2) Data as a product*: In much the same way as teams sees the software they produce as a product (typically in the form of a service) where they have a special responsibility to the end-users, data are also treated as a product. A data product must have the right level of quality and availability, where the owner understands the needs of the consumer of the data product. In practical terms, a data product consists of code (data pipelines to access data), data and metadata (the actual data and metadata that is needed to understand and use data), and infrastructure (to execute the code and to store the data). *3) Self-serve data platform:* In similar ways as teams may use a shared application platform to deliver their software products to consumers, they also need a platform to deliver their data products to consumers, such as other teams or data analysts. The platform offers tools and infrastructure for simplified provisioning as a shared asset in the organization. This can be infrastructure and tools for creating, maintaining, announcing, and sharing data products. *4) Federated computational governance:* Following the distribution of data and the responsibility for data comes a need for a federated approach to govern and improve the data mesh, including common principles and a shared data platform. Governance is a shared responsibility between data product owners, their consumers, and data platform product owners.

However, despite increasing attention among both researchers and partitions, there are to date few peer-reviewed empirical studies that exploring how agile data management and data mesh is addressed by organizations. Apart from informative whitepapers and internal presentations, e.g., from Zalando and Netflix [20], we have only identified one empirical study [18]. The reported transition indicates that the data mesh might increase analytical capabilities, suggesting that more industrial studies of practice are needed.

3 Research Site and Methods

3.1 Case Background

The focus of the study is on how changes to organization, technology architecture, and software development approach is affecting the management of analytical data. The research was performed within the IT department of NAV, short for the Norwegian Labor and Welfare Administration. The IT department has approximately 800 employees that maintain and operate welfare services. The organization uses consultants as needed in development initiatives. NAVs IT system portfolio is made up of several generations of solutions, from mainframe systems to modern web-oriented applications, as well as standard systems that support operations such as accounting, payroll, and document production.

3.2 Data Collection

Data was collected from two main sources: Interviews and document reviews. To capture several aspects of the shift from centralized to decentralized data management, we chose informants from three parts of the organization: 1) data warehouse teams, 2) application development teams, and 3) the data platform team (the team responsible for developing the new data platform). Since these teams were cross-functional, they had members belonging both to the IT department (technical expertise) and to relevant business areas. These three categories of informants were chosen because they cover the various roles involved in data analytics within NAV: The data warehouse teams consumed data, the Application development teams produced data, while the data platform team developed the platform and facilitated the exchange of data. The number of informants, and their distribution across the different types of teams is listed in Table 1.

Although the long-term goal is for application development teams to both produce and consume analytical data, this has not yet occurred. In this first stage of the transition, the organization's focus has been on supporting existing uses of analytical data, rather than using data in new ways.

Table 1. Overview of interviews.

Team type				
		1) Data warehouse	2) Application	3) Data platform
Role	Data scientists		2	
	Data analysts	1		
	Developers	3	3	
	Privacy coach		1	
	Product owners	1	2	3
	Managers			2
		5	8	5

We performed 18 semi-structured interviews. Of these, 12 were recorded and transcribed. Interviews lasted between 30 and 60 min. In the cases where we were unable to record the interviews, one researcher asked questions, while another took extensive notes. Informants were recruited through a snowballing approach, where one informant would suggest another. Typically, we were guided towards respondents that were known to have updated knowledge, competency, and interest in the topics that are relevant to our study, e.g., on the construction of the data platform, domain teams that are early adopters, data scientists looking for data, managers of groups that are impacted by the data mesh initiative, etc.

A second important source of data were document reviews. These documents included project steering documents, descriptions of the new data strategy (as proposed by the NAV IT department, online documentation of the data platform (GitHub), descriptions of NAV's IT ambition, and conference presentations held by members of the development organization (i.e., the Norwegian JavaZone conference[1], and the data

[1] https://javazone.no.

mesh podcast[2]). Many of these presentations have been recorded and published online. They provided insight into the public version of NAV's IT and data strategies.

In addition to the data sources mentioned above, one of the authors studied the transformation of NAV's IT department from 2017 to 2019 as part of her PhD work [23]. In this period, NAV transformed its software development strategy and application architecture. Informants described this transformation as a trigger for the transition from centralized to decentralized data management. Changes to the software development strategy therefore needs to be seen in connection.

3.3 Data Analysis

The data analysis can be described as an iterative three-step process [24]. In the first step, we explored appropriate literature to conceptualize the phenomenon of interest. Initially, we focused on the literature on open data. However, as we began the fieldwork, we learned that NAV's focus was on improved data sharing *inside* NAV. The rationale behind this internal focus was that effective data sharing with external partners, requires efficient data sharing internally. Attention was therefore shifted from external to internal data management, where we paid attention to the data mesh concept [7], which very clearly motivated the IT-organization.

In the second step of data analysis, data was examined inductively through a manual coding process. Among the codes to emerge were "data product", "data platform", and "ownership". The codes were discussed and grouped into meaningful categories. We derived at two overarching categories, namely Centralized data management and Agile data management. We applied a manual approach for coding, where paper prints of transcripts and notes were shared between three of the researchers, sections that were found to exemplify or explain the implementation and viewpoints on the data mesh principles were extracted (cutting out text snippets) and arranged in groups that were given descriptive titles (codes).

In the third step, the inductively derived codes were merged with concepts from the literature. We found that our codes largely overlapping with the principles of "data mesh" [7, 8], leaving us with 3 categories of Agile data management: data ownership and products, data platform, and data governance. This provided us with structured insight into the organization's interpretation and adaption of data mesh.

4 Findings

4.1 Background to the Transition

To increase the efficiency and flexibility of public services, NAV has made substantial changes to the way they develop and disseminate software during the past few years. Handovers between departments have been replaced by continuous development, and hierarchical organization has been replaced by cross-functional teams that take responsibility for the entire software development life cycle. To enable and support these

[2] https://daappod.com/data-mesh-radio/early-platform-insights-goran-berntsen-and-audun-fauchald-strand/.

organizational changes, large and monolithic IT systems are being broken down into smaller applications. By reducing dependencies between applications, teams can work more independently, thus increasing the flexibility and speed of development.

However, the transition towards continuous development and smaller applications is challenging NAV's use of analytical data. Within NAV, analytical data has traditionally been managed by a single unit, the Knowledge department. As the name implies, the Knowledge department has been responsible for producing analytical insight about NAV, ranging from public statistics to internal steering information. By collecting data from various data sources and synthesizing them into a coherent model (data warehouse), the Knowledge department has been able to provide insight across business domains. But the centralized does not scale: As the number of data sources and change rates increase, the Knowledge department has become a bottleneck and a potential source of error. To manage these shortcomings, the NAV IT department has proposed a decentralized data management strategy, where teams take responsibility for preparing and sharing data produced by their applications.

In the following sections, we begin by giving a more detailed description of NAV's centralized data management strategy, and why it is incompatible with agile software development practice. We then continue to describe the ongoing transition towards decentralized data management and the challenges this entails.

4.2 Centralized Data Management

NAV is responsible for presenting statistics and steering information on welfare services and users. Among their customers are the Government, Statistics Norway, as well as the media and the public. Many of these statistics are regulated by law, including the Statistics act[3] and financial regulations[4]. The reported statistics are used for planning and prioritizing and influence internal operations as well as national interests.

"NAV is a large enterprise, and it affects the stock market if our reports are wrong. What is happening [with the data] under our wings is of great importance nationally." (Member of Application development team).

The Knowledge department has traditionally been responsible for gathering analytical data across NAV. These data have been extracted from source systems, transformed, and loaded into a data warehouse. The data warehouse team has been responsible for transforming and compiling data into a coherent data model. This requires extensive knowledge of both source systems and business domains:

"[Data] must be arranged such that you don't put apples and grapes in the same report. You need to understand the concepts which were in the data when they were originally reported. [...] This is addressed in the traditional data warehouse model, with ETL [Extract-Transform-Load] thinking and processes for extracting and transforming data, where you know with certainty what has happened to the data which lay in your centralized data storage." (Member of data warehouse team).

[3] https://lovdata.no/dokument/NLO/lov/1989-06-16-54.

[4] https://www.regjeringen.no/globalassets/upload/fin/vedlegg/okstyring/reglement_for_okonomistyring_i_staten.pdf.

This approach worked reasonably well when the system landscape consisted of large monolithic IT-systems and databases that rarely changed. As formulated by a member of the Data platform team:

"Back then [two to three years back], the data warehouse team could extract all the data, and changes were quite rare. Because changes were a hassle".

However, with the transition towards agile development and micro architectures, applications and databases began to change more frequently. For some systems, change rates increased from yearly to hourly releases.

The data warehouse team was unable to cope with the escalating number of changes, forcing the NAV to look for alternative data management strategies.

"The centralized data warehouse environment cannot keep up with the pace because they are not rigged for it. It was doomed to fail before they tried, because somehow you suddenly have 150 applications instead of a few large monoliths. [...] We have gone from making changes [to our software] four times a year [...] to around 1300 times a week. In other words, continuous deployment, and it is no longer possible for a centralized environment to keep up with all the changes. Things break in pipelines and then things stop working and are not updated. So, this has been the big question: What do we do to fix it? How do we equip ourselves?" (Product owner).

To address the problem, the IT department proposed a distributed model, described as a "data-mesh" [7], where application development teams take responsibility for creating products and sharing data.

4.3 Towards Agile Data Management

The distributed data management model, or "data-mesh" [7], can be described in terms of 1) data products and ownership, 2) data platform, and 3) federated governance. Each of these elements, and their interpretation within NAV are described below.

Data Products and Decentralized Data Ownership. Foundational to the data mesh is the decentralization of data ownership. For NAV, a shift from centralized to decentralized ownership implies that application development teams assume responsibility for their own data:

"It is not a technological change or a technical implementation that is the big change. The big changes come when we say to the teams, for example, the team working with unemployment benefits, that they are also responsible for producing analytical insight into the domain. Reporting and statistics. They don't do this now, because today this is the responsibility of the Knowledge department" (Member of the Platform team).

With the distributed data ownership, interpretations and decisions relating to the data are done by the people closest to the data. In addition to sharing data with other teams, the distributed ownership model is thought to increase the quality of analytical data within the team:

"We not only want the teams to share [their data] with others. We also want the teams to become aware of the possibility of using these data themselves to make decisions. This will result in better data for everyone" (Member of Data platform team).

As a means of implementing data ownership, teams will develop so called "data products". A data product is defined as a dataset and the documentation it. Data products require deliberate design and management, satisfying the needs of prospective users.

The term "data product" is used to show that data needs to be treated as other products or services within the organization, and that the team. This requires insight into the needs of prospective users, and a strategy for maintaining and improving the products.

However, the transition towards distributed data management causes concerns in some parts of the organization. One informant addresses the fear of losing control over the data:

"We are concerned that when individual teams take ownership of data and begin to produce data products, we might lose oversight over the different domains. This means that it must be clear who has responsibility for what, which isn't currently the case" (Product owner, Data warehouse team).

Others were concerned that the teams neither has the competence nor the time to take responsibility for the data and that data consumers would no longer have insight into and control over the extraction and transformation.

"Data won't be prioritized. That's our experience. Developing data products is not something development teams usually think about when they develop systems. They are concerned with the [end] user, and how the case worker will use the system. Data is way down on their list" (Product owner, Data warehouse team).

Self-serve Data Platform. To enable distributed data ownership, the organization has introduced a self-serve data platform called NADA. The new data platform differs from the data warehouse in several ways. Most importantly in the way data is shared: While data in the data warehouse is collected and curated by a single team, the new data platform offers functionality which allows all teams to share their data. The NADA platform is thus a multisided platform where the entire organization can produce and consume data.

Despite the need for alternative ways of managing analytical data, there is not yet consensus across the organization concerning the new data strategy. For distributed data management to be introduced, the IT department must therefore develop a data platform which simplifies data sharing and analysis, as compared to existing solutions:

"If a team is to become responsible for publishing insights concerning their domain, then they must have tools that make it easy. How can they publish a data product that provides insight into changes [within the domain] over time, or the number of cases we have processed per day? How can you publish this information easily?" (Member of platform development team).

The platform will become a marketplace where producers and consumers meet to exchange data. To increase the value of the platform, the platform development team actively encourages data producers to offer their data on the platform. The platform team describes this process of identifying needs and encouraging teams to add data products as "growth hacking" The platform team tries to understand the needs of users, and subsequently going out into the organization to get these needs fulfilled:

"*I ask teams that have data which I know will be useful to others to create data products and deploy them on the platform"* (Member of data platform team).

In addition to facilitating the creation of data products, the platform will have a dashboard and tools for analysis. The output of the analysis can in turn be used to create new data products, thus allowing insights to be shared and reused across the organization. The platform is based on Google Development Platform and data products are created in BigQuery[5]. Although BigQuery is currently the only available technology on the platform, the platform team plans to offer other technologies in the future.

However, developing a multisided platform is challenging, since there is no direct interaction between producers and consumers of data, and a producer is not directly rewarded for preparing and sharing their data.

"With the data platform, on the other hand, you have two types of users: You have those who produce data and those who need data. We therefore use the metaphor 'data marketplace'. We are creating a marketplace where it should be possible to offer and to find data. So, it is a more complex image for us who create the platform because we are not simply a service provider. [...] So, we have more of a chicken and egg problem, where you need some users on the consumer side, since this gives value. But to get some consumers, you also need some data which they can consume" (Member of platform development team).

Federated Computational Governance. At the time of writing, very few rules govern the creation and dissemination of data products on the platform. This follows from the platform team's deliberate intension of minimizing the number of rules enforced:

"As the data platform provider, we do not wish to become a large, centralized decision-maker. We wish to listen to our users to understand their needs, and we aim to be very restrictive with implementing rules" (member of data platform team).

The creation of rules thus happens through ongoing negations, where rules are formed in collaboration with data producers and consumers:

"So far, we don't have many rules that apply, because we have very few users both on the consumer and producer side, but this is an ongoing discussion. How do we agree on the rules? For example, should we use one type of key to identify a person? Should we use birth number? Should that be the key for all data, or should each individual domain be able to have its own? We have several keys identifying a person today. [...]. These are rules we must agree on. But to know what [rules] to make, we need to know what users need. For this, I need a forum where producers and consumers of data can meet and agree on the rules" (Member of platform team).

The IT department is exploring how they can maintain the privacy and security of citizens, while simultaneously stimulating teams to share and use data. To address this challenge, domain teams have access to a "privacy coach", which gives them legal counseling in the use of data. The IT department also has a "Data treatment catalog", where the use of sensitive data is recorded and justified. However, the data treatment catalogue has not yet been linked to the data platform:

"All teams that treat data should register this treatment in the Data treatment catalogue and make the information available to the rest of NAV and to the authorities. It can also be used for other purposes, but so far, it is not linked to the [new] data platform. So, the ability to describe datasets and the legal justification for use has not yet been linked to the platform" (Data analyst).

Whereas some data products only involved data from a single domain, the most valuable data products are those that involve multiple products and domains. One example of such domain-spanning products are unemployment figures. Unemployment figures

[5] https://cloud.google.com/bigquery.

cannot be calculated from a single system but are based on "all the things which a person is not". For instance, an unemployed person is not under education, is not on sick leave, and is not temporarily laid off, elements of information which is gathered from a series of different information systems. Other examples of compilations of data from multiple domains are average case processing time, and the number of erroneous payments made by NAV. Using data from different domains require knowledge of these domains. In the centralized model, this competence is held by the Knowledge department, and there is concern that cross-organizational insight and the ability to analyze data across domains will be reduced with distributed data management and local ownership of data.

"It requires a lot of competence to use data from other domains. So, if you are to use data from another domain, it must be well documented. The data must be processed in a way that makes it easy to understand and user-friendly. In addition, what does it mean to connect data [from different domains]? This is a type of competence which takes time, and which must be acquired by the teams that work with source systems" (Employee in the Knowledge department).

Another concern relates to the willingness of teams to invest in data products, as they have no direct benefit in sharing the data. Some informants therefore believe that data sharing must be compulsory:

"We want there to be established requirements, compelling teams to make data available. And make sure that this data is made available as part of the statistics and steering information. Otherwise, it will be difficult for us, because we cannot involve ourselves with all the 120 teams" (Product owner, Data warehouse team).

5 Discussion

Our initial involvement with NAV has provided some early insights regarding both the need and motivation for considering data mesh as a strategy for becoming data-driven, but also insights into challenges that follow from such a transition. NAV is the largest service organization in the country and administers data on – literally – every Norwegian citizen. However, technical, and organizational legacy is challenging the organization's agility, and their ability to convert data into actionable insights.

NAV's journey towards increased agility has so far taken the organization through two transitions. First, the IT department enabled autonomous and cross-functional teams that build domain knowledge in product areas such as Work, Health, and Family. Cross-functional teams within each domain have autonomy and responsibility for the continuous development and deployment of related IT services, e.g., caseworker support systems.

Second, to match this way of organizing software teams, the system architecture has been transformed over time: Large and monolithic systems have been broken down into micro-services, enabling independent and loosely coupled applications that can be managed by single teams. However, although these transitions have increased the agility of the software development organization, they have also triggered the need for alternative ways of managing data. Traditional data management models, where analytical data are gathered in data silos and interpreted by a centralized unit, are unsuited for a distributed and continuous reality of agile software development. As monolithic systems are broken down into smaller applications, change rates have increase, and data pipelines are broken. Hence, applying the ideas of data mesh and distributed data management might be considered an imperative in further increasing the agility of software organizations and creating a data-driven organization.

The transition towards the principles of data mesh and distributed data management is however in an early stage and does not come without challenges. Our findings reveal that such a radical change creates uncertainties in various parts of the organization, depending on their need for and use of data.

Challenge 1 – Change of Control of Data Extraction and Transformation. Analysts in the knowledge department are concerned that they might lose control of data sources, resulting in erroneous data and reduced quality. They argue that having overview of the various domains is necessary when producing national statistics and providing analysis to national authorities and policy makers. They argue that such an overview cannot be obtained in application development teams, which role is precisely to specialize in a specific domain. The question thus remains: When the traditional centralized data management model where a single unit is responsible for gaining cross-organizational insight is replaced by local ownership and data products – how will the organization be able to support compiled data products which require cross-organizational insights? How should new needs for data be communicated to many data-controlling domain teams?

Challenge 2 – Managing Rightful and Legal Access to, and Use of Data. The principle that domain teams are responsible for offering “their” data as data products via a data platform that everyone can access raises concerns regarding control and rightful use of data according to regulations on data protection. The General Data Protection Regulation (GDPR), which all the countries in the European Union (EU) and European Economic Area (EEA) are covered by, is a good example of such a regulation that may problematize the data mesh mentality. How does one incorporate the data minimization principle in GDPR, which says that you should only collect and process the minimum amount of data possible to fulfill your purpose, in the data mesh where one wishes to collect and process as much data as possible? Or how can one provide a transparent description of how a person’s data is processed as demanded by the lawfulness, fairness, and transparency principle in GDPR when the aim of the data mesh is to provide the data to everyone with the goal of continuously discovering new innovative ways of utilizing the data? The principles of data mesh and GDPR does not fully harmonize, and it is unknown how a data mesh should be managed in practice to reap the fruits of Dehghani’s [7] data mesh principles as well as being in line with the General Data Protection Regulation.

Challenge 3 – Creating Data Products. Traditionally, software product teams have given little thought to the data stored in their databases beyond their own use in the specific application that is developed. It has been the responsibility of data engineers and analysts in the central unit to gather and prepare data for analytical purposes – using the data warehouse as the main data storage. Developers have traditionally been focused on developing of end-user functionality, lacking both the competence and motivation for prepare and enrich data for analytical use. Hence, there is a need to add competence and capacity to the domain teams. It is however unclear which skillset it requires, and what the cost will be.

Related to this issue is the need for data products that span multiple product domains. Although some insight can be gained by analysing data within a single domain, the most valuable use cases involve a combination of data products from different domains. The autonomy of domains must therefore be combined with some degree of standardization, making it possible for data products to be combined. This requires insight into other business domains, as well as one's own.

Several questions therefore need to be answered: Should cross-functional teams include a data scientist function or role? How much can be automated and supported by the data platform? And how should a team learn about the need for data in other business domains?

Challenge 4 - Establishing a Thriving Ecosystem. A functioning data mesh builds on data owners that publish data products that can be consumed by others. But which incentives does a software team have to invest time in preparing, publishing, and maintaining data products? Of course, in a system where most teams can make use of data from other teams, we could foresee a naturally functioning ecosystem – but do we understand such mechanisms properly? Should the publication of data products be an organizational obligation or are there other mechanisms that could be put into play? For example, could we make use of the same incentives that drive open-source development of code, where opening your data means that other's provide valuable feedback and enriched data in return? Would opening of data mean that the product team as a data provider put extra effort in making data understandable and useful – in short, establish proper quality of the data product?

This overview of challenges is not exhaustive and by no means complete. It merely provides an initial understanding of the many challenges related to the effectuation of the data mesh strategy in a complex and data-intensive software organization as perceived by informants in NAV. Had we studied other organizations within other sectors or countries, these challenges might differ.

There are also some challenges not addressed in the current study: Among these are whether it is possible to host the data platform on a cloud service by a service provider which is located outside of EU/EEA in a lawful way. The Schrems II ruling by the Court of Justice of the European Union states that "companies must verify, on a case-by-case basis, whether the law in the recipient country ensures adequate protection, under EU law, for personal data transferred under SCCs and, where it doesn't, that companies must provide additional safeguards or suspend transfers". The US, where most of the largest cloud providers have their headquarters, is a country that often is not considered a country where personal data is adequately protected.

We have studied a single case organization and a recent phenomenon (the transition towards a data mesh) in an early phase, over a restricted period (approximately 6 months). This naturally restricts generalizability. However, the study provides valuable early insights into a very large and complex organization that seeks to implement increase the use of efficiency of analytical data by introducing distributed data management – a challenge shared by many data-rich organizations. Furthermore, there are yet few reports from practice on how a "data mesh" can be realized and the challenges which organizations might face. We hope that others can build on this in future work. We have aimed to ensure validity by following acknowledged guidance on case studies [6] We have gathered data from more than one source (triangulation); document analysis (e.g. strategy documents) and interviews (covering a wide span in the IT organization), and we have collected data within a real-life context (NAV).

6 Conclusion

Our findings suggest that the organization agrees on the need for alternative ways of managing analytical data. There are however varying views on how this should be done, and how distributed data management and data mesh will affect the creation and use of analytical data. Also, although the main concepts, as laid out by Dehghani [7], are understood and motivates the transition, it is too early to see how these will be implemented, and how they will affect roles and work processes.

The ongoing transition is driven "inside-out", meaning that a data platform team offers a technical solution - the data platform, and supports teams that chose to take the platform into use. Some challenges have been identified and need to be addressed, while others have yet to appear. We hope that the potential benefits of more agile data management inspire researcher to investigate these approaches in the years to come.

6.1 Future Work

We will continue to follow NAV and their transition towards becoming a data-driven organization. In that, we will 1) address the challenges that was identified in this study (as well as new emerging challenges), 2) collect and analyze data to investigate whether the new approach – data mesh – provides the effects that initially motivated the investments, and 3) describe the details on *how* NAV, as a complex organization, implements these principles. This briefly described research agenda has a potential for extending the knowledge on agile data management, and on how organizations can make better use of analytical data in improved insight and services. Furthermore, observing the case over time will give a basis for developing theories concerning the adoption of distributed data management. Leaning on the proposed framework by Eisenhardt [25], we have initiated some of the recommended steps, such as *Getting Started*, *Selecting Cases*, and *Entering the Field*.

Acknowledgements. This work was conducted in the project Transformit supported by the Research Council of Norway through grant 321477. The project was led and sponsored by the company KnowIT AS. We also want to thank NAV for granting us access to the case.

References

1. Kashyap, B., Shinghal, K.: Data democratisation: the evolution of informed decisions (2019)
2. Joshi, D., Pratik, S., Podila, M.: Data governance in data mesh infrastructures: the Saxo Bank case study. EasyChair, 2516–2314 (2021)
3. Janssen, M., Brous, P., Estevez, E., Barbosa, L.S., Janowski, T.: Data governance: organizing data for trustworthy artificial intelligence. Gov. Inf. Q. **37**(3), 101493 (2020)
4. Zahid, H., Mahmood, T., Ikram, N.: Enhancing dependability in big data analytics enterprise pipelines, pp. 272–281 (2018)
5. Ereth, J.: DataOps-towards a definition. LWDA **2191**, 104–112 (2018)
6. Bergh, C., Benghiat, G., Strod, E.: The dataOps cookbook. DataKitchen Hqrs. (2019)
7. Dehghani, Z.: Data mesh principles and logical architecture (2020). https://martinfowler.com/articles/data-mesh-principles.html. Accessed 19 Feb 2022
8. Dehghani, Z.: How to move beyond a monolithic data lake to a distributed data mesh. How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh (2019). https://martinfowler.com/articles/data-monolith-to-mesh.html. Accessed 19 Feb 2022
9. Priebe, T., Neumaier, S., Markus, S.: Finding your way through the jungle of big data architectures, pp. 5994–5996 (2021)
10. Chen, H., Chiang, R.H., Storey, V.C.: Business intelligence and analytics: from big data to big impact. MIS Q., 1165–1188 (2012)
11. Seddon, J.J., Currie, W.L.: A model for unpacking big data analytics in high-frequency trading. J. Bus. Res. **70**, 300–307 (2017)
12. Holsapple, C., Lee-Post, A., Pakath, R.: A unified foundation for business analytics. Decis. Support Syst. **64**, 130–141 (2014)
13. Mikalef, P., Pappas, I.O., Krogstie, J., Giannakos, M.: Big data analytics capabilities: a systematic literature review and research agenda. IseB **16**(3), 547–578 (2017). https://doi.org/10.1007/s10257-017-0362-y
14. McAfee, A., Brynjolfsson, E., Davenport, T.H., Patil, D., Barton, D.: Big data: the management revolution. Harv. Bus. Rev. **90**(10), 60–68 (2012)
15. Sun, E.W., Chen, Y.-T., Yu, M.-T.: Generalized optimal wavelet decomposing algorithm for big financial data. Int. J. Prod. Econ. **165**, 194–214 (2015)
16. Broomfield, H., Reutter, L.M.: Towards a data-driven public administration: an empirical analysis of nascent phase implementation. Scand. J. Public Adm. **25**(2), 73–97 (2021)
17. Fredriksson, C., Mubarak, F., Tuohimaa, M., Zhan, M.: Big data in the public sector: a systematic literature review. Scand. J. Public Adm. **21**(3), 39–62 (2017)
18. Loukiala, A., Joutsenlahti, J.-P., Raatikainen, M., Mikkonen, T., Lehtonen, T.: Migrating from a centralized data warehouse to a decentralized data platform architecture, pp. 36–48 (2021)
19. Machado, I.A., Costa, C., Santos, M.Y.: Data mesh: concepts and principles of a paradigm shift in data architectures. Procedia Comput. Sci. **196**, 263–271 (2022)
20. Machado, I., Costa, C., Santos, M.Y.: Data-driven information systems: the data mesh paradigm shift (2021)
21. Conboy, K., Mikalef, P., Dennehy, D., Krogstie, J.: Using business analytics to enhance dynamic capabilities in operations research: a case analysis and research agenda. Eur. J. Oper. Res. **281**(3), 656–672 (2020)
22. Evans, E., Evans, E.J.: Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley Professional (2004)
23. Vestues, K.: Using digital platforms to promote value co-creation: a case study of a public sector organization (2021)

24. Pan, S.L., Tan, B.: Demystifying case research: a structured–pragmatic–situational (SPS) approach to conducting case studies. Inf. Organ. **21**(3), 161–176 (2011)
25. Eisenhardt, K.M., Graebner, M.E.: Theory building from cases: opportunities and challenges. Acad. Manage. J. **50**(1), 25–32 (2007)

Author Index