Technische Universität München
TUM School of Computation, Information and Technology

Empirical Analysis and Solution Designs for Goals and Metrics in Large-Scale Agile Software Development

Pascal M. Philipp

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Alexander Pretschner

Prüfende der Dissertation:

1. Prof. Dr. Florian Matthes
2. Prof. Dr. Ruth Breu

Die Dissertation wurde am 21.11.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 06.05.2024 angenommen.

Abstract

Research Problem:

Metrics are widely used and extensively researched in traditional software development environments. An important finding of previous research is that the successful use of metrics requires a deep understanding of the context. This is critical to tailoring metrics and associated solution approaches to the specific requirements of an organization. A central research problem of this dissertation arises from the increasing relevance of large-scale agile software development and the simultaneously prevailing research deficit on metrics in this context. So far, it is unclear whether metrics are actually relevant for agile practitioners or if their relevance has only been emphasized by scholars. Moreover, current research on large-scale agile software development lacks basic knowledge about goals, which should always be linked to metrics, and about the challenges and success factors of metrics. This lack of foundational knowledge is particularly problematic as it is useful for developing scientific artifacts that support the adoption of metrics while being adapted to the peculiarities of scaled agile development environments. Symptomatically, no scientific artifacts exist to guide the adoption of metrics, nor is there a comprehensive collection of metrics suitable for large agile organizations. Furthermore, current research lacks a management method for metrics, as well as design principles applicable to measurement programs in scaled agile development environments.

Research Design:

A case study and expert interviews are conducted at the outset to gather empirical data on goals and metrics in scaled agile environments. Then, the empirical findings are used with design-oriented research approaches to develop solution artifacts that support agile practitioners in establishing metrics while being adapted to the peculiarities of scaling agile environments. Our first two artifacts, the minimalistic Metric Management Fact Sheet (MMFS) and metric catalog were developed using design science research, and their demonstration and evaluation were conducted as part of a case study. Next, we implemented an action design research study in conjunction with situational method engineering. This study spanned a two-year period to design a management method for metrics in scaled agile environments in close collaboration with practitioners.

Results:

This dissertation provides empirical findings and solution designs for goals and metrics of scaled agile software development. First, we identify 196 metrics and explain the reasons for their use, the respective organizational level, their type (agile or traditional), and the most commonly used metrics. We then highlight recurring challenges and success factors of metrics in scaled agile software development. Based on these empirical findings, we developed three artifacts to support the implementation of metrics in scaled agile software development. The first artifact is the minimalistic MMFS, which is used in the second artifact, the metric catalog, to systematically capture the identified metrics. As a third artifact, we developed a method for managing metrics and defined ten design principles for measurement programs. Finally, we present an investigation of goal-setting within a large agile case organization. Here, we identified goal-setting practices,

various documentation techniques, and a list of categorized goals. In addition, we show challenges in goal-setting and suggest strategies to overcome them.

Contribution:

Our research results represent a significant contribution to the field of large-scale agile software development. They emphasize the relevance of metrics in such development environments for both researchers and practitioners. Furthermore, we provide support for metrics selection by highlighting the reasons for their adoption. We suggest practitioners to consider recurring challenges and success factors when using metrics, thus supporting their successful implementation. In addition, the identified challenges and success factors form a knowledge base that was used to develop three artifacts. These artifacts support practitioners in dealing with metrics while taking into account the specific requirements of scaled agile organizations. The minimalistic MMFS, in combination with the metric catalog provides practitioners with an overview of potential metrics and supports the selection as well as the organization-specific implementation of metrics. Our method for managing metrics and the design principles derived from it help practitioners select, operate, and scale metrics as well as develop their measurement programs. Finally, we provide in-depth knowledge on how to successfully realize goal-setting in large agile organizations and counter recurring challenges using solution strategies.

Zusammenfassung

Motivation:

Metriken sind in traditionellen Softwareentwicklungsumgebungen weit verbreitet und umfassend erforscht. Eine wichtige Erkenntnis der bisherigen Forschung ist, dass der erfolgreiche Einsatz von Metriken ein tiefgehendes Verständnis des jeweiligen Kontexts erfordert. Dies ist entscheidend, um Metriken und die dazugehörigen Lösungsansätze an die speziellen Anforderungen einer Organisation anzupassen. Ein zentrales Forschungsproblem dieser Dissertation ergibt sich aus der steigenden Relevanz von skalierten agilen Softwareentwicklungsumgebungen und dem gleichzeitig vorherrschenden Forschungsdefizit über Metriken in diesem Kontext. Bislang ist unklar, ob Metriken für agile Praktiker tatsächlich von Bedeutung sind oder ob ihre Relevanz lediglich aus wissenschaftlicher Sicht betont wurde. Zudem mangelt es der aktuellen Forschung über die skalierte agile Softwareentwicklung an grundlegendem Wissen über Ziele, die stets mit Metriken verknüpft werden sollten, sowie über die Herausforderungen und Erfolgsfaktoren von Metriken. Dieser Mangel an Basiswissen ist besonders problematisch, da es für die Entwicklung von wissenschaftlichen Artefakten nützlich ist, welche die Einführung von Metriken unterstützen und gleichzeitig an die Eigenheiten skalierter agiler Entwicklungsumgebungen angepasst sind. Symptomatisch existieren weder wissenschaftliche Artefakte, die den Einsatz von Metriken anleiten, noch eine umfassende Sammlung an Metriken, die für skalierte agile Entwicklungsumgebungen geeignet sind. Darüber hinaus mangelt es in der aktuellen Forschung an einer Management-Methode für Metriken, sowie an Gestaltungsprinzipien, die für Messprogramme in skalierten agilen Umgebungen anwendbar sind.

Forschungsdesign:

Zu Beginn werden eine Fallstudie und Experteninterviews durchgeführt, um empirische Daten über Ziele und Metriken in skalierten agilen Umgebungen zu sammeln. Anschließend werden die empirischen Erkenntnisse mit designorientierten Forschungsansätzen genutzt, um Lösungsartefakte zu entwickeln, die agile Praktiker bei der Festlegung von Metriken unterstützen und gleichzeitig an die Besonderheiten skalierender agiler Umgebungen angepasst sind. Unsere ersten beiden Artefakte, das minimalistische Metric Management Fact Sheet (MMFS) und der Metrikenkatalog, wurden mit Hilfe von Design Science Research entwickelt, und die Demonstration und Evaluierung wurde im Rahmen einer Fallstudie durchgeführt. Weiterführend realisierten wir eine Action Design Research-Studie in Verbindung mit Situational Method Engineering. Diese Studie erstreckte sich über einen Zeitraum von zwei Jahren, um in enger Zusammenarbeit mit Praktikern eine Management-Methode für Metriken in skalierten agilen Umgebungen zu konzipieren.

Ergebnisse:

Diese Dissertation bietet empirische Erkenntnisse und Lösungskonzepte für Ziele und Metriken der skalierten agilen Softwareentwicklung. Zunächst identifizieren wir 196 Metriken und erläutern die Gründe für ihre Anwendung, die jeweilige Organisationsebene, ihre Kategorisierung (agil oder traditionell) und die am häufigsten eingesetzten Metriken. Im Weiteren beleuchten wir wiederkehrende Herausforderungen und Erfolgsfaktoren von Metriken in der skalierten agilen

Softwareentwicklung. Basierend auf diesen empirischen Erkenntnissen haben wir drei Artefakte entwickelt, um die Implementierung von Metriken in der skalierten agilen Softwareentwicklung zu unterstützen. Das erste Artefakt ist das minimalistische MMFS, welches im zweiten Artefakt, dem Metrikenkatalog, dazu verwendet wird, die identifizierten Metriken systematisch zu erfassen. Als drittes Artefakt entwickelten wir eine Methode zum Management von Metriken und definierten zudem zehn Designprinzipien für Messprogramme. Abschließend präsentieren wir eine Untersuchung zur Zielsetzung innerhalb einer großen agilen Fallorganisation. Hierbei haben wir Praktiken der Zielsetzung, verschiedene Dokumentationstechniken und eine Liste kategorisierter Ziele identifiziert. Zusätzlich zeigen wir Herausforderungen bei der Zielsetzung und schlagen Strategien vor, um diese zu bewältigen.

Beitrag:

Unsere Forschungsergebnisse stellen einen bedeutenden Beitrag im Bereich der skalierten agilen Softwareentwicklung dar. Sie betonen die Relevanz von Metriken in solchen Entwicklungsumgebungen sowohl für Forscher als auch für Praktiker. Des Weiteren leisten wir Unterstützung bei der Auswahl von Metriken, indem wir die Beweggründe für ihre Anwendung hervorheben. Wir legen Praktikern nahe, wiederkehrende Herausforderungen und Erfolgsfaktoren bei der Nutzung von Metriken zu beachten, um so deren erfolgreiche Implementierung zu unterstützen. Darüber hinaus bilden die identifizierten Herausforderungen und Erfolgsfaktoren eine Wissensbasis, die für die Entwicklung von drei Artefakten verwendet wurde. Diese Artefakte unterstützen Praktiker im Umgang mit Metriken und berücksichtigen dabei die besonderen Anforderungen skalierter agiler Organisationen. Das minimalistische MMFS in Kombination mit dem Metrikenkatalog gibt Praktikern einen Überblick über potenzielle Metriken und unterstützt die Auswahl sowie die organisationsspezifische Implementierung von Metriken. Unsere Methode zum Management von Metriken und die daraus abgeleiteten Gestaltungsprinzipien helfen Praktikern bei der Auswahl, dem Betrieb und der Skalierung von Metriken sowie bei der Entwicklung ihrer Messprogramme. Schließlich vermitteln wir tiefgehende Kenntnisse darüber, wie die Zielsetzung in großen agilen Organisationen erfolgreich realisiert und wiederkehrenden Herausforderungen mittels Lösungsstrategien entgegengewirkt werden kann.

Acknowledgment

As I sit down to pen this acknowledgment, I'm reminded of the African proverb that says, "If you want to go fast, go alone; but if you want to go far, go together." This dissertation is not just a culmination of years of research but also a testament to the incredible community that has supported, guided, and inspired me every step of the way. From the insightful academic guidance to the unwavering personal support, every individual mentioned here, and many more not named, have been instrumental in making this journey memorable and enriching.

First and foremost, I wish to convey my profound gratitude to my doctoral supervisor, Prof. Dr. Florian Matthes. The privilege of working under a mentor whom I deeply respect and admire has been a foundational aspect of my journey. The lessons you imparted, the trust you placed in me, and the numerous opportunities you provided have all been pivotal. I also extend my thanks to Prof. Dr. Ruth Breu for accepting the role as the second examiner for this thesis.

Sincere appreciation goes to my friends, research partners, and colleagues at the sebis chair. A special mention to Franziska and Ömer for the remarkable teamwork and endless insightful discussions. I hold particular gratitude for Nektarios and Sascha, who have accompanied me throughout my entire time at the chair, from its inception to its conclusion. I'm also grateful to Ahmed, Alexandra, Aline, Anum, Burak, Daniel, Dominik, Fatih, Felix, Gloria, Gonzalo, Ingo, Jian, Juraj, Mahdi, Oliver, Patrick, Peter, Phillip, Stefanie, Stephen, Tim, Tobias, Tri, Ulrich, and Wessel, with whom I've shared cherished memories. My former students, Moritz, Leon, and Karla deserve special recognition for their significant contributions.

I'd like to express my deepest gratitude to my longtime friends, Kai and Basti, who have always stood by my side.

Finally, I reserve my most profound gratitude for my family. To my parents, Rolf and Susanne, your unwavering love and support have been invaluable. I am privileged to be your son. Thank you to my grandmother, Christa, and in loving memory of my grandfather, Richard, and my beloved grandmother, Emma. Special appreciation to my sister Mirjam, her husband Daniel, and my treasured nieces, Leni and Romy. Omnia vincit amor.

Garching b. München, 09.11.2023
Pascal M. Philipp

Table of Contents

List of Figures

List of Tables

Part A

CHAPTER 1

Introduction

In this chapter, we offer an introduction to the topic, identify the gaps in existing research, formulate research questions, and present the structure of this dissertation.

1.1. Motivation

For decades, organizational thinking and practice revolved around issues of stability [Orl96]. This paradigm is becoming less relevant as organizations face an ever-changing environment subject to economic, technological, and political transitions [Orl96]. In this environment, the ability to respond to change is becoming increasingly important, and stability is losing its significance [Orl96]. A relevant technological transition that affects entire industries and businesses is the increasing demand for software solutions [VOWVH06]. This shift is so fundamental that it is often described by the phrase "software is eating the world" [VOWVH06].

Business agility has become an umbrella term for an organization's ability to deal effectively with internal and external change [VOWVH06], and is therefore a necessity for success in the current business environment. According to Van Oosterhout et al. [VOWVH06], an enabler for achieving business agility is the implementation of an agile IT and process architecture.

The increasing demand for agility and software solutions poses challenges to traditional plan-based and sequence-oriented software development methods (e.g., the waterfall method) [HC01, Hig02]. Limitations inherent to these methods include a lack of continuous customer feedback, irregular releases with large batch sizes, and adherence to rigid processes [JGZag+13, PW09]. Generally, traditional methods are based on the assumption that problems can be completely defined and that optimal and predictable solutions exist for each problem [DD08].

In 2001, these shortcomings motivated a group of practitioners to formulate the well-known

Manifesto for Agile Software Development [BBVB+01]. In agile software development, the focus lies on feedback and change, wherein higher rates of change are rather embraced than rejected [WC03]. Nowadays, many agile methods such as Scrum [SB01] and Extreme Programming [Bec00] have emerged and are widely spread across the industry [Dig20, HA13, MJ10].

Agile methods have proven to be effective when deployed in contexts similar to the "agile sweet spot," characterized by small, collaborative teams that continuously deliver software and interact with systems with low to medium criticality and a stable architecture [NOK14, Kru13]. However, if agile methods are applied without adaptations (i.e., "out of the box") in contexts that deviate in some dimensions from the context for which they were originally developed, the probability of failure increases [Kru13]. According to Kruchten [Kru13], size and a lack of focus on architecture in large systems are among the context factors that pose the most significant risk of project failure.

Inspired by the success of agile methods on a small scale, an increasing number of organizations began to scale agile methods to larger contexts by adopting scaling agile frameworks [Dig20, DPL16, Kru13, NOK14]. Multiple scaling agile frameworks (e.g., Scaled Agile Framework (SAFe)) [SA22]) have been developed to help organizations apply agile software development on a larger scale. Simultaneously, the interest of the scientific community in this topic has grown, as evidenced by the increasing number of publications in the last decade [UPP+22].

However, by adding complexity to scaling agile environments (e.g., through adding agile teams), maintaining an appropriate level of situational awareness by purely relying on qualitative feedback loops becomes difficult. For example, it is hard in terms of time for a manager of a large multi-team setting to communicate with each agile team regularly. In such circumstances, metrics can augment or replace qualitative feedback.

In the context of traditional software development, metrics are well-researched and used to measure processes, products, and resources [FN00, FB14]. Several scientific articles propose designs for measurement programs to increase the chances of successful metrics implementation [TRG16]. Most of these measurement programs rely on or extend the Goal Question Metric Approach (GQM) [TRG16]. Among the main findings of metrics research in recent years is the need to link metrics to specific goals and to always perform metrics interpretation based on the organizational context, environment, and goals [CR94].

A central research problem of this dissertation is that agile methodologies have become increasingly popular in scaling agile environments over the past few decades [UPP+22], but researchers still need to develop an understanding of metrics in this context. Specifically, there needs to be more insight into whether metrics are relevant to practitioners or if they have only been selected as an important research topic by academics. In addition, research needs to gain basic knowledge about goals, which should always be set in connection with metrics, and about metrics' challenges and success factors. This basic knowledge could help develop scientific artifacts to support the adoption of metrics while being adapted to the unique requirements of large agile software development environments. At present, however, there are no scientific artifacts to guide the use of metrics, nor is there a comprehensive review of metrics suitable for scaling agile development environments. In addition, current research lacks a method for managing metrics in

large agile organizations and recommendations for developing measurement programs in scaling agile environments.

In this dissertation we aim to make contributions to these research gaps.

1.2. Research Questions

Subsequently, we identify four distinct research gaps and formulate our research questions based on them.

Research Gap 1: *There is a lack of research that investigates the relevance and motivations for metrics across multiple large-scale agile organizations. In detail, no study provides a comprehensive overview and analysis of reasons for using metrics, the types of metrics applied (i.e, traditional or agile metrics), metrics beyond single agile teams, and the most popular metrics among metric experts.*

Metrics have played an important role in traditional software development over decades. However, various researchers argue that the successfully applied metrics in traditional software development are not always applicable in agile settings (e.g., [HD06, OK12, PM14]). For instance, one potential conflict between measurement in traditional versus agile software development is that traditional approaches measure against a predefined plan, whereas agile approaches follow the value of embracing change (e.g., [KTS21, KMI15]). A few studies investigate metrics in agile software development regarding metric categorization (e.g., [KFW18, OK12, KMI15]) and reasons for using metrics (e.g., [KMI15]).

So far, only limited research on metrics in scaling agile environments has been conducted [UPP+22] and proving the relevance of metrics or a comprehensive overview of large-scale agile practitioners' reasons for using metrics are topics that lack sufficient investigation. Current research rather focuses on which metrics are adopted [DTHK05, KLFM19, RRO18, SC20, THDK06, SM11], using metrics to measure success [SKR+20, KR18, MSMB20, OHW+16] and team performance [CJR09, KTS21] or as reporting instrument on the portfolio level [SS18].

To fill the identified research gap, we conducted an expert interview study across multiple large-scale agile organizations. In this study, we identified reasons for using metrics and categorized them. Moreover, we investigated on which scaling levels metrics are implemented, which kind of metrics are used (i.e., agile or traditional metrics), and present the most used metrics among the expert organizations.

Research Question 1 (RQ1)

Why are metrics adopted, on which scaling levels are they implemented, which kind of metrics are used, and what are the most popular metrics across large-scale agile software development organizations?

Research Gap 2: *In the current research, there is no comprehensive analysis of challenges and success factors of adopting metrics in scaling agile environments, which is taking into account*

the viewpoints of multiple organizations. This lack leads to a deficit of fundamental knowledge needed to develop artifacts to support the adoption of metrics in large-scale agile projects.

Challenges and success factors are important data constructs since they can be used to inform the design of solutions helping practitioners to succeed in their scaling agile environments. Some initial work in large-scale agile development has been conducted to identify challenges and success factors. However, the focus of these papers lies on other topics such as agile transformations (e.g., [DPL16]), frameworks (e.g., [CC19]), or stakeholders and initiatives (e.g., [UKCM18]), but not on metrics and the challenges and success factors occurring when metrics are applied in scaling agile environments.

To fill the research gap, we conducted an expert interview study across multiple large-scale agile organizations. In this study, we identified, categorized, and ranked (i.e., according to the number of mentions) challenges and success factors of metrics applied in large-scale agile development.

Research Question 2 (RQ2)

What challenges and success factors have been reported for metrics in large-scale agile development?

Research Gap 3: *Current research lacks artifacts that can support the adoption of metrics and are adapted to the specific requirements of large-scale agile software development environments. Moreover, recent research on metrics in large-scale agile development does not use (action) design science research to build artifacts within real scaling agile contexts in collaboration with agile practitioners. More specifically, no scientific artifact exists that provide guidance on how to adopt metrics or offers a comprehensive set of metrics applicable to scaling agile development contexts. Further, current research is lacking a method for managing metrics in large-agile organizations.*

Design science intends to extend the boundaries of human and organizational capabilities by designing new and innovative artifacts [HCHC10]. However, in large-scale agile development, Uludağ et al. [UPP+22] have shown that design science as research approach is scarcely applied. So far, no artifacts exist that support the documentation and organization-specific implementation or management of metrics in large-scale agile development. Conversely, in traditional software development, research exists for both types of artifacts.

Regarding metric documentation and their organization-specific implementation, we identified best practices from literature on structured collections of metrics [BDV14, Mon14, OLP02, OHW+16], approaches to support metric adoption [HD06, Kan04], and generic metrics documentation structures [iso17, Küt11, NAK02, Par15, PS10]. However, no artifact exists that provides a standard set of metrics and goals identified in large-scale agile organizations or addresses the challenges and success factors mentioned by metric experts working in scaling agile development environments.

To address this research gap, we used the design science approach to develop two artifacts for the large-scale agile development domain: (i) a structured documentation template for metrics in the form of a minimalistic Metric Management Fact Sheet (MMFS), and (ii) a metric catalog, documenting metrics and goals found in the industry using the previously developed MMFS.

We conducted three survey-based iterations to evaluate our design proposals for the MMFS and metric catalog. Moreover, we presented the resulting MMFS and metric catalog during an expert discussion and a case study. In the case study, the catalog was applied as part of a systematic process for metrics adoption and long-term management.

Research Question 3 (RQ3)

How can a generic and structured metrics collection be designed to support large-scale agile software organizations during metric adoption?

With regard to metric management, the systematic literature review by Tahir et al. [TRG16] provides a comprehensive overview of existing measurement planning models and tools, mitigation strategies for their challenges, and success factors in implementing measurement programs. Tahir et al. [TRG16] identified various approaches for measurement in software engineering, such as the Measurement Information Model in ISO/IEC 15939:2007 [fS02], GQM approach [Bas92, CR94, BW84], GQM+Strategies [BLR+10], Goal Argument Metrics [CG07] or the Balanced Scorecard [KN92]. Various models recommended extensions or improvements to the GQM approach. Other authors provide a more technical perspective and lack an extensive exploration of the socio-technical environment in which these measurement systems operate. For instance, Staron et al. [SMN09] introduce a framework for designing measurement systems in software development, highlighting the critical role of automated metrics collection and processing in a large organization. Moreover, while the study offers useful insights into the design and operation of measurement systems, it does not fully address the complexities and unique challenges of managing metrics in large-scale agile settings.

In the domain of agile software development and large-scale agile software development, several authors use the GQM approach to identify a set of metrics serving different purposes. For example, metrics are selected to build a metrics-based reporting model [BLTV15], to establish software process measurement [RRO18], to assess the agile project quality by Kārkliņa et al. [KP18], to measure the success of agile software development among a set of Critical Success Factors [AG18], or to measure the impact of an agile transformation [HWMP13].

Further, Ram et al. [RRO18] identify success factors for operating metrics in agile software development, focusing on factors that facilitate the long-term use of metrics. According to Ram et al. [RRO18], data availability, metric trustworthiness, and development process are particularly important to a successful long-term metrics operation. An alternative approach to designing, developing, sustaining, and evolving measurement programs is the formation of a dedicated metrics team. In alignment with this perspective, Meding et al. [MSS21] introduced the metrics team maturity model, known as MeTeaM, and provided an assessment template to evaluate the effectiveness of such teams.

In summary, the approaches proposed for agile software development focus on supporting metric selection, not long-term metric operation or scaling. Moreover, the proposed approaches for traditional software development were not developed in and adapted to the circumstances of scaling agile environments.

To address this research gap, we present a goal-based method designed for a large agile case

organization to help stakeholders select, operate, and scale metrics. In addition, based on lessons learned in the case organization, we present design principles that can potentially guide the development of methods for other contexts. We conducted this research using an Action Design Research (ADR) approach in combination with Situational Method Engineering (SME).

Research Question 4 (RQ4)

How can metrics be systematically selected, operated, and scaled in large-scale agile software development environments?

Research Gap 4: *In the current literature on large-scale agile development, there is a lack of research on goals. First, there is no deep investigation of how goal-setting takes place and what documentation techniques support it. Second, there is no systematic analysis of goal-setting challenges and associated proposed solutions. Third, there is insufficient insight into concrete goals that have been set and their categories.*

In management literature, the importance of management by objectives was clarified by Peter Drucker [Gre81]. He recommended that the visions and efforts of all managers should be directed toward a common goal [Gre81]. In software development, the GQM approach [CR94] is known for assisting practitioners in setting goals and corresponding measurements for goal achievement. In addition, there are various goal categorization schemes in the literature on traditional software development that assign goals to categories and levels [CR94, BLR+10, KGW13].

There is only some initial research on goals in agile and large-scale agile development. Performance dimensions (e.g., responsiveness to change) for agile projects [KTS21] or success factors in goal-setting (e.g., working closely with customers) [LKL+18] are discussed. Regarding goal-setting in large-scale agile development, several researchers argue that goals in scaling agile environments can be divided into a hierarchy of goals. A hierarchy allows teams to pursue their goals while contributing to higher-level goals [BMS19, MDS+19, KTS21]. However, no study has examined the interplay of relevant stakeholders, events, goal-setting approaches, and documentation techniques in large-scale agile development. Moreover, the current literature only provides high-level performance goals and goal categorization schemes (e.g., [KTS21]) but lacks an overview of concrete goals and their assignment to different scaling levels. In addition, some studies occasionally highlight challenges in goal-setting [MDS+19] but lack a thorough examination of the challenges and suggestions for mitigating the challenges in goal-setting.

To address this research gap, we conduct an embedded single case study at a large German automotive manufacturer to describe how goal-setting is performed and the techniques used to document the goals. We also identify the challenges encountered and propose remedies, which are evaluated by industry experts in this study.

Research Question 5 (RQ5)

How are goals set and documented in a large agile organization, which goals exist, what challenges have been encountered during goal-setting, and how can these challenges be addressed?

1.3. Structure of the Dissertation

As Figure 1.1 illustrates, this dissertation is divided into three parts. Part A, includes the introduction (Chapter 1) to the topic, theoretical background (Chapter 2), and research design (Chapter 3). The introduction motivates the topic, defines the research problem through five research questions, and outlines the structure of this dissertation. The theoretical background provides an overview of relevant concepts, including agile software development, large-scale agile development, metrics and goals, and the GQM approach. The research design outlines the research paradigm and methods we used in the embedded publications.

Part B provides an overview of the five publications that align with the research questions of this dissertation. The first two publications (P1 and P2), explore the relevance, challenges, and success factors for metrics in large-scale agile developments. Publications P3 and P4 exploit the knowledge gained in the first two publications to develop artifacts to support practitioners of large agile organizations in adopting metrics. Publication 5 investigates goal-establishment in large-scale agile development in depth. Figure 1.1 shows the flow of argumentation between the publications. The full papers are available in this dissertation's Appendix A.

In Part C, we summarize the results of the five publications (Chapter 4) and discuss our findings in relation to current literature (Chapter 5). Moreover, we show the implications for research and practice (Chapter 6), describe the limitations (Chapter 7), and conclude the dissertation (Chapter 8). Finally, we outline future research directions (Chapter 9).

In the following paragraphs, we summarize the research problem, methodology, and main contribution of each publication belonging to Part B. Table 1.1 shows these publications' meta-data. Please note that the flow of our argumentation is designed for comprehensibility and does not reflect the actual chronological order of the studies conducted. Studies P1, P2, and P5 were conducted simultaneously. Subsequently, the implementation of study P3 took place, followed by the last realized study P4.

P1 Investigating the Adoption of Metrics in Large-Scale Agile Software Development: Metrics are well-researched in the traditional software development research domain. However, there is little research in the area of large-scale agile development, and it is unclear whether metrics are relevant to practitioners or merely highlighted by academics as a high-priority research topic. Therefore, in the first paper [PTM22b], we analyze the extent to which metrics are relevant in scaling agile software development. For this purpose, we conducted an expert interview study to determine the experts' reasons for using metrics. In addition, we examined which metric types (i.e., agile or traditional) occur at which organizational level to understand whether metrics are relevant beyond the scope of individual agile teams. Finally, we identified the most commonly used metrics in expert organizations.

P2 Challenges and Success Factors for Metrics in Large-Scale Agile Development: The second publication [PTM22a] is based on the same expert interview study as P1 and aims to identify challenges and success factors when adopting metrics. The current literature on metrics in large-scale agile development lacks foundational knowledge from practitioners about the problems encountered when introducing metrics and possible solutions. This knowledge is needed to create artifacts that can support practitioners in successfully adopting metrics.

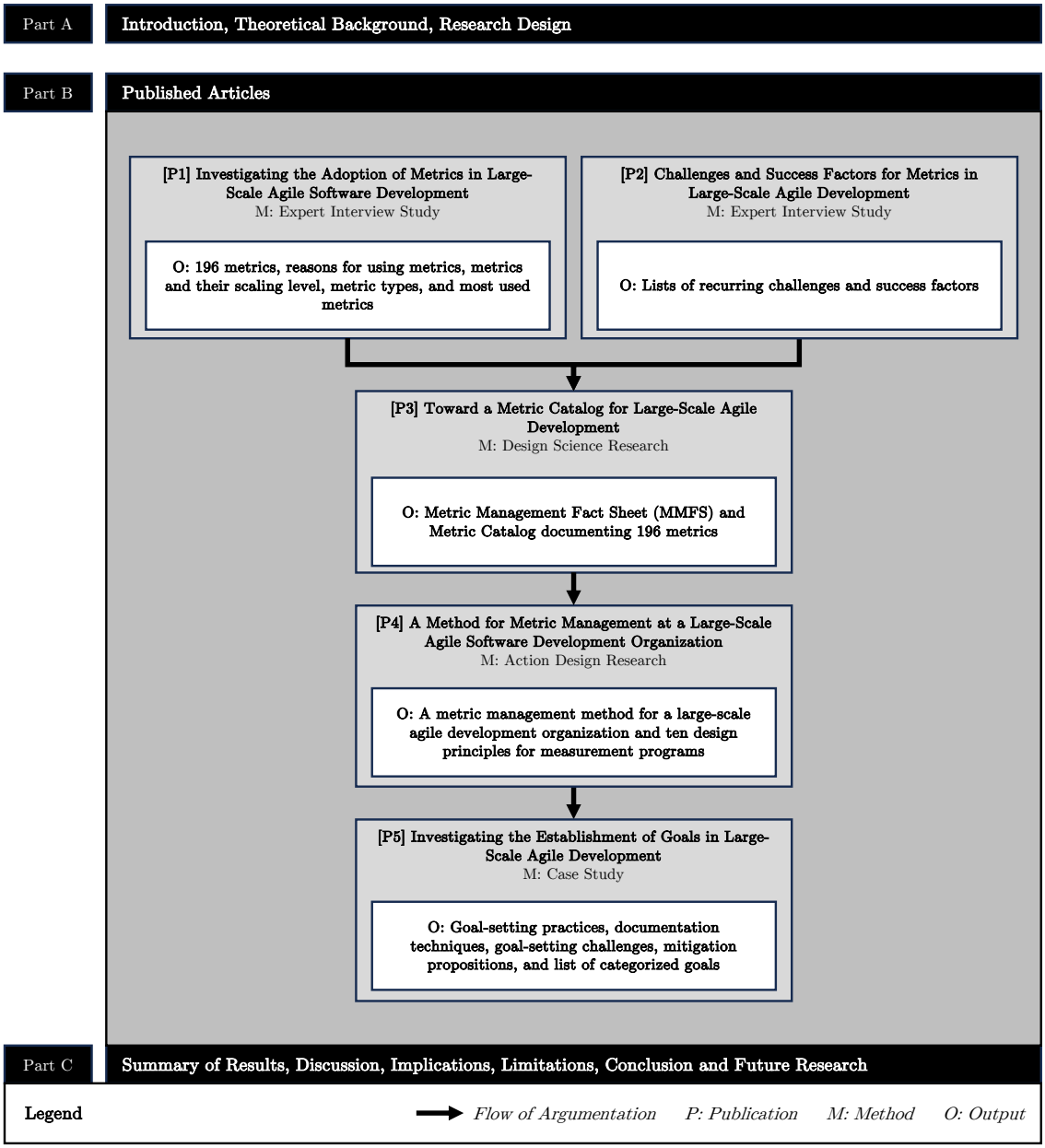


Figure 1.1.: Structure of the Dissertation

Table 1.1.: Overview of Embedded Publications

RQ	#	Authors	Title	Outlet	Type	CORE Ranking
RQ1	P1	Philipp, Tobisch, Matthes	Investigating the Adoption of Metrics in Large-Scale Agile Software Development	PACIS* 2022	CON	A
RQ2	P2	Philipp, Tobisch, Matthes	Challenges and Success Factors for Metrics in Large-Scale Agile Development	AMCIS* 2022	CON	A
RQ3	P3	Philipp, Tobisch, Menzel, Matthes	Toward a Metric Catalog for Large-Scale Agile Development	AMCIS* 2022	CON	A
RQ4	P4	Philipp, Tobisch, Menzel, Matthes	A Method for Metric Management at a Large-Scale Agile Software Development Organization	MENSURA* 2023	CON	C
RQ5	P5	Philipp, Schüll, Matthes	Investigating the Establishment of Goals in Large-Scale Agile Development	PACIS* 2022	CON	A

Outlet:		Type:	
AMCIS	Americas Conference on Information Systems	CON:	Conference
PACIS	Pacific Asia Conference on Information Systems	CORE:	Computing Research & Education
MENSURA	International Conference on Software Process and Product Measurement		

* All publications are published and peer reviewed.

Therefore, we use the expert interview study to identify, categorize, and describe challenges and success factors.

P3 Toward a Metric Catalog for Large-Scale Agile Development: Current research on large-scale agile development lacks artifacts that can help practitioners adopt metrics. Therefore, in the third paper [PTMM22], we applied design science to develop a minimalistic MMFS for documenting metrics with a minimal number of elements. The MMFS incorporates the challenges and success factors identified in **P2** and serves as the foundation for the metric catalog, our second artifact, to document a standard set of 196 metrics identified in our expert interview study **P1**. In addition, the catalog provides navigational aids to find metrics more efficiently. We evaluated both artifacts in several rounds of evaluation. The evaluation included a case study in which the metric catalog was used as a part of the first version of our goal-based metric management method in two case organizations **P4**.

P4 A Method for Managing Metrics in a Large-Scale Agile Software Development Organization: The fourth paper [PTMM23] develops a method for selecting, operating, and scaling metrics in a large-scale agile development organization. This method was developed using action design research over two years in two case organizations. We addressed some of the challenges and success factors identified in **P2** as we developed this method. In addition, the method draws on a digital version of the metric catalog from **P3** and insight gained in **P5** regarding goal-establishment in large-scale agile development.

P5 Investigating the Establishment of Goals in Large-Scale Agile Development: The fifth publication [PSM22] comprises an in-depth analysis on goal-setting in the context of scaled agile software development. Utilizing a single-case embedded study at a large German automotive manufacturer, we describe how goal-setting is performed and which techniques are used to document the goals. We also identify challenges encountered and propose mitigation strategies, which are evaluated by industry experts participating in this study.

Table 1.2.: Overview of Additional Publications

Authors	Title	Outlet	Type	CORE Ranking
Uludağ, Philipp, Putta, Paasivaara, Lassenius, Matthes	Revealing the state of the art of large-scale agile development research: A systematic mapping study	JSS* 2022	JNL	A
Schüll, Hofmann, Philipp, Urbach	Reporting in large-scale agile organizations: insights and recommendations from a case study in software development	ISeB* 2023	JNL	C
Outlet:		**Type:**		
JSS	Journal of Systems and Software	JNL	Journal	
ISeB	Information Systems and e-Business Management	CORE	Computing Research & Education	
* All publications are published and peer reviewed.				

In addition to the key publications P1-P5, we list two publications led by co-authors. Both publications are related to large-scale agile development (see Table 1.2).

CHAPTER 2

Theoretical Background

In this chapter, we introduce the definitions and explain core concepts relevant to this dissertation. First, we review the topic of agile software development. Second, we provide an overview of large-scale agile development. Third, we introduce relevant concepts related to goals and metrics in software development.

2.1. Agile Software Development

Traditional software development is increasingly reaching its limits given the growing demand for agility [HC01, Hig02]. Its severe weaknesses, such as irregular releases with large batch sizes, lack of customer feedback, and rigid processes, acutely constrain organizations [JGZag+13, PW09]. These shortcomings of traditional software development have led to the success of agile software development [JGZag+13, PW09]. Nowadays, the application of agile methods is the predominant way of software development [DM14]. Agile methods are based on the values and principles summarized in the Agile Manifesto [BCS+10, Sch18], which was published in 2001 to provide "an alternative to documentation-driven, heavyweight software development processes" [BBVB+01]. Currently, Scrum is the most widely used agile approach in practice [Dig20]. The benefits of agile methods, such as responsiveness to fluctuating customer requirements, flexibility, and resilience in changing environments, are undisputed [Dig20, KB12]. However, according to Kruchten [Kru13], the success of agile methods depends on the context in which they are used. Implementing agile methods out of the box is likely to be successful in the agile sweet spot, a context that is identical or similar to the context in which agile methods were originally developed [Kru13].

2.1.1. Agile Manifesto and Definitions

The agile movement was triggered by the creation of the Manifesto for Agile Software Development in 2001 [ASRW02]. Seventeen professionals gathered to set up a document consisting of four values and twelve principles to enhance how software development was conducted at that time [HKvB+18]. The four values are [BBVB+01]:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

The creators of the manifesto formulated the four values together with an important note. While items on the right are valuable, they value the items on the left more [BBVB+01]. The formulations of the values are tersely ambiguous. Thus, there is room for interpretation. Below, we present Abrahamsson et al. [ASRW02] interpretations for each value:

- The first value emphasizes the relationship and communality between software developers and the human aspect of contracts instead of focusing on processes and tools. This value leads to close team relationships and tight working environment arrangements.
- The second value emphasizes that software teams should continuously aim to produce working software. Therefore, software releases happen frequently. The release intervals can range from hours to months, depending on the context. Further, software teams must focus on code simplicity and technical excellence to reduce the documentation effort to its required minimum.
- The third value embodies a clear preference for customer collaboration and maintaining a good relationship instead of strict contracts. However, the need for well-drafted contracts grows with the increasing size of software projects. Nevertheless, the contract negotiation process can serve as an opportunity to achieve and maintain a viable relationship.
- The fourth value highlights that the development group, comprising both software developers and customer representatives, should be well-informed, competent, and authorized to consider possible adjustment needs that may emerge during the development process life-cycle. Therefore, the participants are prepared to make changes, and existing contracts are formed with tools that allow these enhancements to be conducted.

In addition to the four values, twelve principles were formulated [BBVB+01] (see Table 2.1).

Over the years, the Agile Manifesto was criticized [LSA13]. For instance, Coplien [LSA13] suggests replacing the term “working software” with “usable software” which is part of the second value and the principles P3 and P7. Another critique is that the manifesto cannot be used as a basis for scientific work since it is too vague and lacks a proper grounding in management theory and philosophy [CF04].

The scientific literature has yet to reach a consensus in defining agile and agility. Typically,

Table 2.1.: Overview of Agile Principles [BBVB+01]

ID	Principle
P1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
P2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
P3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
P4	Business people and developers must work together daily throughout the project.
P5	Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
P6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
P7	Working software is the primary measure of progress.
P8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
P9	Continuous attention to technical excellence and good design enhances agility.
P10	Simplicity--the art of maximizing the amount of work not done--is essential.
P11	The best architectures, requirements, and designs emerge from self-organizing teams.
P12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

the Agile Manifesto is used to define these concepts [LSA13]. Regardless, multiple researchers proposed definitions that adhere to the values and principles of the Agile Manifesto to a differing degree [LSA13, Ket09, DNBM12]. Lanti et al. [LSA13] provide an analysis of how different definitions, identified by [Ket09], emphasize key ideas from the Agile Manifesto (see Table 2.2).

2.1.2. Contextualizing Agile Software Development

Kruchten [Kru13] stresses the importance of viewing agile software development methods in light of their context. He clarifies which types of contexts exist in a corporate setting and shows how different contexts relate. Further, he introduces the agile sweet spot, a context in which the odds for a successful application of agile software development are high [Kru13]. His ideas will be explained subsequently.

Context can be divided into two sets of factors (see Figure 2.1). The first set of factors is shown on the left side of the figure. This set includes culture, business domain, number of instances, maturity of organization, and level of innovation. These factors apply to the entire organization. The right side of the figure shows factors that apply to single projects. This set includes size, criticality, business model, stable architecture, team distribution, governance, rate of change, and age of the system. The interested reader can access a more detailed description of each

Table 2.2.: Overview of Agile Definitions [LSA13]

Source	Definition of agile	Emphasis of the corresponding definition
Anderson 2003	Ability to expedite.	Speed
Larman 2003	Rapid and flexible response to change.	Speed, flexibility, responsiveness
Schuh 2004	Building software by empowering and trusting people. Acknowledging change as a norm and promoting constant feedback. Producing more valuable functionality faster.	People, empowerment, change, feedback, value, speed
Lyytinen 2006	Discovery and adoption of multiple types of Information Systems Development innovations through garnering and utilizing agile sensing and response capabilities.	Delivery, innovations, responsiveness
Subramaniam 2005	Uses feedback to make constant adjustments in a highly collaborative environment.	Feedback, adaptability, collaboration
Ambler 2007	Iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams with "just enough" ceremony that produces high-quality software in a cost-effective and timely manner which meets the changing needs of its stakeholders.	Iterative, incremental, self-organizing, less process-driven, collaborative, cost-conscious, speed, customer-driven
Nerur and Balijepally 2007	Define Agile software development via strategic thinking (of uncertainty), holographic organization theory, "emergent metaphor of design" and Agile Methods as people-centric, competent people and their relationships, high customer satisfaction through quick delivery of quality software, active participation of concerned stakeholders; creating and leveraging change. Evolutionary delivery through short iterative cycles, intense collaboration, self-organizing teams and high degree of developer discretion. Learning, teamwork, self-organization and personal empowerment. Responsiveness and flexibility. Interchangeability of roles and jobs based on autonomy.	Strategic thinking, uncertainty, chaos theories, holographic organization, non-traditional, emergent design, people-centric, competent people and their relationships, high customer satisfaction, quick delivery, active participation, creating and leveraging change, short iterative cycles, intense collaboration, self-organizing teams, developer discretion, learning, teamwork, self-organization, personal empowerment, responsiveness, flexibility, heterarchy, role interchangeability and autonomy.
IEEE 2007	Capability to accommodate uncertain or changing needs up to a late stage of the development (until the start of the last iterative development cycle of the release).	Iterative, responsive
Wikipedia 2007	Conceptual framework for software engineering that promotes development iterations throughout the lifecycle of the project.	Iterative, conceptual framework
Cockburn 2001	Being effective and maneuverable. Use of light-but-sufficient rules of project behavior and the use of human and communication-oriented rules.	Effective, steerable, rule-based, people, communication

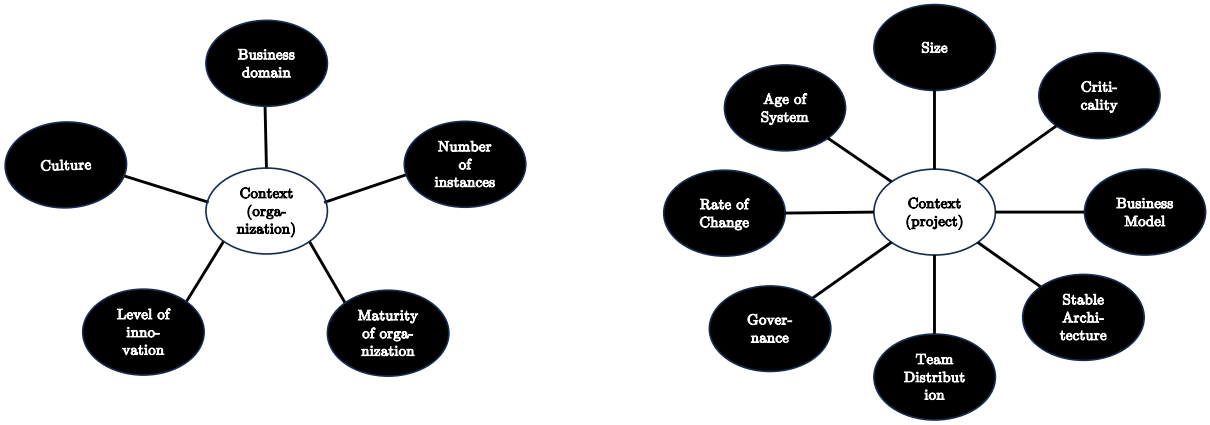


Figure 2.1.: Factors Related to the Organization (Left) and Project Context (Right) [Kru13]

factor in the original publication [Kru13]. Both sets relate to each other since the factors that apply to the entire organization are driving and constraining the single project factors. For instance, if the business domain is software for aerospace onboarding, this precondition will influence the project-context factors (e.g., criticality, size, business model).

Below, the agile sweet spot is defined by assigning values to the project-context factors [Kru13]. The bold values within the brackets are typical conditions for the agile sweet spot. Agile methods originated from these conditions and are most likely to succeed under such circumstances [Kru13].

- Size: 0 .. **[12 people]** .. 300
- Stable architecture: **[Stable]**, changed, new
- Business model: **[In house]**, Open source, ...
- Team distribution: **[Collocated]**, .., .., offshore outsource
- Rate of change: Low, **[medium]**, **[high]**
- Age of system: Exploratory, **[greenfield]**, legacy maintenance
- Criticality: **[Simple]**, **[$ losses]**, deaths
- Governance: **[Simple rules]**, .., sOX, ...

Understanding the agile sweet spot is crucial when scaling agile methods since characteristics of large-scale agile development environments deviate. Therefore, practitioners can not expect agile methods to succeed in contexts for which they were not created without adaptations.

2.1.3. Comparison of Agile Methods

Many agile methods have been proposed over the years [DNBM12]. These methods share some common characteristics, such as an iterative development process focusing on interaction, com-

Table 2.3.: Overview of Agile Methods [ASRW02]

Method name	Key points	Special features
Adaptive Software Development	Adaptive culture, collaboration, mission-driven component based iterative development.	Organizations are seen as adaptive systems. Creating an emergent order out of a web of interconnected individuals.
Agile Modeling	Applying agile principles to modeling: Agile culture, work organization to support communication, simplicity.	Agile thinking applies to modeling also.
Crystal	Family of methods. Each has the same underlying core values and principles. Techniques, roles, tools and standards vary.	Method design principles. Ability to select the most suitable method based on project size and criticality.
Dynamic systems development method	Application of controls to RAD, use of timeboxing, empowered DSDM teams, active consortium to steer the method development.	First truly agile software development method, use of prototyping, several user roles: "ambassador", "visionary" and "advisor".
Extreme programming	Customer driven development, small teams, daily builds.	Refactoring – the ongoing redesign of the system to improve its performance and responsiveness to change.
Feature Driven Development	Five-step process, object-oriented component (i.e., feature) based development. Very short iterations: from hours to 2 weeks.	Method simplicity, design and implement the system by features, object modeling.
Open Source Software	Volunteer based, distributed development, often the problem domain is more of a challenge than a commercial undertaking.	Licensing practice; source code freely available to all parties.
Pragmatic Programming	Emphasis on pragmatism, theory of programming is of less importance, high level of automation in all aspects of programming.	Concrete and empirically validated tips and hints, i.e., a pragmatic approach to software development.
Rational Unified Process	Complete SW development model including tool support. Activity driven role assignment.	Business modeling, tool family support.
Scrum	Independent, small, self-organizing development teams, 30-day release cycles.	Enforce a paradigm shift from the "defined and repeatable" to the "new product development view of Scrum."

munication, and reducing costly intermediate artifacts [CC04]. Abrahamsson et al. [ASRW02] analyzed agile methods, their key points, and special features (see Table 2.3).

2.1.4. Scrum

The agile methods differ regarding their relevance for practitioners. According to the non-scientific but popular Version One State of Agile Report, Scrum is the leading agile method used by 87% in 2020 of respondents compared to 58% in 2019 [Ver20]. Table 2.4 shows that Scrum is also the underlying method of many scaling agile frameworks. Due to Scrum's significance to agile and large-scale agile, we will introduce the method below based on the 2020 Scrum Guide [SS20].

Scrum is defined as a "lightweight framework that helps people, teams and organizations generate

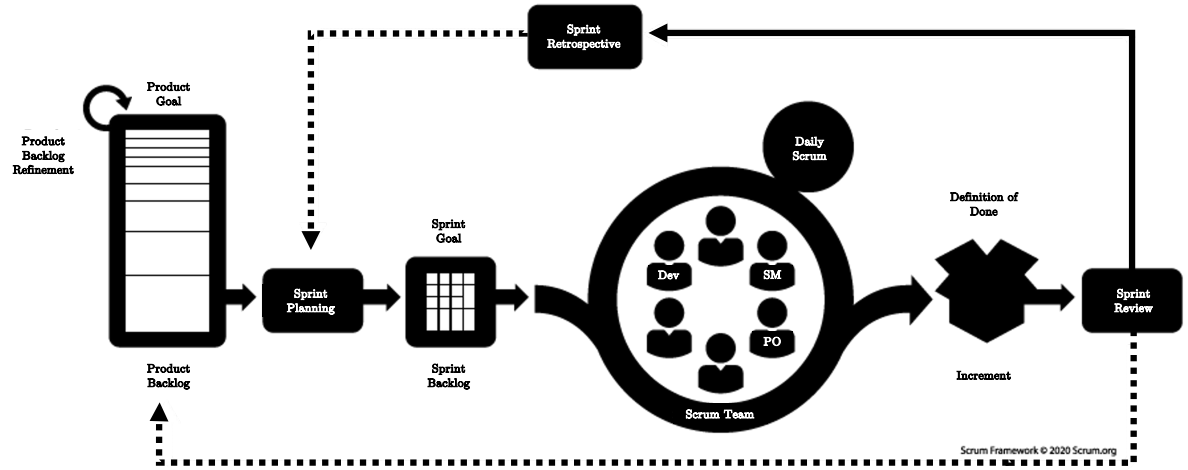


Figure 2.2.: Overview of the Scrum Framework [Scr22]

value through adaptive solutions for complex problems" [SS20]. The framework comprises the Scrum Team, Scrum Events, and Scrum Artifacts.

Scrum Team:

The Scrum Team is a cross-functional, autonomous, and cohesive unit consisting of Developers, one Product Owner, and one Scrum Master. Typically, the Scrum Team consists of fewer than ten people. At each point, the Scrum Team focuses on a single objective, the Product Goal. Within the Scrum Team, no sub-teams or hierarchies exist. However, the Scrum Team can reorganize into multiple separate teams. The Scrum Team takes responsibility for all product-related tasks and is accountable for creating a valuable Increment each Sprint. The Developers commit to creating aspects that are part of the usable Increment. The responsibilities of the developers include creating a Sprint Backlog, adhering to the Definition of Done, adapting the plan to meet the Sprint Goal, and holding each other accountable. The Product Owner is responsible for maximizing the product's value resulting from the Scrum Team's work. The primary responsibility of the Product Owner is effective Product Backlog management. The Scrum Master is responsible for the Scrum Team's effectiveness and ensures that the Scrum Team adheres to the rules of the Scrum Guide. Therefore, they act as educators, equipping everyone within the Scrum Team and the organization with the required knowledge.

Scrum Events:

Sprints are recurring events of fixed length (i.e., less than one month) that transform ideas into value by performing the work necessary to achieve the Product Goal. The Sprint Event contains all other events, namely the Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective. During each Sprint, it is ensured that no changes that endanger the Sprint Goal are made, quality is sustained, the Product Backlog is refined when necessary, and with

increasing knowledge, the scope is clarified and renegotiated with the Product Owner. An increasing Sprint length may lead to the Sprint Goal becoming invalid, a rise in complexity, and an increase in risk. Contrarily, shorter sprint cycles increase the number of learning cycles and limit the risk of cost and effort to a smaller time frame. In case the Sprint Goal becomes obsolete, only the Product Owner can cancel the Sprint. During the Sprint Planning Event, the Scrum Team comes together to collaboratively define the work to be accomplished during the Sprint. This is the first event during a Sprint. For the event to be effective, the Product Owner facilitates the discussion of the most critical Product Backlog items and how they are aligned with the Product Goal. The output of the Sprint Planning Event is a Sprint Backlog that incorporates the Sprint Goal, the selected Product Backlog items, and a plan on how to deliver these items. The Daily Scrum event is performed to inspect progress toward the Sprint Goal, adapt the Sprint Backlog whenever necessary, and define an actionable plan for the next day. During the Sprint Review, the Scrum Team reflects on the Sprint outcome and evaluates future adaptations. The event is a means to present the accomplished work to key stakeholders, discuss the progress towards the Product Goal, and possibly adjust the Product Backlog. The Sprint Retrospective is performed to identify and plan the most impactful actions to increase quality and effectiveness. Areas worth assessing for improvement potential are individuals, interactions, processes, tools, and the Definition of Done. The Scrum Team discusses what worked during the Sprint, what problems occurred, and how problems were solved or not solved. The Sprint Retrospective finishes the Sprint and is the event before the Sprint Planning event.

Scrum Artifacts

The Scrum Artifacts include the Product Backlog, Sprint Backlog, and Increment. Each artifact is used to maximize the transparency of relevant information. Therefore, these artifacts provide a common basis for inspection and adaptation. Each artifact has a commitment. The Product Goal is the commitment of the Product Backlog. Respectively, the Sprint Goal is the commitment of the Sprint Backlog. The Definition of Done serves as a commitment for the Increment. The Product Backlog is the single source of items required to improve the product. These items emerge over time and are ordered. Product Backlog refinement is the process of making the items ready to be pulled by the Scrum Team to accomplish them during a Sprint. This is achieved through resizing, adding details, or changing the order. The sizing is done by the Developers. The Product Goal describes an admirable future state of the product. It is used for planning purposes and to prioritize the Product Backlog items. The Sprint Backlog contains the Sprint Goal, the items that were selected for the current sprint from the Product Backlog, and an actionable plan to deliver the Increment. The Sprint Backlog can be seen as a plan for the developers created by themselves. The Sprint Goal is the single objective for the Sprint. The Increment contributes towards achieving the Product Goal. Each Increment must be verified since it is an addition to all prior Increments and has to work together with them. Further, an Increment must be usable in order to be valuable. Multiple Increments may be created during a Sprint. Work can only be part of an Increment if it adheres to the Definition of Done. The Definition of Done is a description of the future state of the product that meets all quality requirements belonging to the product.

2.2. Large-Scale Agile Software Development

Inspired by the success of agile methods at a small scale, an increasing number of large organizations have begun to scale agile practices to larger contexts that depart significantly from the agile sweet spot [Dig20, DPL16, DMFS18]. Scaling agile methods is referred to as large-scale agile software development [DM14, RDFS16]. Several large-scale agile practices and frameworks, such as the Scaled Agile Framework (SAFe) [Lef18] and Large Enterprise Scrum [LV16], have been developed to support organizations adopting agile approaches at scale [DPL16, NMM05].

2.2.1. Definition

The term "large-scale agile development" has been used in a great variety of contexts, including large teams and large multi-team settings, but also when agile methods are applied within the entire organization [DMFS18]. Several researchers proposed definitions for large-scale agile development [UPP+22]. Usually, these definitions are based on recurring characteristics such as project costs, time, lines of code, or the number of requirements [DFI14]. Dingsøyr et al. [DFI14] showed that these characteristics are ineffective as part of a definition since they are unreliable predictors of scale. More recently, Uludağ et al. synthesized the prevailing definitions and suggested defining large-scale agile development as the "adoption of agile methods in large agile multi-team settings with at least two teams or the large-scale adoption of agile methods on the organizational level comprising multiple large agile multi-team settings" [UPP+22]. In this dissertation, we follow this definition.

2.2.2. Comparison of Scaling Agile Frameworks

Theobald et al. extracted twelve scaling agile frameworks to conduct a comparative review [DST18]. In the first step, an overview of these frameworks is provided and a comparison based on the frameworks' flexibility and underlying agile method is conducted (see Table 2.4). While no clear trend regarding the flexibility is recognizable, Scrum as underlying method is used across all frameworks. In the second step, the practices of all frameworks were identified and assigned to broader categories. For example, the category Scaled Backlog contains the practices Program Backlog, Sync Backlog, Portfolio Backlog, Nexus Sprint Backlog [TSD19]. Subsequently, the occurrences of these categories across the twelve frameworks were identified (see Table 2.5). The comparison shows that the team level Scrum practices (e.g., Sprint Planning) and their scaled version (e.g., Scaled Planning) are predominant across the frameworks. Further, some practices (e.g., Architectural Runway) only occur in few frameworks.

Table 2.4.: Overview of Scaling Agile Frameworks [DST18]

Frameworks	Degree of flexibility	Underlying agile method
Scrum-of-Scrums	High	Scrum
Large Scale Scrum	Medium	Scrum
Scaled Agile Framework	Low	Scrum / Kanban / Lean, specific XP practices "mandated"
Nexus / Scaled Professional Scrum	Medium	Scrum
Disciplined Agile Delivery	Medium	Scrum / Lean, mixed set of methods
Spotify "model"	High	Scrum/Kanban, Focus on Culture
Dynamic Systems Development Method	Medium	Scrum
Scrum at Scale	Medium	Scrum
Agile Scaling Cycle	High	Scrum / Lean
Agility Path	High	Scrum
FAST Agile Scaled Technology	Medium	Scrum / XP / Kaizen / specific Practices like Story Mapping
Recipes for Agile Governance	High	Scrum / Lean

Table 2.5.: Overview of Practices Across Scaling Agile Frameworks [TSD19]

#	Practice	#	Practice	#	Practice
11	Sprint Planning	4	Increment	1	Manage Impediments
11	Sprint	4	Scaled Review	1	Scaling Impediments Management
10	Retrospective	4	Strategic Activities	1	Architectural Runway
10	Review/“Demo”	4	Estimation	1	Architecture Envisioning
9	Daily Stand-Up	3	Agile Release Train	1	Internal Open Source
8	Product Backlog	3	Release Planning	1	Delivery/Release Plan
7	Definition of Done	3	Scaled Retrospective	1	Release Handoffs
6	Scrum of Scrums	2	Prioritization	1	Product Deploy Validation
6	Sprint Backlog	2	Transition Backlog	1	Release Review
6	Backlog Refinement	2	Scaled Backlog Refinement	1	Beta Codex
5	Scaled Planning	2	Collective Ownership	1	Facilitated Workshop
5	Scaled (Sprint) Goal	2	Portfolio/Program Kanban Board	1	Flex-Teaming
5	Requirements Documentation	1	Product Owner Sync	1	Initiative Assessment
4	Scaled Backlog	1	Timeboxing	1	Benefits Assessment
4	Community of Practice	1	Release Map		

2.3. Goals and Metrics in Software Development

In the following, we introduce definitions for metrics and goals. Subsequently, we describe the GQM approach, which serves as the basis of our metrics management methodology (RQ4).

2.3.1. Definitions

Measurements and Metrics in Software Development: A metric can be defined as "a method of quantitatively determining the extent to which a software process, product, or project possesses a certain attribute" [Das92]. According to Daskalantonakis [Das92], a metric does not only include the formula used to derive a metric value but also the diagram to present the metric values and the guidelines to interpret the metric values within a specific context.

Goals in Software Development: We use the definition for goals proposed in the Goal-Based Requirements Analysis Method (GBRAM). This method aims at identifying and refining goals into operational requirements for software-based information systems [Ant97]. In GBRAM, goals are defined as "targets for achievement which provide a framework for the desired system. Goals are high-level objectives of the business, organization, or system. They express the rationale for proposed systems and guide decisions at various levels within the enterprise".

2.3.2. Goal Question Metric Approach

The GQM approach proposed by Basili et al. [CR94] assumes that each organization that wants to measure purposefully must first specify the goals for itself and its projects. Subsequently, these goals must be traced to data that define the goals operationally. Finally, a framework has to be established to interpret the data with respect to the stated goals.

Applying the GQM approach results in a model consisting of three levels. On the first level, the conceptual level, goals are defined for an object. Objects of measurement can be Products, Processes, or Resources. Products are artifacts resulting from the system lifecycle (e.g., programs). Processes are activities related to software that are typically associated with time (e.g., testing). Processes use Resources to produce outputs (e.g., personnel). On the second level, the operational level, a set of questions is derived to characterize how the assessment or achievement of a specific goal is performed using a characterizing model. These questions select a quality issue of an object and determine its quality from a viewpoint. On the third level, the quantitative level, metrics (data) are assigned to each question to provide a quantitative answer. The assigned metric can be either objective or subjective. Objectivity is given when the data only depends on the object to be measured and not a specific viewpoint (e.g., staff hours spent on a task). Subjectivity is given if the data depends on the object and viewpoint (e.g., level of user satisfaction).

The GQM approach defines a hierarchical structure of goals, questions, and metrics. On the highest level, the goals specify the measurement purpose, the object and issue to be measured, and the viewpoint from which the measurement is conducted. Each goal is refined into several questions, which are, in turn, refined into metrics. Moreover, it is possible that a single metric

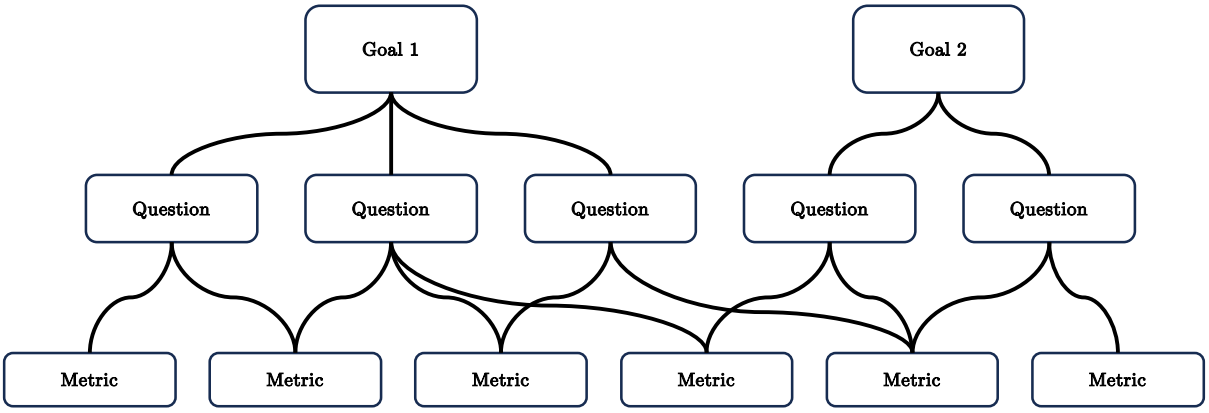


Figure 2.3.: Overview of the GQM Approach [CR94]

is used to answer multiple questions belonging to the same goal. It is also possible that multiple GQM models have questions and metrics in common.

CHAPTER 3

Research Design

In this chapter, we describe the research design, outlining the research paradigm and methods we used in the embedded publications.

3.1. Research Paradigm

For a robust research design, it is obligatory to select a research paradigm that is congruent with the researchers' beliefs about the nature of reality [Lev13, MBF06]. In research, the term paradigm represents a "system of ideas, or world view, used by a community of researchers to generate knowledge. It is a set of assumptions, research strategies and criteria for rigour that are shared, even taken for granted, by that community" ([FHMD02], as cited in [Lev13]). Commonly, a research paradigm is described by an ontology that defines what the nature of reality is, an epistemology that defines the nature of the relationship between the inquirer and the known, and a methodology that defines how the inquirer should go about gaining knowledge of the world [Gub90].

In this dissertation, we have chosen the interpretivist paradigm. This paradigm is based on a relativistic ontology with a subjectivist epistemology [Lev13]. A relativist ontology embodies the belief that reality is a finite and subjective experience [DL05, Lev13]. Subjectivism holds the view that knowledge is "always filtered through the lenses of language, gender, social class, race, and ethnicity" [DL05].

Table 3.1.: Overview of Embedded Publications with Applied Research Methods

#	Title	CS	EI	DSR	ADR
P1	Investigating the Adoption of Metrics in Large-Scale Agile Software Development		●		
P2	Challenges and Success Factors for Metrics in Large-Scale Agile Development		●		
P3	Toward a Metric Catalog for Large-Scale Agile Development	○	○	●	
P4	A Method for Metric Management at a Large-Scale Agile Software Development Organization	○			●
P5	Investigating the Establishment of Goals in Large-Scale Agile Development	●			

Legend:

● Primary method used in publication — **CS:** Case Study — **DSR:** Design Science Research

○ Secondary method used in publication — **EI:** Expert Interviews — **ADR:** Action Design Research

3.2. Research Methods

Several methodologies were used in the publications embedded in this dissertation (see Table 3.1). Whereas the methodologies in P1, P2, and P5 are qualitative in nature, we used mainly design-oriented methodologies in P3 and P4. Sometimes, it was necessary to combine several methodologies within a single publication (cf. P3 and P4). Whenever this was the case, we classified the methods used into primary and secondary methods. In the following sections, we characterize the methods used and describe the guidelines we followed during their implementation to ensure high rigor and relevance.

3.2.1. Case Study Research

Yin [Yin18] defines case studies in terms of the scope and features of a study. Regarding scope, a case study is an empirical method that aims at "investigating contemporary phenomenon in their context" [RH09], especially when "the boundaries between phenomenon and context may not be clearly evident" [Yin18]. In addition, case study research is characterized by the following features. A case study

- copes with the technically distinct situation of many more variables than data points (i.e., cases)
- and as one result can benefit from existing theories to guide design, data collection, and analysis and as another result
- requires multiple sources of evidence, where the data collection must rely on triangulation.

Yin [Yin18] suggests adhering to a six-step process in conducting case study research (see Figure 3.1).

During the planning activity, the researchers determine if case study research is suitable as a research methodology for their planned research endeavor. Choosing case study research instead of another research methodology is appropriate if (1) the research questions are "how" and "why" questions, (2) the researchers cannot control behavioral events, and (3) the research is focused on a contemporary phenomenon instead of being entirely historical.

During the design activity, the researchers (1) identify the case(s) they will study, (2) develop theory, propositions, and related issues to guide the case study and generalize its results, (3) identify the case study's design, and (4) test the identified design against internal validity, external validity, construct validity, and reliability. The research design connects the research questions and conclusions of the study with the data to be collected. The researchers can choose from four case study designs (see Figure 3.2). The first pair of the 2x2 matrix shows single and multiple case study designs. The second pair of the matrix differentiates between holistic and embedded designs. We used single embedded case studies in P4 and P5.

During the preparation activity, the researchers (1) will obtain institutional approval regarding the case study procedures to protect the human subjects participating in the case study. Subsequently, (2) the entire case study team will undergo a training. (3) The candidate cases will be screened for suitability for the case study. (4) A pilot case study is conducted. The training involves developing a case study protocol that guides the data collection.

During the collection activity, the data can stem from six different sources: interviews, direct observations, participant observation, documents, archival records, or physical artifacts. The different data types require the researchers to conduct interviews with the same participant on multiple occasions or to make field observations. Moreover, four principles are essential to any data collection effort as part of case study research. (1) Multiple sources of evidence must be included, (2) a case study database must be established, (3) a chain of evidence must be maintained and (4) a high level of sensitivity when including social media must be maintained. Adhering to the principles will significantly increase the quality of the case study.

During the analysis activity, the researchers can begin by gaining an initial understanding of the data while searching for promising patterns, insights or concepts. Other starting points for the analysis are the following general strategies: (1) leveraging theoretical propositions, (2) working through the data from the "ground up," (3) creating a case description, or (4) examining rival explanations. Each of these strategies can be adapted using the five techniques for analyzing case studies: (i) cross-case synthesis, (ii) logic models, (iii) time-series analysis, (iv) explanation building, and (v) pattern matching. Throughout the case study, it is crucial to consider the entire collected evidence, investigate alternative interpretations, address the most significant aspects of the case study, and show an understanding of prevailing thinking and literature regarding the investigated topic.

During the reporting activity, the researchers aim to share the conclusions from the case study by bringing its results and findings to closure. The following steps contribute to this goal. The researchers (1) identify the audience, (2) define the case study's format, (3) and let their drafts be reviewed by others. In addition, six compositional structures can be considered: (i) comparative, (ii) chronological, (iii) theory-building, (iv) linear-analytic, (v) "suspense," and (vi) unsequenced structures. Each of these structures can shape the entire substantive composition. Moreover,

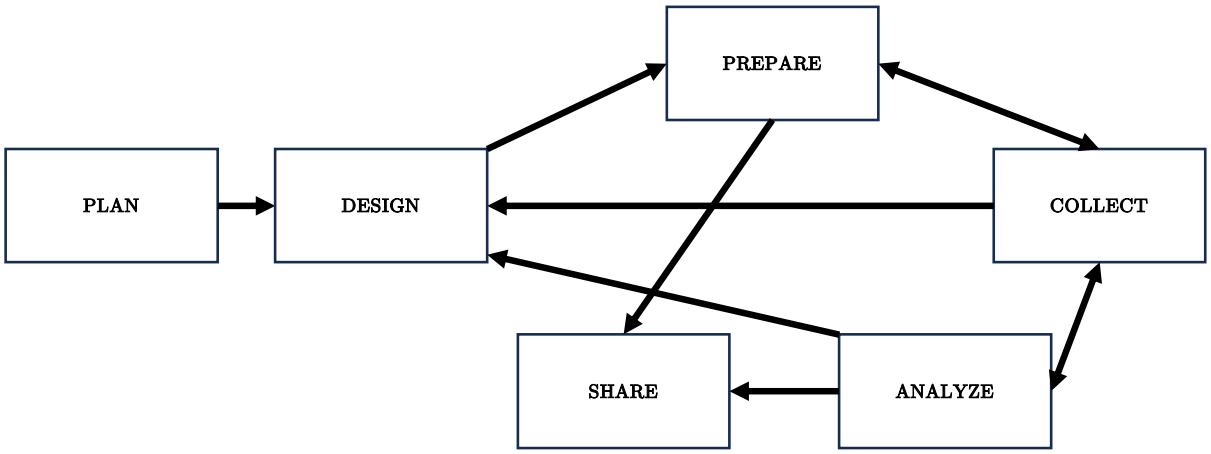


Figure 3.1.: Overview of Case Study Process [Yin18]

the case study methodology must be described. The final steps of the case study research aim at deciding how and when to start composing, choices regarding the disclosure or anonymity of the case subjects, and reviews of the draft case study.

3.2.2. Semi-Structured Expert Interviews

Semi-structured interviews mix open-ended with specific questions to collect not only information foreseen but also unexpected types of information [Sea99]. As opposed to structured interviews where the interviewer is in complete control of the questions asked, during semi-structured interviews, in addition to answers, questions can also arise from the interviewee [LG85]. In the following, we describe seven guidelines (see Figure 3.3) for qualitative interviews provided by Myers and Newman [MN07].

1. Situating the researchers as actors. The researchers should begin by "situating" themselves and the interviewee. Situating can be accomplished by asking questions regarding identity, role, experience, and background. Situating is necessary since the interview is a social encounter, and the data gathered is idiographic. The researchers should consider documenting the answers since they can help readers assess the study's validity.

2. Minimize social dissonance. This involves tactics to make the interviewee feel as comfortable as possible, which can positively impact the disclosure quality. For example, the researchers could manage first impressions, use the appropriate language and jargon, or dress appropriately.

3. Represent various "voices". Qualitative research usually requires interviewing a variety of people within an organization to overcome various biases (e.g., elite bias). This process of identifying different interviewees is termed "triangulation of subjects" [RR11], which aims to avoid only one voice emerging. As best practice, the researchers could use a table of interviewees, including their organizational position.

4. Everyone is an interpreter. This guideline sensitizes the interviewer and interviewees as creative interpreters of their worlds. Moreover, the interview is an artificial and rare event for

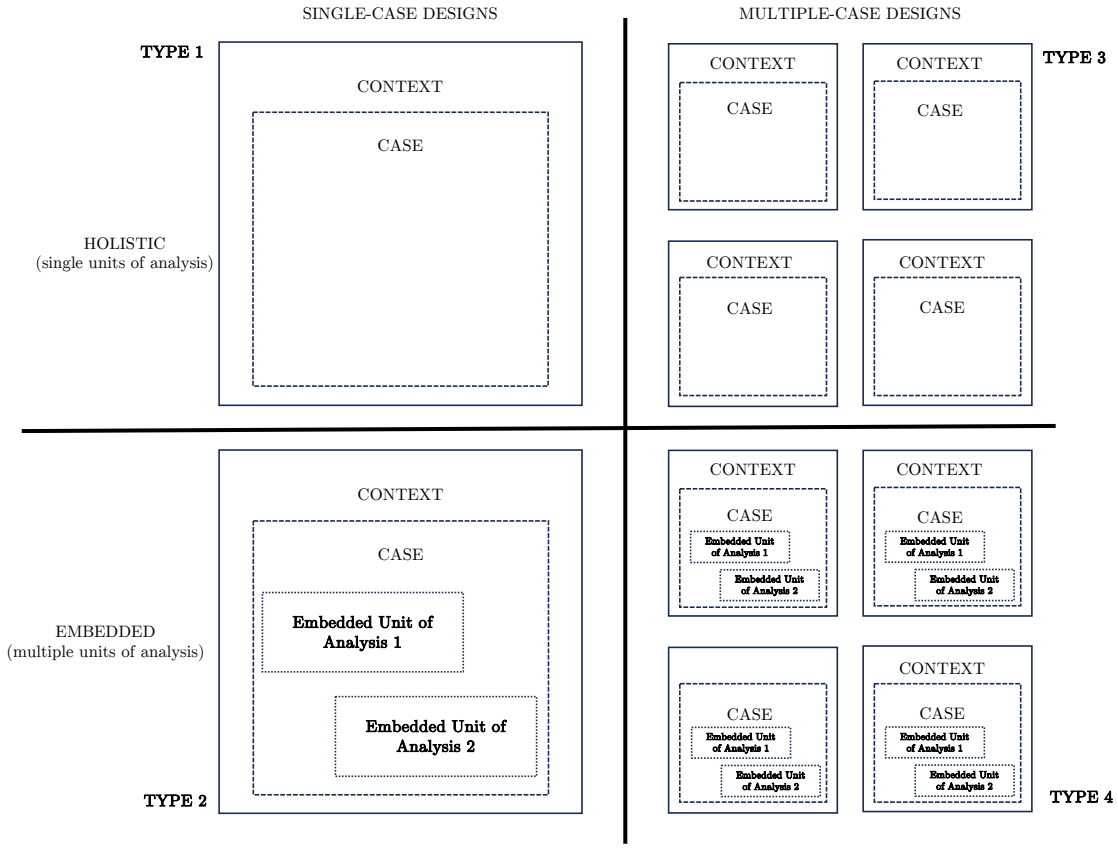


Figure 3.2.: Overview of Case Study Designs [Yin18]

most interviewees. The interview creates one or several texts, including the interview transcript as the first text.

5. Use Mirroring in questions and answers. Mirroring is the practice of taking the words and phrases of the interviewees into account when forming a subsequent question or answer. Thus, the researchers focus on the interviewees' world instead of forcing their own terminology and worldview on the subjects. Usually, it is good practice to use open-ended questions and to move from the general to the specific.

6. Flexibility. Mixing semi-structured and unstructured interview techniques leaves the interviewer with a minimal script, which necessitates a certain degree of flexibility, openness, and improvisation. This is also true for the interviewee, who usually does not have a script.

7. Confidentiality of disclosures. The final guideline highlights the importance of maintaining the confidentiality and security of transcripts, records, and technology. Taking the feedback of subjects regarding factual matters into account can help.

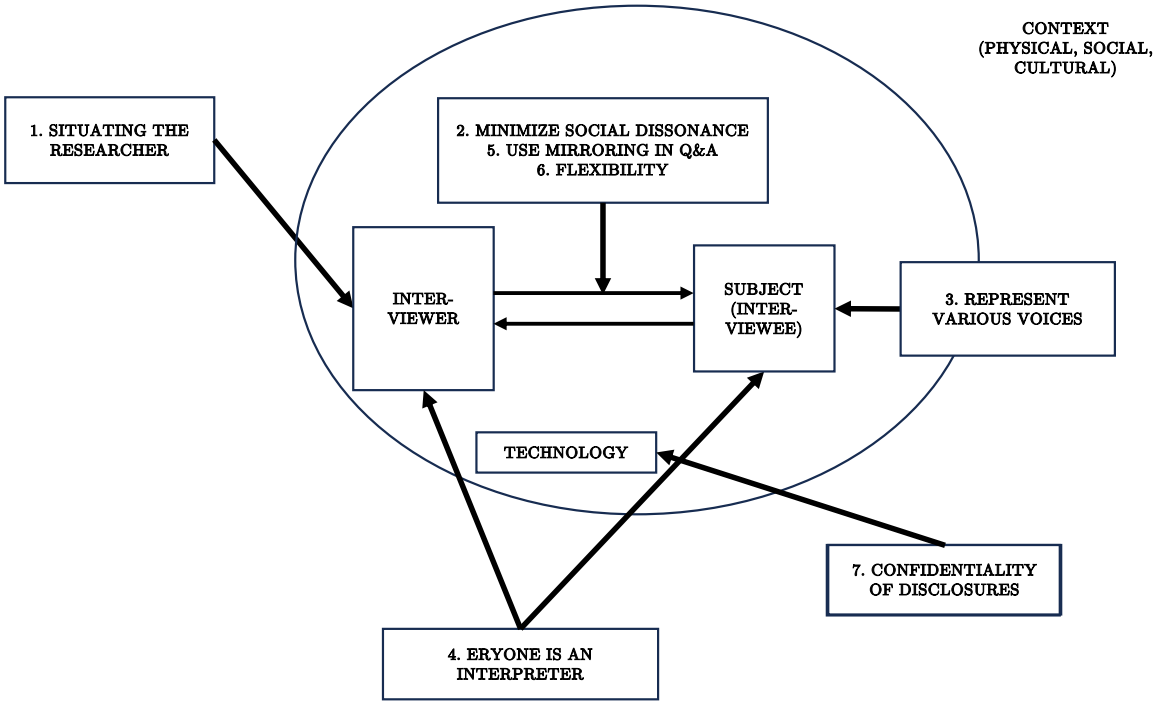


Figure 3.3.: Guidelines for Qualitative Interviews [MN07]

3.2.3. Coding Qualitative Data

Coding can be described as the "critical link" between the collected data and its explanation of meaning [Cha01]. According to Saldaña [Sal21], typically, a code is a "word or short phrase that symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute for a portion of language-based [...] data" [Sal21]. The data to be coded can stem from, for example, interview transcripts, field notes, or documents. Figure 3.4 shows an idealized streamlined coding process. In reality, the process of theory building is more complex and confusing but irrelevant to this dissertation. Ideally, codes are summarized in categories, which in turn can be used to derive concepts and theories. According to Richards and Morse [RM07], "categorizing is how we get 'up' from the diversity of data to the shapes of the data, the sorts of things represented. Concepts are how we get up to more general, higher-level, and more abstract constructs".

Coding is a cyclical act during which existing codes are recoded. Often, the First Cycle Coding can be improved by conducting more coding cycles that further manage, filter, highlight, and focus on the most important features of the qualitative data [Sal21]. The magnitude of data during the First Cycle Coding ranges between a single word and a whole page of text. During the Second Cycle Coding, the codes can be of the same size, greater, or a reconfiguration of the previously created codes [Sal21].

Through coding, qualitative information can be transformed into quantitative data to perform quantitative or statistical analysis as shown by the following example [Sea99]:

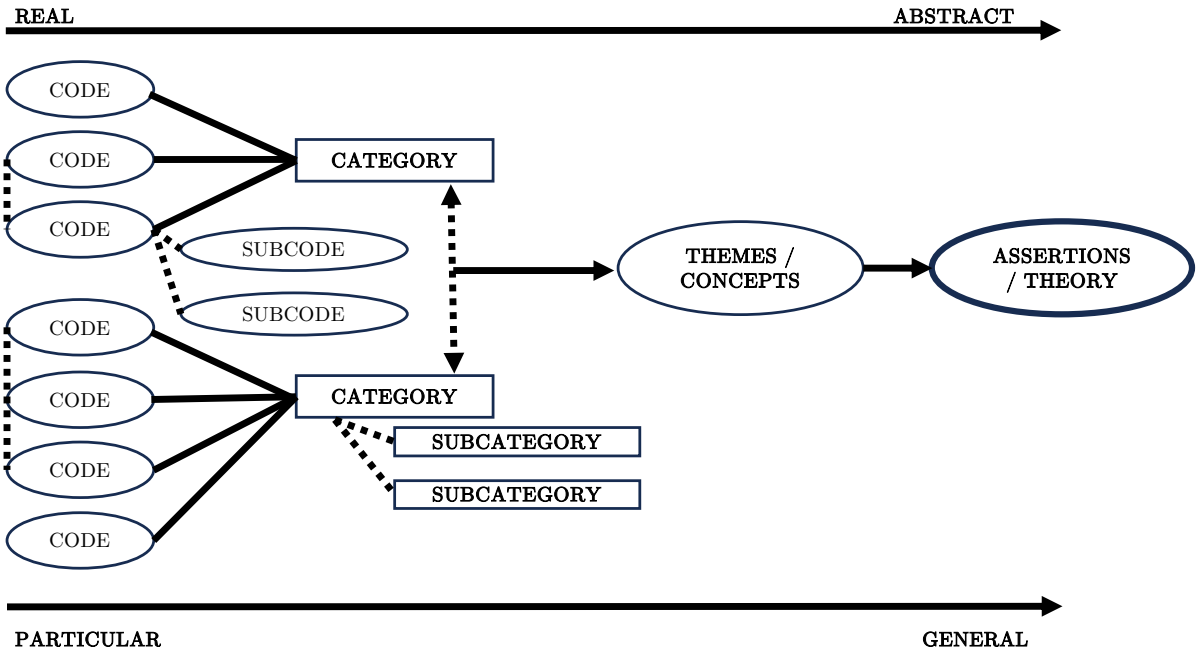


Figure 3.4.: Overview of an Idealized Coding Process [Sal21]

Example:

Tom, Shirley, and Fred were the only participants in the meeting.
↓
participantCount = 3

Such transformations must be done carefully if the concept to be coded is subjective, the terminology to describe the concept varies or is difficult to interpret, or the different data sources disagree [Sea99]. Transformations of qualitative information into quantitative data were performed in publications P1 and P2.

3.2.4. Design Science Research

Design science research aims to extend the boundaries of human and organizational capabilities by creating novel and innovative artifacts [HMPR08]. Generally, IT artifacts can be classified as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems) [HMPR08]. The design science research paradigm stems from engineering and the sciences of the artificial [Sim96]. At its core, design science is a problem-solving paradigm [HMPR08]. Researchers create knowledge and understanding of a problem domain and its solution while building and applying the designed artifact [HMPR08]. However, the designed artifacts are not exempt from behavioral theories or natural laws but rely on existing kernel theories that are implemented, tested, adapted, and extended by the researchers [WWES92, MMG02]. In this dissertation, we

followed the conceptual process (see Figure 4.6) and mental model by Peffers et al. [PTRC07] for conducting design science and presenting it.

During the first activity *problem identification & motivation*, the specific research problem is defined, and the value of a solution is justified. Second, the *objectives of a solution* are inferred from the problem definition. In this regard, an objective can be qualitative (i.e., the artifact supports solutions to problems not addressed yet) or quantitative (i.e., the desirable solution is better than current ones). To infer the objectives of a solution, the researchers require knowledge of the current problems and solutions and their efficacy. Third, the artifact is created to support the solution. This activity includes determining the desired functionality and architecture as well as creating the artifact. Fourth, the efficacy of the artifact to solve the problem is demonstrated. This dissertation used a case study in P4 to demonstrate the developed artifacts. Fifth, the researchers observe to which degree the artifact supports the inferred solutions to solve the problem. Therefore, the objectives of a solution must be compared with the actual results observed while using the artifact. After finishing this activity, the researchers can decide if it is worthwhile to redo the third activity to enhance the effectiveness of the artifact or to continue with the communication. The researchers must communicate their research through scholarly publications during the last activity. They describe the problem, its relevance, the designed artifact, and its usefulness and novelty. Moreover, they argue for the rigor of their design science research.

This process does not have to be executed in sequential order. The researchers can choose different entry points and move outward. Researchers who choose a problem-centered approach start with the first activity. This entry point is particularly suitable when the research idea emerges from observation of the problem or is based on suggestions for future work highlighted in previous research. Another option to start a design science project is an objective-centered solution. Here, the inspiration could come from development experiences where client expectations were unmet. A design and development-centered approach can be chosen if an artifact already exists but has not been formally thought through as a solution for the explicit problem domain. This research artifact possibly comes from a different research domain, was leveraged to solve a different problem, or appeared as an analogical idea. We chose a design and development-centered approach in P3 since our artifact was inspired by the work of Monahov et al. [Mon14], who developed a metric management fact sheet and metric catalog for the problem domain of Enterprise Architecture Management. The last research entry option is observing an existing solution that already works and working backward to increase rigor.

3.2.5. Action Design Research

Sein et al. [SHP+11] argue that design science pays insufficient attention to the artifacts' shaping by its organizational context. Existing design science methods concentrate on building the artifact and conducting the evaluation during a subsequent and isolated phase [SHP+11]. This approach usually results in artifacts with high technological rigor. However, it fails to recognize that the artifact emerges from "interaction with the organizational context" and, therefore, insufficiently optimizes for organizational relevance [SHP+11]. To address this problem, Sein et

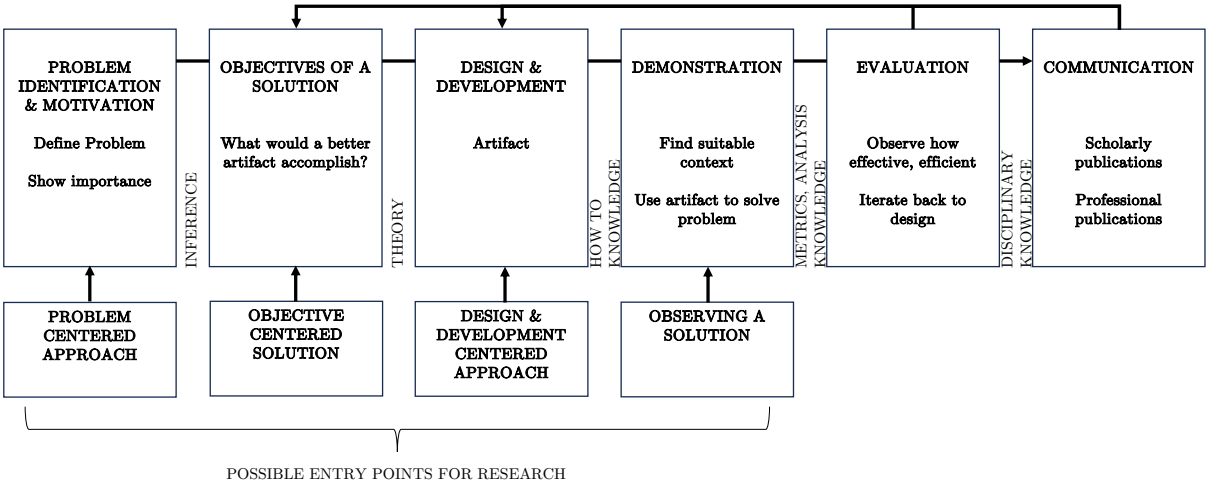


Figure 3.5.: Overview of Design Science Approach [PTRC07]

al. [SHP+11] propose action design research as a novel design science method. Action Design Research consists of four stages and seven principles (see Figure 3.6).

During the Problem Formulation stage, the researchers identify and conceptualize the research problem. In this stage, the initial research questions are formulated and the problem is described as an instance of a class of problems. In addition, relevant theories and technology are searched. Since an action design research project is conducted together with a partner that provides the organizational setting, the long-term commitment of both parties must be ensured. Further roles and responsibilities must be defined. Two principles guide the Problem Formulation stage. Principle 1, Practice-Inspired Research, encourages the researchers to view the research project as an opportunity to contribute knowledge to the higher class of problems instead of only solving the problem. Principle 2, Theory-Ingrained Artifact, highlights the importance of leveraging existing theories to structure the problem, identify possible solutions, or guide the artifact design.

The Building, Intervention, and Evaluation stage leverages the problem framing and theories from stage one to build an initial design of the IT artifact. This initial design is shaped through its organizational use and subsequent design iterations. In this stage, it is crucial to treat the building, intervention, and evaluation as inseparable and interwoven activities. Hence, the outcome of the Building, Intervention, and Evaluation stage is the realized design of the artifact. Three principles guide the researchers while conducting this stage. Principle 3 emphasizes that the IT artifact and organizational context mutually influence each other and are inseparable. Principle 4 states the importance that the project participants can and should learn from each other. Principle 5 points out that the artifact evaluation is not a separate step following the building activity.

During the Reflection and Learning stage, the researchers apply their learning to the broader class of problems instead of only to the particular problem instance they are interacting with. This stage is performed in parallel to the first two stages. Principle 6, Guided Emergence, accompanies this stage. This principle states that emergence and design are not mutually incompatible. Therefore, the designed artifact is a product of its initial design (cf. Principle 2),

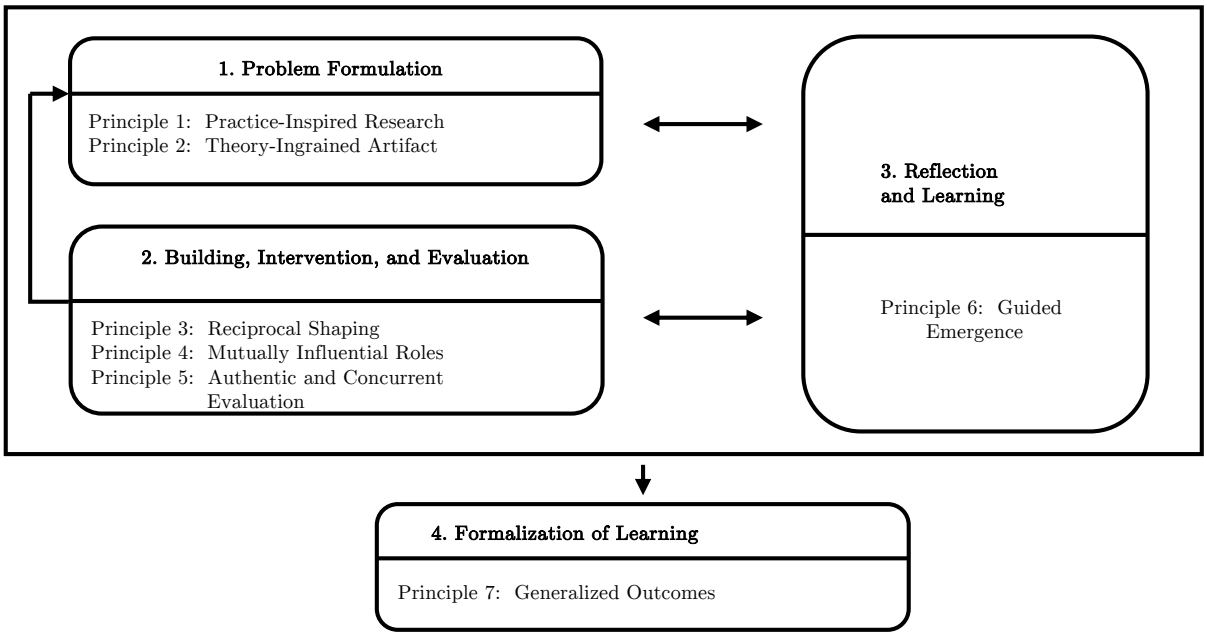


Figure 3.6.: Overview of Action Design Research [SHP+11]

organizational shaping including perspectives, and participants [GJT08, Iiv03] (cf. Principle 3 and 4), and evaluation (cf. Principle 5).

During the last stage, Formalization of Learning, the learning must be formalized into "general solution concepts for a class of field problems" [Ake04]. The seventh principle, Generalized Outcomes, accompanies this stage and stresses the importance of conceptually moving from the specific and unique to the general and abstract. This can be accomplished by generalizing the problem instance or by deriving design principles from the research outcomes.

3.2.6. Situational Method Engineering

Method Engineering can be defined as the "engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems" [Bri96]. However, in reality, projects are different and usually cannot be supported by standard methods from textbooks or manuals [Bri96]. Situational Method Engineering (SME) is a flavor of Method Engineering that tackles this problem by performing project-specific method construction [RDR03]. The main idea behind SME is the "in-house construction of an organization-specific or project-specific methodological approach" [HSR10] instead of implementing ready-made methods from some supplier.

Many strategies exist to perform SME [RDR03]. This dissertation followed the guidelines for performing an assembly-based SME approach [RDR03]. The researchers must perform two main tasks at the beginning of each SME effort. First, they have to set a method engineering goal. Second, they must construct a method that satisfies this goal. The assembly-based method

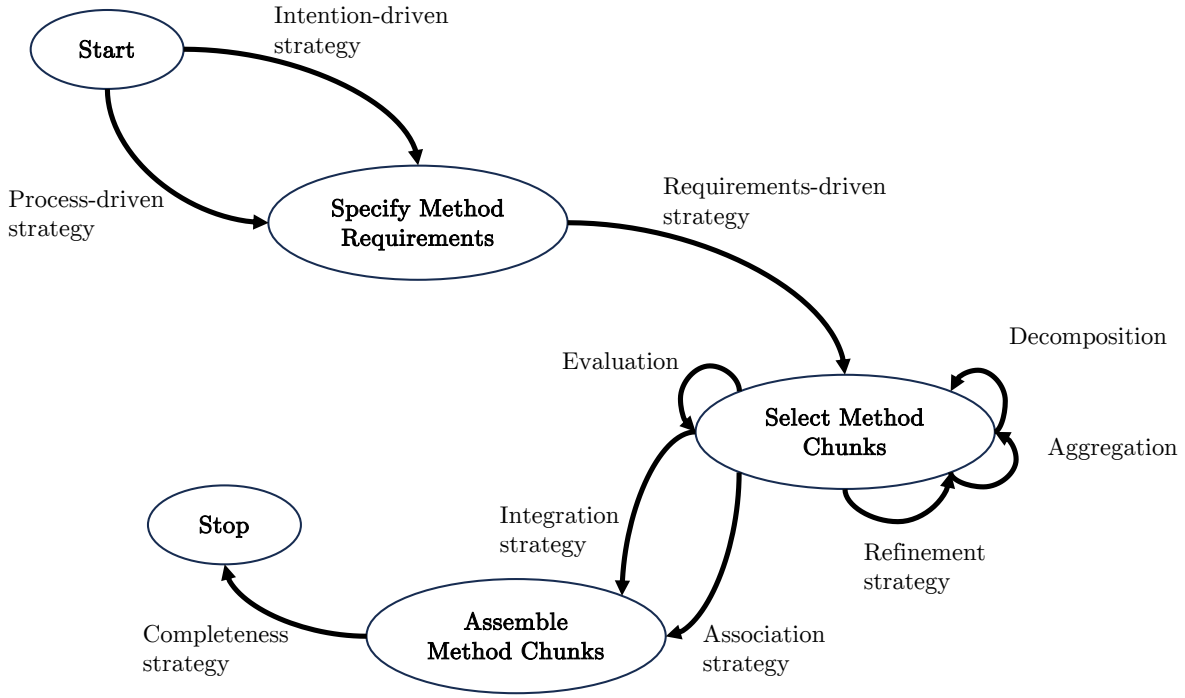


Figure 3.7.: Overview of an Assembly-Based SME Approach [RDR03]

construction consists of three main tasks, which are represented by the large nodes in Figure 3.7.

First, the method requirements must be specified. These requirements are derived from a method engineering goal, which is defined to clarify if an existing method is adapted or a new method is designed. Two strategies can be applied to elicit the requirements. We used the intention-driven strategy since it is suitable for method adaptation. An adaptation can be either in the form of adding or deleting functionality.

Second, the method chunks that match the identified requirements best are selected by following a requirements-driven strategy. An evaluation strategy is performed to validate the retrieved chunks, which determines the degree to which candidate chunks fulfill the requirements. The researchers can apply decomposition, aggregation, and refinement strategies to refine the candidate chunk selection. Decomposition is a means to resolve aggregated chunks into their sub-chunks to decide for each sub-chunk if it is selected or dismissed. In opposition, aggregation combines retrieved chunks that only partly match with other chunks that might fulfill the requirements. The refinement strategy encapsulates the search for another chunk that satisfies the same intention but with different guidelines for achieving it.

Part B

1. Investigating the Adoption of Metrics in Large-Scale Agile Software Development

Table 4.1. Fact Sheet Publication P1

Authors	Philipp, Pascal* Tobisch, Franziska* Matthes, Florian* *Technische Universität München, Chair of Software Engineering for Business Information Systems, Boltzmannstraße 3, D-85748 Garching, Germany
Outlet	26th Pacific Asia Conference on Information Systems (PACIS)
Status	Published
Contribution of First Author	Problem Definition, Research Design, Data Collection, Data Analysis, Interpretation, Reporting

Abstract. Many organizations have started using agile methods to develop software in small projects. The success of agile methods on a small scale inspired organizations to scale them to larger contexts. However, adopting agile practices at scale can complicate the monitoring, coordination, and steering of the multi-layered development process. Metrics can address this challenge but are controversial since their implementation is a challenge in itself. Hitherto, research on large-scale agile development lacks publications investigating the adoption of metrics. We conducted an expert interview study to explore their (i) reasons for adopting metrics (ii), what metrics they use on the team, program, and portfolio level (iii), and the most occurring metrics in the expert organizations. Our results show that metrics are mainly used for transparency, improvement, and controlling. Most metrics occur on the program level. Finally, we identified story point estimation, velocity, and sprint burn-down, as the most established metrics.

2. Challenges and Success Factors for Metrics in Large-Scale Agile Development

Table 4.2. Fact Sheet Publication P2

Authors	Philipp, Pascal* Tobisch, Franziska* Matthes, Florian* *Technische Universität München, Chair of Software Engineering for Business Information Systems, Boltzmannstraße 3, D-85748 Garching, Germany
Outlet	28th Americas Conference on Information Systems (AMCIS)
Status	Published
Contribution of First Author	Problem Definition, Research Design, Data Collection, Data Analysis, Interpretation, Reporting

Abstract. Contemporary organizations widely use agile software development to react to unpredictable changes in their business environment. Due to the success of agile methods in contexts similar to the agile sweet spot, organizations have been applying them on a larger scale. However, maintaining oversight in large-scale agile development remains a problem. Metrics can tackle this problem by increasing transparency, but organizations have struggled with their adoption. Furthermore, extant research on large-scale agile development lacks publications investigating metric challenges and success factors. Against this backdrop, we conducted an expert interview study with 23 experts from 13 organizations. The most mentioned challenges are data collection challenges, lack of metric usefulness, and metric calculation challenges. On the other hand, the most occurring success factors are context-dependent metric adoption, implementing an agile metric management process, and ensuring understanding of the metric purpose.

3. Toward a Metric Catalog for Large-Scale Agile Development

Table 4.3. Fact Sheet Publication P3

Authors	Philipp, Pascal* Tobisch, Franziska* Menzel, Leon* Matthes, Florian* *Technische Universität München, Chair of Software Engineering for Business Information Systems, Boltzmannstraße 3, D-85748 Garching, Germany
Outlet	28th Americas Conference on Information Systems (AMCIS)
Status	Published
Contribution of First Author	Problem Definition, Research Design, Data Collection, Data Analysis, Interpretation, Reporting

Abstract. Nowadays, organizations use agile software development to remain competitive in their frequently changing business environment. Inspired by the success of agile methods on a small scale, organizations have started to apply them in larger contexts. However, the limited scalability of agile methods is a problem. Metrics can be a success factor for achieving agility at scale, thus adopting them is promising. Most scaling agile frameworks provide few recommendations regarding metrics. Likewise, research on metrics in large-scale agile development lacks concrete guidance for metrics or their organization-specific adoption. To fill this gap, we propose two artifacts. We present the design of a minimalistic metric management fact sheet (MMFS) for large-scale agile development to support practitioners in using metrics in their organization-specific development environment. Furthermore, the MMFS is the basis for our metric catalog documenting 196 metrics identified in an expert study to provide a comprehensive metric set for scaling agile environments.

4. A Method for Metric Management at a Large-Scale Agile Software Development Organization

Table 4.4. Fact Sheet Publication P4

Authors	Philipp, Pascal* Tobisch, Franziska* Menzel, Leon* Matthes, Florian* *Technische Universität München, Chair of Software Engineering for Business Information Systems, Boltzmannstraße 3, D-85748 Garching, Germany
Outlet	17th International Conference on Software Process and Product Measurement (MENSURA)
Status	Published
Contribution of First Author	Problem Definition, Research Design, Data Collection, Data Analysis, Interpretation, Reporting

Abstract. The benefits of agile methods for small projects inspired organizations to scale these methods to more extensive settings consisting of multiple agile teams. Such scaling agile settings are more complex, which can make maintaining situational awareness difficult. Metrics can alleviate this problem by increasing insight into the development organization. However, adopting metrics comes with various socio-technical challenges, and current research is missing guidance on metric management in large agile organizations. Therefore, we present a goal-based method designed for a large agile case organization to support stakeholders in selecting, operating, and scaling metrics. Moreover, based on the learnings at the case organization, we present design principles that can potentially guide the development of methods suitable for other contexts. We conducted this research following an action design research (ADR) approach combined with situational method engineering (SME). Our findings indicate that our method proved effective for the case organization. This was accomplished by combining well-established elements (e.g., goal-orientation and tool support) from measurement programs designed for traditional software engineering with unique elements of our method (e.g., metric scaling activities and alignment with agile software development). With this study, we provide deep insights into how metrics are managed at a large agile case organization. Researchers and practitioners can use this work as a foundation for designing measurement programs suitable to other scaling agile organizations.

5. Investigating the Establishment of Goals in Large-Scale Agile Development

Table 4.5. Fact Sheet Publication P5

Authors	Philipp, Pascal* Schüll, Moritz Matthes, Florian* *Technische Universität München, Chair of Software Engineering for Business Information Systems, Boltzmannstraße 3, D-85748 Garching, Germany
Outlet	26th Pacific Asia Conference on Information Systems (PACIS)
Status	Published
Contribution of First Author	Problem Definition, Research Design, Data Collection, Data Analysis, Interpretation, Reporting

Abstract. Maintaining competitiveness in future business environments requires agile organizations to implement systematic changes such as adapting existing goal-setting practices. Such changes can only be accomplished by taking a holistic view of the development organization. In large-scale agile development, the whole development organization is considered, and some scaling agile frameworks support practitioners to establish goals by providing recommendations for goal-setting practices. However, so far, only a limited amount of research investigating the establishment of goals in scaling agile environments exists. Against this backdrop, we present a case study to explore how goals are set and documented within eight programs at a large German automobile manufacturer. Moreover, we identify and categorize goals, present goal-setting challenges encountered by the case organization, and formulate seven mitigation propositions to address these challenges. Finally, we evaluate these mitigation propositions and discuss the evaluation results by incorporating the qualitative feedback provided by the interviewees.

Part C

CHAPTER 4

Summary of Results

In this chapter, we summarize the results of the five publications.

Summary of main findings related to RQ1: To answer RQ1, we conducted an expert interview study across multiple large-scale agile organizations. In this study, we identified reasons for using metrics and categorized them. Moreover, we investigated at which scaling levels metrics are implemented, which type of metrics are used (i.e., agile or traditional metrics), and present the most used metrics among the expert organizations.

Research Question 1 (RQ1)

Why are metrics adopted, on which scaling levels are they implemented, which kind of metrics are used, and what are the most popular metrics across large-scale agile software development organizations?

Reasons for Metrics in Large-Scale Agile Software Development

We asked experts for their reasons for using metrics in their large-scale agile software development environments. Table 5.1 summarizes the reason categories and the contexts in which those reasons occur. The most frequently cited reasons were Transparency (39%), Improvement (22%), Controlling (13%), Planning (8%), Employee motivation (4%), Agility (3%), Problem identification (3%), and Stakeholder involvement (3%). To enhance clarity and conciseness, other less commonly cited reasons are not detailed in this summary. Each major reason category can relate to one of the following contexts: Development organization and processes, Product, Entire organization, Customer, Finance, Employee, and Stakeholder. For example, in our study, experts noted increasing transparency in the context of Development organization and processes as a reason 56 times. When a main category was mentioned, but the expert did not relate it to a

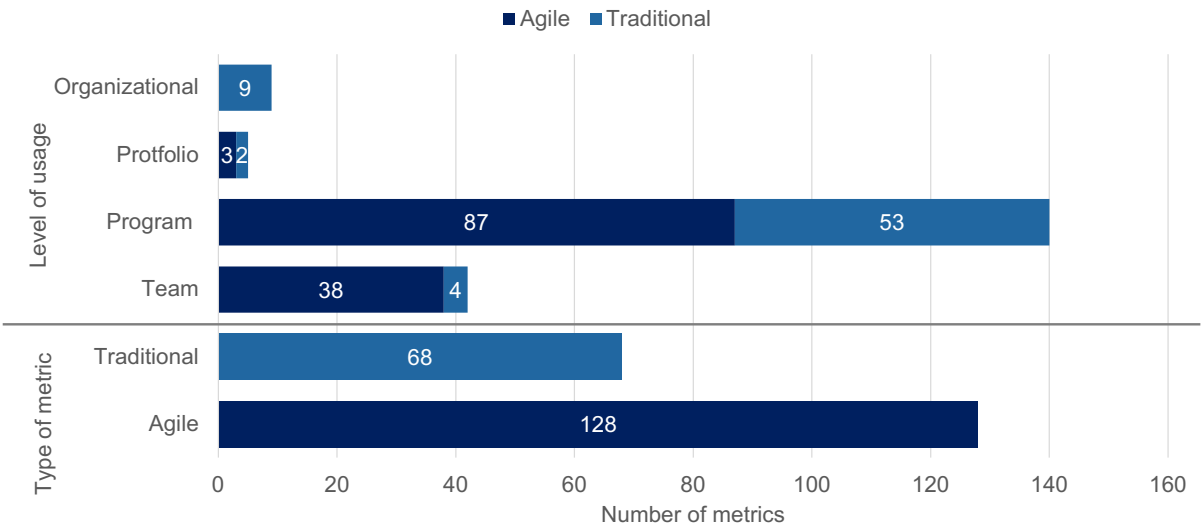


Figure 4.1.: Distribution of Metrics per Organizational Level of Usage and Type (Based on P2)

context, we assigned the reason to the context Generic. In the following, we explain each main category in more detail.

Agile and Traditional Metrics on Organizational Levels

Moreover, we examined the use of metrics at different organizational levels (i.e., team, program, portfolio, and organization) and their type (i.e., traditional or agile). Our results are not quantitative since no mathematical models were used for analysis. Instead, we focused on capturing the frequency of specific metrics in expert organizations within different categories (see Figure 4.1).

The study results showed that most metrics (140, corresponding to 71.43% of the total) were implemented at the program level, followed by the team level with 42 metrics (21.43%), and the organization level where 9 metrics (4.59%) were captured. With only five metrics captured (2.55%), the portfolio level was the area where the lowest number of metrics were identified.

In terms of the type of metrics, it appeared that at all levels studied, except for the organizational level, agile metrics were implemented more frequently than traditional metrics. In particular, at the team level, the dominance of agile metrics was clear: of the total 42 metrics implemented at this level, 38 (90.48%) were agile. In contrast, only traditional metrics could be identified at the organizational level.

Most Used Metrics Among the Expert Organizations

To identify the most widely used metrics among the expert organizations, we asked the experts to name the metrics used in their respective organizations. Similar to the previous section, we do not apply quantitative research methods in investigating the most used metrics. Following the rule of three [Cop96], we included only those metrics in table 5.2 that are applied in at least three organizations. Most organizations (10, or 91%) measured Story Point Estimation and Velocity

Table 5.1. Reasons for Metrics in Large-Scale Agile Software Development (Based on P2)

Reason	# Occurrences	# Experts
Transparency	**111 (39%)**	
Development organization and processes	56 (20%)	18 (90%)
Product	18 (6%)	10 (50%)
Generic	13 (5%)	10 (50%)
Entire organization	9 (3%)	8 (40%)
Customer	7 (2%)	6 (30%)
Finance	5 (2%)	5 (25%)
Employee	2 (1%)	2 (10%)
Stakeholder	1 (<1%)	1 (5%)
Improvement	**62 (22%)**	
Development organization and processes	28 (10%)	12 (60%)
Generic	13 (5%)	9 (45%)
Product	12 (4%)	7 (35%)
Entire Organization	4 (1%)	2 (10%)
Employee	4 (1%)	4 (20%)
Customer	1 (<1%)	1 (5%)
Controlling	**38 (13%)**	
Generic	14 (5%)	7 (35%)
Development organization and processes	10 (3%)	8 (40%)
Entire organization	5 (2%)	2 (10%)
Employee	4 (1%)	3 (15%)
Product	3 (1%)	3 (15%)
Finance	2 (1%)	1 (5%)
Planning	**24 (8%)**	
Development organization and processes	12 (4%)	8 (40%)
Generic	7 (2%)	5 (25%)
Product	6 (2%)	6 (30%)
Employee motivation	**11 (4%)**	
Development organization and processes	9 (3%)	9 (45%)
Entire organization	2 (1%)	2 (10%)
Agility	**9 (3%)**	
Generic	8 (3%)	4 (20%)
Development organization and processes	1 (<1%)	1 (5%)
Problem identification	**9 (3%)**	
Generic	3 (1%)	3 (15%)
Product	3 (1%)	3 (15%)
Development organization and processes	3 (1%)	3 (15%)
Stakeholder involvement	**8 (3%)**	
Customer	6 (2%)	6 (30%)
Generic	1 (<1%)	1 (5%)
Entire organization	1 (<1%)	1 (5%)

Table 5.2. Most Used Metrics Among the Expert Organizations (Based on P2)

Metric			#org.
Story point estimation: Sum of the story points estimated for a user story.	A	T	10 (91%)
Velocity: Sum of the story points of all user stories completed (Done) by a team in a sprint.	A	T	10 (91%)
Sprint burn-down: Sum of the story points of all open user stories in a sprint of a team in relation to the time left in the sprint reduced for each completed (Done) user story.	A	T	7 (64%)
Defects in production: Number of defects in the production environment(s) of the software product for each severity category.	TD	P	4 (36%)
Planned velocity: Sum of the story points of all user stories planned to be completed by a team in a sprint.	A	T	4 (36%)
Program velocity per sprint: Sum of the story points of all user stories completed (Done) by all teams in a program in a sprint.	A	P	4 (36%)
Number of features per PI: Number of features realized (Done) in a PI by a program.	A	P	3 (27%)
Planned program velocity per sprint: Sum of the story points of all user stories planned to be completed by all teams in a program in a sprint.	A	P	3 (27%)
Release-burn-down: Sum of the story points of all user stories planned for a release that still need to be realized (Done) by all teams in a program in relation to the time left until the release.	A	P	3 (27%)
Test coverage: Percentage of the number of lines of code all tests currently execute in relation to the total number of lines of code of the software that is to be tested.	TD	P	3 (27%)
Legend: T = Team, P = Program; A = Agile, TD = Traditional			

metrics. Notably, these metrics are typically implemented at the team level. The second most common performance metric is the Burn-Down Chart, which is used in seven organizations (or 64% of cases). In third place are the metrics Defects in Production, Planned Velocity, and Program Velocity per Sprint, which are used in four companies (36%).

Summary of main findings related to RQ2: To answer RQ2, we again use the results of the expert interview study, which was conducted across multiple large-scale agile organizations. In this study, we identified, categorized, and ranked (i.e., according to the number of occurrences) challenges and success factors for metrics applied in large-scale agile development.

Research Question 2 (RQ2)

What challenges and success factors have been reported for metrics in large-scale agile development?

Challenges for Metrics in Large-Scale Agile Development

We identified eleven challenges, as summarized in Table 5.3.

(1) Data collection challenges: Data collection for metrics was recognized as a major problem among the experts. Related challenges are data quality, data availability, data collection complexity, and anonymity of data collection. In particular, data quality is highlighted as a major challenge, as a metric can lose its value due to irrelevant data or biased human data sources. Additionally, the lack of discipline among employees regarding entering the data was identified as another challenge related to data availability.

(2) Lack of metric usefulness: A large number of experts identify ensuring metric utility as a critical problem. Related challenges are deriving value from metrics and error-prone metric-based planning. In deriving value from metrics, experts experience a lack of insight into the reasons for changing metric values or deviations from planned metric values and that metrics were only a starting point for identifying problems. Regarding the challenge of error-prone metrics-based planning, one expert expressed the difficulty of making reliable forecasts based on metrics.

(3) Metric calculation challenges: Metric calculation is another pervasive problem. Experts cited the inclusion of all relevant variables, the provision of resources for calculation, the aggregation of team-based values to program-based values, the complexity of metric calculation, and the lack of adequate tool support as related challenges. Including all relevant variables was particularly emphasized, as failure to account for variables such as fluctuations in velocity planning, vacation time, and changes in the size of Epics resulted in a reduction in metric usefulness. In addition, the significant time required to create metric tests and measure the metric is perceived as challenging.

(4) Metric-based decision-making challenges: Decision-making based on metric values is perceived as problematic. For example, experts cited difficulty in deriving the next actions from metrics, ineffective management based on metric values, and misleading information derived from metrics as associated challenges. For example, one organization had problems identifying actions and responsible parties from the metric to remove obstacles. In addition, ineffective management can occur when managers make infeasible comparisons between teams or programs based on metrics (e.g., story points).

(5) Negative effects for employees: Negative impact on employees is another common problem and includes challenges such as adverse psychological effects, internal competition, and the feeling of being controlled. An example of adverse psychological effects is increased pressure and stress due to deviations from planned metric values. In addition, metric-based criticism is sometimes perceived as a personal attack. Further, metric-based comparisons can lead to unwanted internal competition, thus negatively affecting collaboration between teams.

(6) Lack of metric understanding: A lack of understanding of metrics is another common problem among experts. It is caused by a lack of awareness of dependencies, a general lack of metric understanding, a lack of understanding of metric calculation, insufficient resource allocation to gain metric understanding, and complexity. Occasionally, a lack of knowledge about the existence of metrics is also a related challenge. Flawed understanding often leads to metric misuse. In particular, a lack of understanding by managers is described as problematic.

(7) Change resistance: Resistance to change within the organization is also a frequently cited problem by the experts. Related challenges include difficulty in convincing stakeholders, lack

of management commitment, establishing a common understanding of metrics within the organization, investment in change, and lack of common reporting standards. An example of how change resistance can manifest itself is the difficulty of convincing managers with a traditional mindset to use agile metrics and justify metrics to the works council.

(8) Focus on local instead of global optimum: Working toward goals that do not contribute to an organization-wide optimum presents another challenge. Several experts noted that the target values of metrics often do not align with key objectives on higher organizational levels, such as customer satisfaction, product quality, or cost efficiency.

(9) Metric definition challenges: Several experts articulated the challenges associated with defining metrics. In particular, defining adequate target values, inherent complexities of metric definition, determining appropriate tolerance values, determining appropriate granularity for calculation parameters, and devising appropriate calculation rules were identified as problematic. Challenges in the context of target value definition include setting unrealistic target values and significant discrepancies between programs in defining target values at the portfolio level. Further, the metric definition can be complex due to difficulties in harmonizing metrics across teams.

(10) Manipulation of metric values: Another challenge discussed by several experts is employees' manipulation of metric values. Conservative estimation was given as an example, which serves as a strategy to achieve the target metric value. According to the experts, motivators for manipulating metric values include competition between teams and employees' sense of being controlled.

(11) Imbalance between control and team autonomy: Determining the right balance between control and team autonomy is a recurring challenge. In particular, balancing governance and team autonomy and avoiding excessive management involvement present related challenges. The balance between governance and team autonomy is problematic because self-organization and control can be perceived as opposing concepts. Regarding management involvement, both program management interference and metrics-based micromanagement are perceived to hinder employees' work.

Success Factors for Metrics in Large-Scale Agile Development

We identified 22 success factors, of which eleven are explained in detail (see Table 5.4). The remaining success factors are summarized in the last paragraph.

(1) Context-dependent metric adoption: An effective implementation of metrics requires contextualization of metrics. According to experts, context significantly impacts various aspects, including the configuration of metrics, decision-making based on metric data, interpretation of metric values, and measurement and selection of relevant metrics. Furthermore, organizations need to address the specific requirements of the context of large-scale agile development. This context fitting requires a consistent consideration of agile values and principles.

(2) Implementing an agile metric management process: Several experts identified implementing an agile metrics management process as a critical success factor. From their perspective, continuous improvement, collaborative inspection, collaborative decision-making concerning the metrics lifecycle, ensuring metrics transparency, and using lightweight tools are essential. A

Table 5.3. Identified Metric Challenges (Based on P3)

ID	Challenge	% Occurrences	# Experts (Occurrences)
C1	Data collection challenges	21%	17 (85%)
C2	Lack of metric usefulness	15%	17 (85%)
C3	Metric calculation challenges	13%	12 (60%)
C4	Metric-based decision-making challenges	11%	18 (90%)
C5	Negative effects for employees	11%	16 (80%)
C6	Lack of metric understanding	7%	10 (50%)
C7	Change resistance	6%	10 (50%)
C8	Focus on local instead of global optimum	5%	10 (50%)
C9	Metric definition challenges	5%	10 (50%)
C10	Manipulation of metric values	4%	7 (35%)
C11	Imbalance between control and team autonomy	2%	4 (20%)

collaborative inspection could be fostered by integrating discussions about metrics into regular events.

(3) Ensuring understanding of metric purpose: An additional success factor is ensuring a deep understanding of the purpose of the metric, thereby minimizing potential resistance to change and eliminating unproductive comparisons between teams. Experts also emphasized the importance of improving understanding at the management level and throughout the organization. In addition, creating an understanding of the metric's target values is considered crucial.

(4) Managing the interplay of metric and its environment: Each metric is embedded in a context, and managing its dependencies within this context is essential for a successful implementation. From an expert perspective, it is necessary to manage the relationships between metrics and their dependencies on the surrounding context. In addition, the experts highlighted the combination of related metrics and the awareness of the distinction between causality and correlation as important.

(5) Keep metric adoption simple: An additional success factor considered significant by the experts concerns simplicity in adopting metrics. The experts highlighted the relevance of simplifying both the measurement of metrics and metrics in general. They emphasized automation and the selection of low-overhead metrics as strategies for streamlining metrics implementation. A straightforward strategy to simplify metric adoption is limiting the number of metrics to be implemented.

(6) Measures for ensuring effective data collection: Various experts recommended implementing measures for effective data collection. Therefore, establishing a data collection strategy, ensuring data quality, and an effective testing environment are key factors to be considered. Maintaining anonymity during data collection and defining responsibilities within the data collection process were also considered crucial. In addition, clarifying the effort required for data collection is another success factor.

(7) Enabling metric adoption by providing sufficient resources: Ensuring adequate resources, including, for example, through the support of appropriate tools, the provision of a metric expert team, the availability of experts, and the allocation of appropriate time and budget, were identified as critical.

(8) Empowerment of teams in metric adoption: Strengthening the capacity of teams to implement metrics is an additional success factor. Therefore, it is important to delegate extensive responsibility to the respective team. Furthermore, it is essential to ensure autonomy to the team in selecting and deselecting, calculating the metric as well as configuring it. Finally, the team-centric perspective in adopting metrics is highlighted as another key factor.

(9) Standardization of metric adoption: Most metric experts advocate using best practices in adopting metrics. In addition, achieving standardization in calculation, configuration, and reporting is critical. Tools serve here as a means of standardizing the adoption of metrics.

(10) Combining metric with qualitative feedback: An additional success factor lies in the fusion of the quantitative information provided by metrics with complementary qualitative feedback to enhance inspection. Integrating qualitative feedback can help improve understanding of quantitative values or provide a more comprehensive view of the situation. Surveys can serve as a means of acquiring qualitative feedback.

(11) Ensuring goal-orientation of metrics: The use of metrics to support goal achievement can be realized if the target values of the metrics are themselves defined as goals or if the metrics provide transparency with respect to the fulfillment of the goals. Experts identified setting realistic target values as a key factor. Furthermore, defining objectives and key results and quantifying them through the use of metrics can benefit organizations. In addition, the target values of metrics should focus on the outcome rather than on the output delivered.

(12) Other: Additional success factors include providing training, guidance, and documentation; understanding what each metric measures; using metrics at higher organizational levels; communicating the metrics' value to reduce resistance to change; providing access to metrics information only to authorized stakeholders; assigning clearly defined responsibilities; collaborating teams to achieve shared metric target values; establishing a community of practice for metrics; supporting alignment between management and teams on metrics; ensuring workforce acceptance of the tools; and ensuring realistic expectation management based on the metrics.

Summary of main findings related to RQ3: To answer RQ3, we used the design science approach [HMPR08] to develop two artifacts for the domain of large-scale agile development: (i) a structured documentation template for metrics in the form of a minimalistic MMFS and (ii) a metric catalog, documenting metrics found in our expert interview study using the previously developed MMFS. We conducted three survey-based iterations to evaluate our MMFS and

Table 5.4. Identified Metric Success Factors (Based on P3)

ID	Success Factor	% Occurrences	# Experts (Occurrences)
S1	Context-dependent metric adoption	16%	19 (95%)
S2	Implementing an agile metric management process	15%	14 (70%)
S3	Ensuring understanding of metric purpose	9%	14 (70%)
S4	Managing interplay of metric and its environment	8%	9 (45%)
S5	Keep metric adoption simple	8%	13 (65%)
S6	Measures for ensuring effective data collection	7%	12 (60%)
S7	Enabling metric adoption by providing sufficient resources	7%	13 (65%)
S8	Empowerment of teams in metric adoption	6%	10 (50%)
S9	Standardization of metric adoption	5%	8 (40%)
S10	Combining metric with qualitative feedback	3%	5 (25%)
S11	Ensuring goal-orientation of metrics	3%	8 (40%)
	Other	13%	18 (90%)

metric catalog. Finally, we presented the resulting metric catalog at two case organizations, where the catalog was demonstrated and evaluated as part of the initial version of our metric management method (cf. RQ4).

Research Question 3 (RQ3)

How can a generic and structured metrics collection be designed to support large-scale agile software organizations during metric adoption?

During the design and development phase of the design science approach [HMPR08], we carefully reviewed the related work to identify description elements for metrics. Moreover, for each identified description element, we evaluated its applicability to the domain of large-scale agile development. Similar to Monahov [Mon14], we provide an overview of all elements from the literature, including their source (see Table 5.5). The elements without checkmarks (i.e., calculation type, reporting frequency, definition responsible, implementation responsible, and monitoring responsible) are unique to our design and result from the insights we were able to derive from the expert interview study. Our minimalistic MMFS is most similar to Monahov's [Mon14] design, as both structures have the most elements in common. The calculation rule and the title are the most frequently proposed elements among all sources. Like Monahov [Mon14], we divided the elements into two categories. We characterize general elements as "independent of the context

of a specific organization" [Mon14]. We classify elements that "describe the configuration of the metric in a specific organization" as organization-specific elements [Mon14]. Our expert interviews confirm the relevance of organization-specific elements, as organizations have implemented similar metrics but have used multiple configurations in terms of tools, people responsible, terminology, and target values of the metric. Furthermore, organization-specific elements enable the preservation of confidentiality of certain information (e.g., names of employees).

Elements of the minimalistic MMFS: Like Monahov [Mon14], we have included only a "minimal number of description elements, which are required to ensure a comprehensive metric documentation as a starting point for an organization-specific metric implementation". Below, we provide a detailed description and rationale for each element included. The following seven elements were classified as general elements:

(1) Title: The title, written in natural language, provides a concise and accurate overview of the metric. It assists in the retrieval of metrics. Omitting the title could hinder the speed and simplicity of understanding a metric [Mon14].

(2) Description: The description comprehensively explains the purpose and meaning of the metric. It contributes to a common understanding of the motivations for a particular metric. The omission of the description could affect the collective understanding of the stakeholders involved regarding the motivation, expected benefits, and assumptions of the metric [Mon14].

(3) Calculation rule: The calculation rule must include all relevant variables and transparently state how a particular metric is calculated. A calculation rule can be expressed in either natural or formal language. The validity of both approaches has been confirmed in the studies of Hartmann and Dymond [HD06] and Bouwers et al. [BDV14]. Our work follows the approach of Monahov [Mon14] by expressing the calculation rule in natural language but always aligned with its formal definition. In this way, we provide an accessible understanding of the metric without losing essential information. Furthermore, the results of our interviews indicate that neglecting the calculation rule can lead to complications, as certain tools require it.

(4) Information model: In line with the suggestion of Monahov [Mon14], we document and visualize the information model underlying each metric using a UML class diagram. These information models facilitate the identification of the specific data items needed to calculate the metrics and promote an understanding of the relationships between these data items. Furthermore, the information model enables mapping each metric to a company-specific context and terminology. Without an appropriate information model, the data items and their relational connections could not be adequately visualized [Mon14].

(5) Code: The code acts as an identifier for the specific metric and facilitates the efficient retrieval of metrics. It also ensures unambiguous use by all stakeholders involved [Mon14]. Eliminating the code from the minimalistic MMFS significantly complicates metrics retrieval [Mon14].

(6) Goals: Consistent with Monahov [Mon14], we assign each metric to at least one management goal in large-scale agile development. Each metric contributes to the achievement of these goals. Our interview study identified five main goal categories and 27 subcategories. Excluding goals from the minimalistic MMFS results in a disconnect between goals and metrics, an aspect of essential importance to the success of metrics [Bas92].

Table 5.5. Final MMFS Elements and their Literature Sources (Based on P3)

MMFS element	Neely et al.	Popova and Sharpanskykh	Parmenter	Kütz	ISO/IEC/IEEE 15939:2017(E)	Hartmann and Dymond	Monahov	Bouwers et al.	Olsina et al.	Kaner et al.	Olszewska et al.
General elements											
Title	X	X		X	X	X	X	X	X		X
Description			X	X			X		X		
Calculation rule	X	X		X	X	X	X	X	X	X	X
Information model							X				
Code			X				X				
Goals							X				
Organizational level						X				X	X
Organization-specific elements											
Organizational instance			X								
Data mapping											
Data item							X				
Mapped name							X				
Data source	X			X			X				
Data contact				X			X				
Properties											
Measurement tool									X	X	X
Calculation type											
Measurement frequency/event	X		X	X			X		X		
Monitoring frequency/event	X										
Reporting frequency/event											
Owner				X					X		
Definition responsible(s)											
Implementation responsible(s)											
Achievement responsible(s)	X			X			X				
Monitoring responsible(s)											
Target value(s)	X		X	X			X	X	X		
Planned value(s)	X			X			X	X	X		
Tolerance value(s)		X		X	X		X	X	X		
Escalation rule(s)				X	X		X				
Interested stakeholder						X					

(7) Organizational level: Each metric is linked to the organizational level at which it is applied. According to Hartmann and Dymond [HD06], the organizational level is particularly relevant in an agile context because it can reveal potential limits to metric application. Furthermore, it supports the efficient retrieval of metrics. Metrics and their associated goals are often linked to organizational levels in large-scale agile development [KTS21, SS18]. We linked each metric with either team, program, portfolio, or enterprise level, consistent with Korpivaara et al. [KTS21] and Stettina and Schoemaker [SS18]. Eliminating the organizational level from the minimalistic MMFS would make it impossible for stakeholders to directly and unambiguously link a metric to its corresponding organizational level.

(8) Organizational instance: Our expert interviews suggest that most organizational environments have a high diversity of organizational structures (i.e., multiple teams, programs, or portfolios). Metrics may be implemented multiple times within a single organization. As a result, the organizational instance aims to document the specific organizational entity that implements a particular metric, such as a single team. Identifying metric affiliation would be complicated should the organizational instance be eliminated from the minimalistic MMFS.

(9) Data mapping: This element supports the organization-specific metrics adoption. According to our interview-based research, organizations use diverse terminologies to define the data elements of the information model. Therefore, we adapt Monahov's [Mon14] approach by assigning each data element of the information model to its organization-specific context. By using the mapped name, the internal term or concept is linked with a class, attribute, or relationship. In addition, each data element is linked to its associated data source to ensure complete documentation of the data sources required for metrics calculation. The responsible data contact can also be documented. This role is responsible for ensuring data quality and availability. Eliminating the element from the minimalistic MMFS would lead to complications regarding organization-specific terminology mapping, documentation of required data sources, and data contacts [Mon14].

(10) Properties: We propose a property element with 15 organization-specific attributes to better contextualize the metric. All tools used in the adoption of the metric should be documented in the first attribute, the measurement tool. Following the recommendation of Olsina et al. [OLP02], we suggest specifying the calculation method to determine whether a metric is calculated manually or automatically. Our expert study found that such specification is appropriate because some tools require manual intervention while others are fully automated. Further, we recommend the measurement frequency or event attribute for interpreting the metric's underlying time series and to connect the metric to agile events such as sprint or program planning. The frequency or event of reporting represents another recommended attribute. This attribute encourages the data collected to be regularly prepared in report form and shared with relevant stakeholders. The findings from our interviews support defining events for monitoring and reporting, which are often performed during recurring agile events. In addition, we advocate defining responsibilities for activities related to metric adoption. A metrics owner is responsible for ensuring sufficient resources, such as budget or staff, to implement the metric. The owner is the recipient of reporting information. Four additional roles can be defined to clarify responsibilities for defining, implementing, monitoring, and achieving the metric. The metric achievement responsible ensures that the specified target values of the metric are achieved on time. One per-

son can take on multiple roles. We recommend designating one person per area of responsibility. The remaining attributes - target value(s), planned value(s), and tolerance value(s) - can be used to determine when to follow the escalation rule. The planned value(s) establish intermediate goals at specific points in time. The escalation rule documents the responsible person's action plan once the tolerance values are reached. Interested stakeholders, the final attribute, may include additional metric stakeholders. If the attribute element is disregarded, several critical elements essential to successfully implementing a metric will remain undocumented [Mon14].

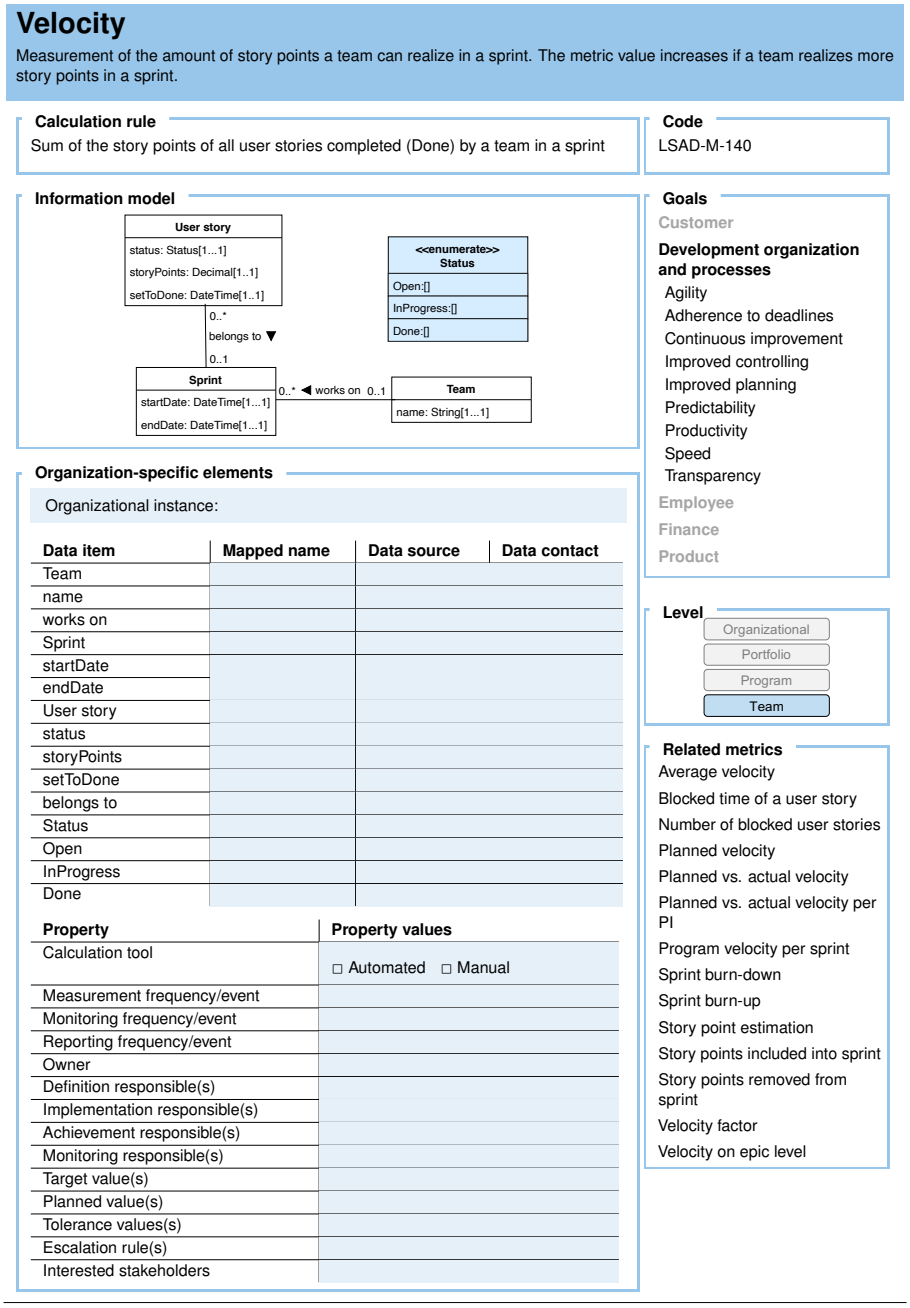
Metric Catalog[1]: We propose a metric catalog design to guide the establishment of metrics in large-scale agile software development environments. The structure of the catalog was significantly influenced by Matthes et al. [MMSS12]. Our catalog begins with an introductory explanation focusing on the motivation for creating the catalog, its purpose, and the intended audience (i.e., practitioners and researchers). We also provide navigation support to ensure that relevant metrics can be retrieved efficiently. The navigation structure includes three different elements to support user navigation. The first element, navigation based on large-scale agile software development goals, uses a Goal-Metric-Matrix analogous to Monahov [Mon14]. The second element, navigation based on the organizational level of the metric usage, corresponds to the organizational level element of the developed minimalistic MMFS. The final navigation element, navigation based on related metrics, can be used to explore metrics based on relationships. The core of the catalog is the collection of 196 metrics we identified during the expert interview study. Each metric is documented using the MMFS, with all related metrics additionally listed and linked for each metric (see Figure 4.2).

Demonstration and Evaluation

During an initial presentation, the artifacts were demonstrated to three experts. All experts confirmed the overall idea of both artifacts, the well-thought-out structure and design of the minimalistic MMFS, and the importance of linking metrics to goals. We then conducted three iterations of demonstrations and evaluations based on surveys. The first two iterations regarding the MMFS and metric catalog involved expert evaluations [PRTV12]. The first iteration was used to solicit feedback during a presentation. This feedback was integrated before the second iteration. In the second iteration, we verified if potential users and key stakeholders found the solution artifacts valuable. In addition to the first two iterations, we conducted a third iteration for the metric catalog, deploying it in two case organizations as part of a systematic metrics implementation and long-term management process. Thus, the evaluation in the third iteration constituted an action research evaluation [PRTV12]. In total, 14 experts participated in the evaluation of the minimalistic MMFS, and 24 experts participated in evaluating the metric catalog.

(1) The artifacts were shown to five experts during the initial demonstration. The experts made four recommendations. First, one of the experts suggested that the calculation rule should be revised in mathematical formulation, as it would be easier to verify and relate to other metrics or assumptions. However, since understanding the mathematical definition depends on the prior mathematical knowledge of the users, we did not consider this suggestion for improvement. Second, an expert suggested that we expand the five goal categories initially proposed for the goal element because they did not provide sufficient granularity for a meaningful connection

[1]Metric Catalog: `https://wwwmatthes.in.tum.de/page/filesandSubPages?id=8attq54y5jaa`



Velocity

Measurement of the amount of story points a team can realize in a sprint. The metric value increases if a team realizes more story points in a sprint.

Calculation rule

Sum of the story points of all user stories completed (Done) by a team in a sprint

Code

LSAD-M-140

Information model

Goals

Customer

Development organization and processes

- Agility
- Adherence to deadlines
- Continuous improvement
- Improved controlling
- Improved planning
- Predictability
- Productivity
- Speed
- Transparency

Employee

Finance

Product

Organization-specific elements

Organizational instance:

Data item	Mapped name	Data source	Data contact
Team			
name			
works on			
Sprint			
startDate			
endDate			
User story			
status			
storyPoints			
setToDone			
belongs to			
Status			
Open			
InProgress			
Done			

Property	Property values
Calculation tool	☐ Automated ☐ Manual
Measurement frequency/event	
Monitoring frequency/event	
Reporting frequency/event	
Owner	
Definition responsible(s)	
Implementation responsible(s)	
Achievement responsible(s)	
Monitoring responsible(s)	
Target value(s)	
Planned value(s)	
Tolerance values(s)	
Escalation rule(s)	
Interested stakeholders	

Level

- Organizational
- Portfolio
- Program
- **Team**

Related metrics

- Average velocity
- Blocked time of a user story
- Number of blocked user stories
- Planned velocity
- Planned vs. actual velocity
- Planned vs. actual velocity per PI
- Program velocity per sprint
- Sprint burn-down
- Sprint burn-up
- Story point estimation
- Story points included into sprint
- Story points removed from sprint
- Velocity factor
- Velocity on epic level

Figure 4.2.: Example Metric Documented in the Metric Catalog with the Minimalistic MMFS (Based on P3)

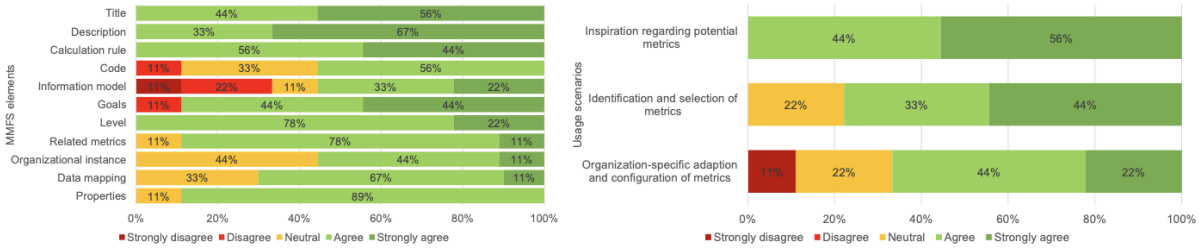


Figure 4.3.: Evaluation of the MMFS Elements and the Catalog's Usefulness in Different Usage Scenarios (Based on P3)

between metrics and goals. We implemented this feedback and expanded the five goal categories to include 27 subgoals. Third, one expert recommended including "a range for every metric showing you if the metric is useful or not useful." We did not follow this advice either because we lacked the required data. The final suggestion for improvement from an expert was to specify the purpose of each metric. We did not add an additional element for this requirement because the second suggested improvement (adding the 27 subgoals) already provided information about the metric's purpose.

(2) The artifacts were presented to nine experts during the second demonstration. After considering feedback from the first iteration of our survey-based assessment, we evaluated whether the minimalistic MMFS and metric catalog were sufficiently valuable to be used in practice or needed further adaptation. We first assessed the relevance and utility of each element of the minimalistic MMFS (see Figure 4.3). The survey results show that each element is considered relevant and useful. There was disagreement among some experts regarding the code, the information model, and the goals. Therefore, we decided not to modify the elements of the minimalistic MMFS. We then presented three potential usage scenarios of the catalog to the experts. The usage scenarios included inspiration regarding potential metrics, identification and selection of metrics, and organization-specific customization and configuration of metrics. Experts were then asked to rate the usefulness of each of these three usage scenarios. Although most experts could envision using the catalog in all three scenarios, using the catalog for inspiration as well as identification and selection were the scenarios with the most positive feedback. Therefore, we chose inspiration regarding potential metrics as well as identification and selection of metrics as the usage scenarios for the third iteration of the demonstration and survey-based evaluation of the catalog.

(3) In the context of our metrics adoption process, we used the catalog in two case study organizations (see Figure 4.4). This process is focused on a goal-based introduction of metrics. The metric catalog was applied in two procedural steps as part of the process. In the second step, teams determined the goal coverage based on the previously identified metrics. To do this, they performed a bottom-up search within the catalog. In addition, in step five, the teams applied the catalog's Goal-Metric-Matrix in a top-down application to create a list of potential metrics and then selected the metrics that best matched the defined goals and questions. We received feedback that some relevant goals (e.g., improve collaboration) and metrics (e.g., security-related story points per sprint) were missing in the catalog. Both of these suggested improvements were

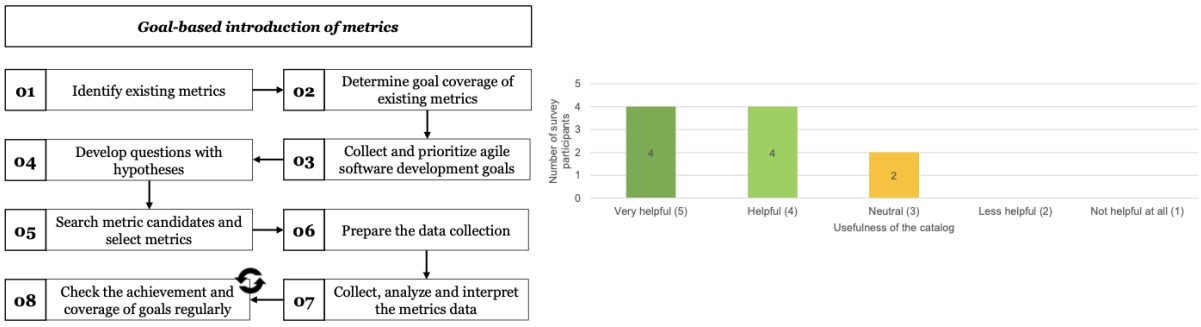


Figure 4.4.: Metric Adoption Process (Left) and Evaluation of the Catalog's Usefulness (Right) (Based on P3)

incorporated into the catalog. The results illustrate that most survey participants considered the catalog to be very helpful or helpful.

Summary of main findings related to RQ4: To answer RQ4, we present a goal-based method designed for a large agile case organization to help stakeholders select, operate, and scale metrics. In addition, based on lessons learned in the case organization, we present design principles that can potentially guide the development of methods for other contexts. We conducted this research using an Action Design Research approach in combination with Situational Method Engineering.

Research Question 4 (RQ4)

How can metrics be systematically selected, operated, and scaled in large-scale agile software development environments?

Method to Select, Operate, and Scale Metrics in Large-Scale Agile Software Development

We have developed a comprehensive metrics selection, operation, and scaling method at a large German software company. This method is facilitated through a web application that digitally represents the developed metric catalog and minimalistic MMFS. A metric expert team executes the activities of the method in close collaboration with relevant stakeholders that have or represent specific information needs. In the following, the web application, the roles within the presented methodology, and the corresponding individual activities are described in detail.

Metric Catalog Web Application:

The Metric Catalog Web Application was developed based on the results of P3. It can be viewed as a comprehensive metrics repository that contains over 200 predefined metrics. The core functions of this web application are the search for potential metric candidates and the documentation of new metrics together with their associated goals. The documentation of a metric is done by specifying general metric attributes (e.g., title or calculation rule) as well as the organization-specific metric attributes (e.g., owner and measurement frequency).

Roles:

Performing the activities of the method requires the collaborative efforts of two essential roles: the metric expert team and the stakeholders. Both roles are interdependent. The metric expert team requires the stakeholders' specific knowledge of their agile software development context. In contrast, the stakeholders depend on the metric expert team because they generally do not have the expertise and resources necessary to implement metrics effectively and efficiently. The method is suitable for different stakeholder roles, allowing for both a top-down (i.e., by a Domain Manager) and a bottom-up (i.e., by a Scrum Master) adoption of metrics. Both roles are explored in more detail below.

Stakeholders: A stakeholder is an employee within the case organization who has or represents an information need. In a metaphorical view, stakeholders could be seen as information consumers who consume measurements as a service the metric expert team provides. Once stakeholders can operate the metrics independently, the metric expert team reduces its involvement and acts as an ad-hoc resource. The method's success can be measured primarily by the usefulness of the adopted metrics in achieving the stakeholders' goals. Stakeholders can hold different agile roles (e.g., Scrum Masters, Product Owners, Program Managers) and participate at different organizational levels (e.g., Team, Program, Portfolio, Organization).

Metric Expert Team: The metric expert team should consist of dedicated experts characterized by long-term availability, a sound understanding of metrics adoption as a socio-technical problem, excellent communication and collaboration skills, and extensive knowledge of the agile way of working. As a sparring partner to stakeholders, the metric expert team must precisely understand each method activity and its rationale. Furthermore, the metric expert team strives to increase the value of the method by scaling it within the organization. Scaling can be done, for example, by sharing success stories within Communities of Practice (CoPs) - a recurring event that invites stakeholders with similar interests to share their knowledge. Moreover, the metric expert team should be able to inspire stakeholders who are reluctant to change with a standard set of metrics that are typical for the role profile of the respective stakeholder. For example, in the case companies, it appeared that Scrum Masters were particularly interested in process metrics, Product Owners and Developers were interested in product metrics, and Program Managers were interested in all metrics (i.e., Product, Resource, and Process). With an inventory of more than 200 standard metrics, the Metric Catalog Web Application could potentially assist the metric expert team in finding a relevant set of metrics.

Method Main Activities

To improve clarity, an aggregation of the 20 activities was made in a simplified diagram, which can be viewed in Figure 4.5. In this diagram, the activities were divided into the three primary activity categories of metric selection, operation, and scaling. A comprehensive Unified Modeling Language (UML) activity diagram illustrating all activities, their interrelationships and their links to the Metric Catalog Web Application can be found in P4.

Metric Selection:

- *Ensure Management Commitment:* Commitment from management, such as the Program Manager, is essential for the method's initial and subsequent iterations. Management commitment includes the provision of required resources, such as the sponsorship of a dedicated metric expert team that serves as a valuable resource to stakeholders throughout the execu-

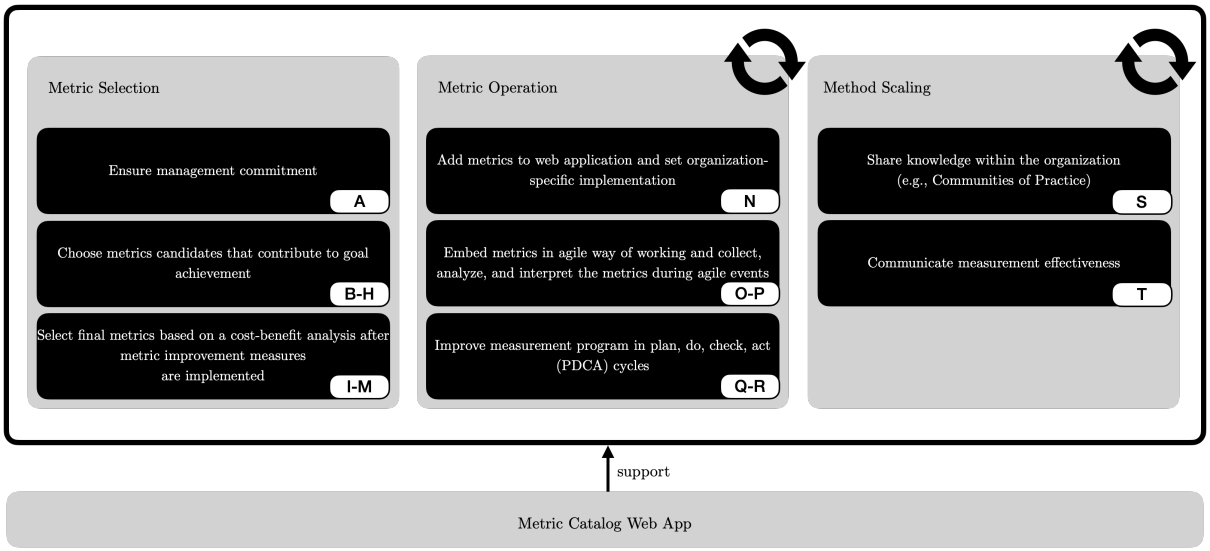


Figure 4.5.: Method to Manage Metrics in Large-Scale Agile Software Development (Based on P4)

tion of the method activities. Furthermore, the proclamation of the goal to transform into a metrics-driven organization might encourage stakeholders to fulfill their responsibilities along the method activities. However, the method will be discontinued if a reassessment concludes that it is no longer viable and leadership withdraws its commitment.

- *Choose metrics candidates that contribute to goal achievement:* Linking metrics to strategic goals ensures that only relevant metrics are adopted. Metrics facilitate goal achievement by making progress toward a goal more transparent or by serving as conditions for meeting a goal, i.e., as metric target values. Goals can be collected from multiple sources, including documents, tools, or stakeholders. The accumulated goals must be documented in the Metric Catalog Web Application. Afterward, a measurement priority is established for each goal. The measurement priority is based on two factors. The first factor, metric potential, characterizes the potential of a metric to contribute to the realization of the goal in question, for example, by providing visibility into whether or not goal realization is likely. The second factor is defined by goal priority, which describes how essential the goal is to fulfill the business strategy. Discussions can be leveraged using Likert scales [Lik32] to estimate metric potential, goal priority, and measurement priority. We consider a detailed estimation process to be unnecessary and potentially inefficient, as discussions have been shown to be effective in the case organization. Following the GQM approach [Bas92], questions and hypotheses are formulated for the selected goals (i.e., goals with the highest measurement priorities). Finally, the Metric Catalog Web Application is used to search for metric candidates.
- *Select final metrics based on a cost-benefit analysis after improvement measures are implemented:* A cost-benefit analysis is used for each candidate metric to decide if it is selected. The benefit of a metric is determined by exploring its metric potential through discussion. On the cost side, several variables, such as data collection and data quality improvement,

are relevant depending on the metric context. Discussion sessions can again be used to estimate benefits and costs. These discussions should also evaluate the potential for cost improvement measures. However, since any improvement measure is also associated with costs, implementation should only occur if the measure's potential to reduce metric costs exceeds the implementation costs.

Metric Operation: The activities below are performed regularly and are aligned with the cadence of agile events:

- *Add metrics to the web application and set organization-specific implementation:* The final set of metrics is added to the Metric Catalog Web Application. In this context, the corresponding goals are assigned to the respective metrics, and the general as well as organization-specific elements are assigned for each metric.
- *Embed metrics in agile way of working and collect, analyze, and interpret the metrics during agile events:* Metrics should be assigned to agile events that correspond with the most appropriate purpose. For example, planning metrics (such as velocity) are particularly appropriate for sprint planning events, while sprint retrospectives provide an appropriate forum to discuss team happiness metrics. Before each event, data collection should be completed to allow for analysis and interpretation of the data. Moreover, declaring the metric discussion as a regular agenda item can help ensure that metrics become an integral part of the agile development process.
- *Improve measurement program in plan, do, check, act (PDCA) cycles:* Continuous execution of PDCA cycles to improve the measurement program is necessary to increase its value over time. Key objectives for the PDCA cycles could be, for example, to optimize the quality of individual metrics or to ensure the relevance of the current set of metrics.

Method Scaling: The following activities are repeated regularly.

- *Share knowledge within the organization (e.g., Communities of Practice):* Another effective strategy for adding value to the measurement program is to scale it at the organizational level. This process could lead to economies of scale, as implementing new metrics could lead to reduced long-run average costs over time (e.g., through increased efficiency of the metric expert team). Scaling can be achieved by attracting new stakeholders to participate in the measurement program. An effective tactic for promoting the measurement program could be sharing success stories about how other stakeholders have benefited from the metrics. Communities of Practice (e.g., Scrum Master CoP) could serve as a platform for efficient communication with various potential stakeholders.
- *Communicate measurement effectiveness:* Regular reporting on the effectiveness of the measurement program provides an important basis for management to decide whether to continue or withdraw its commitment. Therefore, qualitative and quantitative feedback should be obtained and shared continuously to assess how valuable stakeholders consider the measurement program to be.

We subsequently present ten design principles derived from the leanings gained during method development. These principles could potentially serve as guidelines for developing analogous methods in other organizations.

- *Design Principle 1:* Ensure management commitment: Management must provide the resources necessary to implement the method. In addition, management should set and communicate becoming a metric-driven organization as a goal to motivate stakeholders to fulfill the responsibilities under the method activities.
- *Design Principle 2:* Establish a metric expert team who views stakeholders as customers: The metric expert team should view stakeholders as customers and measurements as a service. If stakeholders can operate the metrics on their own, the metric expert team should reduce the involvement and act only as an ad-hoc resource.
- *Design Principle 3:* Build stakeholder understanding: The metric expert team must ensure that stakeholders understand the metrics they are using. Stakeholder understanding can be achieved by involving stakeholders in setting goals, questions, and hypotheses.
- *Design Principle 4:* Align metrics with agile software development goals: Metrics must align with agile software development goals derived from the business strategy to ensure coherence.
- *Design Principle 5:* Conduct a cost-benefit analysis to select metrics: The metrics with the best cost-benefit ratio should be selected first. On the cost side, ensuring information quality has been found to be a major factor, while on the benefit side, measurement priority serves as the main decision criterion. The cost-benefit ratio can be improved by implementing improvement measures (e.g., improving information quality through defining Jira best practices).
- *Design Principle 6:* Provide a metric repository: Searching for potential metrics and maintaining the selected set of metrics using a software solution can increase the method's efficiency. Search efficiency can be optimized by providing search and filtering functionalities. Maintainability is improved by providing standard functions for creating, reading, updating, and deleting metric attributes.
- *Design Principle 7:* Align metric operation with the agile way of working: Adopting the measurement program should not cause the benefits of agile methods to be lost by disrupting the agile way of working. In particular, the measurement program must be aligned with the agile process (e.g., development cadence, agile events, and agile roles).
- *Design Principle 8:* Maintain and improve the measurement program: The measurement program must be continuously maintained and improved. At its core, the quality and relevance of the selected metrics must be constantly ensured. Relevance and quality are critical factors in ensuring the added value of the measurement program for stakeholders.
- *Design Principle 9:* Scale the measurement program: Its sole maintenance and improvement limits its value to the current stakeholders. Therefore, scaling the measurement program could be an option to increase its value as more stakeholders can benefit from it. In addition, scaling the measurement program within the organization can lead to economies of scale that reduce the cost of implementing additional metrics.
- *Design Principle 10:* View the measurement program as an investment: the measurement

program should only be continued if the expected future value justifies the investment required to run it.

Summary of main findings related to RQ5: To answer RQ5 we conducted a single-case embedded study at the IT department of a large German car manufacturer. We explored how goals are set and documented within eight programs at a large German automobile manufacturer. Moreover, we identified and categorized goals, collected goal-setting challenges, and formulated seven mitigation propositions to address these challenges. Finally, we evaluated these mitigation propositions and discussed the evaluation results by incorporating the qualitative feedback provided by the interviewees.

Research Question 5 (RQ5)

How are goals set and documented in a large agile organization, which goals exist, what challenges have been encountered during goal-setting, and how can these challenges be addressed?

Goal-Setting Approaches

(1) Goal Management Process (GMP): A GMP is implemented throughout the organization to break down the strategic goals along the scaling levels. This process is mandatory in all areas, regardless of the agile methods or scaling agile frameworks used.

At the highest level, strategic goals are defined by the board and strategic management circles with the support of a balanced scorecard. A Strategic Management Circle is a focus group for issues irrelevant to the entire board. It comprises representatives of the Board of Management and the management level immediately below. In the Strategic Management Circle for the IT department, for example, the Chief Executive Officer, Chief Operating Officer, and Chief Information Officer from the Executive Board meet with the heads of the IT department to discuss IT-related issues that are important for strategic decision-making.

Breaking down the goals is done through the Product Owner structure, as shown in Figure 4.6. Here, workshops are a means for Product Owners to meet with lower-level Product Owners to discuss sub-goals and their distribution across the lower levels. This approach applies to each scaling level. The GMP also has bottom-up input from agile teams, which is especially helpful in goal-setting to clarify how the goals will be achieved. In contrast, the top-down process focuses more on what should be achieved.

Workshops are held regularly to ensure alignment between top-down goals and bottom-up input. Before the case study organization implemented agile methods, the GMP was held once a year. Currently, strategic goals are set annually and can span multiple years. Similarly, goals for individual areas are derived annually. However, the frequency of deriving product-level goals varies by product. Generally, this is done every quarter. At the Product level, the derivation of goals depends on the development rate of the agile teams. For example, when Scrum is applied at the Team level, the derivation of sprint goals may depend on the length of the sprint. Figure 4.7 shows how the GMP process is implemented in an exemplary case organization program.

(2) Dual-track agile goal-setting process: Despite mandatory compliance with GMP, specific

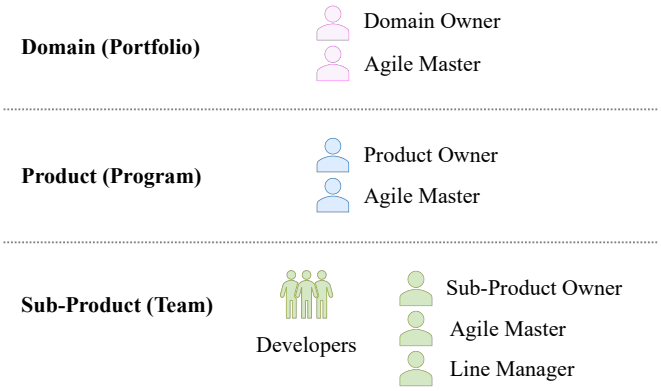


Figure 4.6.: Overview of Scaling Levels at the Case Organization [SS18] (Based on P1)

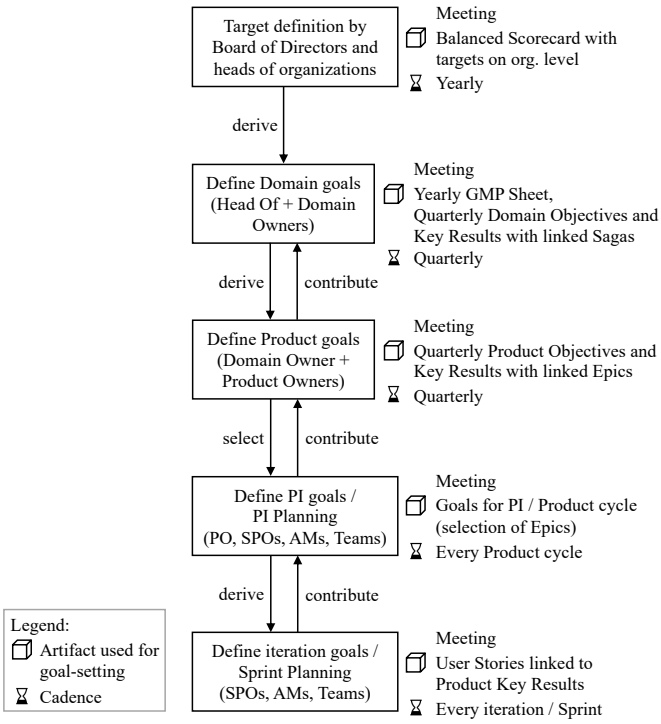


Figure 4.7.: Overview of GMP (Based on P1)

goal-setting practices differ across the case organization. For example, in one program at the case organization, a dual-track agile goal-setting process is implemented. This process defines how the program implements goal-setting within the boundaries of the GMP. The process focuses on defining the program goals for the next quarter and decomposing the defined goals into iteration goals. Active participation of agile teams is guaranteed throughout the process. Their input is especially important for product-related goals. During the goal collection, higher-level GMP goals and goals gathered from the agile teams, customers, partners, or other sources are included. Subsequently, the Product Owners reconcile all recorded goals. If a goal is assessed as desirable, corresponding Sagas are derived and documented in the Product Backlog. The agile teams then conduct initial research and ideation to identify possible solutions to the Saga(s). If necessary, a prototype is created to better understand the problem. The results of the research phase and potential solutions are recorded as Epics in the Program Backlog. Together with the Product Owners, the agile teams estimate the initial effort for the Epics. Finally, a "Vision, Roadmap and Direction Focus Day" is held before the start of the next quarter. At this event, led by the Product Owner, the stakeholders jointly prioritize and plan the Epics for the upcoming quarter based on their goals. Afterward, the agile teams define and specify their individual sprint goals, which are aligned with the Epics for the upcoming quarter. Compared to the GMP implementation, the dual approach does not use Objectives and Key Results (OKRs) and requires more planning and exploration.

Goal Definition and Documentation Techniques

(1) Pre-defined templates for goal documentation (GMP Sheets): GMP sheets must be used to document the goals of the GMP at the Domain level. The GMP sheets are used for goal definition, review, and reporting. The respective Product Owner at the corresponding scaling level is responsible for completing the GMP sheet. Although using GMP sheets at scaling levels below the Domain level is not mandatory, their adoption varies across the case organization. The templates differentiate between performance and change goals. While performance goals represent operational targets and must be quantitative and measurable, change goals illustrate necessary modifications within the case organization to meet the overall strategy. The GMP sheet recommends documenting each goal's name, summary, and goal achievement date. Moreover, for quantitative goals, the target value is recorded.

(2) Backlogs with a linked chain of sub-goals: Backlogs are used within all programs investigated at the case organizations. Typically, these backlogs are managed using tools such as Jira or CodeBeamer. Numerous experts see it as critical to create a linked chain of sub-goals within the backlogs across the different levels of scaling. The top-level backlog items, often referred to as Sagas, are set at the Domain level and contribute to the long-term goals of that domain. One level below, at the Product level, the middle-level backlog items known as Epics are set. Each Epic should be linked to the higher-level Saga to which it contributes. The same logic applies to the Sub-product level: each User Story must be linked to its corresponding Epic. This process not only promotes the breakdown of goals but also improves traceability of how the completed work contributes to the higher-level goals.

(3) SMART technique: The SMART method is mandatory in the context of the GMP. This method ensures that each goal is specific, measurable, achievable, relevant, and time-bound.

Table 5.6. Overview of Goal Categories (Based on P1)

Goal Content	Goal Content Types
Product Goals	Goals related to artifacts, documents or deliverables produced during the lifecycle of the offered product or service. Based on Product goals from GQM (Basili et al. 1994).
Process Goals	Goals related to internal work processes of the organization, typically associated with time. Based on Process goals from GQM (Basili et al. 1994).
Resource Goals	Goals related to resources of the organization that are used by processes for product development. Based on Process goals from GQM (Basili et al. 1994).
Strategic Goals	Long-term goals, usually relevant to multiple programs and products, that steer decisions and overall direction of the organization.
Legal, Security & Compliance Goals	Obligatory goals that must be attained; often set / defined by entities outside of the organization.

(4) Objectives and Key Results (OKRs): The OKRs are documented utilizing special tools within the organigram, which enables visibility across all departments for all employees who use these tools. The implementation of OKRs can be decided individually by each department. Below is an explanation of how OKRs are implemented in a particular program. Domain goals provide the basis for deriving quantitative quarterly goals. The goal definition process involves the domain owner and key external stakeholders. In this process, the definition is not exclusively top-down. Agile teams can contribute their issues as bottom-up input to the discussion. However, this bottom-up input is only considered up to the main department, not at higher levels. Once the goals are defined, the quantitative Key Results are set for the upcoming quarter. Product Owners are tasked with defining the product OKRs based on the domain OKRs. A maximum of five Objectives and four Key Results per Objective are allowed at the Domain and Product level. Furthermore, each Key Result must include a Key Performance Indicator. Backlog items are linked to Key Results to assess the completed backlog items quantitatively. When all assigned Backlog Items are processed, the corresponding Key Result is considered achieved. As soon as all Key Results are fulfilled, the set Objective is achieved.

Identified Goals

We identified 51 goals (see Table 5.7) in the case organization and categorized them according to goal content and scaling level dimensions. For goal content, we selected Product, Process, and Resource categories and added two additional categories, namely Strategic and Legal, Security and Compliance. Regarding the scaling level, we relied on the levels shown in Figure 4.6 and added the organizational categories of Enterprise and Line Management to better represent our goal data. Table 5.6 illustrates the goal content dimension and associated goal content types. In analyzing the goal content, we found that most identified goals were classified as either Process goals (22 goals) or Product goals (15 goals). In addition, we recorded seven goals as Strategic goals, four as Resource goals, and three as Legal, Security, and Compliance goals. Process and Product goals occur with approximately equal frequency at both the Domain and Product level. Regarding scaling level, most goals occurred at the Product level (26 goals), followed by the Domain level with 21 goals. At the Sub-product level, we identified only one goal. Furthermore, we identified seven goals that are relevant to the Entire organization and two goals that are specifically relevant to Line Management.

Table 5.7. Identified Goals (Based on P5)

ID	Goal	Category	Scaling Level	Source
G1	System stability	Product	Product	B2, B3, C1, C2
G2	Customer value	Product	Product	B1, B3
G3	Customer satisfaction	Product	Domain	B2, B3
G4	Service availability	Product	Product	B2, C1
G5	Transition to cloud	Strategic	Enterprise	B2, C2
G6	Process efficiency	Process	Domain	C2, D1
G7	Process standardization	Process	Enterprise	C2, D1
G8	Release of product	Strategic	Domain, Product	A1, C2
G9	New market entry	Strategic	Enterprise	D1
G10	Product enablement	Process	Domain	B2
G11	Security	Legal, Security & Compliance	Product	B2
G12	Scalability	Product	Product	B2
G13	Service measurability	Product	Product	B2
G14	Reduce cost	Resource	Product	B2
G15	Increase employee engagement	Process	Domain	B2
G16	Reduction of days-to-sell	Process	Domain	B1
G17	Legal compliance	Legal, Security & Compliance	Domain, Product	B2, C1, C2, D2
G18	From specialist roles to Feature Teams	Process	Product	B3, C1
G19	Transition from car leasing to mobility leasing	Strategic	Enterprise	D1
G20	Knowledge transfer and community enablement	Process	Domain	B2
G21	System performance	Product	Product, Sub-Product	B2
G22	Clear definition of rights and responsibilities of each role	Process	Product	B3
G23	From coordinate to integrate to coordination through integration	Process	Product	B3
G24	From independent teams to continuous cross-team cooperation	Process	Product	B3
G25	From organizing around technology to organizing around customer	Process	Product	B3
G26	From resource thinking to people thinking	Process	Product	B3
G27	Improve quality	Product	Domain	B2
G28	Increase productivity	Process	Product	B2
G29	Decrease time to market	Process	Product	B2
G30	Improve collaboration	Process	Domain	B2
G31	Return on investment	Resource	Product	B3
G32	From project to product	Process	Product	B3
G33	Reduce lead-time	Process	Product	B3
G34	Serviceability	Product	Product	B3
G35	Market share	Strategic	Enterprise	B3
G36	Upgradeability	Product	Domain, Product	B2
G37	Always fresh	Product	Product	B2
G38	Verification of product	Product	Product	A1
G39	Reduction of complexity	Product	Domain	C1
G40	Replacing old IT system	Product	Domain	C1
G41	Generate customer leads	Process	Product	B1
G42	Flexibility	Process	Domain	B2
G43	Pursue agile working model	Process	Domain	B2
G44	Customer-orientation via user journeys	Product	Domain	C1
G45	Product consolidation and harmonization	Strategic	Domain	B1
G46	Reduce days employees call in sick	Resource	Domain, Line Organization	C2
G47	Euros spent per employee	Resource	Domain, Line Organization	C2
G48	Create a new roles & rights concept	Process	Domain	D2
G49	Implement strong authentication	Legal, Security & Compliance	Enterprise	D2
G50	Improve structure of requirements	Process	Domain	B2
G51	Competitiveness in the market	Strategic	Enterprise	B2

Table 5.8. Identified Goal-Setting Challenges (Based on P1)

ID	Construct	Name	Description	Interviewees
C1	Interdependencies	External dependencies limiting autonomous goal-setting	External dependencies like stakeholder commitment to goals, dependencies on other products or teams make it hard to define clear goals that can be achieved autonomously by the Product / Domain.	PO1, STE1, LM1, PO3
C2	Interdependencies	Prioritization conflicts between goals	Balancing different needs of stakeholders.	LM1, AM1, AM4, AM5
C3	Reporting	No goals for individual employees	Reporting on individual employees is not allowed by workers union or too resource- intensive in large programs.	PO1, STE1
C4	Adaption	Management control limits team autonomy	Fixed yearly / quarterly goals limit autonomy of Product Owners and Agile Teams. Reporting demanded by management is perceived as intrusive by Agile Teams.	AM4, AM5, STE1, DEV1, PO2
C5	Communication	Unclear goals from higher levels	Goals received from higher org. levels are not clearly defined and explained to all stakeholders.	AM2
C6	Communication	Define current state and target state for qualitative goals	For qualitative, non-measurable goals it is often hard to clearly define the current state and the target state to be achieved.	AM2
C7	Commitment	Missing attachment of teams to goals	Agile Teams lack attachment and commitment to goals they did not define themselves.	AM4
C8	Adaption	Too rigid fixation on goals	Focusing on goals causes lack of appreciation for necessary routine / operational work.	AM3, LM1
C9	Interdependencies	External contracts limiting Feature Team working model	External contractors are not allowed to operate as cross-functional Feature Teams. Instead, they own specific components only.	AM3
C10	Communication	Missing link between organizational goals and realization	Organizational goals often do not clearly define what is to be done and implemented to achieve the goal.	AM3
C11	Lack of Investment	Resource constraints for goal-setting and reporting	The goal-setting and reporting processes consume a lot of resources.	BE1, PO3, DEV1

Goal-Setting Challenges

In total, we identified eleven challenges (see Table 5.8) and assigned these challenges to the constructs Interdependencies, Reporting, Adaption, Communication, Commitment, and Lack of Investment. Especially in the areas of Communication and Interdependencies, we encountered the most problems. The most mentioned challenges related to goal-setting in the case organization are external dependencies limiting autonomous goal-setting, prioritization conflicts between goals, and management control that limits team autonomy.

Propositions to Mitigate Goal-Setting Challenges

After carefully reviewing the academic literature and analyzing our interview data, we identified a collection of mitigation propositions that might be appropriate for addressing the designated goal-setting challenges. In the analysis phase, we consolidated similar mitigation propositions and eliminated duplicates, resulting in seven mitigation propositions (see Table 5.9).

(M1) Goal-setting responsibility should be shared among actors to facilitate collaborative goal-setting practices: We recommend that all roles in Agile development (i.e., Agile Teams, Agile Masters, Product Owners, and Line Managers) should be allowed to contribute to and propose goals of any type. This proposition facilitates collaborative goal-setting activities and practices. Therefore, this proposition addresses the challenges of agile teams' lack of attachment to goals (C7) and unclear goals received from higher levels (C5). Additionally, this proposition is sup-

ported by literature findings that highlight the importance of shared goals and collaborative goal-setting [BMS19, MDS+19]. Similarly, Schnabel and Pizka [SP06] emphasize collaborative goal-setting in their processes. Classical goal-setting theory also postulates improved group performance due to collaborative goal-setting [LL06]. Implementing this proposition can address challenges C5 and C7 and thus improve Communication and Commitment.

(M2) Goal definition practices (e.g., SMART, OKRs, GQM) ensure a clear understanding of goals: It is advisable that the program agrees on methods for defining goals and establish a Definition of Ready (DoR) (cf. Power et al. [Pow14]) for goals, analogous to the Definition of Done (DoD) for backlog items in Scrum. In our study, we identified OKRs, the SMART technique, and program press releases that describe the expected situation in program D1 once a specific goal is achieved. In addition, we postulate that goal definition practices can overcome the challenges of unclear goals (C5) and the challenges associated with defining the current state and target state for qualitative goals (C6). By implementing this mitigation proposition, challenges C5 and C6 can be overcome, leading to improved Communication.

(M3) A linked chain of sub-goals across scaling levels should be established to facilitate transparency: We suggest linking each goal or backlog item to the goals at the next higher scaling level. This proposition can address the problem of lack of linkage between organizational goals and their realization (C10). In addition, according to Berntzen et al. [BMS19], breaking down goals into a hierarchy can improve coordination. Implementing this mitigation proposition can address the challenge C10 and facilitate Communication.

(M4) Goals should be documented publicly for all actors and stakeholders to facilitate transparency: We have already presented the practice of the GMP sheet within the case organization as a method of documentation that implements our proposition. Another documentation practice within the case organization is the Confluence-based enterprise wiki, which is used to document goals in several programs. We claim that the problem of unclear goals (C5) can be solved by documenting goals publicly. Since implementing this mitigation proposition can address challenge C5, Communication can be improved.

(M5) Goals from external stakeholders should be broken top-down along the Product Owner hierarchy to ensure consideration of dependencies and coordination: This proposition can address the challenge of external dependencies limiting the autonomous goal-setting of teams (C1). This proposal is consistent with Berntzen et al.'s [BMS19] finding that the Product Owner is essential in coordinating the large-scale agile development effort. Furthermore, deriving subgoals from higher-level goals allows teams to implicitly adhere to higher-level dependencies. Implementing this mitigation proposition can counter challenges C1 and C2 so that issues related to Interdependencies can be resolved.

(M6) Definition, prioritization, and communication of middle- to lower-level goals should involve Agile Teams to ensure consideration of technical aspects and acceptance of goals by the teams: By implementing this proposition, the problem of missing attachment between teams and goals (C7) can be effectively addressed. In this regard, Schnabel and Pizka's [SP06] concept of "top-down thinking and bottom-up action" is similar to the combination of our propositions M5 and M6. In contrast to Schnabel and Pizka [SP06], who only differentiate between stakeholders ("top") and developers ("bottom"), we take a more detailed view and consider multiple scaling levels and

roles that are specifically relevant for large-scale agile development. Moreover, unlike Schnabel and Pizka [SP06], we argue that agile teams should not only act according to predefined goals ("bottom-up acting") but also take an active role in defining these goals ("thinking"). We are convinced that implementing this proposition can effectively address challenges C2, C4, and C7. Consequently, Interdependencies can be optimized, and both Adaption and Commitment can be improved.

(M7) All goals should be maintained in Backlogs to facilitate clear understanding and transparency: Schwaber and Sutherland [SS20] emphasize that only transparently documented artifacts can provide a basis for transparent decisions and goal-setting. The predominant practice for documenting goals is the backlog, which is used in all of the investigated programs in our case organization. In addition, several scaling agile frameworks recommend the use of backlogs (e.g., Larman and Vodde [LV16]; Scaled Agile Inc. [SA22]; Schwaber [Sch18]). The use of a backlog can address the problem of ill-defined goals from higher organizational levels that are not sufficiently explained to all stakeholders (C5). Consequently, implementing this risk-mitigating measure can combat challenge C5 and thus optimize Communication.

Evaluation of Mitigation Propositions

As part of our research, we conducted eleven semi-structured interviews to evaluate our mitigation propositions. We intended to determine the extent of practitioner agreement with our mitigation proposals and collect qualitative feedback. We asked interview participants to what extent they accepted our suggestions using a five-point Likert scale [Lik32]. Respondents were given the opportunity to provide additional thoughts on each proposition. Table 5.9 provides a comprehensive overview of the evaluation results. The evaluation results show that the experts agreed with most of the propositions. In this regard, proposal M4 received the most favorable evaluation results, while M7 met with the lowest level of agreement. Regarding propositions M5 and M7, no clear consent among the experts exists.

Table 5.9. Identified Mitigation Propositions (Based on P1)

ID	Description	Addressed Challenges	Explanation
M1	Goal-setting responsibility should be shared among actors to facilitate collaborative goal-setting practices	C5, C7	All actors and stakeholders can define goals of any kind (but not prioritize them); Based on statements by LM1, AM5; Literature finds shared, collaborative goal-setting facilitates group performance (Locke and Latham 2006) and coordination in LSAD (Berntzen et al. 2019; Moe et al. 2019; Schnabel and Pizka 2006)
M2	Goal definition practices (e.g., SMART, OKRs, GQM) ensure a clear understanding of goals	C5, C6	SMART is used in programs B1, B2, C1, D1; OKRs is used by programs B2, C1, C2, D2; The used goal definition technique should be documented in the Definition of Ready document
M3	A linked chain of sub-goals across scaling levels should be established to facilitate transparency	C10	Link team goals to program goals, and program goals to portfolio goals; Commonly implemented with linked Backlog Items, using Stories (team), Epics (program), Sagas (portfolio); Jira or similar solutions are used by programs A1, B2, C1, C2, D1; Berntzen et al. (2019) make a similar observation
M4	Goals should be documented publicly for all actors and stakeholders to facilitate transparency	C5	Usage of public documentation, e.g., Wiki-pages, allows everyone to access the most recent goals (AM2, AM3, AM5, PO3, DEV1, LM1, STE1); Ensures transparency and mitigates unclear goals
M5	Goals from external stakeholders should be broken top-down along the Product Owner hierarchy to ensure consideration of dependencies and coordination	C1, C2	Breaking down goals for the next lower level should be done collaboratively with Product Owners from both levels; e.g., via regular workshops according to development cycles similar to GMP process at case org.; Influenced by findings on importance of POs for coordination by Berntzen et al. (2019)
M6	Definition, prioritization, and communication of middle- to lower-level goals should involve Agile Teams to ensure consideration of technical aspects and acceptance of goals by the teams	C2, C4, C7	Agile Teams should always be involved in the 'how' of goals (AM2); E.g., via participation in workshops for breaking down goals in addition to participation in Refinements and Plannings on team-level; Murphy and Cormican also suggest to involve developers (Murphy and Cormican 2015)
M7	All goals should be maintained in Backlogs to facilitate clear understanding and transparency	C5	Goals of all types are documented and maintained in Backlogs by programs A1, B1, B2, B3, C1, C2, D1, and D2

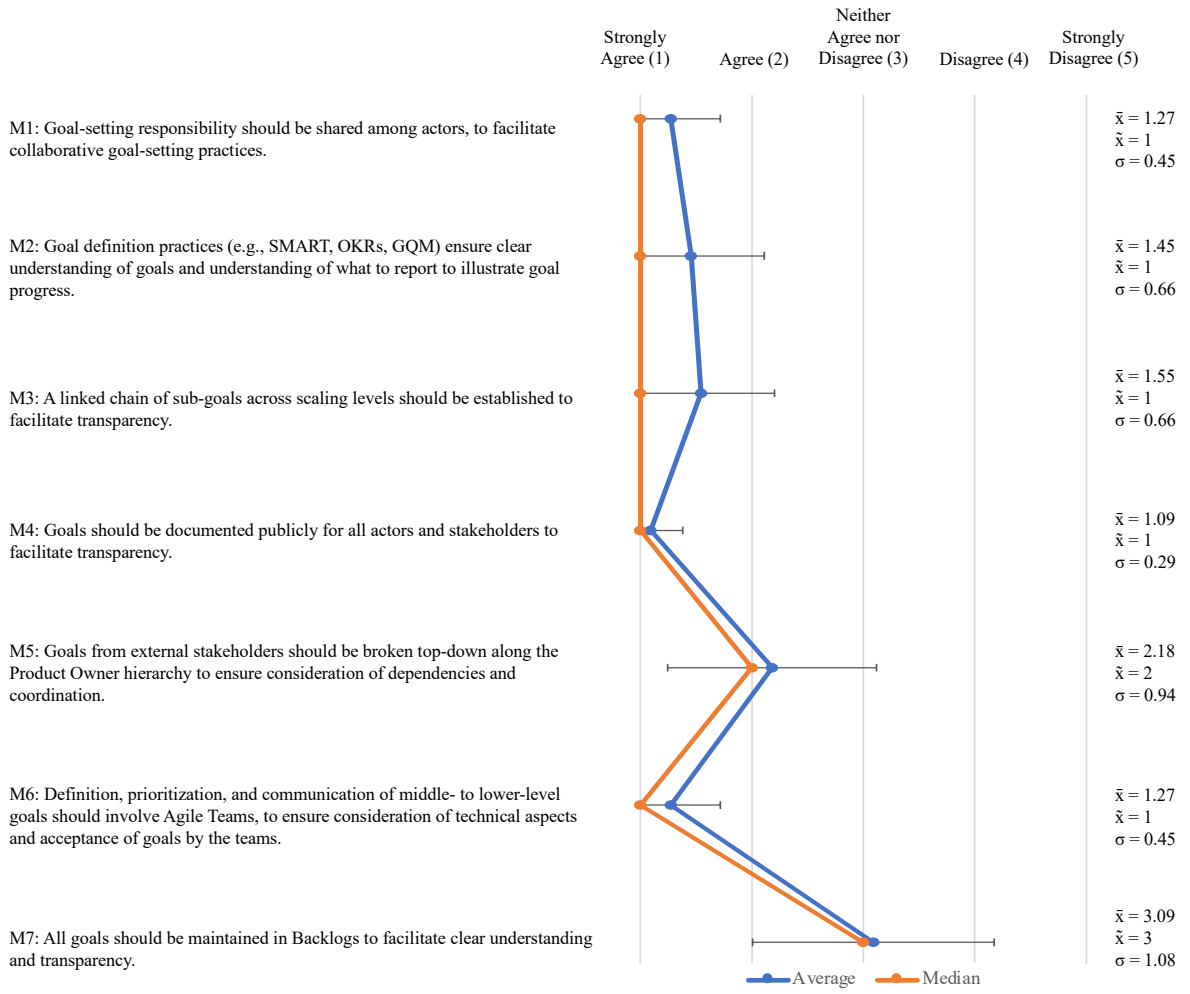


Figure 4.8.: Evaluation of Mitigation Propositions (Based on P1)

CHAPTER 5

Discussion

In this chapter, we discuss our findings in relation to current literature.

Discussion related to RQ1:

Six key findings can be derived from our results. First, metrics are mainly implemented to increase transparency within the development organization at the program and team level. This observation is consistent with the studies of Ebert et al. [EDBS05] and Jethani [Jet13], who argue that metrics can be used to monitor the software process. Furthermore, we found that metrics provide transparency regarding progress tracking, similar to Kupiainen et al. [KMI15]. Our results show that experts used release and epic burn-down charts to determine the progress of the program and deviations from the plan. Second, in contrast to Basili et al. [Bas92, BLR+10], who emphasize the importance of aligning metrics with goals, using metrics for goal-setting was among the least cited reasons. Only some experts reported using metrics to define targets and provide information on the progress toward meeting goals. Third, our results suggest that in scaling agile environments, metrics are more likely to be used to fulfill traditional use cases from general management such as steering, controlling, reporting, early warning, and identifying problems [KN92, Küt11, NAK02]. Fourth, however, primarily agile metrics are used at the team level. We infer from this finding that organizations tend to refrain from implementing traditional metrics at the team level because they can undermine the cultural shift toward an agile way of working [HD06]. This finding is also supported by our fifth key finding, which shows that the most commonly used metrics by the organization are estimation of user story points, velocity, and sprint burn-down, which is consistent with the results of Kupiainen et al. [KMI15], who identified velocity and effort estimates as the most important metrics in the agile industry. Furthermore, this finding is consistent with Schwaber [Sch04] that metrics such as the burn-down chart or velocity are used within individual agile teams. Also, according to the State of Agile Report by Digital.ai [Dig20], velocity and iteration burn-downs are the most commonly used metrics

in practice. Sixth, most companies use program-level metrics. This finding is consistent with Kettunen et al. [KLFM19], who report that program metrics such as cycle time, release cycle, feature and epics lead time, and defects are the most commonly used metrics. Furthermore, this finding is consistent with previous research showing that managers operating at higher organizational levels need metrics to improve their decision-making [Küt11, Mil88, NAK02].

Discussion related to RQ2:

We extend the existing literature by thoroughly investigating the challenges and success factors in scaled agile environments. In contrast to previous studies that focus predominantly on isolated challenges or individual success factors, our research provides a comprehensive analysis of challenges and success factors based on empirical data from various large agile organizations. We identified most challenges regarding data collection, lack of metrics usefulness, and challenges in metrics calculation. Korpivaara et al. [KTS21] and Ram et al. [RRO18] also identified challenges during data collection. Korpivaara et al. [KTS21] identified issues related to the time and effort required for data collection, insufficient data quality, and data availability as problems. Ram et al. [RRO18] mention a lack of instruments as a cause of problems in data collection. Regarding the usefulness of metrics, Oza and Korkala [OK12] and Korpivaara et al. [KTS21] found that strong external factors influence metrics to the extent that their outcome can no longer be controlled. Regarding metrics calculation, Korpivaara et al. [KTS21] state the lack of skills and guidance as a problem. Our findings suggest that adopting metrics in scaling agile environments is challenging because tools that can facilitate data collection and calculation are often missing or used for other purposes. However, since many of the expert organizations that participated in the study are in the early phases of agile transformation, the challenges can also be attributed to a low maturity level of metrics that may be improved in a later phase of the transformation. The success factors with the most occurrences are contextual adoption of metrics, implementing an agile metrics management process, and ensuring an understanding of the purpose of metrics. These success factors are consistent with the findings of previous research. Regarding contextual metrics adoption and implementing an agile metrics management process, Ram et al. (2019) emphasize adopting flexible development processes to accommodate new practices and enable metrics adoption. Additionally, Korpivaara et al. [KTS21] and Ram et al. [RROMF19] highlight the importance of an iterative and agile approach to the metrics program, which includes discussions and short feedback cycles. With regard to ensuring an understanding of the purpose of metrics, Oza and Korkala [OK12] and Ram et al. [RROMF19] state that metrics can be used for trend analysis. The elicited success factors suggest that metrics adoption fails when metrics are introduced without further customization. We recommend careful and ongoing assurance of transparency, review, and customization of the metric and its context and ensuring an understanding of why metrics are being implemented. These factors are critical to the success of metrics in large-scale agile development.

Discussion related to RQ3:

Our results extend previous research on structured collections of metrics, approaches to support the adoption of metrics, and generic structures for documenting metrics. Similar research has already been conducted in the area of Enterprise Architecture Management. Monahov [Mon14] developed a metric catalog that is also based on a minimalistic MMFS. The work by Monahov [Mon14] served as inspiration for a catalog for the area of large-scale agile software develop-

ment. Regarding structured collections of metrics, our results represent an extension. The metric catalog documents 196 metrics that are actually used in scaled agile organizations. Traditional solutions, on the other hand, present metrics with a different focus such as software metrics (e.g., Bouwers et al. [BDV14]) or metrics for web app development (e.g., Olsina et al. [OLP02]). Furthermore, our metric catalog differentiates itself from previous research insofar as it considers the challenges and success factors relevant to the adoption of metrics in scaled agile software development (cf. [PTM22a, PTMM22]). Thus, the catalog may be better adapted to the context of large agile organizations compared to other catalog templates (cf. Olsina et al. [OLP02]). So far, only Olszewska et al. [OHW+16] have explored a metrics description structure potentially applicable to large-scale agile software development. However, this is intended for documenting metrics that can measure agile transformation and not for metrics used in agile development programs. Furthermore, the metric catalog is a contribution to support the adoption of metrics. The evaluation results show that the catalog is suitable for inspiring practitioners regarding potential metrics, helping them to select metrics through different navigation elements, and supporting their organization-specific implementation. Previous research on large-scale agile development provided approaches to support the implementation of metrics in the form of checklists (e.g., Hartmann and Dymond [HD06]) and a framework consisting of ten questions, each related to characteristics of metrics (cf. Kaner et al. [Kan04]). In addition, we contribute to generic structures for documenting metrics, since we assessed which best practices from literature on structured collections of metrics (cf. [BDV14, Mon14, OLP02, OHW+16]), approaches to support metric adoption (cf. [HD06, Kan04]), and generic metrics documentation structures (cf. [iso17, Küt11, NAK02, Par15, PS10]) are applicable to our minimalistic MMFS. The final set of elements and their source of origin is illustrated in Table 5.5. The elements without a checkmark are unique to our design and stem from insights gained in the expert interviews.

Discussion related to RQ4:

We contribute to the existing body of knowledge by demonstrating that elements of measurement programs designed for traditional software development can also be effective in large-scale agile software development. For example, we build on the GQM approach proposed by Basili et al [Bas92], incorporate activities to improve the measurement program (cf. [BV01]), prioritize goals (cf. [FLM+98, GPMI13, TG10]), perform a cost-benefit analysis (cf. [VSB01]), and propose to set measurement as a goal (cf. [Kil01, GMK05, VLVSO+98]). However, we also contribute to existing research by presenting elements unique to our method. First and foremost, our method aligns metrics operation with the agile software development process. For example, metrics are mapped to agile events, stakeholders can take on different agile roles, and PDCA cycles are integrated into the agile software development process (cf. Design Principle 7). What is also unique about our method is that the responsibilities of measurement experts are extended in our method (cf. Design Principle 9), as they guide not only the selection of metrics but also the operation and scaling (cf. Tahir et al. [TRG16] on measurement programs). In addition, our method also focuses on scaling the measurement program in the organization (cf. Design Principle 9). In contrast, measurement programs developed for traditional software engineering tend to focus on sustainability and reusability [TRG16]. Another unique feature of our method is determined by the metric catalog web application, which provides a standard set of metrics and corresponding goals for large-scale agile software development (cf. Design Principle 6).

Discussion related to RQ5:

First, the involvement of agile teams in goal-setting becomes more important in large agile organizations. Current literature states that the bottom-up contribution of teams tends to focus on implementing the goals but not helping to define them. Instead, goal-setting is more the responsibility of Product Owners or management, who hold a top-down perspective [KTS21, SP06]. Our results show that this separation of responsibilities may no longer be appropriate in scaling agile organizations. This assertion is supported in particular by assessment results M5 and M6. We recommend that the agile team should be involved in all aspects of goal-setting at the team and program levels to address the challenges of increased product complexity and scope. Second, most of the identified goals can be attributed to the portfolio and program level. In total, we documented 21 objectives at the portfolio level and 26 objectives at the program level. In addition, we identified only one team-level objective and seven objectives related to the entire organization. These results indicate that in large agile organizations, there is a higher relevance for goal-setting at the middle level (i.e., portfolio and program level) than at the highest (i.e., enterprise level) and lowest levels (i.e., team level). Furthermore, our results show that process and product goals are equally common at the portfolio and program levels. This observation contradicts Korpivaara et al. [KTS21], who showed that process efficiency goals are set at the program level rather than at higher levels in the organization. Third, there is usually no clear distinction between goals and requirements in practice. This claim is supported by evaluation result M7, where we observed two viewpoints on whether backlog items can be goals. We recommend that organizations try to clearly document objectives and clarify for each requirement the link to the associated objective. For this purpose, several of our suggestions (e.g., M3 and M4) may be helpful.

CHAPTER 6

Implications

In this chapter, we provide an overview of the implications for research and practice associated with answering our research questions.

6.1. Implications for Research

Implications for research related to RQ1: Current literature on large-scale agile software development lacks a thorough assessment of the circumstances under which metrics are applied and the underlying reasons for their application. Our expert interview study contributes to the current state of research by demonstrating the relevance of metrics in the context of large-scale agile development, as it proves the broad application of metrics, even beyond the level of individual agile teams. Moreover, our detailed analysis of reasons for using metrics contributes to the understanding of application scenarios for metrics. This knowledge can be used to focus future research efforts on the most prevailing reason categories. For example, to develop customized metrics sets that specifically address a reason category. In addition, by revealing the most used metrics we enable researchers to target their research on the most relevant metrics. This study also adds to the body of knowledge in the area of metrics research across contexts, as it offers specific insights into the characteristics of metric application within large-scale agile software development as opposed to, for example, traditional software engineering.

Implications for research related to RQ2: The existing literature on large-scale agile software development does not provide a comprehensive overview of challenges and success factors in different organizations. Our work is an important resource in this area since researchers can use it to develop solutions that both address the specific challenges and consider the success factors. It also adds to the body of knowledge in the area of metrics research across contexts, as

it offers specific insights into the challenges and success factors relevant within large-scale agile software development as opposed to, for example, traditional software engineering.

Implications for research related to RQ3: In the current literature on large-scale agile software development, there are no artifacts that facilitate both documentation and organization-specific implementation of metrics. Inspired by the work of Monahov et al. [Mon14], we created a minimalistic MMFS and a metric catalog. Both artifacts are specifically designed for the context of large-scale agile software development. Our metric catalog contains 196 metrics that are actually used in scaling agile organizations. Researchers can use this set of metrics, for example, to define targeted metrics subsets for specific stakeholder groups. Furthermore, researchers can use these two artifacts to conduct further design-oriented research. For this purpose, it would be possible to integrate the two artifacts into one overarching artifact or to adapt the two artifacts for further contexts. The minimalistic MMFS also contributes to the body of knowledge in cross-context metrics research by providing specific insights into MMFS elements and metrics that are relevant in large-scale agile software development, as opposed to, for example, traditional software development.

Implications for research related to RQ4: In the current literature on large-scale agile software development, there are no methods that support the management of metrics. We developed such a method over the course of two years together with two case organizations. This work provides researchers with in-depth insights on how a successful measurement program is operated at a large agile organization. The developed method and the derived design principles can serve as inspiration for the development of measurement programs suited for other contexts. It also adds to the body of knowledge in the area of metrics research across contexts, as it offers specific insights into which method activities are unique to large-scale agile software development as opposed to, for example, traditional software engineering.

Implications for research related to RQ5: Current literature on large-scale agile software development lacks an in-depth investigation of goal establishment in large-scale agile software development. Our case study enables researchers to better understand goal-setting and documentation techniques in large-scale agile development projects. We describe in detail how goal-setting is conducted alongside the mandatory GMP, which can serve as an example for developing goal-setting approaches for other contexts. Moreover, the challenges and mitigation propositions we identify can help researchers to develop best practices and solutions. In addition, the identified goals could help in developing goal categorization schemes. Finally, our case study can help understand large-scale agile organizations' inner workings. This study also adds to the body of knowledge in the area of goal-setting research across contexts, as it offers specific insights into the characteristics of goal-establishment within large-scale agile software development as opposed to, for example, traditional software engineering.

6.2. Implications for Practice

Implications for practice related to RQ1: Our research shows that many large agile organizations use metrics at the team and higher levels. In organizations that view metrics as a paradigm that contradicts the agile approach, their widespread use in agile contexts may serve

as an argument for implementing them. In addition, practitioners may see the widespread use of metrics across multiple organizations as an opportunity to create industry-wide networks or communities to share their inclinations about metrics. They can also draw inspiration from the reasons other organizations use metrics and derive deployment scenarios for metrics. In particular, the most commonly used metrics could provide a useful entry point for practitioners exploring metrics for the first time.

Implications for practice related to RQ2: Practitioners can refer to the detailed overview of challenges and success factors to increase the likelihood of successful deployment of their metrics, especially if they are relevant to their specific context. The results also enable practitioners to better estimate the effort required for a metrics implementation, as corresponding requirements can be derived from the identified challenges and success factors. Furthermore, practitioners can use the insights gained to design artifacts or decide on the introduction of artifacts. In doing so, those should be chosen or adapted that take into account the relevant challenges and success factors.

Implications for practice related to RQ3: The two artifacts, the minimalistic MMFS and the metric catalog, can be used directly by practitioners to document metrics and support their organization-specific instantiation. Our evaluation has shown that the catalog is particularly useful for getting inspiration about possible metrics or helping select metrics. In addition, practitioners can use our findings to adapt our catalog to their organization or be inspired to design their own metrics repositories, possibly even in a digital form similar to our metric catalog web app [PTMM23].

Implications for practice related to RQ4: Practitioners can potentially integrate the metric management method and the design principles derived from it into their own measurement program. Even if only certain aspects of the method are of interest, it adds value because the complete method does not necessarily have to be applied.

Implications for practice related to RQ5: Practitioners can draw inspiration from our detailed presentations of goal-setting processes and documentation techniques to optimize their goal-setting strategies. The challenges we uncover and proposed solutions can help them increase the likelihood of success of their goal-setting processes, especially if they are relevant to their particular context. Agile practitioners can use the studies' findings to build an argument for stronger participation of agile teams in not only how goals are achieved but also in what goals are set in the first place. In addition, our comprehensive overview of goals may include goals that practitioners previously were unaware of but are relevant to them.

CHAPTER 7

Limitations

In this chapter, we assess the limitations of this dissertation. Therefore, we use the quality criteria construct validity, internal validity, external validity, and reliability from Yin [Yin18].

Construct validity requires identifying correct operational measures for the concepts under study. In all studies (P1-P5), we took measures to increase construct validity. As suggested by Yin [Yin18], we always included multiple sources of evidence (e.g., documents, interviews, observations). Furthermore, we discussed ambiguities, especially regarding the definition of the constructs and the interpretation of the data with the study participants. Triangulation of the data was achieved by always involving several researchers in the analyses of the data. Despite the measures taken, bias in the interpretation of the data on the part of the researchers and study participants cannot be completely ruled out.

Internal validity requires establishing a causal relationship in which certain conditions are assumed to lead to other conditions instead of spurious relationships. Since the case study from P5 and the expert interviews from P1 and P2 are exploratory and descriptive, internal validity is irrelevant for these studies. In the context of design science research (P3 and P4), the internal validity of the developed artifacts is measured by the extent to which the design, implemented in a problem context, meets the criteria established in the problem investigation [Wie09]. To ensure the internal validity of the three artifacts, we adapted the evaluations to measure internal validity accordingly. We conducted three iterations of demonstrations and survey-based evaluations for the minimalistic MMFS and metric catalog (P3) to analyze if the MMFS and metric catalog are useful in their most suitable application scenarios. As typical for action design research, the metric management method (P4) was evaluated continuously during the development process within the problem context to analyze if it can support the case organization's selection, operation, and scaling of metrics.

External validity requires that the results of the research are generalizable. All studies are

subject to major limitations with regard to the generalizability of the results. This is due to the fact that each study was only conducted in a limited number of scaled agile organizations. It is notable that the study partners are mainly from Germany. These limitations can be attributed to the choice of a qualitative research strategy and cannot be completely eliminated. To increase external validity, we ensured that all participating organizations and interviewees were representative of large-scale agile software development. In P5, as recommended by Yin [Yin18], a variety of interviewees from different programs (i.e., multiple cases) participated in the study. In our expert interview study (P1 and P2), we had a variety of experts from different companies and industries participate. We developed our artifacts based on existing theories to increase generalizability [Yin18]. The metric catalog is inspired by Monahov et al. [Mon14]. Our metric management method is based on the Goal Question Metric approach [Bas92]. In addition, for all studies, we have taken into account the recommendation of Yin [Yin18] to report all results in a detailed manner.

Reliability requires to demonstrate that the operations of a study, such as its data collection procedures, can be repeated with the same results. To achieve this, we used standardized data collection and analysis methods in all studies (P1-P5). In P1, P2 and P5, we performed systematic coding to ensure a chain of evidence. Furthermore, the results of all studies (P1-P5) were reviewed by several researchers in the research team and the research process was documented in detail.

CHAPTER 8

Conclusion

Inspired by the success of agile methods on a small scale, an increasing number of organizations began to scale agile methods to larger contexts [DPL16]. However, when large-scale agile development organizations become complex (e.g., through adding agile teams), maintaining an adequate level of situational awareness by relying solely on qualitative feedback loops becomes challenging. In such environments, metrics gain importance as they can complement or replace qualitative feedback.

In traditional software development, metrics are well-researched and used to measure processes, products, and resources [FN00, FB14]. Among the most important findings of metrics research in recent years is the need to link metrics to specific goals and to always interpret metrics based on organizational context, environment, and goals [CR94].

Thus, a central research problem of this dissertation is that agile methodologies have become increasingly popular in scaling agile environments over the past few decades [UPP+22], but researchers still need to develop an understanding of metrics in this context. Specifically, there needs to be more insight into whether metrics are relevant to practitioners or whether they have only been selected as an important topic by academics (RQ1). In addition, research needs to gain basic knowledge about goals (RQ5), which should always be set in connection with metrics, and about metrics' challenges and success factors (RQ2). The lack of this basic knowledge is particularly problematic because it could help develop scientific artifacts to support the adoption of metrics while being adapted to the unique requirements of large agile software development environments. Currently, there are no scientific artifacts to guide the use of metrics, nor is there a comprehensive overview of metrics that are suitable for scaling agile development environments (RQ3). In addition, current research lacks a method for managing metrics in large agile organizations, and recommendations for developing measurement programs in scaling agile environments are missing (RQ4).

Conclusions related to RQ1: We present a comprehensive overview of the reasons for using metrics in different expert organizations and describe them in detail. Moreover, we provide insight into the distribution of agile and traditional metrics in expert organizations and analyze which agile scaling level metrics are used. Furthermore, the most commonly used metrics within the organizations are presented.

Conclusions related to RQ2: This dissertation provides detailed insight into the challenges and success factors relevant to implementing metrics in large-scale agile organizations. Our findings offer valuable suggestions for the development of artifacts that promote the use of metrics in large-scale agile development.

Conclusions related to RQ3: This dissertation presents a minimalistic MMFS and an associated metric catalog. In the catalog, 196 metrics have been documented in a structured way based on the MMFS. The catalog offers suggestions for potential metrics and supports both their identification and selection as well as their organization-specific implementation.

Conclusions related to RQ4: This dissertation presents a goal-based method designed for a large agile case organization to support stakeholders in selecting, operating, and scaling metrics. Moreover, based on the learnings at the case organization, we present design principles that can potentially guide the development of methods suitable for other contexts.

Conclusions related to RQ5: We provide an in-depth study of how goal-setting is conducted across eight programs of a case organization. Moreover, goal definition and documentation techniques are described, challenges occurring during goal-setting are identified, and mitigation propositions to address these challenges are presented and evaluated. In addition, we have identified 51 goals which we assigned to categories and scaling levels on which they occurred.

CHAPTER 9

Future Research

In this chapter, we outline future research directions.

Future research avenues related to RQ1: The scope of our research is limited to the experts and their organizations who participated in our interview study. Future research could aim to identify more reasons for using metrics, determine more metrics, and specify their types and implementation levels. Since our results are not quantitative, as we do not use mathematical models, future research could focus on a survey-based study to draw statistical inferences about using metrics in large-scale agile development.

Future research avenues related to RQ2: Given that RQ2 draws upon the same empirical data as RQ1, the scope of our research is once again confined to the experts and their respective organizations involved in our interview study. Respectively, future research could be conducted in different contexts to identify more challenges and success factors for using metrics. Moreover, future research could focus on developing artifacts to address the identified challenges and success factors. Another possible research direction is to evaluate this research's challenges and success factors regarding relevance and effectiveness.

Future research avenues related to RQ3: We have evaluated the minimalistic MMFS and the metric catalog using discussions, surveys, and a case study. Subsequently, both artifacts were integrated into our goal-based metric management method and evaluated in another case organization. Nevertheless, future research could identify improvement suggestions for the two artifacts or evaluate the catalog in additional contexts. Another research opportunity is to collect additional metrics and document them using the metric catalog. A subsequent step could be to investigate the compatibility of both artifacts with existing scaling agile frameworks and other best practices for large-scale agile development.

Future research avenues related to RQ4: Our metric management method is adapted to

the context of the case organization. Therefore, future research could aim to adapt it to other contexts. In addition, future research could evaluate the ten design principles or incorporate and test them in measurement programs within different organizations. A subsequent step could be to investigate the compatibility of the developed method with existing scaling agile frameworks and other best practices for large-scale agile development.

Future research avenues related to RQ5: An inherent characteristic of case study research is that the meaningfulness of the results is limited to the boundaries of the case organization. Therefore, future research projects could aim to study goal-setting in other large agile organizations and compare their practices with our findings. Another starting point for future research would be to evaluate our mitigation strategies in other organizations. In addition, future research could consider developing artifacts that address the identified mitigation strategies. Furthermore, researchers could collect goals in different organizations and document them using the categorization scheme we used to provide a more complete view of goals in large-scale agile development.

Bibliography

[AG18] Abdullah M Aldahmash and Andrew Gravell. Measuring success in agile software development projects: a gqm approach. In *The Thirteenth International Conference on Software Engineering Advances*. IARIA, 2018.

[Ake04] Joan E van Aken. Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. *Journal of management studies*, 41(2):219–246, 2004.

[Ant97] Ana I Anton. *Goal identification and refinement in the specification of software-based information systems*. Georgia Institute of Technology, 1997.

[ASRW02] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, and Juhani Warsta. Agile software development methods: Review and analysis. *Proc. Espoo 2002*, pages 3–107, 01 2002.

[Bas92] Victor R Basili. Software modeling and measurement: the goal/question/metric paradigm. Technical report, University of Maryland, 1992.

[BBVB+01] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. Manifesto for agile software development. 2001.

[BCS+10] Mike Beedle, James O Coplien, Jeff Sutherland, Jens C Østergaard, Ademar Aguiar, and Ken Schwaber. Essential scrum patterns. In *14th European Conference on Pattern Languages of Programs*, pages 1–17. Citeseer, 2010.

[BDV14] Eric Bouwers, Arie van Deursen, and Joost Visser. Towards a catalog format for software metrics. In *Proceedings of the 5th international workshop on emerging trends in software metrics*, pages 44–47, 2014.

[Bec00] Kent Beck. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.

[BLR+10] Victor R Basili, Mikael Lindvall, Myrna Regardie, Carolyn Seaman, Jens Heidrich, Jürgen Münch, Dieter Rombach, and Adam Trendowicz. Linking software

development and business strategy through measurement. *Computer*, 43(4):57–65, 2010.

[BLTV15] Martin P Boerman, Zeeger Lubsen, Damian A Tamburri, and Joost Visser. Measuring and monitoring agile development status. In *6th International Workshop on Emerging Trends in Software Metrics*, pages 54–62. IEEE, 2015.

[BMS19] Marthe Berntzen, Nils Brede Moe, and Viktoria Stray. The product owner in large-scale agile: an empirical study through the lens of relational coordination theory. In *20th International Conference on Agile Software Development (XP)*, pages 121–136. Springer, 2019.

[Bri96] Sjaak Brinkkemper. Method engineering: engineering of information systems development methods and tools. *Information and software technology*, 38(4):275–280, 1996.

[BV01] Michael Berry and Michiel F Vandenbroek. A targeted assessment of the software measurement process. In *Proceedings Seventh International Software Metrics Symposium*, pages 222–235. IEEE, 2001.

[BW84] Victor R Basili and David M Weiss. A methodology for collecting valid software engineering data. *IEEE Transactions on software engineering*, (6):728–738, 1984.

[CC04] David Cohen and Patrícia Costa. An introduction to agile methods. *Advances in Computers*, 62:1–66, 12 2004.

[CC19] Kieran Conboy and Noel Carroll. Implementing large-scale agile frameworks: challenges and recommendations. *IEEE software*, 36(2):44–50, 2019.

[CF04] Kieran Conboy and Brian Fitzgerald. Toward a conceptual framework of agile methods: a study of agility in different disciplines. In *Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research*, pages 37–44, 2004.

[CG07] Łukasz Cyra and Janusz Górski. Extending gqm by argument structures. In *IFIP Central and East European Conference on Software Engineering Techniques*, pages 26–39. Springer, 2007.

[Cha01] K. Charmaz. Grounded theory. In R.M. Emerson, editor, *Contemporary Field Research*, pages 335–352. Waveland Press, Inc., Long Grove, 2001.

[CJR09] Tjan-Hien Cheng, Slinger Jansen, and Marc Remmers. Controlling and monitoring agile software development in three dutch product software companies. In *ICSE workshop on software development governance*, pages 29–35. IEEE, 2009.

[Cop96] James O. Coplien. *Software Patterns*. Wiley, 1996.

[CR94] Victor R Basili Gianluigi Caldiera and H Dieter Rombach. The goal question metric approach. *Encyclopedia of software engineering*, pages 528–532, 1994.

[Das92] Michael K Daskalantonakis. A practical view of software measurement and implementation experiences within motorola. *IEEE Transactions on Software Engineering*, 18(11):998, 1992.

[DD08] Tore Dybå and Torgeir Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9-10):833–859, 2008.

[DFI14] Torgeir Dingsøyr, Tor Erlend Fægri, and Juha Itkonen. What is large in large-scale? a taxonomy of scale for agile software development. In *15th International Conference on Product-Focused Software Process Improvement (PROFES)*, pages 273–276, Helsinki, Finland, 2014. Springer.

[Dig20] Digital.ai. 14th annual state of agile report. URL: `https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf`, 2020. [Online; accessed 09-10-2023].

[DL05] N. K. Denzin and Y. S. Lincoln, editors. *The Sage Handbook of Qualitative Research*. Sage Publications Ltd., 3rd edition, 2005.

[DM14] Torgeir Dingsøyr and Nils Brede Moe. Towards principles of large-scale agile development: A summary of the workshop at xp2014 and a revised research agenda. In *XP 2014 Workshops*, pages 1–8. Springer, 2014.

[DMFS18] Torgeir Dingsøyr, Nils Brede Moe, Tor Erlend Fægri, and Eva Amdahl Seim. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 23:490–520, 2018.

[DNBM12] Torgeir Dingsøyr, Sridhar Nerur, VenuGopal Balijepally, and Nils Brede Moe. A decade of agile methodologies: Towards explaining agile software development. *Journal of systems and software*, 85(6):1213–1221, 2012.

[DPL16] Kim Dikert, Maria Paasivaara, and Casper Lassenius. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119:87–108, 2016.

[DST18] Philipp Diebold, Anna Schmitt, and Sven Theobald. Scaling agile: how to select the most appropriate framework. In *Proceedings of the 19th international conference on agile software development: companion*, pages 1–4, 2018.

[DTHK05] Yael Dubinsky, David Talby, Orit Hazzan, and Arie Keren. Agile metrics at the israeli air force. In *Agile Development Conference (ADC'05)*, pages 12–19. IEEE, 2005.

[EDBS05] C. Ebert, R. Dumke, M. Bundschuh, and A. Schmietendorf. *Best Practices in Software Measurement: How to Use Metrics to Improve Project and Process Performance*. Springer, Heidelberg, Germany, 1 edition, 2005.

[FB14] Norman Fenton and James Bieman. *Software metrics: a rigorous and practical approach*. CRC press, 2014.

[FHMD02] Ellie Fossey, Carol Harvey, Fiona McDermott, and Larry Davidson. Understanding and evaluating qualitative research. *Australian & New Zealand journal of psychiatry*, 36(6):717–732, 2002.

[FLM+98] Alfonso Fuggetta, Luigi Lavazza, Sandro Morasca, Stefano Cinti, Giandomenico Oldano, and Elena Orazi. Applying gqm in an industrial software factory. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 7(4):411–448, 1998.

[FN00] Norman E Fenton and Martin Neil. Software metrics: roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 357–370, 2000.

[fS02] International Organization for Standardization. Software engineering - software measurement process. Technical Report 15939, ISO/IEC, 2002.

[GJT08] Raghu Garud, Sanjay Jain, and Philipp Tuertscher. Incomplete by design and designing for incompleteness. *Organization studies*, 29(3):351–371, 2008.

[GMK05] Anandasivam Gopal, Tridas Mukhopadhyay, and Mayuram S Krishnan. The impact of institutional forces on software metrics programs. *IEEE Transactions on Software Engineering*, 31(8):679–694, 2005.

[GPMI13] Cigdem Gencel, Kai Petersen, Aftab Ahmad Mughal, and Muhammad Imran Iqbal. A decision support framework for metrics selection in goal-based measurement programs: Gqm-dsfms. *Journal of Systems and Software*, 86(12):3091–3108, 2013.

[Gre81] Ronald C Greenwood. Management by objectives: As developed by peter drucker, assisted by harold smiddy. *Academy of Management Review*, 6(2):225–230, 1981.

[Gub90] E. G. Guba. *The Paradigm Dialog.* Sage Publications, Inc., 1990.

[HA13] Amani Mahdi Mohammed Hamed and Hisham Abushama. Popular agile approaches in software development: Review and analysis. In *International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)*, pages 160–166. IEEE, 2013.

[HC01] Jim Highsmith and Alistair Cockburn. Agile software development: The business of innovation. *Computer*, 34(9):120–127, 2001.

[HCHC10] Alan Hevner, Samir Chatterjee, Alan Hevner, and Samir Chatterjee. Design science research in information systems. *Design research in information systems: theory and practice*, pages 9–22, 2010.

[HD06] Deborah Hartmann and Robin Dymond. Appropriate agile measurement: using metrics and diagnostics to deliver business value. In *AGILE 2006*, pages 6–pp. IEEE, 2006.

[Hig02] James A Highsmith. *Agile software development ecosystems.* Addison-Wesley Professional, 2002.

[HKvB+18] Philipp Hohl, Jil Klünder, Arie van Bennekum, Ryan Lockard, James Gifford, Jürgen Münch, Michael Stupperich, and Kurt Schneider. Back to the future: origins and directions of the "agile manifesto"–views of the originators. *Journal of Software Engineering Research and Development*, 6:1–27, 2018.

[HMPR08] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *Management Information Systems Quarterly*, 28(1):6, 2008.

[HSR10] Brian Henderson-Sellers and Jolita Ralyté. Situational method engineering: state-of-the-art review. *Journal of Universal Computer Science*, 2010.

[HWMP13] Jeanette Heidenberg, Max Weijola, Kirsi Mikkonen, and Ivan Porres. A metrics model to measure the impact of an agile transformation in large software development organizations. In *14th International Conference on Agile Software Development (XP 2013)*, pages 165–179, Vienna, Austria, 2013. Springer.

[Iiv03] Juhani Iivari. The is core-vii: Towards information systems as a science of meta-artifacts. *Communications of the Association for Information Systems*, 12(1):37, 2003.

[iso17] Iso/iec/ieee international standard - systems and software engineering–vocabulary. *ISO/IEC/IEEE 24765:2017(E)*, pages 1–541, 2017.

[Jet13] K. Jethani. Software metrics for effective project management. *International Journal of System Assurance Engineering and Management*, 4(4):335–340, 2013.

[JGZag+13] Taghi Javdani Gandomani, Hazura Zulzalil, Abdul azim abdul ghani, Abu Bakar Md Sultan, and Reza Parizi. On the current measurement practices in agile software development. *International Journal of Computer Science Issues*, 9, 01 2013.

[Kan04] Cem Kaner. Software engineering metrics: What do they measure and how do we know? In *10th International Symposium on Software Metrics*, pages 1–12, 2004.

[KB12] Gaurav Kumar and Pradeep Kumar Bhatia. Impact of agile methodology on software development process. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, 2(4):46–50, 2012.

[Ket09] Petri Kettunen. Adopting key lessons from agile manufacturing to agile software product development—a comparative study. *Technovation*, 29(6-7):408–422, 2009.

[KFW18] Reni Kurnia, Ridi Ferdiana, and Sunu Wibirama. Software metrics classification for agile scrum process: A literature review. In *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 174–179. IEEE, 2018.

[KGW13] Mahvish Khurum, Tony Gorschek, and Magnus Wilson. The software value map—an exhaustive collection of value aspects for the development of software intensive products. *Journal of software: Evolution and Process*, 25(7):711–741, 2013.

[Kil01] Tapani Kilpi. Implementing a software metrics program at nokia. *IEEE software*, 18(6):72–77, 2001.

[KLFM19] Petri Kettunen, Maarit Laanti, Fabian Fagerholm, and Tommi Mikkonen. Agile in the era of digitalization: a finnish survey study. In *20th International Conference on Product-Focused Software Process Improvement (PROFES)*, pages 383–398. Springer, 2019.

[KMI15] Eetu Kupiainen, Mika V Mäntylä, and Juha Itkonen. Using metrics in agile and lean software development–a systematic literature review of industrial studies. *Information and software technology*, 62:143–163, 2015.

[KN92] Robert S. Kaplan and David P. Norton. The balanced scorecard: Measures that drive performance. *Harvard Business Review*, 70(1):71–79, 1992.

[KP18] Kristīne Kārkliņa and Rūta Pirta. Quality metrics in agile software development projects. *Information Technology & Management Science*, 21, 2018.

[KR18] Filip Kišš and Bruno Rossi. Agile to lean software development transformation: A systematic literature review. In *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 969–973. IEEE, 2018.

[Kru13] Philippe Kruchten. Contextualizing agile software development. *Journal of software: Evolution and Process*, 25(4):351–361, 2013.

[KTS21] Ida Korpivaara, Tuure Tuunanen, and Ville Seppänen. Performance measurement in scaled agile organizations. In *Proceedings of the Annual Hawaii International Conference on System Sciences*. University of Hawai'i at Manoa, 2021.

[Küt11] Martin Kütz. *Kennzahlen in der IT: Werkzeuge für Controlling und Management*. dpunkt. verlag, 2011.

[Lef18] Dean Leffingwell. *Safe 4.5 Reference Guide: Scaled Agile Framework for Lean Enterprises*. Addison-Wesley Professional, 2018.

[Lev13] Merry-Jo D Levers. Philosophical paradigms, grounded theory, and perspectives on emergence. *Sage Open*, 3(4):2158244013517243, 2013.

[LG85] Yvonna S Lincoln and Egon G Guba. *Naturalistic inquiry*. sage, 1985.

[Lik32] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[LKL+18] Teemu Lappi, Teemu Karvonen, Lucy Ellen Lwakatare, Kirsi Aaltonen, and Pasi Kuvaja. Toward an improved understanding of agile project governance: A systematic literature review. *Project Management Journal*, 49(6):39–63, 2018.

[LL06] Edwin A Locke and Gary P Latham. New directions in goal-setting theory. *Current directions in psychological science*, 15(5):265–268, 2006.

[LSA13] Maarit Laanti, Jouni Similä, and Pekka Abrahamsson. Definitions of agile software development and agility. In *20th European Conference on Software Process Improvement (EuroSPI)*, pages 247–258, Dundalk, Ireland, 2013. Springer.

[LV16] Craig Larman and Bas Vodde. *Large-scale scrum: More with LeSS*. Addison-Wesley Professional, 2016.

[MBF06] Jane Mills, Ann Bonner, and Karen Francis. The development of constructivist grounded theory. *International journal of qualitative methods*, 5(1):25–35, 2006.

[MDS+19] Nils Brede Moe, Bjørn Haug Dahl, Viktoria Stray, Lina Sund Karlsen, and Stine Schjødt-Osmo. Team autonomy in large-scale agile. In *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*, pages 6997–7006. Association for Information Systems (AIS), 2019.

[Mil88] E. E. Mills. *Software Metrics*, 1988.

[MJ10] Neil Maiden and Sara Jones. Agile requirements can we have our cake and eat it too? *IEEE software*, 27(3):87–88, 2010.

[MMG02] M Lynne Markus, Ann Majchrzak, and Les Gasser. A design theory for systems that support emergent knowledge processes. *MIS quarterly*, pages 179–212, 2002.

[MMSS12] Florian Matthes, Ivan Monahov, Alexander Schneider, and Christopher Schulz. Eam kpi catalog v 1.0. Technical report, 2012.

[MN07] Michael D Myers and Michael Newman. The qualitative interview in is research: Examining the craft. *Information and organization*, 17(1):2–26, 2007.

[Mon14] Ivan Monahov. *Integrated software support for quantitative models in the domain of Enterprise Architecture Management*. PhD thesis, Technische Universität München, 2014.

[MSMB20] Marius Mikalsen, Viktoria Stray, Nils Brede Moe, and Idun Backer. Shifting conceptualization of control in agile transformations. In *XP Workshops*, pages 173–181, 2020.

[MSS21] Wilhelm Meding, Miroslaw Staron, and Ola Söder. Meteam—a method for characterizing mature software metrics teams. *Journal of Systems and Software*, 180:111006, 2021.

[NAK02] Andy D Neely, Chris Adams, and Mike Kennerley. *The performance prism: The scorecard for measuring and managing business success*. Prentice Hall Financial Times London, 2002.

[NMM05] Sridhar Nerur, RadhaKanta Mahapatra, and George Mangalaraj. Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5):72–78, 2005.

[NOK14] Robert L Nord, Ipek Ozkaya, and Philippe Kruchten. Agile in distress: Architecture to the rescue. In *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014 International Workshops*, pages 43–57, Rome, Italy, 2014. Springer.

[OHW+16] Marta Olszewska, Jeanette Heidenberg, Max Weijola, Kirsi Mikkonen, and Ivan Porres. Quantitatively measuring a large-scale agile transformation. *Journal of Systems and Software*, 117:258–273, 2016.

[OK12] Nilay Oza and Mikko Korkala. Lessons learned in implementing agile software development metrics. In *UK Academy for Information Systems Conference Proceedings 2012*, number 38, 2012.

[OLP02] Luis Olsina, Guillermo Lafuente, and Oscar Pastor. Towards a reusable repository for web metrics. *Journal of Web engineering*, pages 061–073, 2002.

[Orl96] Wanda J Orlikowski. Improvising organizational transformation over time: A situated change perspective. *Information systems research*, 7(1):63–92, 1996.

[Par15] David Parmenter. *Key performance indicators: developing, implementing, and using winning KPIs*. John Wiley & Sons, 2015.

[PM14] Aroun Poligadu and Raj Kishen Moloo. An innovative measurement programme for agile governance. *International Journal of Agile Systems and Management*, 7(1):26–60, 2014.

[Pow14] Ken Power. Definition of ready: An experience report from teams at cisco. In *International Conference on Agile Software Development (XP)*, pages 312–319. Springer, 2014.

[PRTV12] Ken Peffers, Marcus Rothenberger, Tuure Tuunanen, and Reza Vaezi. Design science research evaluation. In *Proceedings of the 7th international conference on Design Science Research in Information Systems: advances in theory and practice*, pages 398–410. Springer, 2012.

[PS10] Viara Popova and Alexei Sharpanskykh. Modeling organizational performance indicators. *Information systems*, 35(4):505–527, 2010.

[PTRC07] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[PW09] Kai Petersen and Claes Wohlin. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of systems and software*, 82(9):1479–1490, 2009.

[RDFS16] Knut Rolland, Torgeir Dingsoyr, Brian Fitzgerald, and Klaas-Jan Stol. Problematizing agile in the large: alternative assumptions for large-scale agile development. In *39th International Conference on Information Systems (ICIS)*, pages 1–21. Association for Information Systems (AIS), 2016.

[RDR03] Jolita Ralyté, Rébecca Deneckère, and Colette Rolland. Towards a generic model for situational method engineering. In *International Conference on Advanced Information Systems Engineering (CAiSE)*, pages 95–110, Klagenfurt/Velden, Austria, 2003. Springer.

[RH09] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14:131–164, 2009.

[RM07] L. Richards and J.M. Morse. *Readme First for a User's Guide to Qualitative Methods*. Sage Publications, Thousand Oaks, 2007.

[RR11] Herbert J Rubin and Irene S Rubin. *Qualitative interviewing: The art of hearing data*. sage, 2011.

[RRO18] Prabhat Ram, Pilar Rodriguez, and Markku Oivo. Software process measurement and related challenges in agile software development: A multiple case study. In *19th International Conference on Product-Focused Software Process Improvement (PROFES)*, pages 272–287. Springer, 2018.

[RROMF19] Prabhat Ram, Pilar Rodriguez, Markku Oivo, and Silverio Martínez-Fernández. Success factors for effective process metrics operationalization in agile software development: a multiple case study. In *IEEE/ACM International Conference on Software and System Processes (ICSSP)*, pages 14–23. IEEE, 2019.

[SA22] Inc. Scaled Agile. SAFe 5. `https://www.scaledagileframework.com`, 2022. [Online; accessed 08-11-2023].

[Sal21] Johnny Saldaña. *The coding manual for qualitative researchers*. sage, 2021.

[SB01] Ken Schwaber and Mike Beedle. *Agile software development with Scrum*. Prentice Hall PTR, 2001.

[SC20] Adriana Storti and Tony Clear. The contradiction of agile measures: Customer as focus, but process as measured? In *Australasian Conference on Information Systems (ACIS)*. Association for Information Systems (AIS), 2020.

[Sch04] Ken Schwaber. *Agile project management with Scrum*. Microsoft press, 2004.

[Sch18] Ken Schwaber. The definitive guide to scaling scrum with nexus: The rules of the game, 2018. [Online; accessed 10-10-2023].

[Scr22] Scrum.org. The scrum framework poster, 2022. [Online; accessed 10-10-2023].

[Sea99] Carolyn B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering*, 25(4):557–572, 1999.

[SHP+11] Maung K Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. Action design research. *MIS quarterly*, pages 37–56, 2011.

[Sim96] Herbert A Simon. *The sciences of the artificial*. MIT press, 1996.

[SKR+20] Svetlana Shirokova, Elizaveta Kislova, Olga Rostova, Anastasiia Shmeleva, and Lene Tolstrup. Company efficiency improvement using agile methodologies for managing it projects. In *Proceedings of the International Scientific Conference-Digital Transformation on Manufacturing, Infrastructure and Service*, pages 1–10, 2020.

[SM11] Miroslaw Staron and Wilhelm Medig. Factors determining long-term success of a measurement program: An industrial case study. *e-Informatica Software Engineering Journal*, 5(1), 2011.

[SMN09] Miroslaw Staron, Wilhelm Meding, and Christer Nilsson. A framework for developing measurement systems and its industrial evaluation. *Information and Software Technology*, 51(4):721–737, 2009.

[SP06] Ingo Schnabel and Markus Pizka. Goal-driven software development. In *30th Annual IEEE/NASA Software Engineering Workshop*, pages 59–65. IEEE, 2006.

[SS18] Christoph Johann Stettina and Lennard Schoemaker. Reporting in agile portfolio management: Routines, metrics and artefacts to maintain an effective oversight. In *Agile Processes in Software Engineering and Extreme Programming: 19th International Conference*, pages 199–215. Springer International Publishing, 2018.

[SS20] Ken Schwaber and Jeff Sutherland. The 2020 scrum guide, 2020. [Online; accessed 10-10-2023].

[TG10] Touseef Tahir and Cigdem Gencel. A structured goal based measurement framework enabling traceability and prioritization. In *6th International Conference on Emerging Technologies (ICET)*, pages 282–286. IEEE, 2010.

[THDK06] David Talby, Orit Hazzan, Yael Dubinsky, and Arie Keren. Reflections on reflection in agile software development. In *AGILE 2006*, pages 11–pp. IEEE, 2006.

[TRG16] Touseef Tahir, Ghulam Rasool, and Cigdem Gencel. A systematic literature review on software measurement programs. *Information and Software Technology*, 73:101–121, 2016.

[TSD19] Sven Theobald, Anna Schmitt, and Philipp Diebold. Comparing scaling agile frameworks based on underlying practices. In *International Conference on Agile Software Development (XP)*, pages 88–96. Springer, 2019.

[UKCM18] Ömer Uludag, Martin Kleehaus, Christoph Caprano, and Florian Matthes. Identifying and structuring challenges in large-scale agile development based on a structured literature review. In *22nd International Enterprise Distributed Object Computing Conference (EDOC)*, pages 191–197. IEEE, 2018.

[Ver20] VersionOne. 16th annual state of agile survey, 2020. [Online; accessed 09-10-2023].

[VLVSO+98] Frank Van Latum, Rini Van Solingen, Markku Oivo, Barbara Hoisl, Dieter Rombach, and Gunther Ruhe. Adopting gqm based measurement in an industrial environment. *IEEE software*, 15(1):78–86, 1998.

[VOWVH06] Marcel Van Oosterhout, Eric Waarts, and Jos Van Hillegersberg. Change factors requiring agility and implications for it. *European journal of information systems*, 15(2):132–145, 2006.

[VSB01] Rini Van Solingen and Egon Berghout. Integrating goal-oriented measurement in industrial software engineering: industrial experiences with and additions to the goal/question/metric method (gqm). In *Proceedings Seventh International Software Metrics Symposium*, pages 246–258. IEEE, 2001.

[WC03] Laurie Williams and Alistair Cockburn. Agile software development: It's about feedback and change. *Computer*, 36(6):39–43, 2003.

[Wie09] Roel Wieringa. Design science as nested problem solving. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, pages 1–12, Philadelphia, USA, 2009.

[WWES92] Joseph G Walls, George R Widmeyer, and Omar A El Sawy. Building an information system design theory for vigilant eis. *Information systems research*, 3(1):36–59, 1992.

[Yin18] Robert K Yin. Case study research and applications design and methods. 2018.

Publications

[PSM22] Pascal Philipp, Moritz Schüll, and Florian Matthes. Investigating the establishment of goals in large-scale agile development. In *26th Pacific Asia Conference on Information Systems (PACIS)*, Taipei/Sydney, 2022. Association for Information Systems (AIS).

[PTM22a] Pascal Philipp, Franziska Tobisch, and Florian Matthes. Challenges and success factors for metrics in large-scale agile development. In *28th Americas Conference on Information Systems (AMCIS)*, Minneapolis, USA, 2022. Association for Information Systems (AIS).

[PTM22b] Pascal Philipp, Franziska Tobisch, and Florian Matthes. Investigating the adoption of metrics in large-scale agile software development. In *26th Pacific Asia Conference on Information Systems (PACIS)*, Taipei/Sydney, 2022. Association for Information Systems (AIS).

[PTMM22] Pascal Philipp, Franziska Tobisch, Leon Menzel, and Florian Matthes. Toward a metric catalog for large-scale agile development. In *28th Americas Conference on Information Systems (AMCIS)*, Minneapolis, USA, 2022. Association for Information Systems (AIS).

[PTMM23] Pascal Philipp, Franziska Tobisch, Leon Menzel, and Florian Matthes. A method for metric management at a large-scale agile software development organization. In *17th International Conference on Software Process and Product Measurement*, Rome, Italy, 2023. IWSM-Mensura.

[UPP+22] Ömer Uludağ, Pascal Philipp, Abheeshta Putta, Maria Paasivaara, Casper Lassenius, and Florian Matthes. Revealing the state of the art of large-scale agile development research: A systematic mapping study. *Journal of Systems and Software (JSS)*, 194:1–43, 2022. `https://doi.org/10.1016/j.jss.2022.111473`. The accepted version of this paper is included in Appendix A of the dissertation.

Abbreviations

AMCIS Americas Conference on Information Systems

GBRAM Goal-Based Requirements Analysis Method

GMP Goal Management Process

GQM Goal Question Metric Approach

MENSURA International Conference on Software Process and Product Measurement

MMFS Metric Management Fact Sheet

OKRs Objectives and Key Results

PACIS Pacific Asia Conference on Information Systems

SAFe Scaled Agile Framework

SME Situational Method Engineering

APPENDIX A

Embedded Publications in Original Format

Investigating the Adoption of Metrics in Large-Scale Agile Software Development

Completed Research Paper

Pascal Philipp
Chair of Software Engineering for Business Information Systems (sebis), Department of Informatics, Technical University of Munich
Boltzmannstraße 3, 85748 Garching, Germany
pascal.philipp@tum.de

Franziska Tobisch
Chair of Software Engineering for Business Information Systems (sebis), Department of Informatics, Technical University of Munich
Boltzmannstraße 3, 85748 Garching, Germany
franziska.tobisch@tum.de

Florian Matthes
Chair of Software Engineering for Business Information Systems (sebis), Department of Informatics, Technical University of Munich
Boltzmannstraße 3, 85748 Garching, Germany
matthes@tum.de

Abstract

Many organizations have started using agile methods to develop software in small projects. The success of agile methods on a small scale inspired organizations to scale them to larger contexts. However, adopting agile practices at scale can complicate the monitoring, coordination, and steering of the multi-layered development process. Metrics can address this challenge but are controversial since their implementation is a challenge in itself. Hitherto, research on large-scale agile development lacks publications investigating the adoption of metrics. We conducted an expert interview study to explore their (i) reasons for adopting metrics (ii), what metrics they use on the team, program, and portfolio level (iii), and the most occurring metrics in the expert organizations. Our results show that metrics are mainly used for transparency, improvement, and controlling. Most metrics occur on the program level. Finally, we identified story point estimation, velocity, and sprint burn-down, as the most established metrics.

Keywords: Agile software development, large-scale agile, metrics, reasons

Introduction

Organizations act in an environment in which rapid changes in technology, markets, and regulations increasingly force them to adapt quickly to remain competitive (Preiss et al. 1996; Van Oosterhout et al. 2006). Simultaneously, products and services increasingly depend on software (Attaran 2004; Highsmith and Highsmith 2002; Van Oosterhout et al. 2006). In dealing with the growing demand for agility,

traditional software development is ever more reaching its limits (Highsmith and Cockburn 2001; Highsmith and Highsmith 2002). Its severe weaknesses, such as infrequent releases with large batch sizes, a lack of customer feedback, and rigid processes, acutely restrict organizations (Javdani et al. 2012; Petersen and Wohlin 2009). These shortcomings of traditional software development gave rise to the success of agile software development (Javdani et al. 2012; Petersen and Wohlin 2009). Developing software in an agile manner has lastingly changed and coined existing software development practices (Dingsøyr et al. 2012; Rajlich 2006). Accordingly, agile approaches like Scrum (Schwaber and Beedle 2002) and Extreme Programming (XP) (Beck 2000) have been widely adopted by practitioners (Digital.ai 2020; Hamed and Abushama 2013; Maiden and Jones 2010). These approaches work well on a small scale in contexts close to the 'agile sweet spot' comprising ideal conditions like co-located teams of 5–12 members; a stable underlying architecture; medium to low system criticality, and frequent deliveries (Kruchten 2013; Nord et al. 2014). Inspired by the success of agile methods on a small scale, an increasing number of large organizations have started to scale agile practices up to larger contexts which depart significantly from the 'agile sweet spot' (Digital.ai 2020; Dikert et al. 2016; Dingsøyr et al. 2018; Kruchten 2013; Nord et al. 2014). Scaling agile methods is coined as large-scale agile software development (Dingsøyr and Moe 2014; Rolland et al. 2016). Large-scale agile software development can refer to settings where multiple teams work together on a software product (Dingsøyr and Moe 2014; Dumitriu et al. 2019). Further, large-scale agile software development comprises implementing agile methods and principles across a whole organization, including higher organizational levels (Dingsøyr and Moe 2014; Dumitriu et al. 2019). Several large-scale agile practices and frameworks, such as the Scaled Agile Framework (SAFe) (Leffingwell 2018), and Large Enterprise Scrum (LeSS) (Larman and Vodde 2016), have been developed to support organizations (Paasivaara 2017) applying agile approaches at scale (Dikert et al. 2016; Nerur et al. 2005).

The term '*metric*' is generally understood to mean measurement (Geras et al. 2004; ISO 2017). Measurement can be described as "the empirical, objective assignment of numbers, according to a rule derived from a model or theory to attributes of objects or events with the intent of describing them" (Geras et al. 2004). Metrics can serve managers as feedback instruments by providing concise quantitative information. Therefore, metrics can be seen as an alternative to or be combined with qualitative information (i.e., reports, group discussions, or interviews). In traditional software development, metrics have played an essential role for decades (Fenton and Neil 2000) and are used to measure different attributes (e.g., costs) of products, processes, and resources (Fenton and Bieman 2014). Several researchers argue that metrics implemented in traditional software development approaches are not always applicable for using them within agile settings (Hartmann and Dymond 2006; Oza and Korkala 2012; Poligadu and Moloo 2014). As measurements are essential drivers for behavior, there is the risk that using improper metrics negatively impacts team behavior, productivity and undermines the cultural change towards an agile way of working (Hartmann and Dymond 2006). However, within single agile teams, metrics like the burn-down chart (Schwaber 2004; Scrum.org 2021) or velocity (Scrum.org 2021) are applied and can support a key principle of agile software development: data-driven continuous feedback and learning (Janus et al. 2012; Liechti et al. 2017). In scaling agile environments, the increased organizational scope (Brown et al. 2013; Dingsøyr et al. 2018) and complexity (Ambler and Lines 2014; Brown et al. 2013) can lead to various challenges, such as the difficulty of monitoring, coordination, and steering of a multi-layered development process (Talby and Dubinsky 2009). Metrics can address this challenge because they allow to monitor, control, assess, manage, and improve the software process (Ebert et al. 2005; Jethani 2013). Moreover, metrics are means to identify problems and risks early, support communication, and enable fact-based planning, estimating decisions, and optimization (Jethani 2013). However, introducing metrics in scaling agile environments can be challenging itself. Numerous organizations struggle with efficiently adopting metrics (Korpivaara et al. 2021), due to difficulties in reducing fragmented project specific data and measurement (Digital.ai 2020), connecting metrics belonging to different organizational levels while considering level-specific, varied requirements and appropriate metrics scaling (Korpivaara et al. 2021). Besides these challenges specific to scaling agile environments, implementing metrics in general can lead to challenges like lacking credibility of measurements caused by inadequate data quality (Eckerson 2010), misunderstandings, and misinterpretations (Bird et al. 2005; Eckerson 2010). Metrics are well-researched in general management (cf., Kütz 2011; Mills 1988; Neely et al. 2002) and traditional software engineering (cf., Fenton and Neil 2000). On the contrary, expert insights investigating the adoption of metrics in scaling agile environments are scarce (Uludağ et al. 2021). Moreover, examining the usage of metrics beyond the team level was identified as a burning research topic (Dingsøyr and Moe 2014). Therefore, we intend to conduct an expert interview study to collect and analyze their reasons for using metrics in scaling agile

environments. Moreover, we aim to observe on which scaling levels metrics occur and how important traditional metrics (i.e., stemming from general management) and agile metrics (i.e., related to agile concepts, events, and artifacts) are on these levels, according to the experts. Our final goal is to present the most mentioned metrics occurring in at least three expert organizations. Based on these goals, we formulate the three research questions (RQs):

- *RQ1: Why do practitioners adopt metrics in large-scale agile software development?*
- *RQ2: Which metrics do practitioners use on the team, program, and portfolio level?*
- *RQ3: Which metrics do practitioners use the most in large-scale agile software development?*

Theoretical Background

Large-scale Agile Software Development

Nowadays, applying agile methodologies in software development is the predominant mode of developing software (Dingsøyr and Moe 2014). Agile methods rely on the values and principles summarized in the agile manifesto (Beedle et al. 2010; Schwaber 2018), published in 2001 to provide "an alternative to documentation driven, heavyweight software development processes" (Beck et al. 2001). Multiple agile methods and frameworks, including Feature-Driven Development, Dynamic System Development Method (DSDM), and Crystal Methods, have been developed to help organizations implement agility (Uludağ et al. 2021). Presently, Scrum, ScrumBan, and hybrid approaches, like a combination of Scrum and XP, are the most widely used agile approaches in the industry (Digital.ai 2020). The advantages of agile methods like responsiveness to volatile customer requirements, flexibility, and resilience in altering environments are undisputed (Digital.ai 2020; Kumar and Bhatia 2012). However, according to Kruchten (2013), the success of agile methods depends on the context in which practitioners deploy them. Implementing agile methods 'out of the box' is likely to be successful in the 'agile sweet spot', a context identical or similar to the context in which agile methods have been initially created (Kruchten 2013). According to Nord et al. (2014), a favorable context is characterized by co-located teams of 5-12 members; a stable underlying architecture; medium to low system criticality; and frequent deliveries. Kruchten (2013) identified eight key factors to assess whether a given context departs from the sweet spot. The key factors are size, stable architecture, business model, team distribution, rate of change, system age, criticality, and governance (Kruchten 2013). Applying agile approaches without modification in unfavorable contexts may lead to failure (Kruchten, 2013). The success of agile methods within the 'agile sweet spot' inspired large organizations to scale agile methods and practices to larger settings (Digital.ai 2020; Dikert et al. 2016; Dingsøyr et al. 2018). The approach of scaling agile methods is often referred to as large-scale agile software development (Dingsøyr and Moe 2014; Rolland et al. 2016). In the last years, multiple frameworks have been proposed to support employing agile approaches at scale (Dikert et al. 2016; Paasivaara 2017; Turetken et al. 2017). Most frameworks are based on Scrum and lean principles (Uludağ et al. 2017). The Annual State of Agile Report by Digital.ai (2020) classified Scaled Agile Framework (SAFe), Scrum-of-Scrums, Disciplined Agile Delivery (DAD), Large Scale Scrum (LeSS), Enterprise Scrum (ES), and Lean Management as the most popular frameworks in the industry.

Metrics in Large-scale Agile Software Development

Previous research shows that managers require metrics to enhance their decision-making (Kütz 2011; Mills 1988; Neely et al. 2002). Adopting metrics can support organizations with steering, controlling, reporting, early warning, and identifying problems (Kaplan and Norton 1992; Kütz 2011; Neely et al. 2002). A well-known approach for adopting metrics within organizations is the balanced scorecard (BSC) by Kaplan and Norton (1992). In the domain of traditional software development, metrics have played an essential role for decades (Fenton and Neil 2000). Fenton and Neil (2000) use the term 'software metrics' to summarize a wide range of activities dealing with measurements in traditional software engineering. According to Fenton and Bieman (2014), software metrics are used to measure or predict attributes (i.e., measurable characteristics) of processes, products, or resources. Multiple researchers emphasize that metrics successfully applied in traditional software development are not always applicable within agile settings (Hartmann and Dymond 2006; Oza and Korkala 2012; Poligadu and Moloo 2014). A potential conflict between measurement in traditional and agile software development is that traditional approaches measure

against a predefined plan, whereas agile approaches follow the value of embracing change (Korpivaara et al. 2021; Kupiainen et al. 2015). Kupiainen et al. (2015) identify the main differences of measurements between agile and traditional software development as empowerment of teams, customer focus, simplicity, orientation on trends, fast feedback, and responsiveness to change. The Annual State of Agile Report conducted by Digital.ai (2020) identified the most used metrics to measure the success of agile transformations within individual agile projects in the industry. Sorted according to their popularity, these metrics are business value delivered, customer satisfaction, velocity, budget vs. actual cost, and planned vs. actual stories per iteration (Digital.ai 2020). Likewise, the research investigated metrics in non-scaled agile software development environments. According to Kupiainen et al. (2015), the main reasons for using metrics in agile software development are planning and progress tracking for sprints and projects, understanding and improving quality, fixing software process problems, and motivating people. Hossain et al. (2021) provide an overview of agile software development process metrics and identify performance on delivery, cycle completion rate, and development speed as the most important high-level metrics. Aldahmash and Gravell (2018) propose an instrument for measuring the success of agile projects with the help of metrics. Moreover, some research was conducted to identify metric categories. Oza and Korkala (2012) assign metrics in agile software development to the categories Project Management, Engineer, Test, Automation, Strategy, Iteration, and Code. Another proposed approach is mapping metrics in agile software development to the Scrum Events Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective (Kurnia et al. 2018). Kupiainen et al. (2015) suggest a categorization of metrics in agile software development based on reasons and agile principles. Also, research has been conducted to investigate metrics in scaled agile software development environments. Several studies investigated which metrics are adopted by practitioners in general (Dubinsky et al. 2005; Kettunen et al. 2019; Ram et al. 2018; Storti and Clear 2020; Talby et al. 2006) and within measurement programs (Staron and Medig 2011). Further metrics are applied as an indicator for the success in large-scale agile software development based on goals (Shirokova et al. 2020) or to measure the success of the transformation (Kišš and Rossi 2018; Mikalsen et al. 2020; Olszewska et al. 2016). Moreover, metrics are used to measure team performance (Cheng et al. 2009; Korpivaara et al. 2021), to support (Talby and Dubinsky 2009), or report on the portfolio level (Stettina and Schoemaker 2018).

Methodology

Study Design and Overview

We conducted semi-structured expert interviews (Fontana and Frey 2000; Myers and Newman 2007; Seaman 1999) to answer our research questions. To ensure a rigorous collection of empirical evidence, we designed our research methodology based on the guidelines of Myers and Newman (2007) for semi-structured interviews. The chosen research design relies on qualitative data collection and is appropriate since adopting metrics in large-scale agile development is a problem in practice (Seaman 1999). The design of the interview questions was derived during a workshop from our research questions. This study has an exploratory character but also includes descriptive and explanatory elements. Table 1 presents an overview of the study participants. In total, we conducted 23 interviews with practitioners from thirteen organizations. The selection of the interviewees was intentional, and the purpose was to identify interviewees that work in environments typical for large-scale agile development. Large-scale agile software development refers to environments where multiple agile teams work together on a software product (Dingsøyr and Moe 2014; Dumitriu et al. 2019) or to organizations that apply agile methods across the entire organization, also on higher organizational levels. The expert organizations apply to this definition of large-scale agile software development, since within each organization multiple agile teams work together on a software product. Moreover, we only interviewed preferred roles, like Scrum Master, manager, or developer, to ensure enough topic-related knowledge. However, three interviews at one organization were excluded since the interviewees had no sufficient experience regarding the usage of metrics in large-scale agile software development. The included interviewees are characterized by a broad spectrum of job roles and industry backgrounds, leading to a large "variety of voices" covering multiple viewpoints (Myers and Newman 2007).

Data Collection

The interviews were conducted via videotelephony between January and March 2021. On average, the interviews had a length of 82 minutes, resulting in a large amount of empirical data totalling 1.640 minutes of audio recordings. In particular, the interview length depended on the extent of metric usage within the organizations. The interview durations exclude the time at the beginning of each expert interview required for the introduction and clarifications. Whereas the introduction covered the research goal, the clarifications addressed the interview structure and the permission to record the interview. At least two researchers guided each interview to enhance observer triangulation (Runeson and Höst 2009). All interviews followed the same outline, and the questions did not change throughout the course of the interviews. However, due to the interviews' semi-structured nature, the questions were asked in changing the order, or slight variations of the questions' wording did occur. First, questions regarding the professional background and the large-scale agile environment at the corresponding organization were asked. It is worth mentioning that some of the experts answered the questions related to the first part of the questionnaire before the actual interview. Subsequent questions targeted the adoption of metrics (e.g., reasons for using metrics and applied metrics at the organization). In addition to the interviews, we included relevant files shared by our interview partners to facilitate the triangulation of data sources. The results presented in this study are a subset of the overall collected data. We plan to communicate the remaining evidence in separate publications.

No.	Role(s), Large-scale agile software development experience (years)	No.	Role(s), Large-scale agile software development experience (years)
E1	Agile Coach, 11-15	**E11**	Manager, 3-5
E2	Manager, 3-5	**E12**	Scrum Master and Agile Coach, 3-5
E3	Manager, 6-10	**E13**	Agile Coach, 6-10
E4	Scrum Master and Agile Coach, 1-2	**E14**	Scrum Master and Agile Coach, 3-5
E5	Manager and Agile Coach, 6-10	**E15**	Release Train Engineer, 1-2
E6	Developer and Scrum Master, 3-5	**E16**	Scrum Master and Agile Coach, 1-2
E7	Product Owner and Business Analyst, 1-2	**E17**	Manager and Agile Coach, 3-5
E8	Software Architect, Solution Architect, 3-5	**E18**	Agile Coach, 3-5
E9	Scrum Master, Agile Coach and Developer, 11-15	**E19**	Scrum Master and Agile Coach, 6-10
E10	Software Architect, 11-15	**E20**	Release Train Engineer, 6-10

Table 1. Interview Partners

Data Analysis

Two researchers transcribed the audio recordings, and another researcher ensured quality by reviewing the completeness and consistency of each transcript. We guaranteed the anonymity of the interviewees by dismissing sensitive information. In addition, we assigned unique pseudonyms to refer to the experts and organizations throughout this paper. We combined deductive and inductive coding (Miles et al. 2020) to code the interview transcripts, the pre-filled questionnaire parts, and the additional files. The coding was performed following the two-cyclic approach described by Saldaña (2021). First, we assigned descriptive (Miles et al. 2020) first-cycle codes to text segments, which in turn were assigned to the higher-order second-cycle codes. For the second-cycle deductive coding, we used an initial list of higher-order codes (Saldaña 2021) as a starting point. Our list was aligned with the questions of the interview questionnaire. We created additional second-cycle codes inductively whenever new findings were made while analyzing or a text fragment could not be assigned to an existing code category (Saldaña 2021). The introduction of new codes required the re-codification of the previously coded data. Consequently, the final coding was

developed incrementally. We used MAXQDA[1] and Microsoft Excel[2] to code and analyze the transcripts. We discussed and resolved conflicts by mutual consent to increase the result validity. Whenever the interpretation of the evidence required clarification, we contacted the interviewees again to resolve ambiguities.

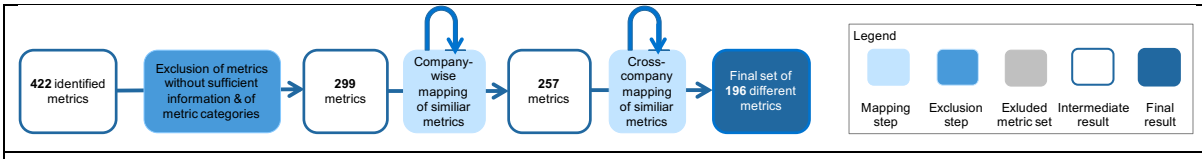


Figure 1. Metric Filtering and Mapping Steps

In the empirical data, we identified an initial set of 422 potential metrics. Based on predefined exclusion criteria, we reduced the initial set of metrics to increase the quality and relevance of the finally included metrics. An overview of the filtering and mapping process is provided in Figure 1. Whenever experts just mentioned a category of metrics like "financial KPIs" (E2) or "code quality metrics" (E14) instead of a specific metric, we did not count the category as a metric. In total, we excluded 78 metrics that did not meet our quality requirements due to insufficient information regarding their organizational level of usage, calculation, and cadence necessary for a profound understanding. From the remaining 299 metrics, we excluded 103 duplicates within the same organization and between the organizations resulting in 196 finally included metrics.

Level	Description
Team level	The organizational scope is a single scrum team with around nine team members and/or a team with a single team backlog shared by all team members.
Program level	The organizational scope is multiple teams that develop one product and/or develop in the same cadence/cycle and/or are grouped into an organizational unit like an ART or a solution train. Moreover, this level covers multiple groups of teams that develop sub-products of one large product and topics directly related to a single product or large solution.
Portfolio level	The organizational scope is a group of programs.
Organizational level	The organizational scope is multiple portfolios, topics related to the corporate strategy influenced by government and regulations, as well as organizational topics that relate to the overall organization and are not directly related to the program and portfolio structures.

Table 2. Categorization of Organizational Levels in the Context of Large-Scale Agile Software Development Inspired by Korpivaara et al. (2021) and Stettina and Schoemaker (2018)

We classified each of the 196 included metrics as either agile or traditional metrics. At least one of the following criteria had to be fulfilled that a metric was classified as an agile metric:

- Calculation based on agile concepts such as story points or user stories (Cohn 2005)
- Measurement of aspects related to continuous integration and delivery (Stolberg 2009)
- Relation to agile events such as the sprint planning (Schwaber and Sutherland 2020), also on a scaled level
- Relation to agile artifacts like the product or sprint backlog (Schwaber and Sutherland 2020), also on a scaled level

The remaining metrics, which did not fulfill any of the mentioned criteria for agile metrics, were classified as traditional metrics.

[1] https://www.maxqda.com

[2] https://www.microsoft.com/sv-se/microsoft-365/excel

Based on the categorization developed by Korpivaara et al. (2021), Table 2 describes the levels we used to assign the identified metrics to the team, program, portfolio, and organizational level. It is worth mentioning, that the expert environments were heterogeneous and different terms were used to describe the same level. Consequently, we performed a mapping between the levels mentioned in the expert environment and our categorization schema.

Results

Overview of the Experts' Development Organizations

Table 3 provides an overview of the of the experts' development organizations.

Company	Description of the experts' development organization
CarCo1	The company's development group is at the beginning of its SAFe transformation and aims to implement the Full SAFe configuration. The programs develop software products for the company's end-users and solutions for internal stakeholders. Within the program of E1, 750 employees work together.
CarCo2	The development organization of E12 uses Scrum@Scale and plans a transition to LeSS in the mid-term future. E12 was not allowed to provide information regarding the implemented software solution. Within the program, approximately 100 employees work together.
ConsultCo	The program of E4, E6, E7, E8 and E11 implements a customized scaling agile approach similar to LeSS. The approach also includes Scrum-of-Scrum elements. Within the program, 50 employees in different teams work together to develop a software solution for a client. The client implements a portfolio level to keep track of all projects conducted with ConsultCo.
BankCo	The development organization of E2 and E18 uses a custom framework similar to SAFe and consists of around 6.000 people grouped into 18 higher-level key areas. Whereas E2 is responsible for the enterprise architecture, E18 works in one of the key areas of the development organization consisting of 400 employees.
HealthCo	The company has multiple heterogeneous development groups using different frameworks. The program of E3 and E5 implements a custom lean approach combining elements of LeSS and SAFe. The program develops software for a medical product that is sold to end customers. Within the program, approximately 450 employees are working together. The development organization of E10 implements Scrum-of-Scrums and develops software for internal stakeholders. Within this program, approximately 500 employees are working together.
InsureCo1	The company's whole IT department is organized in an agile but heterogeneous way. E14 is working in an area implementing SAFe. The program of E14 consists of 75 employees that develop software for the business line health insurances in the area of sales and operation. E17 is working in a development area implementing Nexus to develop software for non-personal non-life insurance. E17's specific program consists of around 90 people. E15 is working in a development area that is using SAFe as well. Within the program of E15, 80 employees develop a cross-sectional solution and several smaller related sub-products used by company-internal customers.
InsureCo2	The development organization of E16 is implementing Scrum@Scale and comprises approximately 35 employees. The program intends to develop software that supports insurance processes.
RetailCo1	The development group of E13 uses a custom framework combining elements of SAFe, Spotify, and Scrum-of-Scrums and comprises 600 people. E13's program consists of 120 people intending to develop a warehouse management system.
RetailCo2	The company's international IT department applies different scaling agile frameworks. Within the development area of E19 work around 3.700-4.700 employees. The program of E19 uses elements of LeSS, Scrum-of-Scrums, and Spotify. Within this program, in total, about 80 internal employees develop a software solution platform in the area of sales.
TeleCo	The development organization of E9 uses elements of LeSS and Scrum-of-Scrums. The program comprises 100 developers grouped into eight teams that develop a software solution to support the implementation of the communication standard Global System for Mobile Communications (GSM).
Transport Co	The development group of E20 implements a Full SAFe configuration and incorporates elements of other frameworks. In total, within the program of E20 approximately 500 people are working from two different subsidiaries of the parent organization. Also, external employees are involved in the program. They work together on software products in the context of sales and ticket booking.

Table 3. Overview of Expert Organizations

Reasons for Using Metrics in Large-Scale Agile Software Development

To answer RQ1, we asked for the experts' reasons for using metrics in their large-scale agile software development environment. In total, we identified and categorized 237 such reasons. Table 4 provides an overview of the reason categories and the contexts each reason category is referring to. The most frequently mentioned reason categories were *Transparency (39%), Improvement (22%), Controlling (13%), Planning (8%), Employee motivation (4%), Agility (3%), Problem identification (3%),* and *Stakeholder involvement (3%)*. On a more detailed level, each reason category can refer to one of the following contexts: *Development organization and processes, Product, Entire organization, Customer, Finance, Employee, and Stakeholder*. For example, the experts in our study mentioned the reason of increasing Transparency

regarding the context *Development organization and processes* 56 times. Whenever a reason category was mentioned, but the expert did not clearly reference it to a context, we assigned the reason to the context *Generic*. Subsequently, we describe each reason category in more detail.

Transparency

In the *Development organization and processes*, metrics are applied to enhance transparency regarding the efficiency of the processes and development environments (E3, E19) on the program (E4, E8, E10, E11, E17) and team levels (E6, E17). For instance, E15 mentioned that the number of features per program increment appears to be constrained by achieving transparency on potential problems in programs. Release (E9) and epic burn-downs (E12) are utilized to clarify the program's progress (E9, E12) and deviations from its plan (E12). Likewise, at the team level, the sprint burn-down is used to create transparency as a basis for self-improvement (E16). According to expert E3 the sprint burn-down helps to *"to stimulate the discussion and create transparency"*. Velocity is predominately implemented on the team level to gain insight into the teams' progress (E17), stability (E1), its performance (E17), and to identify potential problems (E17). Regarding the *Product* context, metrics are used to create transparency with the respect to the value created by developing products (E19) and their performance when used by customers (E3). Expert E3 stated that creating transparency early is beneficial since *"[…] the earlier we find those topics, the easier we can do something against it"*. For example, on the portfolio level, the delivery rate of quarterly business review epics was introduced to create transparency on the actual delivery (E18). According to E9, the release burn-down is adopted to identify the need for de-scoping of the product's functionality. Measuring the total defects is done to gain transparency on the stability of a product (E8). Expert E8 made clear that they do not count the defects to blame the teams: *"[…] we don't try to blame the teams […] it's more like we try to generally aggregate how good we are] "*. Additionally, the defect-feature ratio is implemented to assess product quality (E1). Finally, the metric feature usage is measured to make bad investments into features transparent (E1). Some experts mentioned transparency as a reason for using metrics (E5, E6, E7, E8, E12, E13, E15) without specifying a particular context. Metrics are used to increase transparency regarding the current state (E4), progress (E17), problems, and the need for actions (E2) and reactions (E2). The experts also use metrics to enhance transparency within the *Entire organization* (E18). According to E18 this especially important within large organizations: "[…] *we have 7,000 people working in this agile environment […] to get really an overview where […] we actually stand on the highest level, on the delivery organization level you need […] metrics."* Metrics are used to make the organizations' progress (E2, E18) and current situation (E1, E8, E12, E13) transparent. Within the *Customer* context, metrics are used to enhance transparency on customer-related topics. For example, the release burn-down on the program level is used for realistic customer offers based on detailed planning information (E9). Further, improving the transparency regarding customer satisfaction is one main reason for adopting customer satisfaction metrics (E6, E8). Inside the *Finance* context, metrics are measured to contribute towards financial transparency (E6), for instance, on revenue (E13). Additionally, metrics are used as early indicators for economic success (E11) and to create transparency regarding the profitability of a program (E7). In the *Employee* context achieving transparency of employee-related topics is a reason to implement metrics. For example, employee satisfaction metrics (E6) are implemented to upturn transparency of the employee (E6, E8) and team satisfaction (E6). Within the *Stakeholder* context, metrics are used to surge transparency regarding stakeholders. An example metric is measuring the transparency of the performance of collaboration partners (E19).

Improvement

Metrics are used to improve the efficiency (E3, E19, E20) and team performance (E6) in the program within the *Development organization and processes*. Further metrics serve as the basis for the self-improvement of teams (E2, E6). Sprint report metrics like the number of remaining story points after a sprint help teams improve regarding their way of working and their commitment (E16). The metric planned team velocity is used to avoid overloading the team with work items and thus improve the team's management (E17). To improve the stability of the development flow of programs, the metric feature lead time is applied (E1). Expert 1 stated that using the metric feature lead time is helpful because the *"biggest advantage is we can stabilize […] the flow of production and that is crucial for any success. If you have a chaotic system you won't be successful. […] And so you have to care about the flow of production development. And so, that's the biggest advantage, when we are able to measure that right"*. The experts also use metrics to achieve

improvements in the *Product* context. Particularly metrics are used to improve the quality of the product (E17). The purpose of measuring test errors is to improve the developer's analytical capabilities and achieve more stable software (E5). Metrics for measuring the current product performance or stability are used to improve product competitiveness (E3). Additionally, customer satisfaction is measured to improve the quality of features (E14). Whenever experts mentioned improvement as a reason for using metrics (E3, E14, E18) without specifying a specific context, we assigned the reason to the context *Generic*. Metrics are implemented since they support the identification of improvement potentials (E4, E8, E11, E18). Further, metrics are used to better reach expectations (E9) and goals (E3). Also, metrics are a means to enhance comparability, which supports the identification of best practices (E3). Within the context *Entire organization,* metrics are implemented to contribute to organization-wide improvement. For instance, metrics are used to initiate organizational change (E3) or stabilize the overall organization (E1). Managers also use metrics to enrich the reporting towards defined goals (E1). In the context *Employee,* metrics are adopted to improve employee-related topics like employee satisfaction. Therefore, metrics can be the basis for detecting improvement potentials regarding employee and team satisfaction (E6). Also, metrics can be used to improve the managers' awareness of teams' health (E17) and promote employees' self-responsibility (E15). In the context *Customer,* metrics are mainly used to enhance customer satisfaction. An example is the metric defects in production, which, when minimized, helps improve customer satisfaction (E9).

Controlling

Within the *Development organization and processes*, metrics are used to control the progress of the programs (E20). For example, the metrics release burn-down or number of features per program increment are measured for steering programs in the right direction (E15). Within the program, the metric release burn-down is used to control the complex development processes (E3, E12). The metric velocity is used to steer teams (E16), and the number of completed user stories of a program epic enables the controlling of development goals. Whenever experts mentioned controlling as a reason for using metrics *(E3, E4, E7, E10, E12, E14)* without specifying a particular context, we assigned the reason to the context *Generic*. Expert E3 explained why using metrics to support controlling is important: *"So, for me one of the observations, which is already ten years old, you do not control what you cannot measure"*. In the context *Product,* metrics are used for product-related controlling. For example, the metric total defects enable the controlling of defects (E4). Also, within the *Entire organization,* metrics are used for controlling. For example, metrics are used to control the overall organization (E1) and steer the organization in an agile way (E1). Managers also use metrics to control the achievement of personal goals or bonuses (E1). In the context *Employee,* metrics are applied to control the employees working in an organization (E1, E2). As stated by expert E2 metrics are a suitable instrument to control employees because "*Well, the thing is that the metrics […] make people do the right things"*. Furthermore, metrics are implemented to control employee and team satisfaction as well as team health (E6). As stated by expert E6 controlling is *"also about the team satisfaction, to control […] the team health, how the situation looks like, how to avoid some problems to coordinate some atmosphere in the team, so this also is really important."* In the *Customer* context, metrics are used, for example, to control the competitiveness based on price (E19).

Planning

Metrics such as the remaining story points are used for enabling planning and avoiding over-planning (E16) in the *Development organization and processes*. The metric velocity on the team level is measured to assist sprint capacity planning and to enhance the planning of the program's roadmap (E16). Experts use the program velocity factor per sprint to support the planning in programs (E7, E11), for example, regarding milestones (E8). With the help of the lead time of large initiatives on the program level, the initiatives are prioritized (E13). Whenever experts mentioned planning as a reason for using metrics (E7, E8, E11, E13) without specifying a particular context, we assigned the reason to the context *Generic*. Further, metrics are adopted since they can improve predictability in general (E5). In the *Product* context, metrics are used to support the planning of product-related topics such as planning of delivery roadmaps (E16). Further, metrics are implemented to improve the delivery predictability (E6, E10, E16, E17, E20). Accordingly, expert E17 stated: *"forecasting delivery dates is still a thing. We need to do it, there is no way around it"*. Moreover, expert E20 highlighted the need to improve the delivery predictability: *"[…] we have a hard deadline. We want to replace our existing system at the end of 2022. What we want to do now with the*

metrics is to see if our velocity is fast enough". The metric remaining defects per sprint in a program is measured to support the capacity planning to fix a product's defects (E11).

Reason (Occurrences)	Experts	# Experts
Transparency (111, 39%)		
Development organization and processes, (56, 20%)	E1, E2, E3, E4, E6, E7, E8, E9, E10, E11, E12, E13, E15, E16, E17, E18, E19, E20	18 (90%)
Product, (18, 6%)	E1, E2, E3, E8, E9, E11, E14, E15, E18, E19	10 (50%)
Generic, (13, 5%)	E2, E4, E5, E6, E7, E8, E9, E12, E13, E15	10 (50%)
Entire organization, (9, 3%)	E1, E2, E8, E10, E12, E13, E18, E20	8 (40%)
Customer, (7, 2%)	E1, E7, E8, E9, E14, E17	6 (30%)
Finance, (5, 2%)	E1, E6, E7, E9, E13	5 (25%)
Employee, (2, 1%)	E6, E8	2 (10%)
Stakeholder, (1, <1%)	E19	1 (5%)
Improvement 62 (22%)		
Development organization and processes, (28, 10%)	E1, E2, E3, E5, E6, E10, E12, E14, E16, E17, E19, E20	12 (60%)
Generic, (13, 5%)	E3, E4, E7, E8, E9, E11, E14, E15, E18	9 (45%)
Product, (12, 4%)	E1, E3, E5, E10, E11, E14, E15	7 (35%)
Entire Organization, (4, 1%)	E1, E3	2 (10%)
Employee, (4, 1%)	E6, E13, E15, E17	4 (20%)
Customer, (1, <1%)	E9	1 (5%)
Controlling (38, 13%)		
Generic, (14, 5%)	E3, E4, E7, E10, E12, E14, E15	7 (35%)
Development organization and processes, (10, 3%)	E2, E3, E5, E6, E12, E15, E16, E20	8 (40%)
Entire organization, (5, 2%)	E1, E12	2 (10%)
Employee, (4, 1%)	E1, E2, E6	3 (15%)
Product, (3, 1%)	E4, E7, E10	3 (15%)
Finance, (2, 1%)	E19	1 (5%)
Planning (24, 8%)		
Development organization and processes, (12, 4%)	E3, E4, E7, E8, E11, E13, E16, E17	8 (40%)
Generic, (7, 2%)	E5, E7, E8, E11, E13	5 (25%)
Product, (6, 2%)	E6, E10, E11, E16, E17, E20	6 (30%)
Employee motivation (11, 4%)		
Development organization and processes, (9, 3%)	E1, E4, E6, E7, E8, E15, E16, E17, E20	9 (45%)
Entire organization, (2, 1%)	E13, E15	2 (10%)
Agility (9, 3%)		
Generic, (8, 3%)	E2, E7, E12, E13	4 (20%)
Development organization and processes, (1, <1%)	E13	1 (5%)
Problem identification (9, 3%)		
Generic, (3, 1%)	E4, E12, E17	3 (15%)
Product, (3, 1%)	E7, E11, E12	3 (15%)
Development organization and processes, (3, 1%)	E6, E12, E20	3 (15%)
Stakeholder involvement (8, 3%)		
Customer, (6, 2%)	E6, E8, E11, E14, E15, E16	6 (30%)
Generic, (1, <1%)	E14	1 (5%)
Entire organization, (1, <1%)	E16	1 (5%)

Table 4. Reasons for Using Metrics in Large-Scale Agile Software Development

Employee Motivation

In the *Development organization and processes* metrics are used to enhance the motivation of the employees. Metrics like team velocity (E17) or sprint burn down (E1, E16) are measured to boost motivation. Using the sprint burn down chart as a motivator is described by expert E1: *"[…] the burn down chart is very nice if you have it on a task level, because it is a big motivator. When it's very prominent in the team space […] and every time one task is finished it goes down it's a very big motivator".* The program velocity can strengthen the motivation of employees working in the program (E20). In the context *Entire organization,* metrics are used to encourage and inspire the workforce (E13) and ensure its commitment (E15).

Agility

Most experts mentioned *Agility* as a reason to use metrics without specifying a specific context (E2, E7, E12, E13). Metrics are implemented to enable (E2) and foster reactivity (E2, E7, E12). Expert E12

highlighted the importance to communicate that *"[…] it's mostly about trust and making the people understand you don't want to control them, you just want to understand on a higher level what's going on and be able to react when there are problems""*. Further, adopting metrics allows adaption to change (E7), including the adjustment of goals and strategy (E13).

In the Development organization and processes, metrics are used to increase agility by implementing measures such as velocity to react, replan, and perform readjustments of goals and the strategy in programs (E13).

Problem Identification

Most experts mentioned *Problem identification* as a reason for implementing metrics without specifying a specific context (E4, E12, E17). Thus, we assigned these occurrences to the context *Generic*. In the context *Product,* metrics are implemented to identify product-related problems. For instance, the metric completed user stories of an epic on program level can indicate delivery delays (E12). The metric total reported defects is measured to reveal problems regarding the product's usability (E11). In the *Development organization and processes,* metrics are used to indicate problems in programs. Therefore, for example, the metrics planned vs. actual program velocity (E20) and burn-down on program level (E12) are measured.

Stakeholder Involvement

Metrics are used to facilitate customer communication (E5, E8, E16) and joint planning (E5, E16). For example, to support customer communication and contract negotiation, the metric program velocity factor per sprint is implemented. The metrics number of defects in production is measured since it positively affects the relationship with the customers (E15). One expert (E14) mentioned supporting customer involvement as a reason without assigning it to a specific context. In the *Entire organization,* metrics are used to facilitate organization-wide stakeholder communication. For example, the epic burn-down on the program level is measured since it can support management communication (E16).

Further Reasons

Further reasons for using metrics are *Risk mitigation, Decision making, Goal setting, Creation of trust, and Coordination*. For example, metrics as the planned vs. actual program effort are implemented to mitigate product-related risks by facilitating early escalations (E15). Further, metrics are used to support *Decision making* since they provide relevant information (E5, E14). Metrics are also used for *Goal setting* because they can define goal targets in the form of lagging indicators (E9) and provide information regarding the status of goal-fulfillment (E2). Metrics, such as the release burn-down, can also facilitate *Trust* between the product management and teams of a program (E9). A final reason for using metrics is *Coordination*. In the *Development organization and processes,* metrics are used to support the coordination of multiple agile teams (E12).

Agile and Traditional Metrics on Organizational Levels

We were also curious about the organizational levels (i.e., team, program, portfolio, organizational) on which metrics are implemented and their type (i.e., traditional, or agile). Our results are not of quantitative nature as we do not involve any mathematical models. We rather intend to count the number of occurrences of metrics at the expert organizations within different categories (see Figure 2). We identified most metrics (140, 71.43%) on the program level, followed by the team level (42, 21.43%) and organizational level (9, 4.59%). The least metrics were found on the portfolio level (5, 2.55%). On all levels except the organizational level, agile metrics were rather used than traditional metrics. The dominance of agile metrics in comparison to traditional metrics is very prominent on the team level. Out of the 42 implemented metrics on the team level, are 38 metrics of agile nature (90.48%). On the organizational level, we only identified traditional metrics.

Most Used Metrics Among the Expert Organizations

To identify the most used metrics among the expert organizations, we asked the experts which metrics they use within their organization.

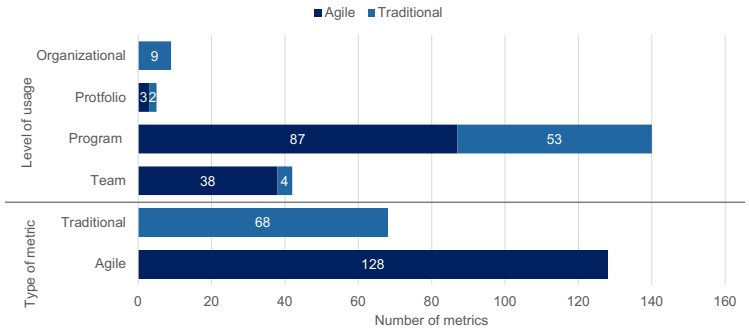


Figure 2. Distribution of Metrics per Organizational Level of Usage

Metric*			Organizations	#org.
Story point estimation: *Sum of the story points estimated for a user story.*	A	T	CarCo1, CarCo2, ConsultCo, HealthCo, InsureCo1, InsureCo2, RetailCo1, RetailCo2, TeleCo, TransportCo	10 (91%)
Velocity: *Sum of the story points of all user stories completed (Done) by a team in a sprint.*	A	T	BankCo, CarCo1, CarCo2, ConsultCo, HealthCo, InsureCo1, InsureCo2, RetailCo1, RetailCo2, TransportCo	10 (91%)
Sprint burn-down: *Sum of the story points of all Open user stories in a sprint of a team in relation to the time left in the sprint reduced for each completed (Done) user story.*	A	T	BankCo, CarCo1, CarCo2, HealthCo, InsureCo1, InsureCo2, TransportCo	7 (64%)
Defects in production: *Number of defects in the production environment(s) of the software product for each severity category.*	TD	P	ConsultCo, HealthCo, InsureCo1, TeleCo	4 (36%)
Planned velocity: *Sum of the story points of all user stories planned to be completed by a team in a sprint.*	A	T	ConsultCo, HealthCo, RetailCo2, TransportCo	4 (36%)
Program velocity per sprint: *Sum of the story points of all user stories completed (Done) by all teams in a program in a sprint.*	A	P	ConsultCo, HealthCo, RetailCo2, TransportCo	4 (36%)
Number of features per PI: *Number of features realized (Done) in a PI by a program.*	A	P	CarCo1, HealthCo, InsureCo1	3 (27%)
Planned program velocity per sprint: *Sum of the story points of all user stories planned to be completed by all teams in a program in a sprint.*	A	P	ConsultCo, RetailCo2, TransportCo	3 (27%)
Release-burn-down: *Sum of the story points of all user stories planned for a release that still need to be realized (Done) by all team in a program in relation to the time left until the release.*	A	P	HealthCo, InsureCo1, TeleCo	3 (27%)
Test coverage: *Percentage of the number of lines of code all tests currently execute in relation to the total number of lines of code of the software that is to be tested.*	TD	P	ConsultCo, InsureCo1, RetailCo2	3 (27%)

Table 3. *T = Team, P = Program; A = Agile, TD = Traditional

Similar to the section before we do not follow a quantitative research strategy to answer this research question. According to the *rule of* three (Coplien and Alexander 1996) we added a metric to Table 3 of the most used metrics whenever it was implemented in at least three organizations. In most organizations (10, 91%), the metrics *Story point estimation* and *Velocity* are measured. It is noteworthy that both metrics are implemented on the team level. The second most used metric is the *Burn-down chart* implemented in seven organizations (64%). The third most used metrics are *Defects in production, Planned velocity,* and *Program velocity per sprint,* which are implemented in four organizations (36%). It is noteworthy that most metrics (166, 84.69%) are custom metrics since they were found in only one organization.

Discussion

Four key findings emerge from this expert interview study. First, metrics are mainly used to increase the transparency within the development organization on the program and team level. This observation is in line with Ebert et al. (2005) and Jethani (2013), who argue that metrics can be used to monitor, control, assess, manage, and improve the software process. Similar to Kupiainen et al. (2015) finding that metrics are used for progress tracking, we observed that the experts adopted release and epic burn-downs to clarify the program's progress and deviations from its plan. In contrast to Basili et al. (1994; 2010), who is highlighting the importance of aligning metrics with goals, using metrics for goal-setting was among the least mentioned reasons. Only a few experts revealed to use metrics to define goal targets and provide information regarding the status of goal-fulfillment. Moreover, our finding suggests that in scaling agile environments, metrics are rather adopted to fulfill traditional use cases from general management like steering, controlling, reporting, early warning, and identifying problems (Kaplan and Norton 1992; Kütz 2011; Neely et al. 2002). Second, on the team level, mainly agile metrics are used. We derive from this finding that the organizations tend to refrain from implementing traditional metrics on the team level, since they can undermine the cultural change towards an agile way of working (Hartmann and Dymond 2006). This finding is also backed by our third key finding revealing that the organization's most-used metrics are the story point estimation of user stories, velocity, and sprint burn-down. This finding corresponds with the results of Kupiainen et al. (2015), who identified velocity and effort estimates as the most important agile industrial metrics. Moreover, this finding is in line with the finding of Schwaber (2004), that within single agile teams, metrics like the burn-down chart or velocity are applied. Likewise, the Annual State of Agile Report by Digital.ai (2020) presents velocity and iteration burn-downs as the most used metrics in the industry. Fourth, most organizations implement metrics on the program level. This finding is in line with Kettunen et al. (2019), who reported that program metrics such as the cycle time, release cycle, lead time of features and epics, and defects are the most frequently used metrics. Further, this finding is in line with previous research showing that managers, which act on higher organizational levels, require metrics to enhance their decision-making (Kütz 2011; Mills 1988; Neely et al. 2002).

This study's results contribute to research on large-scale agile software development. To the best of our knowledge, no research exists investigating reasons for metrics in scaling agile environments. The identified reasons and their associated contexts can serve practitioners to identify usage scenarios for metrics in their organization. Further, we contribute to existing research by highlighting that agile metrics are frequently used on the program level and by identifying the most used metrics among the expert organizations. Consequently, practitioners working in scaling agile environments with low metric maturity regarding the number of implemented metrics should consider adopting agile metrics beyond the team level on the program level. Further, they can select metric candidates from the most used metrics that seem to have a convincing cost-benefit ratio since most expert organizations use them.

We apply the assessment schema of Runeson and Höst (2009) in this study to evaluate possible threats of validity. Since our expert study has an explanatory character and does not aim to determine casual relationships, internal validity is not a considered as a threat. The concern of construct validity is whether the research design is appropriate to investigate our three research questions. We addressed this threat by including several sources during the data collection. In addition to semi-structured interviews with experts working in different roles and industries, we incorporated internal documents of the development organizations in our research. External validity relates to the generalization of our findings. Since we clearly outline how our results relate to the experts and organizations of our study, we effectively countermeasure this threat. Reliability is achieved through the rigorous conduct of this study and by ensuring that a replication of our study would yield the same results (i.e., researchers bias). To counteract researchers bias, three countermeasures were taken. First, at least two researchers were present during each interview. Second, all transcripts were reviewed by a second researcher. Third, whenever the interpretation of the evidence required clarification, we contacted the interviewees again to resolve ambiguities.

Conclusion

Industry and research show an increased interest in scaling agile methods beyond the team level (Digital.ai 2020; Dikert et al. 2016). However, scaling agile methods leads to increased organizational scope as well as complexities (Brown et al. 2013). Adopting metrics can help in managing those challenges (Brown et al.

2013; Talby and Dubinsky 2009). Currently, in the extant literature, publications regarding the application of metrics in scaling agile environments, investigating the reasons for using metrics, and studying which kind of metrics practitioners implement and find important is lacking. Against this backdrop, we conducted an expert interview study comprising 20 experts from 11 organizations and identified 196 metrics and 237 reasons. The identified metrics are not domain specific and can be applied independent of the industry context of an organization. We classified the identified metrics as agile or traditional, identified their organizational level of occurrence, and applied the 'rule of three' to derive reoccurring metrics. Further, we determined eight main reasons for using metrics in large-scale agile software development endeavors and described in detail in which contexts those reasons occur. Finally, this paper leaves some room for future research. First, we aim to publish the remaining evidence regarding goals, challenges, success factors, advantages, and disadvantages of using metrics in large-scale agile development endeavors. Second, we would like to publish all the identified metrics in a metric catalog based on a structured metric fact sheet. Also, we would like to investigate the implementation of the metric catalog within organizations.

References

Aldahmash, A. M., and Gravell, A. 2018. "Measuring Success in Agile Software Development Projects: A Gqm Approach," *Proceedings of the 13th International Conference on Software Engineering Advances (ICSEA)*, Nice, France: IARIA, pp. 38-44.

Ambler, S., and Lines, M. 2014. "Scaling Agile Software Development: Disciplined Agility at Scale," in: *Disciplined Agile Consortium White Paper Series*.

Attaran, M. 2004. "Exploring the Relationship between Information Technology and Business Process Reengineering," *Information & management* (41:5), pp. 585-596.

Basili, V. R., Caldiera, G., and Rombach, H. D. 1994. "The Goal Question Metric Approach," in: *Encyclopedia of software engineering*. pp. 528-532.

Basili, V. R., Lindvall, M., Regardie, M., Seaman, C., Heidrich, J., Münch, J., Rombach, D., and Trendowicz, A. 2010. "Linking Software Development and Business Strategy through Measurement," *Computer* (43:4), pp. 57-65.

Beck, K. 2000. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., and Jeffries, R. 2001. "Manifesto for Agile Software Development."

Beedle, M., Coplien, J. O., Sutherland, J., Østergaard, J. C., Aguiar, A., and Schwaber, K. 2010. "Essential Scrum Patterns." Retrieved 11. November, 2019, from https://www.hillside.net/plop/2010/papers/beedle.pdf

Bird, S. M., Sir David, C., Farewell, V. T., Harvey, G., Tim, H., and Peter C, S. 2005. "Performance Indicators: Good, Bad, and Ugly," *Journal of the Royal Statistical Society: Series A (Statistics in Society)* (168:1), pp. 1-27.

Brown, A. W., Ambler, S., and Royce, W. 2013. "Agility at Scale: Economic Governance, Measured Improvement, and Disciplined Delivery," *Proceedings of the 35th International Conference on Software Engineering (ICSE)*, San Francisco, CA, USA: IEEE, pp. 873-881.

Cheng, T.-H., Jansen, S., and Remmers, M. 2009. "Controlling and Monitoring Agile Software Development in Three Dutch Product Software Companies," *ICSE Workshop on Software Development Governance*, Vancouver, BC, Canada: IEEE, pp. 29-35.

Cohn, M. 2005. *Agile Estimating and Planning*. Pearson Education.

Coplien, J. O., and Alexander, A. W. O. 1996. "Software Patterns."

Digital.ai. 2020. "14th Annual of State of Agile Report." Retrieved 29. January, 2022, from https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf

Dikert, K., Paasivaara, M., and Lassenius, C. 2016. "Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review," *Journal of Systems and Software* (119), pp. 87-108.

Dingsøyr, T., and Moe, N. B. 2014. "Towards Principles of Large-Scale Agile Development," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen (eds.), Rome, Italy: Springer, Cham, pp. 1-8.

Dingsøyr, T., Moe, N. B., Fægri, T. E., and Seim, E. A. 2018. "Exploring Software Development at the Very Large-Scale: A Revelatory Case Study and Research Agenda for Agile Method Adaptation," *Empirical Software Engineering* (23), pp. 490-520.

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B. 2012. "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," *Journal of Systems and Software* (85:6), pp. 1213-1221.

Dubinsky, Y., Talby, D., Hazzan, O., and Keren, A. 2005. "Agile Metrics at the Israeli Air Force," *Proceedings of the Agile Development Conference (ADC'05)*, Denver, CO, USA: IEEE, pp. 12-19.

Dumitriu, F., Meşniţă, G., and Radu, L.-D. 2019. "Challenges and Solutions of Applying Large-Scale Agile at Organizational Level," *Informatica Economica* (23:3), pp. 61-71.

Ebert, C., Dumke, R., Bundschuh, M., and Schmietendorf, A. 2005. *Best Practicesin Software Measurement: How to Use Metrics to Improve Project and Process Performance*, (1 ed.). Heidelberg, Germany: Springer.

Eckerson, W. W. 2010. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*, (2 ed.). John Wiley & Sons.

Fenton, N., and Bieman, J. 2014. *Software Metrics: A Rigorous and Practical Approach*, (3 ed.). CRC Press.

Fenton, N. E., and Neil, M. 2000. "Software Metrics: Roadmap," *Proceedings of the Conference on the Future of Software Engineering*, Limerick, Ireland: ACM, pp. 357-370.

Fontana, A., and Frey, J. H. 2000. "The Interview: From Structured Questions to Negotiated Text," *Handbook of qualitative research* (2:6), pp. 645-672.

Geras, A., Smith, M., and Miller, J. 2004. "A Prototype Empirical Evaluation of Test Driven Development," *Proceedings of the 10th International Symposium on Software Metrics*, Chicago, IL, USA: IEEE, pp. 405-416.

Hamed, A. M. M., and Abushama, H. 2013. "Popular Agile Approaches in Software Development: Review and Analysis," *Proceedings of the International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)*, Khartoum, Sudan: IEEE, pp. 160-166.

Hartmann, D., and Dymond, R. 2006. "Appropriate Agile Measurement: Using Metrics and Diagnostics to Deliver Business Value," *Proceedings of the AGILE 2006 (AGILE'06)*, Minneapolis, MN, USA: IEEE, pp. pp. 6 pp.-134.

Highsmith, J., and Cockburn, A. 2001. "Agile Software Development: The Business of Innovation," *Computer* (34:9), pp. 120-127.

Highsmith, J. A., and Highsmith, J. 2002. *Agile Software Development Ecosystems*. Addison-Wesley Professional.

Hossain, S. S., Ahmed, P., and Arafat, Y. 2021. "Software Process Metrics in Agile Software Development: A Systematic Mapping Study," *Proceedings of the International Conference on Computational Science and Its Applications*, Cagliari, Italy: Springer, Cham, pp. 15-26.

ISO. 2017. "Iso/Iec/Ieee International Standard - Systems and Software Engineering–Vocabulary," in: *ISO/IEC/IEEE 24765: 2017*.

Janus, A., Dumke, R., Schmietendorf, A., and Jäger, J. 2012. "The 3c Approach for Agile Quality Assurance," *3rd International Workshop on Emerging Trends in Software Metrics (WETSoM)*, Zurich, Switzerland: IEEE, pp. 9-13.

Javdani, T., Zulzalil, H., Ghani, A. A. A., Sultan, A. B. M., and Parizi, R. M. 2012. "On the Current Measurement Practices in Agile Software Development," *International Journal of Computer Science Issues (IJCSI)* (9:4), pp. 127-133.

Jethani, K. 2013. "Software Metrics for Effective Project Management," *International Journal of System Assurance Engineering and Management* (4:4), pp. 335-340.

Kaplan, R. S., and Norton, D. P. 1992. "The Balanced Scorecard: Measures That Drive Performance," *Harvard Business Review* (83:7), pp. 71-79.

Kettunen, P., Laanti, M., Fagerholm, F., and Mikkonen, T. 2019. "Agile in the Era of Digitalization: A Finnish Survey Study," *Proceedings of the International Conference on Product-Focused Software Process Improvement*, Barcelona, Spain: Springer, Cham, pp. 383-398.

Kišš, F., and Rossi, B. 2018. "Agile to Lean Software Development Transformation: A Systematic Literature Review," *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS)*, Poznan, Poland: IEEE, pp. 969-973.

Korpivaara, I., Tuunanen, T., and Seppänen, V. 2021. "Performance Measurement in Scaled Agile Organizations," *Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS)*, Kauai, Hawaii, USA, pp. 6912-6921.

Kruchten, P. 2013. "Contextualizing Agile Software Development," *Journal of Software: Evolution and Process* (25:4), pp. 351-361.
Kumar, G., and Bhatia, P. K. 2012. "Impact of Agile Methodology on Software Development Process," *International Journal of Computer Technology and Electronics Engineering (IJCTEE)* (2:4), pp. 46-50.
Kupiainen, E., Mäntylä, M. V., and Itkonen, J. 2015. "Using Metrics in Agile and Lean Software Development – a Systematic Literature Review of Industrial Studies," *Information and Software Technology* (62), pp. 143-163.
Kurnia, R., Ferdiana, R., and Wibirama, S. 2018. "Software Metrics Classification for Agile Scrum Process: A Literature Review," *International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, Indonesia: IEEE, pp. 174-179.
Kütz, M. 2011. *Kennzahlen in der It: Werkzeuge Für Controlling Und Management*. Heidelberg: dpunkt. verlag.
Larman, C., and Vodde, B. 2016. *Large-Scale Scrum: More with Less*. Boston, MA: Addison-Wesley Professional.
Leffingwell, D. 2018. *Safe 4.5 Reference Guide: Scaled Agile Framework for Lean Enterprises*. Addison-Wesley Professional.
Liechti, O., Pasquier, J., and Reis, R. 2017. "Beyond Dashboards: On the Many Facets of Metrics and Feedback in Agile Organizations," *IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, Buenos Aires, Argentina: IEEE, pp. 16-22.
Maiden, N., and Jones, S. 2010. "Agile Requirements Can We Have Our Cake and Eat It Too?," *IEEE Software* (27:3), pp. 87-88.
Mikalsen, M., Stray, V., Moe, N. B., and Backer, I. 2020. "Shifting Conceptualization of Control in Agile Transformations," *Proceedings of the 21st International Conference on Agile Software Development (XP 2020),* M. Paasivaara and P. Kruchten (eds.), Copenhagen, Denmark: Springer, Cham, pp. 173-181.
Miles, M. B., Huberman, A. M., and Saldaña, J. 2020. *Qualitative Data Analysis: A Methods Sourcebook*. SAGE Publications, Inc.
Mills, E. E. 1988. "Software Metrics," Carnegie-Mellon University Pittsburgh - Software Engineering Institute.
Myers, M. D., and Newman, M. 2007. "The Qualitative Interview in Is Research: Examining the Craft," *Information and Organization* (17:1), pp. 2-26.
Neely, A. D., Adams, C., and Kennerley, M. 2002. *The Performance Prism: The Scorecard for Measuring and Managing Business Success*. Prentice Hall Financial Times London.
Nerur, S., Mahapatra, R., and Mangalaraj, G. 2005. "Challenges of Migrating to Agile Methodologies," *Communications of the ACM* (48:5), pp. 72-78.
Nord, R. L., Ozkaya, I., and Kruchten, P. 2014. "Agile in Distress: Architecture to the Rescue," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen (eds.), Rome, Italy: Springer, Cham, pp. 43-57.
Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., and Porres, I. 2016. "Quantitatively Measuring a Large-Scale Agile Transformation," *Journal of Systems and Software* (117), pp. 258-273.
Oza, N., and Korkala, M. 2012. "Lessons Learned in Implementing Agile Software Development Metrics," *Proceedings of the UK Academy for Informaiton Systems Conference (UKAIS)*, Oxford, UK, p. 38.
Paasivaara, M. 2017. "Adopting Safe to Scale Agile in a Globally Distributed Organization," *Proceedings of the 12th International Conference on Global Software Engineering (ICGSE)*, Buenos Aires, Argentina: IEEE, pp. 36-40.
Petersen, K., and Wohlin, C. 2009. "A Comparison of Issues and Advantages in Agile and Incremental Development between State of the Art and an Industrial Case," *Journal of Systems and Software* (82:9), pp. 1479-1490.
Poligadu, A., and Moloo, R. K. 2014. "An Innovative Measurement Programme for Agile Governance," *International Journal of Agile Systems and Management* (7:1), pp. 26-60.
Preiss, K., Goldman, S. L., and Nagel, R. N. 1996. *Cooperate to Compete: Building Agile Business Relationships*. Van Nostrand Reinhold.
Rajlich, V. 2006. "Changing the Paradigm of Software Engineering," *Communications of the ACM* (49:8), pp. 67-70.

Ram, P., Rodriguez, P., and Oivo, M. 2018. "Software Process Measurement and Related Challenges in Agile Software Development: A Multiple Case Study," *Proceedings of the International Conference on Product-Focused Software Process Improvement*, Wolfsburg, Germany: Springer, Cham, pp. 272-287.

Rolland, K., Dingsøyr, T., Fitzgerald, B., and Stol, K.-J. 2016. "Problematizing Agile in the Large: Alternative Assumptions for Large-Scale Agile Development," *Proceedings of the 39th International Conference on Information Systems*, Dublin, Ireland: Association for Information Systems (AIS), pp. 1-21.

Runeson, P., and Höst, M. 2009. "Guidelines for Conducting and Reporting Case Study Research in Software Engineering," *Empirical Software Engineering* (14), pp. 131-164.

Saldaña, J. 2021. *The Coding Manual for Qualitative Researchers*. London: SAGE Publications.

Schwaber, K. 2004. *Agile Project Management with Scrum*. Microsoft press.

Schwaber, K. 2018. "The Nexus Guide." Retrieved 11. November, 2021, from https://www.scrum.org/resources/nexus-guide

Schwaber, K., and Beedle, M. 2002. *Agile Software Development with Scrum*. Prentice Hall Upper Saddle River.

Schwaber, K., and Sutherland, J. 2020. "The 2020 Scrum Guide." Retrieved 11. November, 2021, from https://scrumguides.org/scrum-guide.html

Scrum.org. 2021. "Glossary of Scrum Terms." Retrieved 11. November, 2021, from https://www.scrum.org/resources/scrum-glossary

Seaman, C. B. 1999. "Qualitative Methods in Empirical Studies of Software Engineering," *IEEE Transactions on Software Engineering* (25:4), pp. 557-572.

Shirokova, S., Kislova, E., Rostova, O., Shmeleva, A., and Tolstrup, L. 2020. "Company Efficiency Improvement Using Agile Methodologies for Managing It Projects," *Proceedings of the International Scientific Conference-Digital Transformation on Manufacturing, Infrastructure and Service*, New York, NY, USA, pp. 1-10.

Staron, M., and Medig, W. 2011. "Factors Determining Long-Term Success of a Measurement Program: An Industrial Case Study," *e-Informatica Software Engineering Journal* (5:1).

Stettina, C. J., and Schoemaker, L. 2018. "Reporting in Agile Portfolio Management: Routines, Metrics and Artefacts to Maintain an Effective Oversight," *Proceedings of the 19th International Conference on Agile Software Development (XP 2018),* J. Garbajosa, X. Wang and A. Aguiar (eds.), Porto, Protugal: Springer, Cham, pp. 199-215.

Stolberg, S. 2009. "Enabling Agile Testing through Continuous Integration," *Proceedings of the Agile Conference*, Chicago, IL, USA: IEEE, pp. 369-374.

Storti, A., and Clear, T. 2020. "The Contradiction of Agile Measures: Customer as Focus, but Process as Measured?," *Proceedings of the Australasian Conference on Information Systems*, Wellington, New Zealand: Association for Information Systems (ACIS), pp. 1-12.

Talby, D., and Dubinsky, Y. 2009. "Governance of an Agile Software Project," *Proceedings of the ICSE Workshop on Software Development Governance*, Vancouver, BC, Canada: IEEE, pp. 40-45.

Talby, D., Hazzan, O., Dubinsky, Y., and Keren, A. 2006. "Reflections on Reflection in Agile Software Development," *Proceedings of the AGILE 2006 (AGILE'06)*, Minneapolis, MN, USA: IEEE, pp. 11 pp.-112.

Turetken, O., Stojanov, I., and Trienekens, J. J. 2017. "Assessing the Adoption Level of Scaled Agile Development: A Maturity Model for Scaled Agile Framework," *Journal of Software: Evolution and Process* (29:6), p. e1796.

Uludağ, Ö., Kleehaus, M., Xu, X., and Matthes, F. 2017. "Investigating the Role of Architects in Scaling Agile Frameworks," *Proceedings of the IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)*, Quebec City, QC, Canada: IEEE, pp. 123-132.

Uludağ, Ö., Philipp, P., Putta, A., Paasivaara, M., Lassenius, C., and Matthes, F. 2021. "Revealing the State-of-the-Art in Large-Scale Agile Development: A Systematic Mapping Study," in: *arXiv preprint arXiv:2007.05578*.

Van Oosterhout, M., Waarts, E., and van Hillegersberg, J. 2006. "Change Factors Requiring Agility and Implications for It," *European Journal of Information Systems* (15:2), pp. 132-145.

Challenges and Success Factors for Metrics in Large-Scale Agile Development

Completed Research

Pascal Philipp
Technical University of Munich
pascal.philipp@tum.de

Franziska Tobisch
Technical University of Munich
franziska.tobisch@tum.de

Florian Matthes
Technical University of Munich
matthes@tum.de

Abstract

Contemporary organizations widely use agile software development to react to unpredictable changes in their business environment. Due to the success of agile methods in contexts similar to the agile sweet spot, organizations have been applying them on a larger scale. However, maintaining oversight in large-scale agile development remains a problem. Metrics can tackle this problem by increasing transparency, but organizations have struggled with their adoption. Furthermore, extant research on large-scale agile development lacks publications investigating metric challenges and success factors. Against this backdrop, we conducted an expert interview study with 23 experts from 13 organizations. The most mentioned challenges are data collection challenges, lack of metric usefulness, and metric calculation challenges. On the other hand, the most occurring success factors are context-dependent metric adoption, implementing an agile metric management process, and ensuring understanding of the metric purpose.

Keywords

Agile software development, large-scale agile development, metrics, challenges, success factors.

Introduction

Highly dynamic business environments require organizations to agilely deal with unpredictable internal and external changes (Van Oosterhout et al. 2006). Organizations enable business agility by implementing an agile IT and process architecture (Van Oosterhout et al. 2006). Consequently, developing software agile is crucial. Nowadays, agile software development prevails itself as the dominant software development approach (Dybå and Dingsøyr 2008). Well-known agile methods, such as Scrum and eXtreme programming (XP) are designed for co-located, self-organizing, and small teams that develop software in close collaboration with customers, using regular feedback and rapid development iterations (Kettunen 2007). Agile methods succeed in contexts identical or very similar to the so-called "agile sweet spot" (Kruchten 2013). This sweet spot consists of ideal conditions comprising co-located teams of 5–12 members, frequent deliveries, stable underlying architecture, and medium to low system criticality (Nord et al. 2014). Due to the success of agile methods in the sweet spot, organizations have applied agile software development methods on a larger scale (Digital.ai 2021; Dikert et al. 2016; Dingsøyr et al. 2018; Uludağ et al. 2021a) characterized by single or multiple multi-team settings consisting of at least two agile teams (Dikert et al. 2016; Dingsøyr et al. 2018; Uludağ et al. 2021a). Agile at scale must deal with challenges, such as inter-team coordination (Brown et al. 2013; Dikert et al. 2016; Uludağ et al. 2018) and communication (Dikert et al. 2016; Talby and Dubinsky 2009; Uludağ et al. 2018). Since agile methods, in their original design, cannot deal with these challenges, they must be extended (Uludağ et al. 2021a). Consequently, multiple scaling agile frameworks have been proposed and continuously improved (Uludağ et al. 2021b). The Scaled Agile Framework (SAFe) and Scrum@Scale are the most widely adopted frameworks in the industry (Digital.ai 2021). Adopting metrics beyond the team level has become a relevant research topic in large-scale agile development (Dingsøyr and Moe 2014). Several scaling agile frameworks (e.g., SAFe

(Scaled Agile Inc. 2021) and Scrum@Scale (Sutherland and Scrum Inc. 2021)) are aware of the relevance of metrics and therefore propose metrics and best practices. For example, Scrum@Scale recommends metrics to support data-driven decision-making, streamline the work of teams, stakeholders, or leadership, and reduce decision latency (Sutherland and Scrum Inc. 2021). Despite the relevance of metrics, their implementation remains challenging. Several studies regarding challenges and best practices for implementing metrics in large-scale agile development have been conducted (Korpivaara et al. 2021). However, since the scope of this research included only two cases belonging to the same organization, additional research is required to validate the findings and test their applicability in other organizations and industrial domains (Korpivaara et al. 2021). Therefore, we fill this gap by raising the following research questions:

RQ1: What challenges have been reported for metrics in large-scale agile development?

RQ2: What success factors have been reported for metrics in large-scale agile development?

Background and Related Work

In 2001, the agile manifesto was proposed to improve software development at that time (Beck et al. 2001). In the manifesto, particular importance was assigned to individuals and interactions, working software, customer collaboration, and responding to change. Agile methods, such as Scrum, XP, feature-driven development, and crystal methodologies adhere to the manifesto to varying degrees (Dingsøyr et al. 2012). These methods were created for and succeeded in contexts close to the agile sweet spot (Kruchten 2013). According to Kruchten (2013), more than 70% of software projects operate in contexts comparable to the agile sweet spot. However, agile methods fail in contexts different from what they were created for if they are not adapted (Kruchten 2013). Thus, scaling agile frameworks extend the agile methods to deal with challenges arising in larger contexts (Uludağ et al. 2021a). Typical challenges in larger settings include additional coordination and inter-team dependencies (Dikert et al. 2016). Currently, there are more than 20 frameworks (Uludağ et al. 2021b). The most widely adopted frameworks are SAFe, Scrum@Scale, and Enterprise Scrum (Digital.ai 2021).

In traditional software development, metrics were used for decades (Fenton and Neil 2000) to measure or predict attributes (i.e., measurable characteristics) of processes, products, or resources (Fenton and Neil 2000). Several researchers have stressed that successfully implemented metrics in traditional software development cannot always be adopted in agile development (Hartmann and Dymond 2006; Oza and Korkala 2012; Poligadu and Moloo 2014). Whereas metrics applied in traditional software development measure the achievement of a predefined plan, agile metrics must align with the value of embracing change (Korpivaara et al. 2021; Kupiainen et al. 2015). Metrics implementation is challenging in agile (Ram et al. 2018) and large-scale agile development (Brown et al. 2013; Korpivaara et al. 2021; Stettina and Schoemaker 2018). Several researchers have revealed challenges and success factors for metrics in scaling agile environments. According to Ram et al. (2018), challenges faced in operationalizing metrics include difficulty in deriving actionable inputs, existing processes inhibiting change, and the lack of data or appropriate tools for data production. Similarly, Korpivaara et al. (2021) identified inadequate data availability as a challenge and highlighted challenges including the disconnection between the unit Key Performance Indicators (KPIs) and the teams' work, visibility between team metrics and unit KPIs, and strong external factors influencing the metrics. Additionally, they highlighted inadequate data consistency and a lack of skills and guidance for setting metrics, data processing, and lack of standardization and organizational issues as challenges related to performance metrics (Korpivaara et al. 2021). In their study Oza and Korkala (2012) present the key challenge being that metrics can become interests of individuals rather than contributing to organizational improvement. According to Korpivaara et al. (2021), best practices for performance measurement are regular communication and discussion on unit KPIs and agile metrics. Also, Ram et al. (2019) revealed several success factors for adopting metrics in agile environments, including applying an iterative and agile approach to the metrics program or using familiar metrics to establish trust. More generally, Dikert et al. (2016) identified challenges and success factors for agile transformation such as lack of investment or change resistance which can also apply to the metric adoption.

Study Design

(1) Study Design and Overview: To answer our research questions, we conducted semi-structured interviews and analyzed the collected data. The chosen research design is appropriate since semi-structured interviews can be used to investigate practical problems, such as the adoption of metrics in large-scale agile software development (Seaman 1999). To ensure rigor, we designed our research methodology using the guidelines of Myers and Newman (2007) for semi-structured interviews. Our study, although exploratory, includes descriptive elements. Table 1 provides an overview of the study participants. We conducted 23 interviews with experts from 13 organizations. We recruited the study subjects via an email sent to practitioners and researchers working in the domain of agile software development. Note that three interviews at two organizations were excluded since we only included interviewees that had sufficient topic-related knowledge. The included interviewees are characterized by a broad spectrum of industry backgrounds and job roles, ensuring a large "variety of voices" covering multiple viewpoints (Myers and Newman 2007). All except for one expert organization (TeleCo) have their headquarters in Germany. To the best of our knowledge, the experts had no formal training on scaled agile frameworks.

No.	Company	Role(s), Large-scale agile software development experience (years)	Scaling Agile Framework	Program Size	No.	Company	Role(s), Large-scale agile software development experience (years)	Scaling Agile Framework	Program Size
E1	CarCo1	Agile Coach, 11–15	SAFe	750	**E11**	ConsultCo	Manager, 3–5	LeSS. Scrum-of-Scrums	50
E2	BankCo1	Manager, 3–5	SAFe	6000	**E12**	CarCo2	Scrum Master and Agile Coach, 3–5	Scrum-of-Scrums	100
E3	HealthCo	Manager, 6–10	SAFe, LeSS	450	**E13**	RetailCo1	Agile Coach, 6–10	SAFe	120
E4	ConsultCo	Scrum Master and Agile Coach, 1–2	LeSS, Scrum-of-Scrums	50	**E14**	InsureCo1	Scrum Master and Agile Coach, 3–5	SAFe	75
E5	HealthCo	Manager and Agile Coach, 6–10	SAFe, LeSS	450	**E15**	InsureCo1	Release Train Engineer, 1–2	SAFe	80
E6	ConsultCo	Developer and Scrum Master, 3–5	LeSS, Scrum-of-Scrums	50	**E16**	InsureCo2	Scrum Master and Agile Coach, 1–2	Scrum@Scale	24
E7	ConsultCo	Product Owner and Business Analyst, 1–2	LeSS, Scrum-of-Scrums	50	**E17**	InsureCo1	Manager and Agile Coach, 3–5	Nexus, Scrum-of-Scrums	90
E8	ConsultCo	Software Architect, Solution Architect, 3–5	LeSS, Scrum-of-Scrums	50	**E18**	BankCo1	Agile Coach, 3–5	SAFe	400
E9	TeleCo	Scrum Master, Agile Coach and Developer, 11–15	Scrum-of-Scrums	100	**E19**	RetailCo2	Scrum Master and Agile Coach, 6–10	Scrum-of-Scrums	70
E10	HealthCare Co	Software Architect, 11-15	Scrum-of-Scrums	450	**E20**	TransportationCo	Release Train Engineer, 6-10	SAFe	550

Table 1. Interview Partners

(2) Data Collection: We conducted the expert interviews using videotelephony between January and March 2021. At least two researchers participated in each interview to ensure observer triangulation (Runeson and Höst 2009). All interviews were recorded and had an average duration of 82 minutes. The interviews summed to 1.640 minutes and differed in length due to varying extents of metric usage within the expert organizations. Furthermore, the interview durations exclude the introduction and clarification times at the start of the interview. The research goal was recalled during the introduction, and the clarifications addressed the interview structure. The interview protocol was not reviewed by an ethics committee or human subjects review board. Each interview had the same outline and the interview questions remained unchanged. Though, due to the nature of semi-structured interviews, slight variations regarding the question order or wording occurred. We separated the interview questions into two sections. In the first

section, we asked experts questions on their professional background and large-scale agile development environment. Some interviewees answered questions of the first section upfront and sent their written responses to us. The second section included questions regarding metric adoption. Moreover, we included files shared by the interviewees to facilitate the triangulation of data sources. The results communicated here are a subset of the overall collected data. We plan to publish the remaining empirical evidence in further publications.

(3) Data Analysis: Two researchers took the transcripts, and a third was the quality gate reviewing the completeness and consistency of each transcript. We dismissed sensitive information and ensured anonymity by assigning unique acronyms to each expert. Furthermore, we used deductive and inductive coding (Miles et al. 2020) to code the transcripts, prefilled questionnaire parts, and additional files. We aligned our coding with the two-cyclic approach proposed by Saldaña (2021). Thus, we assigned descriptive first-cycle codes (Miles et al. 2020) to the text segments assigned to the higher-order second-cycle codes. We based the second-cycle deductive coding on an initial list of higher-order codes (Saldaña 2021) and aligned the list with the interview questionnaires. Additional second-cycle codes were inductively created whenever new findings that could not be assigned to an existing code category occurred (Saldaña 2021). Introducing new codes necessitated a re-codification of the previously coded data. Accordingly, we created the final coding incrementally. As tools for coding and analyzing the transcripts we used MAXQDA[1] and Excel[2]. Furthermore, we discussed and resolved all conflicts by mutual consent to enhance result validity. Also, the experts were contacted to resolve ambiguities and clarify interview statements.

Results

Challenges of Metrics in Large-Scale Agile Development

This section answers the first research question: *What challenges have been reported for metrics in large-scale agile development?* We identified eleven challenges, as summarized in Table 2.

Challenge (Occurrences)	Experts	# Experts (Occurrences)
Data collection challenges (21,13%)	E1–E3, E5–E15, E17–E19	17 (85%)
Lack of metric usefulness (15,21%)	E1–E12, E14, E15, E17, E18, E20	17 (85%)
Metric calculation challenges (13,24%)	E1–E5, E7, E8, E10, E11, E13, E15, E17	12 (60%)
Metric-based decision-making challenges (10,99%)	E1–E11, E12, E14, E16–E20	18 (90%)
Negative effects for employees (10,70%)	E1, E2, E4, E6–E9, E11–E17, E19, E20	16 (80%)
Lack of metric understanding (7,04%)	E1–E3, E9, E10, E12–E15, E19	10 (50%)
Change resistance (6,48%)	E1, E2, E7, E10, E12, E15–E17, E19, E20	10 (50%)
Focus on local instead of global optimum (5,07%)	E1, E2, E7–E10, E13–E15, E17	10 (50%)
Metric definition challenges (5,07%)	E1–E3, E5–E7, E9, E11, E14, E16	10 (50%)
Manipulation of metric values (3,10%)	E6–E8, E11, E15, E17, E19	7 (35%)
Imbalance between control and team autonomy (1,97%)	E9, E12, E19, E20	4 (20%)

Table 2. Challenges of Metrics in Large-Scale Agile Development

(1) Data collection challenges: Data collection for the metric is notably a common problem. Several experts mentioned the dependence on data quality (E1–E3, E5, E7, E9, E10, E12–E15, E17–E19), data availability (E1–E3, E6, E7, E10–E14, E18), data collection complexity (E1, E2, E8, E10), and missing anonymity of data collection (E6, E17) as related challenges. In particular, the dependence on data quality is experienced as the main challenge since the metric can lose its value, for example, due to irrelevant data (E1) or biased human data sources (E2). Furthermore, employees missing discipline for entering data was mentioned as a challenge in data availability (E13, E18).

(2) Lack of metric usefulness: Ensuring metric usefulness is experienced as a problem by several experts. Deriving value from the metrics (E1–E7, E10–E12, E14, E17, E18, E20) and error-prone metric-based

[1] https://www.maxqda.com
[2] https://www.microsoft.com/de-de/microsoft-365/excel

planning (E7–E11, E15, E20) were mentioned as related challenges. For deriving value from the metrics, the experts experience a lack of insights on reasons for changes or deviations (E1, E2, E5, E10, E14, E16) and that metrics were only a starting point for problem identification (E10, E14, E22). Concerning the challenge of error-prone metric-based planning, an expert stated that it is difficult to derive stable predictions based on the metric (E8).

(3) Metric calculation challenges: Metric calculation is another common problem. Several experts mentioned the incorporation of all relevant variables (E2, E4, E7, E8, E10, E11, E13, E15), provision of calculation resources (E3, E5), aggregation of team-level values into program-level values (E5, E8, E10, E17), complexity of metric calculation (E1, E3, E11, E17), and lack of adequate tool support (E5) as challenges related to this problem. Notably, incorporating all relevant variables was echoed as a challenge because not considering variables such as fluctuations in velocity planning (E4, E8, E10, E11), holiday times (E7, E8, E11, E13), and size changes of epics (E2, E4, E7, E8, E11) lead to reduced a metric value. Moreover, the high time effort for writing metric tests (E5) and measuring the metric (E3) is perceived as challenging.

(4) Metric-based decision-making challenges: Decision-making based on metric values was perceived as a challenge. For example, multiple experts mentioned the difficulty of deciding on next actions (E1, E3–E5, E8, E9, E12, E14, E17), ineffective management based on metric values (E2, E4, E6, E7, E9, E10, E16, E18–E20), and misleading information derived from metrics (E1, E11) as related challenges. The organization ConsultCo struggled, for instance, to identify actions and responsible persons to resolve blockers (E8). Additionally, ineffective management due to incorrect metric values occurs, for example, when managers conduct unfeasible comparisons between teams (E2, E4, E6, E7, E10, E20) or programs (E19). Expert E10 stressed the importance of avoiding inter-team comparisons based on the velocity: *"[…] reflect that the team velocity is different for every team. So that's one of the biggest things that you can do wrong when it comes to agile […] to say that every team has the same velocity. But if you reflect that already in your planning you can simply sum up all story points and over all teams, the burn-down will go in the right direction in theory, practically of course always something goes wrong."*

(5) Negative effects for employees: Negative effects for employees are another common problem and include challenges, such as negative psychological effects (E1, E2, E4, E7–E9, E11, E13–E17, E19, E20), internal competition (E6, E8, E9, E11), and feeling of being controlled (E1, E12, E15, E16) as described by expert E1: *"[…] you start measuring people, so, there is […] disturbance about the big transparency […] there is always the fear, that the measurement will leads us to control the people"*. An example related to negative psychological effects is increased pressure and stress due to deviations from planned metric values (E2, E13). Also, sometimes metric-based criticism is perceived as a personal attack (E19). Furthermore, such metric-based comparisons lead to undesirable internal competition, thereby decreasing collaboration between teams (E9).

(6) Lack of metric understanding: A lacking metric understanding is another shared problem among the experts. A lack of metric understanding is caused by missing awareness of dependencies (E1–E3, E9, E15), a general lack of metric understanding (E1, E2, E12, E14, E19), lack of understanding the calculation (E10, E12), providing insufficient resources to understand metrics (E12, E13), and complexity (E10, E12). Sometimes missing knowledge on metrics' existence (E12) is another related challenge. Often a lacking understanding leads to misuse of the metric (E10). In particular, the managers' lack of understanding (E2, E19) is experienced as a problem.

(7) Change resistance: Change resistance in the organization is also a shared problem among the experts. Challenges related to change resistance are difficulty in convincing stakeholders (E1, E2, E15, E17, E19), lack of management commitment (E2, E22), establishing a common metric understanding within the organization (E12, E16, E17, E20), investment in change (E1, E7), and lack of common reporting (E10). Convincing stakeholders can be a problem because, for instance, achieving openness (E15, E17) and acceptance (E15), convincing managers with a traditional mindset to use agile metrics (E19), and justifying metrics before the work council (E1, E2) is difficult.

(8) Focus on local instead of global optimum: Working toward goals that do not contribute to an organization-wide optimum is another identified challenge (E1, E2, E7–E10, E13–E15, E17). For example, several experts mentioned that metric target values often did not align with essential goals like customer satisfaction (E14, E15, E17), product quality (E8, E10, E13), or cost-efficiency (E2).

(9) Metric definition challenges: Several experts mentioned challenges during the metric definition. Especially the definition of appropriate target values (E1, E2, E5, E9, E16), complexity of metric definition (E1, E6, E11, E14), definition of appropriate tolerance values (E3, E5), defining appropriate granularity of calculation parameters (E7), and defining appropriate calculation rules (E1, E11) were experienced as challenges. Problems related to defining appropriate target values include setting unrealistic target values (E5) and high variations between programs when setting target values on the portfolio level (E1). Additionally, the metric definition can be complex due to difficulty in harmonizing measurement units across teams (E11).

(10) Manipulation of metric values: Employees' manipulation of metric values is another challenge mentioned by several experts. For example, conservative estimating is a circumvention tactic to reach the metric target value (E15). Driving factors for metric value manipulation are competition between teams (E6, E8, E11, E17) and employees' feeling of being controlled (E15).

(11) Imbalance between control and team autonomy: Finding the right balance between control and ensuring team autonomy is another common challenge. Particularly balancing governance and team autonomy (E12, E19) and avoiding too high management involvement (E9, E20) are related challenges. Balancing governance and team autonomy is challenging because self-organization and control may be experienced as contradictory (E19). Regarding the involvement of management, the intervention of the program management keeps developers from working (E20) and, also, metric-based micro-management hinders people from working (E20).

Success Factors for Metrics in Large-Scale Agile Development

This section answers the second research question: *What success factors have been reported for metrics in large-scale agile development?* We identified 22 success factors. Subsequently, we explain eleven in detail and summarize the remaining in the last paragraph. The success factors are summarized in Table 3.

Success Factor (Occurrences)	Experts	# Experts (Occurrences)
Context-dependent metric adoption (16,15%)	E1–E3, E5–E20	19 (95%)
Implementing an agile metric management process (15,32%)	E1, E3, E5–E7, E9, E10, E12–E17, E20	14 (70%)
Ensuring understanding of metric purpose (8,70%)	E1, E2, E4, E6, E8–E13, E16, E17, E19, E20	14 (70%)
Managing interplay of metric and its environment (8,28%)	E1, E4, E6–E8, E10–E12, E20	9 (45%)
Keep metric adoption simple (7,66%)	E3, E5–E8, E10–E12, E14, E15, E17, E18, E20	13 (65%)
Measures for ensuring effective data collection (7,25%)	E1–E3, E5, E6, E8, E10–E12, E14, E17, E19, E20	12 (60%)
Enabling metric adoption by providing sufficient resources (6,83%)	E1–E5, E7–E11, E14, E17, E20	13 (65%)
Empowerment of teams in metric adoption (5,59%)	E4, E7–E12, E15, E17, E20	10 (50%)
Standardization of metric adoption (4,97%)	E5, E7–E12, E14	8 (40%)
Combining metric with qualitative feedback (3,11%)	E1, E9, E14, E17, E20	5 (25%)
Ensuring goal-orientation of metric (3,11%)	E1, E9, E10, E11, E12, E18, E19, E20	8 (40%)
Other (13,04%)	E1–E7, E9–E17, E19, E20	18 (90%)

Table 3. Success Factors for Metrics in Large-Scale Agile Software Development

(1) Context-dependent metric adoption: A successful metric adoption must be context-specific. According to the experts, the context plays a crucial role during metric configuration (E1–E3, E5–E8, E10–E12, E14–E17, E19, E20), deciding next actions (E1, E3, E5, E11, E14, E15, E19, E20) based on metric values, metric value interpretation (E1, E2, E5, E7, E9, E13), metric measurement (E16–E18), and metric selection (E7). Furthermore, organizations must consider the requirements of the large-scale agile development context by adhering to agile values and principles (E1, E13, E17).

(2) Implementing an agile metric management process: Several experts identified adhering to an agile metric management process as a crucial success factor. According to the experts, continuous improvement (E1, E3, E5, E7, E10, E12, E14, E16, E20), collaborative inspection (E1, E6, E7, E9, E15, E16, E20), collaborative decision-making of stakeholders regarding metric lifecycle (E1, E6, E10, E15, E16), ensuring metric transparency (e.g., by using visualizations) (E7, E9, E10, E13, E16), and using lightweight tools (E12,

E17, E20) are crucial. Collaborative inspection can be facilitated by including metric discussions in regular events (E9, E15, E19).

(3) Ensuring understanding of metric purpose: Another success factor is ensuring an understanding of the purpose of the metric, thereby reducing change resistance (E6, E9, E12, E13, E16, E20) and avoiding comparisons between teams (E4, E6, E8, E10, E11, E17, E19, E20). Also, the experts highlighted the importance of increasing management understanding (E1, E2, E16, E19) and organization-wide understanding (E1, E10, E12). Furthermore, creating an understanding of the metric target values is considered important (E2).

(4) Managing interplay of metric and its environment: Each metric is embedded in an environment, and managing its dependencies within this environment is critical for its successful adoption. According to the experts, it is important to manage the relationships between metrics (E1, E4, E6–E8, E10, E12, E20) and their dependencies to the environment in general (E1, E4, E6, E8, E10, E11, E20). Also, combining related metrics (E12, E20) and raising the awareness of the difference between causality and correlation (E10) was considered important.

(5) Keep metric adoption simple: Another success factor considered by the experts is keeping the metric adoption simple. Experts highlighted the importance of achieving simplicity of metric measurement (E7, E10–E12, E14, E17, E18) and simplicity of metrics in general (E7, E8, E11, E12, E18, E20). They highlighted automation (E3, E5, E10, E20) and choosing metrics with low overhead (E12) as measures to simplify metric adoption. A simple metric adoption is crucial and can be achieved by adopting only a few metrics (E6, E10, E15). Another important factor is ensuring the simplicity of metric scaling (E10).

(6) Measures for ensuring effective data collection: Multiple experts suggested implementing measures for effective data collection. Therefore, establishing a data collection strategy (E1, E2, E6, E14, E17), ensuring data quality (E1, E8, E11, E19, E20), and an effective test environment (E3, E5, E10) are crucial factors to consider. Moreover, safeguarding the anonymity of data collection (E6, E14, E17) and assigning data collection responsibilities (E12) were considered important. Furthermore, clarifying the effort for data collection (E1) is a success factor.

(7) Enabling metric adoption by providing sufficient resources: Providing sufficient resources like appropriate tool support (E2, E4, E5, E7, E10, E11, E14), the provision of a metric support team (E3, E20), ensuring the availability of experts (E1, E7, E8, E11), granting a proper time (E5, E8, E9, E11, E17) and money budget (E3, E5) were considered important.

(8) Empowerment of teams in metric adoption: Empowering the team during metric adoption is another success factor. Thus, assigning extensive team responsibility (E4, E7, E9, E20) is critical. Further, granting the team autonomy regarding selecting and unselecting (E9, E10, E15), the metric calculation (E8, E11, E12), and configuration (E7, E17) is important. Finally, taking a team-centric view (E10) during the metric adoption is another key factor highlighted by experts.

(9) Standardization of metric adoption: Multiple experts recommend using best practices (E5, E7–E9, E11, E12, E14) for the metric adoption. Further, it is considered crucial to achieving standardization regarding calculation (E5, E11), configuration (E10), and reporting (E5, E10). Tools serve to standardize the metric adoption (E5).

(10) Ensuring goal-orientation of metric: Using metrics to support goals (E1, E9, E11, E12, E18, E20) can be achieved if the target values of the metrics are goals or if the metric values create transparency regarding goal-fulfillment. Experts mentioned setting realistic target values as a key factor (E10, E19, E20). Also, implementing Objectives and Key Results (E9) and measuring them using metrics can help organizations. Furthermore, metric target values should focus on the outcome instead of output (E18).

(11) Combining metric with qualitative feedback: Another success factor lies in combining the quantitative information of metrics with further qualitative feedback to enhance inspection (E1, E9, E14, E17, E20) as described by a Scrum Master (E20): *"The number can of course only be an indicator at this point. It always depends on what I do with it. So ultimately, as a Scrum Master […] I must then also look at a qualitative analysis, what were the problems […] in the sprint and must look at how we can make it better through measures that I generate in the retrospective."* Including qualitative feedback can either help to understand the quantitative values (E20) better or get a complete view of the situation. Surveys can serve to collect qualitative feedback (E14, E17, E20).

(12) Other: Further success factors include providing training, guidelines, and documentation (E1, E3, E9, E12, E14, E20), understanding what the metric is measuring (E2, E6, E7, E10, E12, E16, E19, E20), using metrics on higher organizational levels (E7, E9–E11, E13, E15, E20), communicating metrics' value to reduce change resistance (E1, E2, E15, E17, E19), only approved stakeholders can access metric information (E2, E6, E9, E10, E17, E19), assigning clear responsibilities (E5, E15), collaborating teams to achieve shared metric target values (E11), establishing a community of practice for metrics (E4), facilitating alignment of management and teams regarding metrics (E20), acceptance of tools by employees (E15), ensuring realistic expectation management based on metrics (E9).

Discussion

To answer RQ1: "What challenges have been reported for metrics in large-scale agile development?" we identified eleven challenges summarized in Table 2. The challenges with the highest number of total occurrences are data collection challenges (21,13%), lack of metric usefulness (15,21%), and metric calculation challenges (13,24%). Similarly, Korpivaara et al. (2021) identified the time and effort required to collect data, inadequate data quality, and data availability as data collection challenges. Also, Ram et al. (2018) identify a lack of tools as a cause of data collection problems. Regarding metric usefulness, Oza and Korkala (2012) and Korpivaara et al. (2021) revealed that strong external factors influence metrics to a degree where their outcome is no longer controllable. Concerning metric calculation, Korpivaara et al. (2021) identify the lack of skills and guidance as problems. Our findings show that adopting metrics in scaling agile environments is challenging, as often tools easing the data collection and calculation are missing or are used for other purposes. Since many of the expert organizations are in the early phases of the agile transformation, the challenges can also be attributed to a low metric maturity level which will be improved at later stages in the transformation. We identified 22 success factors summarized in Table 3 to answer RQ2: "What success factors have been reported for metrics in large-scale agile development?". The success factors with the highest number of total occurrences are context-dependent metric adoption (16,15%), implementing an agile metric management process (15,32%), and ensuring understanding of metric purpose (8,70%). These success factors confirm the findings of extant research. Similar to the context-dependent metric adoption, Ram et al. (2019) highlight the need to implement flexible development processes to accommodate new practices, enabling the adoption of metrics. Moreover, Korpivaara et al. (2021) and Ram et al. (2019) highlight the importance of applying an iterative and agile way of working to the metrics program, including discussions and short feedback cycles. Finally, Oza and Korkala (2012) and Ram et al. (2019) show the importance of the metric purpose by highlighting that metrics serve as means for trend analysis and propose to establish a long-term vision to drive the technical benefits of metrics. Our success factors indicate that the metric adoption will not succeed if metrics are implemented without further customization. Contrarily, a careful and continuous ensuring of transparency, inspection, and adaption of the metric and its context and safeguarding an understanding of why metrics are implemented is crucial to their success in large-scale agile development.

The results of this study contribute to research on large-scale agile software development. First, in extension to extant literature (cf. Dikert et al. (2016); Korpivaara et al. (2021); Oza and Korkala (2012); Ram et al. (2018); Ram et al. (2019)) and to the best of our knowledge, we present novel challenges and success factors for metrics in scaling agile environments. We identified the three challenges negative effects for employees, manipulation of metric values, and imbalance between control and team autonomy. Practitioners can benefit from these findings by counteracting these challenges. For example, empowering the team during metric adoption is a suitable countermeasure since it demands managers to present metrics as optional means only implemented if compatible with the team's interests. Concurrently, it motivates managers to act as servant leaders by supporting teams during the metric adoption while partly relinquishing decision-power. We identified the three success factors empowerment of teams in metric adoption, combining metrics with qualitative feedback, and ensuring goal-orientation of metrics. Practitioners can benefit from these findings since each success factor offers a concrete starting point to facilitate the metric adoption. We recommend practitioners assess the necessity of introducing new or improving existing success factors based on their context-specific requirements. Second, our research makes a confirmatory contribution since the remaining challenges and success factors are consistent with extant research (cf. Dikert et al. (2016); Korpivaara et al. (2021); Oza and Korkala (2012); Ram et al. (2018); Ram et al. (2019)).

We used the assessment scheme of Runeson and Höst (2009) to evaluate possible validity threats. Since our study is exploratory with descriptive elements and does explain casual relationships, we did not consider internal validity a threat. To counteract the threats to construct validity, we included several sources, i.e., semi-structured interviews and internal documents, during data collection. Whenever the interpretation of the evidence required clarification, we contacted the interviewees again to resolve ambiguities. We countermeasure the threat of external validity by clearly outlining how our results relate to the experts and organizations of our study. We dealt with threats to reliability by implementing three countermeasures. First, each interview had at least two researchers. Second, each transcript was reviewed by a second researcher. Third, whenever the interpretation of the evidence required clarification, we resolved the ambiguities by contacting the experts again.

Conclusion and Future Work

In this paper, we presented an expert interview study exploring challenges and success factors for metrics in large-scale agile development. We conducted an interview study with 23 experts from 13 organizations. The challenges with the highest occurrences are data collection challenges, lack of metric usefulness, and metric calculation challenges. The success factors with the highest occurrences are context-dependent metric adoption, implementing an agile metric management process, and ensuring understanding of the metric purpose. As a future research topic, we suggest using the identified challenges and success factors to inform the design of artifacts that support metric adoption in large-scale agile development. Further, researchers could use the success factors to develop a maturity grid to assess the metric adoption in scaling agile environments.

REFERENCES

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., and Jeffries, R. 2001. "Manifesto for Agile Software Development."

Brown, A.W., Ambler, S., and Royce, W. 2013. "Agility at Scale: Economic Governance, Measured Improvement, and Disciplined Delivery," *Proceedings of the 35th International Conference on Software Engineering (ICSE)*, San Francisco, CA, USA: IEEE, pp. 873-881.

Digital.ai. 2021. "15th State of Agile Survey." Retrieved 16. February, 2022, from https://info.digital.ai/rs/981-LQX-968/images/RE-SA-15th-Annual-State-Of-Agile-Report.pdf_ga=2.241760322.1357558838.1645022323-2137180703.1645022323

Dikert, K., Paasivaara, M., and Lassenius, C. 2016. "Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review," *Journal of Systems and Software* (119), pp. 87-108.

Dingsøyr, T., and Moe, N.B. 2014. "Towards Principles of Large-Scale Agile Development," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen (eds.), Rome, Italy: Springer, Cham, pp. 1-8.

Dingsøyr, T., Moe, N.B., Fægri, T.E., and Seim, E.A. 2018. "Exploring Software Development at the Very Large-Scale: A Revelatory Case Study and Research Agenda for Agile Method Adaptation," *Empirical Software Engineering* (23), pp. 490-520.

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N.B. 2012. "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," *Journal of Systems and Software* (85:6), pp. 1213-1221.

Dybå, T., and Dingsøyr, T. 2008. "Empirical Studies of Agile Software Development: A Systematic Review," *Information and Software Technology* (50:9-10), pp. 833-859.

Fenton, N.E., and Neil, M. 2000. "Software Metrics: Roadmap," *Proceedings of the Conference on the Future of Software Engineering*, Limerick, Ireland: ACM, pp. 357-370.

Hartmann, D., and Dymond, R. 2006. "Appropriate Agile Measurement: Using Metrics and Diagnostics to Deliver Business Value," *Proceedings of the AGILE 2006 (AGILE'06)*, Minneapolis, MN, USA: IEEE, pp. pp. 6 pp.-134.

Kettunen, P. 2007. "Extending Software Project Agility with New Product Development Enterprise Agility," *Software Process: Improvement and Practice* (12:6), pp. 541-548.

Korpivaara, I., Tuunanen, T., and Seppänen, V. 2021. "Performance Measurement in Scaled Agile Organizations," *Proceedings of the 54th Annual Hawaii International Conference on System Sciences (HICSS)*, Kauaii, Hawaii, USA: University of Hawai'i at Manoa, pp. 6912-6921.

Kruchten, P. 2013. "Contextualizing Agile Software Development," *Journal of Software: Evolution and Process* (25:4), pp. 351-361.

Kupiainen, E., Mäntylä, M.V., and Itkonen, J. 2015. "Using Metrics in Agile and Lean Software Development – a Systematic Literature Review of Industrial Studies," *Information and Software Technology* (62), pp. 143-163.

Miles, M.B., Huberman, A.M., and Saldaña, J. 2020. *Qualitative Data Analysis: A Methods Sourcebook*. SAGE Publications, Inc.

Myers, M.D., and Newman, M. 2007. "The Qualitative Interview in Is Research: Examining the Craft," *Information and Organization* (17:1), pp. 2-26.

Nord, R.L., Ozkaya, I., and Kruchten, P. 2014. "Agile in Distress: Architecture to the Rescue," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen (eds.), Rome, Italy: Springer, Cham, pp. 43-57.

Oza, N., and Korkala, M. 2012. "Lessons Learned in Implementing Agile Software Development Metrics," *Proceedings of the UK Academy for Informaiton Systems Conference (UKAIS)*, Oxford, UK, p. 38.

Poligadu, A., and Moloo, R.K. 2014. "An Innovative Measurement Programme for Agile Governance," *International Journal of Agile Systems and Management* (7:1), pp. 26-60.

Ram, P., Rodriguez, P., and Oivo, M. 2018. "Software Process Measurement and Related Challenges in Agile Software Development: A Multiple Case Study," *Proceedings of the International Conference on Product-Focused Software Process Improvement*, Wolfsburg, Germany: Springer, Cham, pp. 272-287.

Ram, P., Rodriguez, P., Oivo, M., and Martínez-Fernández, S. 2019. "Success Factors for Effective Process Metrics Operationalization in Agile Software Development: A Multiple Case Study," *Proceedings of the International Conference on Software and System Processes (ICSSP)*, Montreal, QC, Canada: IEEE, pp. 14-23.

Runeson, P., and Höst, M. 2009. "Guidelines for Conducting and Reporting Case Study Research in Software Engineering," *Empirical Software Engineering* (14), pp. 131-164.

Saldaña, J. 2021. *The Coding Manual for Qualitative Researchers*. London: SAGE Publications.

Scaled Agile Inc. 2021. "Metrics." Retrieved 29. January, 2022, from https://www.scaledagileframework.com/metrics/

Seaman, C.B. 1999. "Qualitative Methods in Empirical Studies of Software Engineering," *IEEE Transactions on Software Engineering* (25:4), pp. 557-572.

Stettina, C.J., and Schoemaker, L. 2018. "Reporting in Agile Portfolio Management: Routines, Metrics and Artefacts to Maintain an Effective Oversight," *Proceedings of the 19th International Conference on Agile Software Development (XP 2018),* J. Garbajosa, X. Wang and A. Aguiar (eds.), Porto, Protugal: Springer, Cham, pp. 199-215.

Sutherland, J., and Scrum Inc. 2021. "The Scrum at Scale Guide - the Definitive Guide to the Scrum@Scale Framework." Retrieved 29. January 2022, from https://www.scrumatscale.com/scrum-at-scale-guide-online/#Metrics-and-transparency

Talby, D., and Dubinsky, Y. 2009. "Governance of an Agile Software Project," *Proceedings of the ICSE Workshop on Software Development Governance*, Vancouver, BC, Canada: IEEE, pp. 40-45.

Uludağ, Ö., Kleehaus, M., Caprano, C., and Matthes, F. 2018. "Identifying and Structuring Challenges in Large-Scale Agile Development Based on a Structured Literature Review," *Proceedings of the 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, Stockholm, Sweden: IEEE, pp. 191-197.

Uludağ, Ö., Philipp, P., Putta, A., Paasivaara, M., Lassenius, C., and Matthes, F. 2021a. "Revealing the State-of-the-Art in Large-Scale Agile Development: A Systematic Mapping Study," *arXiv preprint arXiv:2007.05578*).

Uludağ, Ö., Putta, A., Paasivaara, M., and Matthes, F. 2021b. "Evolution of the Agile Scaling Frameworks," *Proceedings of the International Conference on Agile Software Development,* P. Gregory, C. Lassenius, X. Wang and P. Kruchten (eds.), Virtual Event: Springer, Cham, pp. 123-139.

Van Oosterhout, M., Waarts, E., and van Hillegersberg, J. 2006. "Change Factors Requiring Agility and Implications for It," *European Journal of Information Systems* (15:2), pp. 132-145.

Toward a Metric Catalog for Large-Scale Agile Development

Completed Research

Pascal Philipp
Technical University of Munich
pascal.philipp@tum.de

Franziska Tobisch
Technical University of Munich
franziska.tobisch@tum.de

Leon Menzel
Technical University of Munich
leon.menzel@tum.de

Florian Matthes
Technical University of Munich
matthes@tum.de

Abstract

Nowadays, organizations use agile software development to remain competitive in their frequently changing business environment. Inspired by the success of agile methods on a small scale, organizations have started to apply them in larger contexts. However, the limited scalability of agile methods is a problem. Metrics can be a success factor for achieving agility at scale, thus adopting them is promising. Most scaling agile frameworks provide few recommendations regarding metrics. Likewise, research on metrics in large-scale agile development lacks concrete guidance for metrics or their organization-specific adoption. To fill this gap, we propose two artifacts. We present the design of a minimalistic metric management fact sheet (MMFS) for large-scale agile development to support practitioners in using metrics in their organization-specific development environment. Furthermore, the MMFS is the basis for our metric catalog documenting 196 metrics identified in an expert study to provide a comprehensive metric set for scaling agile environments.

Keywords

Agile software development, large-scale agile development, metrics.

Introduction

In today's highly competitive market environment, software companies must instill high flexibility and ensure short lead times in order to respond to volatile requirements (Petersen and Wohlin 2010). As a result, an increasing number of software companies have begun to use incremental and agile development methods (Petersen and Wohlin 2010). Agile methodologies are currently widely adopted (Maiden and Jones 2010). The limited scalability of agile methods was identified as a limitation (Cohen et al. 2004) in a comprehensive systematic review of empirical studies of agile software development (Dybå and Dingsøyr 2008). A problem contributing to this scalability issue lies in agile methods' original design adapted to contexts similar to the "agile sweet spot" (Kruchten 2013). Frequent deliveries, medium to low system criticality, a stable underlying architecture, and co-located teams of 5–12 members are all part of this sweet spot (Nord et al. 2014). Adopting agile methods in larger settings leads to challenges, such as coordinating multiple teams and stakeholders, building trust (Brown et al. 2013; Dikert et al. 2016; Uludağ et al. 2018), and communicating with concerned parties (Dikert et al. 2016; Uludağ et al. 2018). Although using agile methods in larger projects and organizations is more difficult (Dybå and Dingsøyr 2009), the benefits of using them on a small scale convinced organizations to scale them up (Dikert et al. 2016). Several large-scale agile practices and frameworks, such as the Scaled Agile Framework (SAFe) (Leffingwell 2018) and Large Enterprise Scrum (Larman and Vodde 2016), have been developed to help organizations apply agile methods in larger settings (Paasivaara 2017).

Metrics have played a role in software development for decades (Fenton and Neil 2000). Metric adoption appears to have potential for large-scale agile development as well. Introducing measurement in agile software development, according to a variety of scaling agile frameworks (Ambler and Lines 2012; Beedle 2018; Scaled Agile Inc. 2021; Thompson 2013) and research (Brown et al. 2013), is a success factor for achieving agility at scale. However, most scaling agile frameworks offer limited guidance related to metrics. Particularly, SAFe (Scaled Agile Inc. 2021) covers metrics to a greater extent and among different organizational levels. Similarly, much of the research on metrics in large-scale agile development lacks concrete recommendations for metrics or does not provide enough guidance and information about an organization-specific adoption. Against this backdrop and inspired by Monahov (2014), we propose the design of a minimalistic metric management fact sheet (MMFS) for the domain of large-scale agile development. It serves as the basis for documenting 196 metrics, which we identified in an expert study, in our metric catalog. Our study has two objectives. First, by proposing a minimalistic MMFS, we hope to provide practitioners with guidance on how to adopt metrics in a large-scale agile development environment that is unique to their organization. Second, we want to provide a comprehensive set of metrics for scaling agile development environments. To achieve our research goals, we deduce the following research questions that guide the solution development:

RQ 1: How can a generic and minimalistic MMFS for the domain of large-scale agile software development be designed?

RQ 2: How can a structured collection of metrics help organizations execute large-scale agile software projects?

Related Work

Large-scale agile software development can refer to settings where several teams work together on a software product (Dingsøyr and Moe 2014; Dumitriu et al. 2019) or to organizations where agile methods and principles are implemented across a whole organization, including higher organizational levels (Dingsøyr and Moe 2014; Dumitriu et al. 2019). Subsequently, we present related work that presents structured collections of metrics, approaches to support metric adoption, and generic metrics documentation structures for large-scale agile development and related disciplines. Several authors investigated structured collections of metrics. Olszewska et al. (2016) propose a metrics model to quantitatively measure the impact of an agile transformation in a large software development organization. This metric model consists of a description structure for metrics which is based on the framework for evaluating metrics by Kaner et al. (2004) and the property-based software engineering measurement approach suggested by Briand et al. (1996). Bouwers et al. (2014) propose a catalog format to provide a consolidated overview of available software metrics. The format is designed to give a quick overview of each metric's current status while still providing enough information to make informed decisions about how to use it. Olsina et al. (2002) want to make it easier to understand and choose metrics. Therefore, the authors propose a sound and flexible mechanism for metric documentation and consultation. The authors present a catalog template for metrics based on a conceptual model (Olsina et al. 2002). The template is also used to document web development metrics (Olsina et al. 2002). For the domain of enterprise architecture management (EAM), Monahov (2014) proposes a metric catalog. Each metric in the catalog is documented with a description structure with a minimal number of description elements (Monahov 2014). A two-part navigation structure based on goals and concerns in EAM enables readers to identify relevant metrics efficiently (Monahov 2014). Some researchers examined approaches to support metric adoption. Hartmann and Dymond (2006) discuss aspects of appropriate agile metrics and propose tools to encourage discussion about the metrics' suitability in a particular context. The tools aim to establish measurements that are more congruent with the objectives of agile teamwork. The authors propose a checklist as a tool to improve understanding of the metrics' intent and use, as well as to mitigate the risk of misuse. The checklist, for example, covers the level of usage to indicate the intended usages for each metric at various levels of the organization. Olszewska et al. (2016) and Bouwers et al. (2014) build on the framework of Kaner et al. (2004) which aim to evaluate if proposed metrics do not conflict with construct validity. The framework consists of ten questions that address topics such as the metric's purpose and measurement instrument. Multiple authors studied generic metric documentation structures. The standard ISO/IEC/IEEE 15939:2017 (ISO 2017) provides an elaboration of the measurement process from ISO/IEC/IEEE 15288:2015 (ISO 2015) and ISO/IEC 12207:2008 (ISO 2008), which is applicable to management

disciplines and systems and software engineering. The procedure is described using a model that specifies the process's required activities, how the measures and analysis results should be applied, and how to determine whether the analysis results are valid. In addition, a process is identified to assist in the definition of an appropriate set of measures for specific information needs. Moreover, the standard describes activities and tasks to identify, define, select, apply, and improve measurement. Furthermore, the standard defines commonly used metric terms. In the field of IT controlling, Kütz (2011) discusses metrics and metric models as a basis for controlling and management approaches. Additionally, a MMFS structure was proposed. Neely et al. (2002) discuss available measurement frameworks (e.g., The Balanced Scorecard (Kaplan and Norton 1992)), and introduce their own measurement framework, called the performance prism, to counteract the shortcomings of existing measurement concepts. Moreover, tools, techniques, and methodologies for assessing the various aspects of the performance prism are discussed. In addition, a ten-element generic metric description structure is presented. Popova and Sharpanskykh (2010) define a framework for modeling performance indicators and the relationships between them. The framework is part of a broader organization modeling framework that encompasses a performance-oriented, process-oriented, organization-oriented, and agent-oriented perspective. A description structure is presented in the performance-oriented perspective of the framework for modeling performance indicators. Parmenter (2015) proposes a model for simplifying Key Performance Indicators (KPIs) and avoiding common pitfalls for organizations. Part of the model are critical success factors that serve as guidance. Also, a description structure for KPIs consisting of ten elements is presented.

Step	**Output towards the design of the artifacts**
Problem identification and motivation	We recognized that extant research on large-scale agile development lacked guidance on how to adopt metrics and did not offer a comprehensive set of metrics applicable to scaling agile development contexts.
Objectives of a solution	Our first objective was the development of an MMFS to solve the problem of lacking guidance regarding metric adoption in scaling agile environments. This MMFS should be consistent with best practices offered by literature on structured collections of metrics, approaches to support metric adoption, and generic metrics documentation structures. Further, the MMFS should incorporate best practices and goals from practice. Our second objective was to develop a metric catalog to provide a comprehensive set of metrics for large-scale agile development. Each metric should be collected from practice and documented with the MMFS. Moreover, the metric catalog should provide navigational aids to select metrics efficiently.
Design and development	The design and development process of the MMFS is illustrated in Figure 1. We conducted an expert interview study to collect metrics, goals, and best practices for metric adoption in scaling agile environments. Moreover, we assessed which best practices from literature on structured collections of metrics (Bouwers et al. 2014; Monahov 2014; Olsina et al. 2002; Olszewska et al. 2016), approaches to support metric adoption (Hartmann and Dymond 2006; Kaner et al. 2004), and generic metrics documentation structures (ISO 2017; Kütz 2011; Neely et al. 2002; Parmenter 2015; Popova and Sharpanskykh 2010) are applicable to our MMFS. Finally, we combined the final set of best practices and goal categories into an MMFS. The final set of elements and their source of origin is illustrated in Figure 1. During the expert interview study, we collected 196 metrics. Subsequently, all metrics were documented with help of the MMFS and combined into a metric catalog providing navigational aids (e.g., Goal-Metric-Matrix) to find relevant metrics efficiently.
Demonstration	We demonstrated the resulting MMFS and metric catalog during an expert discussion and within two case organization as part of a systematic process for metrics adoption and long-term management.
Evaluation	We used three survey-based iterations to compare our design proposals of the MMFS and metric catalog with the objectives described above.
Communication	Communication is being done through this paper.

Table 1. Design Science Research Steps According to Peffers et al. (2007)

Research Methodology

We developed two artifacts for the domain of large-scale agile development using the design science methodology proposed by Peffers et al. (2007): (i) a structured documentation template for metrics in the form of an MMFS, and (ii) a metric catalog, documenting metrics and goals found in the industry using the previously developed MMFS. We chose the design science research approach since it is well-known in information systems research and can be used to solve important organizational problems by developing and evaluating purposeful IT artifacts (Hevner et al. 2004). The design of the artifacts is partly based on the insights of an expert study. We describe the expert study's research methodology in another publication (Philipp et al. 2022). Table 1 summarizes each step of the design science approach.

Results

MMFS

In this section, we answer RQ1: *How can a generic and minimalistic metric management fact sheet (MMFS) for the domain of large-scale agile software development be designed?* Therefore, we present the design process of the MMFS (see Figure 1) and propose a minimalistic and generic structure to document metrics and support their organization-specific implementation. By searching current software development literature, we were able to identify description structures and their associated elements during design and development. Following that, we evaluated each identified description element's suitability for the domain of large-scale agile development. Inspired by Monahov (2014), we provide an overview of all elements from literature, including their origin (see Figure 1). The elements without a checkmark are unique to our design and stem from insights gained in the expert interviews. In total, we used ten elements in our MMFS. Our MMFS is the most similar to Monahov's (2014) design because both structures share the most elements. The calculation rule and title are the most proposed elements among all sources. Like Monahov (2014), we structured the elements into two categories. We declared elements as general elements if they were "independent from the context of a given organization" (Monahov 2014). We consider elements that "describe the configuration of the metric in a given organization" to be organization-specific elements (Monahov 2014). The findings of our expert interview study confirm the importance of organization-specific elements, as organizations adopted similar metrics but used different configurations in terms of tools, responsible persons, terminology, and metric target values on multiple occasions. Further, organization-specific elements allow keeping information confidential (e.g., employee names).

Elements of the MMFS

Like Monahov (2014), we only included "a minimal number of description elements, which are required to ensure a comprehensive metric documentation as a starting point for an organization-specific metric implementation." Subsequently, we provide a description and a rationale for each element included. We included the following seven elements as general elements:

(1) Title: The title is written in natural language and provides a concise and precise metric summary. It aids in the search for metrics. Excluding the title can make it more difficult for metric stakeholders to understand a metric quickly and easily (Monahov 2014).

(2) Description: The purpose and meaning of the metric are explained in detail in this description. As a result, it contributes to a common understanding of a metric's motivation. Dismissing the description obstructs the involved stakeholders' shared understanding of the motivation, expected benefits, and assumptions (Monahov 2014).

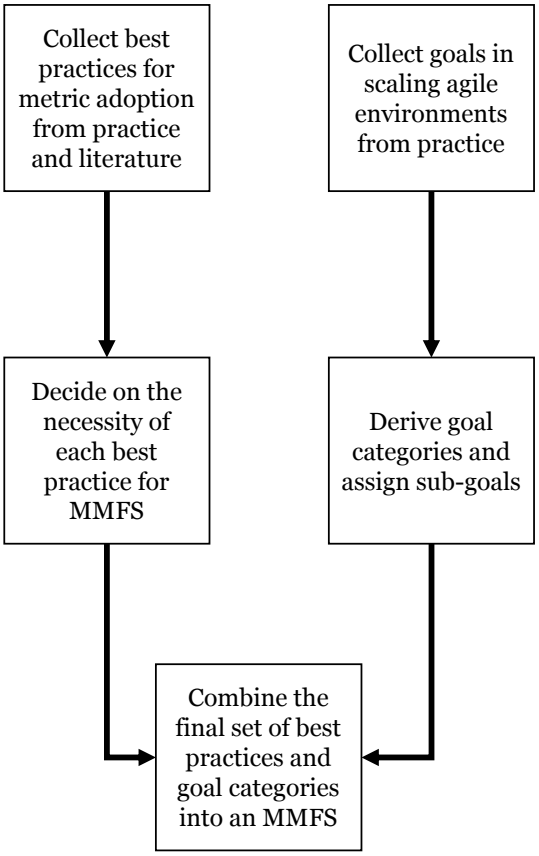
(3) Calculation rule: The calculation rule should include all relevant variables and document how a metric is calculated. A calculation rule can be expressed in natural or formal language. Both approaches are valid, according to Hartmann and Dymond (2006) and Bouwers et al. (2014). We take Monahov's (2014) approach and formulate the calculation rule in natural language while keeping it aligned with its formal definition. Thus, an easy understanding of the metric without losing information is possible, in our opinion. Furthermore, our interview research revealed that ignoring the calculation rule can cause issues because some tools require it.

(4) Information model: We document and visualize the information model underlying a metric using a UML class diagram, as suggested by Monahov (2014). The information models aid in the identification of each data item required for the calculation as well as the comprehension of the relationships between the data items. Furthermore, the information model allows for the mapping of each metric to a context and terminology specific to the organization. The data items and their relationships cannot be visualized without the information model (Monahov 2014).

(5) Code: The code functions as a metric identifier. It facilitates the quick retrieval of a metric and ensures an unambiguous use by all stakeholders (Monahov 2014). The removal of the code from the MMFS makes retrieving metrics more difficult (Monahov 2014).

(6) Goals: We assign each metric to at least one management goal within the large-scale agile development environment, similar to Monahov (2014). Each metric contributes to the achievement of the goals. There are five goal categories and 27 sub-categories that we identified in our interview study. By excluding goals from the MMFS, the link between goals and metrics is interrupted, which is a critical aspect of ensuring the efficiency of metrics (Basili et al. 1994).

(7) Organizational level: Each metric is linked to the organizational level on which it is used. According to Hartmann and Dymond (2006), the organizational level is particularly important in an agile context because it can indicate the metrics usage level's limits. In addition, the organizational level helps in the retrieval of metrics. Metrics and related objectives are often linked to organizational levels in the domain of large-scale agile development (Korpivaara et al. 2021; Stettina and Schoemaker 2018). We connected each metric to the team, program, portfolio, or enterprise level in alignment with Korpivaara et al. (2021) and Stettina and Schoemaker (2018). Removing the organizational level from the MMFS would make it impossible for stakeholders to link a metric to its organizational usage level directly and unambiguously.



MMFS element	Neely et al. [61]	Popova and Sharpanskykh [104]	Parmenter [62]	Kiitz [131]	ISO/IEC/IEEE 15939:2017(E) [130]	Hartmann and Dymond [63]	Monahov [129]	Bouwers et al. [64]	Olsina et al. [128]	Kaner et al. [106]	Olszewska et al. [90]
General elements											
Title	✗	✗		✗	✗	✗	✗	✗	✗		✗
Description			✗	✗			✗		✗		
Calculation rule	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗
Information model							✗				
Code			✗				✗				
Goals							✗				
Organizational level						✗				✗	✗
Organization-specific elements											
Organizational instance			✗								
Data mapping											
Data item							✗				
Mapped name							✗				
Data source	✗			✗			✗				
Data contact				✗			✗				
Properties											
Measurement tool									✗	✗	✗
Calculation type											
Measurement frequency/event	✗		✗	✗			✗		✗		
Monitoring frequency/event	✗										
Reporting frequency/event											
Owner				✗					✗		
Definition responsible(s)											
Implementation responsible(s)											
Achievement responsible(s)	✗			✗			✗				
Monitoring responsible(s)											
Target value(s)	✗		✗	✗			✗	✗	✗		
Planned value(s)	✗			✗			✗	✗	✗		
Tolerance value(s)		✗		✗	✗		✗	✗	✗		
Escalation rule(s)				✗	✗		✗				
Interested stakeholder						✗					

Figure 1. Visualization of the MMFS Design Process (left) and Final MMFS Elements (right)

We included the following three elements as organization-specific elements:

(8) Organizational instance: Our expert interviews revealed that most organizational environments have a variety of organizational structures (i.e., multiple teams, programs, or portfolios). Metrics may be implemented multiple times within a single organization. As a result, the organizational instance aims to document the organizational unit of each metric, such as a single team implementing a specific metric. It would be more difficult to identify the metrics' affiliation if the organizational instance was removed from the MMFS.

(9) Data mapping: This element aids in the organization-specific adoption of metrics. Organizations use different terminology for the data items of the information model, according to our interview study. As a result, we took Monahov's (2014) approach and mapped each data item in the information model to its organization-specific context. The internal term or concept is linked to a class, attribute, or relationship by

the mapped name. In addition, each data item is associated with the appropriate data source in order to document all data sources required for metric calculation. The data contact responsible can also be specified. This role is in charge of ensuring data quality and availability. The element's removal from the MMFS would result in issues with organization-specific terminology mapping, documentation of required data sources, and data contacts (Monahov 2014).

(10) Properties: To better embed the metric into its context, we propose a property element with 15 organization-specific attributes. All tools used during the metric adoption should be documented using the first attribute measurement tool. We recommend specifying the calculation type, as suggested by Olsina et al. (2002), to indicate whether a metric is calculated manually or automatically. Our expert study showed such a specification is reasonable since some tools require manual actions, whereas others are fully automated. Moreover, we recommend using an attribute for measurement frequency or event to understand the metrics underlying time series and bound it to agile events such as the sprint or program planning. The frequency or event of reporting is another recommended attribute. This attribute encourages bringing in collected data on a regular basis in the form of a report and sharing it with relevant stakeholders. The evidence from our interviews supports the definition of events for monitoring and reporting. These routines are often carried out during recurring agile events. Furthermore, we advocate defining responsibilities for activities related to metric adoption. A metric owner is responsible for ensuring sufficient resources for adopting the metric such as budget or personal. The owner is a receiver of reporting information. Four additional roles can be defined to clarify responsibilities regarding the metric definition, implementation, monitoring, and achievement. The metric achievement responsible is in charge of ensuring that the defined metric target values are met on time. One person can have multiple roles. For each responsibility, we recommend naming only one person. The remaining attributes, the target value(s), planned value(s), tolerance value(s), can be used to know when to follow the escalation rule. The planned values indicate intermediate goals at particular points in time. The escalation rule documents the responsible person's action plan when tolerance values are reached. Interested stakeholders, the final attribute, can include additional metric stakeholders. By ignoring the properties element, multiple critical elements required for successfully implementing a metric are left undocumented (Monahov 2014).

Metric Catalog[1]

The second research question is: *How can a structured collection of metrics help organizations execute large-scale agile software projects?* As a result, we propose a metric catalog design to aid in the establishment of metrics in large-scale agile software development. Matthes et al. (2012) influenced the catalog structure. Our catalog begins with a brief introduction concentrating on the motivation for designing it, its purpose, and the intended target group (i.e., practitioners and researchers). Finally, we provide navigation support to ensure that relevant metrics are retrieved quickly. Three different navigation support elements make up the navigation structure. The first element, navigation based on large-scale agile software development goals, uses a Goal-Metric-Matrix like Monahov (2014). The second element, navigation based on the metric's organizational level of usage, is connected to the organizational level element of the designed MMFS. The final navigation element, navigation through related metrics, can be used to discover metrics based on relationships. The core of the catalog is the collection of 196 metrics. Each metric is documented using the MMFS, except that for each metric, additionally all related metrics are listed and linked (see Figure 1).

Demonstration and Evaluation

During an initial demonstration, the artifacts were presented to three experts. All experts confirmed the overall idea for both artifacts, the clear structure and design of the MMFS, and the importance of connecting metrics with goals. Thereafter, we conducted three iterations of demonstrations and survey-based evaluations. The first two iterations for the MMFS and metric catalog included expert evaluations since multiple experts participated in the assessment (Peffers et al. 2012). We used the first iteration to receive feedback during a demonstration which was incorporated before performing the second iteration. In the second iteration we assessed if potential users and essential stakeholders perceive the solution artifacts as valuable. In addition to the first two iterations, we performed a third iteration for the metric catalog where

[1] Metric Catalog for Large-Scale Agile Development: https://bit.ly/39bEJRk

we applied it in two case organizations as part of a systematic process for metric adoption and long-term management. Thus, the evaluation in the third iteration was an action research evaluation (Peffers et al. 2012). In total, 14 experts participated in the MMFS evaluation and 24 experts in the metric catalog evaluation.

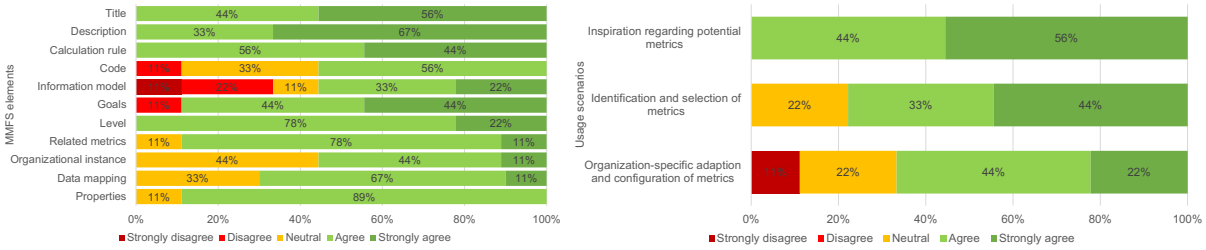


Figure 2. Evaluation of the MMF Elements and the Catalog's Usefulness in Different Usage Scenarios

(1) During the first demonstration, the artifacts were presented to 5 experts. Four recommendations were made by the experts. First, one expert suggested reformulating the calculation rule in mathematical form because it is easier to prove and relate to other metrics or assumptions this way. Because users' understanding of the mathematical definition is dependent on prior mathematical knowledge, we did not incorporate this suggestion for improvement. Second, one expert suggested extending the five goal categories we initially suggested for the goal element since it did not provide sufficient granularity for a reasonable connection between metrics and goals. We incorporated the feedback by extending the five goal categories with 27 sub-goals. Third, one expert proposed to include "a range for every metric showing you if the metric is useful or not useful." Again, we did not incorporate this feedback due to a lack of required data. Forth, including the purpose of each metric, was the final improvement suggestion of one expert. We did not add an extra element for this requirement because the second improvement suggestion (adding the 27 sub-goals) already includes information about the metric's purpose.

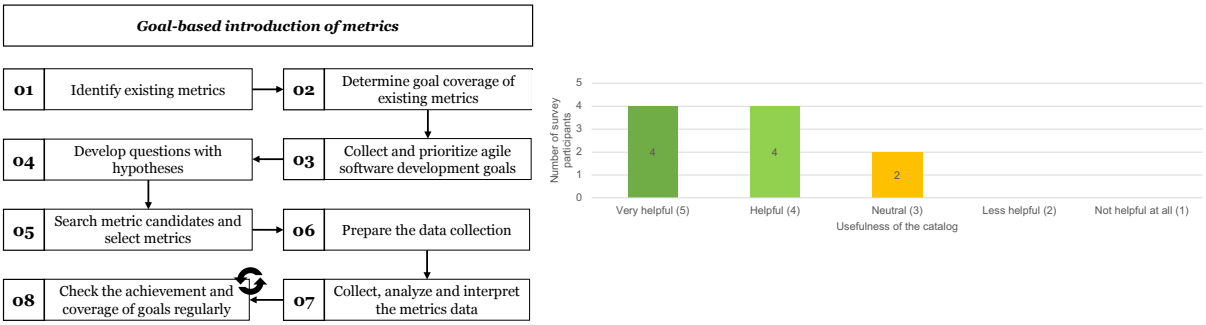


Figure 3. Metric Adoption Process (left) and Evaluation of the Catalog's Usefulness (right)

(2) During the second demonstration, the artifacts were presented to 9 experts. We evaluated whether the MMFS and metric catalog are valuable enough to be used in practice after incorporating feedback from the first iteration of our survey-based evaluation, or if they require further modification. First, we evaluated the relevance and usefulness of each element of the MMFS (see Figure 2). The survey responses show that each element is considered relevant and valuable. Only vis-à-vis the code, information model, and goals did some experts exhibit disagreement. Therefore, we decided not to apply changes to any MMFS elements. Subsequently, we introduced the experts to three possible usage scenarios of the catalog. The usage scenarios are inspiration in terms of potential metrics, identification and selection of metrics, and organization-specific adaption and configuration of metrics. Then, the experts were asked to evaluate the usefulness of each of the three usage scenarios. Even if most experts can imagine applying the catalog for all three scenarios, using the catalog for inspiration, identification and selection were the scenarios with the most agreeing answers. Therefore, we chose inspiration regarding potential metrics and identification and selection of metrics as usage scenarios for the catalog's third iteration of demonstration and survey-based evaluation.

(3) We used the catalog as part of our process for metric adoption within two case organizations (see Figure 3). The process aims at a goal-based introduction of metrics. Within the process the metric catalog was used in two process steps. In Step 2, teams determined the goal coverage based on previously identified metrics. Therefore, they conducted bottom-up searches within the catalog. Moreover, teams applied the catalog's Goal-Metric-Matrix top-down in Step 5 to compile a list of metric candidates and then select the metrics that best matched the goals and questions. We received the feedback that some relevant goals (e.g., improving collaboration) and metrics (e.g., security-related story points per sprint) are not included in the catalog. Both improvement suggestions were incorporated into the catalog. The results show that most survey participants experienced the catalog as very helpful or helpful (see Figure 3).

Discussion

Our research yielded two major conclusions. First, our study shows that a metric catalog for the domain of large-scale agile software development can aid practitioners in metric adoption. As a result, the catalog can serve as a source of ideas for potential metrics. It also makes metric identification, selection, and organization-specific configuration and adaption a lot easier. The helpfulness of the MMFS for the organization-specific configuration and adoption was also a finding of Monahov (2014) in a case study in the domain of enterprise architecture management. Second, we contend that the MMFS can facilitate metric adoption by addressing some of the challenges and success factors identified in our expert interview study. In total, we identified eleven challenges and 22 success factors (Philipp et al. 2022). The information model and data mapping elements can help overcome data collection challenges. Some experts experienced a lack of metric usefulness, which can be overcome by the description element highlighting the metric motivation and purpose. Practitioners can tackle metric calculation challenges with the calculation rule. All elements counteract a lack of metric understanding. The alignment of metrics to higher-level goals helps overcome the challenge of focusing on local instead of a global optimum. Each element of the MMFS supports the avoidance of metric definition challenges. Due to its organization-specific elements, the MMFS considers the success factor of a context-dependent metric adoption. Further, it incorporates the success factor of understanding the metric purpose. The organization-specific elements help achieve the success factor of managing the interplay of the metric and its environment. Furthermore, employing the MMFS simplifies metric adoption, contributing to the success factor of keeping metric adoption simple. The MMFS helps implementing the success factor standardization of metric adoption by providing a structure for each metric to be documented in the same format. Moreover, the goal element helps achieve the success factor of ensuring goal orientation of each metric. Furthermore, all elements contribute to the success factors of providing documentation. The title, description, and calculation rule all support accomplishing the success factor of understanding what the metric is measuring. With the responsibility attributes, the properties element achieves the success factor of assigning clear responsibilities. Finally, the attributes target, planned, and tolerance values ensure the success factor of realistic expectation management. Regarding the design science research, the limited number of experts participating in the evaluation is a limitation. Further, we created both artifacts based on the insights from a limited number of expert interviews and scientific studies. Additionally, only two case organizations from the same industry applied the catalog. We evaluated the threats to the validity of our expert interviews in another publication (Philipp et al. 2022).

Conclusion

In this study, we designed and used an MMFS to create a metric catalog for large-scale agile development that includes 196 metrics and their goals. By investigating related work and conducting 23 experts interviews in thirteen organizations, we were able to inform the design choices for both artifacts. We showed that practitioners could use the catalog to get inspired regarding potential metrics and ease the identification, selection, organization-specific metric adoption and configuration. Furthermore, the MMFS can assist in overcoming challenges that arise during metric adoption. In the future, new metrics and goals should be identified and added to the metric catalog. Moreover, as an additional navigation element, a stakeholder-based metric identification could extend the catalog. We recommend testing the catalog in further case organizations to receive more suggestions for improvement. Moreover, we plan to implement a prototypical web application of the metric catalog to ease the management of metrics and respective goals.

REFERENCES

Ambler, S.W., and Lines, M. 2012. *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press.

Basili, V.R., Caldiera, G., and Rombach, H.D. 1994. "The Goal Question Metric Approach," in: *Encyclopedia of software engineering*. pp. 528-532.

Beedle, M. 2018. "Enterprise Scrum Definition - Agility of the 21st Century." Retrieved 16. February, 2022, from http://static1.1.sqspcdn.com/static/f/608893/27795406/1515443164947/Enterprise+Scrum+Definition.pdf?token=bHxw30Cxx%2BfiftPIBq50%2F8RLYWU%253

Bouwers, E., Deursen, A.v., and Visser, J. 2014. "Towards a Catalog Format for Software Metrics," *Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics (WETSoM)*, Hyderabad, India: ACM, pp. 44-47.

Briand, L.C., Morasca, S., and Basili, V.R. 1996. "Property-Based Software Engineering Measurement," *IEEE Transactions on Software Engineering* (22:1), pp. 68-86.

Brown, A.W., Ambler, S., and Royce, W. 2013. "Agility at Scale: Economic Governance, Measured Improvement, and Disciplined Delivery," *Proceedings of the 35th International Conference on Software Engineering (ICSE)*, San Francisco, CA, USA: IEEE, pp. 873-881.

Cohen, D., Lindvall, M., and Costa, P. 2004. "An Introduction to Agile Methods," in *Advances in Computers, Advances in Software Engineering,* M.V. Zelkowitz (ed.). Amsterdam: Elsevier, pp. 1-66.

Dikert, K., Paasivaara, M., and Lassenius, C. 2016. "Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review," *Journal of Systems and Software* (119), pp. 87-108.

Dingsøyr, T., and Moe, N.B. 2014. "Towards Principles of Large-Scale Agile Development," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen (eds.), Rome, Italy: Springer, Cham, pp. 1-8.

Dumitriu, F., Meşniţă, G., and Radu, L.-D. 2019. "Challenges and Solutions of Applying Large-Scale Agile at Organizational Level," *Informatica Economica* (23:3), pp. 61-71.

Dybå, T., and Dingsøyr, T. 2008. "Empirical Studies of Agile Software Development: A Systematic Review," *Information and Software Technology* (50:9-10), pp. 833-859.

Dybå, T., and Dingsøyr, T. 2009. "What Do We Know About Agile Software Development?," *IEEE Software* (26:5), pp. 6-9.

Fenton, N.E., and Neil, M. 2000. "Software Metrics: Roadmap," *Proceedings of the Conference on the Future of Software Engineering*, Limerick, Ireland: ACM, pp. 357-370.

Hartmann, D., and Dymond, R. 2006. "Appropriate Agile Measurement: Using Metrics and Diagnostics to Deliver Business Value," *Proceedings of the AGILE 2006 (AGILE'06)*, Minneapolis, MN, USA: IEEE, pp. pp. 6 pp.-134.

Hevner, A.R., March, S.T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *MIS Quarterly* (28:1), pp. 75-105.

ISO. 2008. "Iso/Iec International Standard - Systems and Software Engineering - Software Life Cycle Processes," in: *ISO/IEC 12207:2008*.

ISO. 2015. "Iso/Iec/Ieee International Standard - Systems and Software Engineering - System Life Cycle Processes," in: *ISO/IEC/IEEE 15288:2015*.

ISO. 2017. "Iso/Iec/Ieee International Standard - Systems and Software Engineering - Measurement Process " in: *ISO/IEC/IEEE 15939:2017*. pp. 1-49.

Kaner, C., Member, S., and Bond, W.P. 2004. "Software Engineering Metrics: What Do They Measure and How Do We Know?," *Proceedings of the 10th International Symposium on Software Metrics*, Chicago, Illinois: IEEE.

Kaplan, R.S., and Norton, D.P. 1992. "The Balanced Scorecard: Measures That Drive Performance," *Harvard Business Review* (83:7), pp. 71-79.

Korpivaara, I., Tuunanen, T., and Seppänen, V. 2021. "Performance Measurement in Scaled Agile Organizations," *Proceedings of the 54th Annual Hawaii International Conference on System Sciences (HICSS)*, Kauaii, Hawaii, USA: University of Hawai'i at Manoa, pp. 6912-6921.

Kruchten, P. 2013. "Contextualizing Agile Software Development," *Journal of Software: Evolution and Process* (25:4), pp. 351-361.
Kütz, M. 2011. *Kennzahlen in Der It: Werkzeuge Für Controlling Und Management*. Heidelberg: dpunkt. verlag.
Larman, C., and Vodde, B. 2016. *Large-Scale Scrum: More with Less*. Boston, MA: Addison-Wesley Professional.
Leffingwell, D. 2018. *Safe 4.5 Reference Guide: Scaled Agile Framework for Lean Enterprises*. Addison-Wesley Professional.
Maiden, N., and Jones, S. 2010. "Agile Requirements Can We Have Our Cake and Eat It Too?," *IEEE Software* (27:3), pp. 87-88.
Matthes, F., Monahov, I., Schneider, A., and Schulz, C. 2012. "Eam Kpi Catalog V 1.0," Technische Universität München, Munich, Germany.
Monahov, I. 2014. "Integrated Software Support for Quantitative Models in the Domain of Enterprise Architecture Management." Technische Universität München.
Neely, A.D., Adams, C., and Kennerley, M. 2002. *The Performance Prism: The Scorecard for Measuring and Managing Business Success*. Prentice Hall Financial Times London.
Nord, R.L., Ozkaya, I., and Kruchten, P. 2014. "Agile in Distress: Architecture to the Rescue," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen (eds.), Rome, Italy: Springer, Cham, pp. 43-57.
Olsina, L., Lafuente, G., and Pastor, O. 2002. "Towards a Reusable Repository for Web Metrics," *Journal of Web Engineering* (1:1), pp. 61-73.
Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., and Porres, I. 2016. "Quantitatively Measuring a Large-Scale Agile Transformation," *Journal of Systems and Software* (117), pp. 258-273.
Paasivaara, M. 2017. "Adopting Safe to Scale Agile in a Globally Distributed Organization," *Proceedings of the 12th International Conference on Global Software Engineering (ICGSE)*, Buenos Aires, Argentina: IEEE, pp. 36-40.
Parmenter, D. 2015. *Key Performance Indicators: Developing, Implementing, and Using Winning Kpis*. John Wiley & Sons.
Peffers, K., Rothenberger, M., Tuunanen, T., and Vaezi, R. 2012. "Design Science Research Evaluation," *Proceedings of the International Conference on Design Science Research in Information Systems*, Las Vegas, NV, USA: Springer, pp. 398-410.
Peffers, K., Tuunanen, T., Rothenberger, M.A., and Chatterjee, S. 2007. "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems* (24:3), pp. 45-77.
Petersen, K., and Wohlin, C. 2010. "The Effect of Moving from a Plan-Driven to an Incremental Software Development Approach with Agile Practices," *Empirical Software Engineering* (15:6), pp. 654-693.
Philipp, P., Tobisch, F., and Matthes, F. 2022. "Challenges and Success Factors for Metrics in Large-Scale Agile Development," in: *28th Americas Conference on Information Systems*. Minneapolis, Minnesota: AIS.
Popova, V., and Sharpanskykh, A. 2010. "Modeling Organizational Performance Indicators," *Information systems* (35:4), pp. 505-527.
Scaled Agile Inc. 2021. "Metrics." Retrieved 29. January, 2022, from https://www.scaledagileframework.com/metrics/
Stettina, C.J., and Schoemaker, L. 2018. "Reporting in Agile Portfolio Management: Routines, Metrics and Artefacts to Maintain an Effective Oversight," *Proceedings of the 19th International Conference on Agile Software Development (XP 2018),* J. Garbajosa, X. Wang and A. Aguiar (eds.), Porto, Protugal: Springer, Cham, pp. 199-215.
Thompson, K. 2013. "Recipes for Agile Governance in the Enterprise." Retrieved 16. February, 2022, from https://www.cprime.com/wp-content/uploads/woocommerce_uploads/2013/07/RAGE-Final-cPrime1.pdf
Uludağ, Ö., Kleehaus, M., Caprano, C., and Matthes, F. 2018. "Identifying and Structuring Challenges in Large-Scale Agile Development Based on a Structured Literature Review," *Proceedings of the 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, Stockholm, Sweden: IEEE, pp. 191-197.

A Method for Metric Management at a Large-Scale Agile Software Development Organization

Pascal Philipp [1], Franziska Tobisch [1], Leon Menzel [1] and Florian Matthes [1]

[1] *Technical University of Munich (TUM), School of Computation, Information and Technology (CIT), Munich, Germany*

Abstract

The benefits of agile methods for small projects inspired organizations to scale these methods to more extensive settings consisting of multiple agile teams. Such scaling agile settings are more complex, which can make maintaining situational awareness difficult. Metrics can alleviate this problem by increasing insight into the development organization. However, adopting metrics comes with various socio-technical challenges, and current research is missing guidance on metric management in large agile organizations. Therefore, we present a goal-based method designed for a large agile case organization to support stakeholders in selecting, operating, and scaling metrics. Moreover, based on the learnings at the case organization, we present design principles that can potentially guide the development of methods suitable for other contexts. We conducted this research following an action design research (ADR) approach combined with situational method engineering (SME). Our findings indicate that our method proved effective for the case organization. This was accomplished by combining well-established elements (e.g., goal-orientation and tool support) from measurement programs designed for traditional software engineering with unique elements of our method (e.g., metric scaling activities and alignment with agile software development). With this study, we provide deep insights into how metrics are managed at a large agile case organization. Researchers and practitioners can use this work as a foundation for designing measurement programs suitable to other scaling agile organizations.

Keywords

Metrics, Large-scale Agile Software Development, Action Design Research

1. Introduction

Agile methods undeniably succeed when applied in the 'agile sweet spot' characterized by small co-located teams which frequently deliver software and interact with systems with a low to medium criticality and a stable underlying architecture [1, 2]. The demonstrated advantages of agile methods make their application also attractive in larger projects [3]. The relevance of scaling agile software development is evident in practice and research. Numerous organizations scale agile methods by adopting scaling agile frameworks [4, 5]. In parallel, within the scientific community, the number of publications on this topic has increased significantly over the past decade [5].

However, since scaling agile environments are more complex (e.g., multiple teams) than single agile team settings, sustaining an adequate level of communication becomes more difficult. Therefore, to maintain situational awareness, purely relying on qualitative feedback loops is insufficient for effective decision-making. For example, it is hard (i.e., timewise) for a program manager of a large development organization to gain insight by communicating with each agile team regularly. In such circumstances, metrics can augment or replace qualitative feedback.

In traditional software development, metrics are well-researched [6, 7] and heavily used to measure processes, products, and resources' attributes (e.g., costs). Various researchers suggested designs for measurement programs to increase the likelihood of a successful metric adoption [8]. Such measurement programs often rely on or extend the GQM approach [8]. Contrary, in large-scale agile

software development, only a few studies on metrics exist [5] even though researching metrics was declared as a high-priority research topic [9].

A simple strategy to adopt metrics in scaling agile environments could be to appropriate elements from measurement programs designed for traditional software development environments. However, several authors argue against replicating such approaches without modification [10, 11, 12]. For example, a potential risk is that metrics used in traditional software engineering undermine the agile way of working [7, 10]. Generally, embedding metrics in large-scale agile software development organizations presents considerable difficulties, as demonstrated by Philipp et al. [13], who identified eleven challenges (e.g., data collection challenges) that occurred while metrics were adopted in large-scale agile organizations.

However, although introducing metrics in large agile organizations comes with various socio-technical challenges [13], and metrics are widely applied in scaling agile organizations [9], neither scaling agile frameworks nor the scientific literature provide a method to guide practitioners in selecting, operating, and scaling metrics. Therefore, we aim to close this research gap by answering the following research questions (RQs):

RQ1*: How can metrics be selected, operated, and scaled systematically within the scaling agile organization of a large German software company?*

RQ2: *Which design principles can be derived at a large German software company to potentially support the design of future measurement programs at large-scale agile software development organizations?*

We utilized the action design research framework proposed by Sein et al. [14] as our methodological approach to address our research questions. Our investigation leads to the development of a novel method for the goal-based selection, operation, and scaling of metrics, and the formulation of ten design principles with potential applicability across different contexts. The details and implications of these findings will be discussed in the subsequent sections.

The remainder of this paper is structured as follows. In section two, we lay the foundation of this study and provide an overview of related work on metrics in large-scale agile software development. In section three, we introduce the selected research methodology, while in section four, we provide an overview of the research process and present the method we developed for our case organization. In section five, we continue by presenting design principles. After that, we discuss our results and identify future research directions in section six before we conclude this study in section seven.

2. Related Work

Management relies on metrics since they can provide quantitative and concentrated facts to support decision-making [15, 16, 17]. A prominent example of using metrics in organizations is the Balanced Score Card developed by Kaplan et al. [18]. Also in software engineering [17], agile software development (ASD) [19], and large-scale agile software development [20] a widespread application of metrics has been observed.

At first glance, using metrics in ASD and scaling agile development environments seems counterintuitive. For instance, whereas traditional measurement focuses on tracking progress against pre-made plans and measurable goals, embracing change is crucial in ASD [19]. Moreover, various challenges (e.g., data collection challenges, lack of metric usefulness, and metric calculation challenges) add to the difficulty of introducing metrics in scaled agile organizations [13].

Hence, we surveyed the literature on software engineering, ASD, and large-scale agile software development to identify studies that guide metrics selection, operation, or scaling. For software engineering, the systematic literature review by Tahir et al. [8] provide a comprehensive overview of existing measurement planning models and tools, mitigation strategies for their challenges, and success factors in implementing measurement programs.

Tahir et al. [8] identified various approaches for measurement in software engineering, such as Measurement Information Model in ISO/IEC 15939:2007 [21], Goal Question Metrics (GQM) [22, 23,

24], GQM+Strategies [25], Goal Argument Metrics [26] or the Balanced Scorecard [18]. Various models recommended extensions or improvements to the GQM approach [8]. The GQM approach builds upon the assumption that successful measurement must be initiated top-down by identifying the organization's overall and project-related goals [23]. Following goal identification, questions are assigned to goals to characterize the specific object of measurement. Finally, the questions are further refined into metrics which are used to answer the questions quantitatively [27].

Other authors provide a more technical perspective and lack an extensive exploration of the socio-technical environment in which these measurement systems operate. For instance, Staron et al. [28] introduce a framework for designing measurement systems in software development, highlighting the critical role of automated metrics collection and processing in a large organization. Moreover, while the study offers useful insights into the design and operation of measurement systems, it does not fully address the complexities and unique challenges of managing metrics in large-scale agile settings.

In the domain of ASD and large-scale agile software development, several authors use the GQM approach to identify a set of metrics serving different purposes. For example, metrics are selected to build a metrics-based reporting model [29], to establish software process measurement [30], to assess the agile project quality by Kārkliņa et al. [31], to measure the success of agile software development among a set of Critical Success Factors [32], or to measure the impact of an agile transformation [33].

Further, Ram et al. [34] identify success factors for operating metrics in ASD, focusing on factors that facilitate the long-term use of metrics. According to Ram et al. [34], data availability, metric trustworthiness, and development process are particularly important to a successful long-term metrics operation.

An alternative approach to designing, developing, sustaining, and evolving measurement programs is the formation of a dedicated metrics team. In alignment with this perspective, Meding et al. [35] introduced the metrics team maturity model, known as MeTeaM, and provided an assessment template to evaluate the effectiveness of such teams.

In summary, the approaches proposed for ASD focus on supporting metric selection, not long-term metric operation or scaling. Moreover, the proposed approaches for traditional software development were not developed in and adapted to the circumstances of scaling agile environments.

3. Methodology

We chose action design research (ADR) as a design science approach to ensure that the organizational context of the case organization shapes the method developed in this paper. According to Hevner et al. [36], methods can be typical outcomes of design science. The utilization of ADR often demonstrates superiority over traditional design science approaches, for instance due to its capacity to address immediate practical problems [37]. However, it is important to consider the potential drawbacks of this approach, as it typically requires more substantial time commitment and financial resources [37].

During our research, we followed the four stages and seven principles of ADR proposed by Sein et al. [14] (see Figure 1). The first phase, Problem Formulation, identifies and conceptualizes the research problem. During this phase, initial research questions are formulated, and the problem is presented as an instance of a class of problems. Moreover, theoretical foundations and prior technological advances, are identified. This phase contains two principles: Principle 1 declares field problems as opportunities for knowledge creation, and Principle 2 emphasizes that theories inform artifacts created through ADR.

The second phase, Building, Intervention, and Evaluation [7], uses the problem formulated in the first phase and applies the theoretical premises to generate the first draft of the IT artifact. This is further shaped through organizational use and subsequent design cycles. Three principles are characteristic of this phase: Principle 3 emphasizes the mutual influence of artifact and organizational context. Principle 4 highlights the importance of mutual learning among the different project participants. Principle 5 emphasizes the need for authentic and simultaneous evaluation of the artifact in the context of its application.

The third phase is Reflection and Learning. This phase involves reflecting on and documenting what was learned during the intervention and assessment. Principle 6, related to this phase, emphasizes that the artifact reflects not only the preliminary design (c.f., Principle 2) but also the ongoing shaping

through organizational use, perspectives, and participants (c.f., Principles 3 and 4), as well as the results of the authentic, concurrent evaluation (Principle 5).

The fourth phase is the Formalization of Learning. Here, the results are generalized and formalized so that they can be applied to similar situations. Principle 7, associated with this phase, emphasizes the need to generalize the research findings so that they can be applied to other similar situations.

The design principles we outline are not randomly chosen but result from an ongoing ADR process. These principles have been systematically refined and adjusted during each step to effectively reflect the knowledge obtained from our research. Moreover, the principles emerged from a thorough and systematic analysis of our qualitative data.

This process involved transcribing and coding semi-structured interviews, as suggested by Miles et al. [38]. The analytical tool MAXQDA supported our data analysis, helping us delve deeply into the gathered data. We also included insights from numerous workshops and team meetings while deriving these design principles.

In sections four and five, we provide a comprehensive description of the four-stage implementation in this research project, accompanied by the rationale for the principles applied.

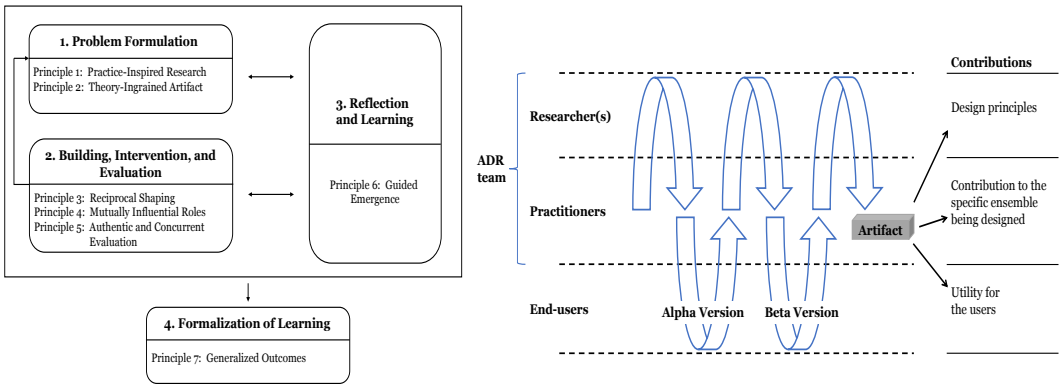


Figure 1: Action Design Research Approach (left) and ADR Stage 2: Building, Intervention, and Evaluation (right) [14]

To design our method in a structured way, we followed the recommendations of Ralyté et al. [39] for conducting situational method engineering (SME). SME effectively complements ADR, since it focuses on project-specific method construction [39]. This approach of integrating SME in ADR also resonates with other studies (e.g., Fridgen et al. [40]).

SME differentiates between method configuration (i.e., adapting a generic method to a specific situation) and method composition (i.e., selecting parts of existing methods that meet the situation's specifics). In this study, we followed the method composition strategy, also known as the assembly-based strategy, since we use elements from existing methods (i.e., GQM Approach) in addition to the elements unique to our method.

The method engineering process consists of three steps. First, we defined the method engineering goal as creating a new method for selecting, operating, and scaling metrics at our large agile case organization. From this goal, we derived the associated requirements. Second, we selected the method chunks that aligned with the requirements. Finally, we assembled these chunks into a new method.

To achieve a robust method design, we incorporated suggested method attributes and elements from Braun et al. [41] and Gutzwiller [42] into our design (see Table 1). To illustrate the method, we use the UML standard for activity diagrams.

4. Results

Below, we describe how we conducted the ADR. First, we outline the class of problems of designing measurement programs in large-scale agile software development, exemplified by the specific problem at the case organizations. After that, the respective building, intervention, and evaluation [7] cycles are described. We started the first BIE stage at InsuranceCo. (March 2021 to September 2021) where we developed an initial design of our method in two organizational units simultaneously. Building on this

initial design, we continued the ADR project at SoftwareCo. (November 2021 to February 2023). Consequently, the final design of our method is rather suited to the context of SoftwareCo. Table 2 provides an overview of both partner organizations.

Table 1
Incorporated Method Attributes and Elements [41, 42]

Fundamental Attributes of Methods
Goal orientation
Systematic approach
Principles
Repeatability
Fundamental Elements of Methods
Role
Activity / Procedure Model
Tool

Table 2
Case Organizations and Involved Roles

Research Partner	Organizational Unit	Description	Partners
InsuranceCo.	IT Provider	The IT Provider is a subsidiary of the InsuranceCo. and employs 30 people in distributed teams to operate and develop software for internal and external customers. The teams use the Scrum-at-Scale framework [43], with sprints lasting 14 days.	Scrum Master Product Owner Principal Product Owner Chief Technology Officer Information Security Officer
	Agile Release Train	The Agile Release Train comprises 80 employees responsible for developing sales software. Since it functions across organizational divisions, it has multiple interfaces with other units. The team uses the Scaled Agile Framework [44] in its Essential configuration.	Scrum Master Developer Release Train Engineer Product Manager
SoftwareCo.	Product Department	The product department consists of more than 40 agile teams working distributed on a large software solution. Within the department, the Scrum of Scrums [45] framework is implemented.	Release Train Engineer Product Manager Developer Scrum Master

4.1. Problem Formulation

We defined the problem to address as "the lack of a systematic method to select, operate, and scale metrics in scaling agile case organizations". We verified this problem and the relevance of a potential solution during focus group discussions and semi-structured interviews with our industry partners InsuranceCo. and SoftwareCo. None of our industry partners had an agreed-upon method to introduce metrics (ADR Principle 1). In both organizations, the few existing metrics were introduced autonomously by single stakeholders or agile teams and were not aligned with the organizational goals.

This problem is also present in other organizations. For example, even though Philipp et al. [13] identified implementing an agile metric management process as the second most mentioned success factor for adopting metrics in large agile software development organizations, none of the practitioner organizations defined such a process. In addition, the industry partners did not have access to expert knowledge, which became apparent across multiple dimensions. For instance, stakeholders within the organizations had limited knowledge regarding metric candidates and the associated challenges of implementing these metrics in scaling agile environments.

To verify the research gap, we reviewed existing literature on large-scale agile software development, which confirmed that previous research did not develop a method to select, operate, and scale metrics in scaling agile environments yet. Moreover, ADR Principle 2 applies to the identified problem since we classify our artifact as theory-ingrained as it builds on the GQM approach [23] and uses the SME approach to guide the design stage.

As suggested by Sein et al. [14] we relate the identified problem at the case organizations to its higher class of problems of measurement programs in large-scale agile software development. This relation is necessary because the goal of ADR is not only to design an artifact that is adapted to the specific conditions of the case organization but also to leverage the learnings obtained at the case organization to derive design principles that apply to a broader class of problems.

Multiple approaches have been proposed to support software organizations in planning their measurement programs [46]. These approaches aim to develop cost-effective measurement programs that only collect useful data with a clear purpose [46]. In essence, these frameworks focus on aligning metrics with organizational goals and the information needs of decision-makers [46].

The suspicion that existing approaches are not explicitly adapted to the requirements of large-scale agile software development organizations served as motivation to form a research program to investigate this problematic situation. For example, it needs to be clarified how measurement programs can be aligned with the development cadence, the agile events, or which agile roles could serve as stakeholders.

4.2. Building, Intervention, and Evaluation

In the second stage, we chose the generic schema for organization-dominant BIE (see Figure 1) to conduct this ADR. Therefore, we deployed the artifact early to challenge organizational participants' existing ideas or assumptions about the artifact's specific use context and to evaluate it iteratively [14]. As suggested by Sein et al. [14], we treated the building of the artifact, its intervention, and evaluation [7] as inseparable and interwoven activities executed concurrently (ADR Principle 3, 5, and 6).

We conducted the BIE cycles (i.e., Alpha and Beta Version) in two target environments at InsuranceCo. and SoftwareCo. (see Table 2). At InsuranceCo., our efforts resulted in an Alpha Version, an early and lightweight design of the method. We based the design of Version Alpha on the GQM approach, as it has proved to be rigorous and adaptable to many different environments [46, 47, 48]. At InsuranceCo. the evaluation sessions were completed during the agile team events, stakeholder discussions, and by a questionnaire.

After that, we built on the Alpha Version at SoftwareCo., where we verified our problem again before we started the next BIE cycle. Our efforts at SoftwareCo. resulted in the eventual and more mature artifact (Beta Version). Hence, the final design of our method is rather suited to the context of SoftwareCo., but it potentially encapsulates some generalizability level since it was developed in two organizations from different industries. At SoftwareCo. we performed the evaluation sessions in group discussions and during the agile team events.

In both target environments, we had the opportunity to interact with practitioners characterized by various software development and management roles (see Table 2). Our research team acted as a sparring partner to the practitioners. The practitioners contributed with their practical hypotheses and knowledge about the organization (ADR Principle 4). In February 2023, we ended the second stage of the ADR at SoftwareCo. since the value of additional cycles seemed marginal.

Figure 2 shows the timeline of our ADR project and the stages conducted at InsurcanceCo. and SoftwareCo. Below, we describe the realized design of our artifact as the outcome of the BIE stage.

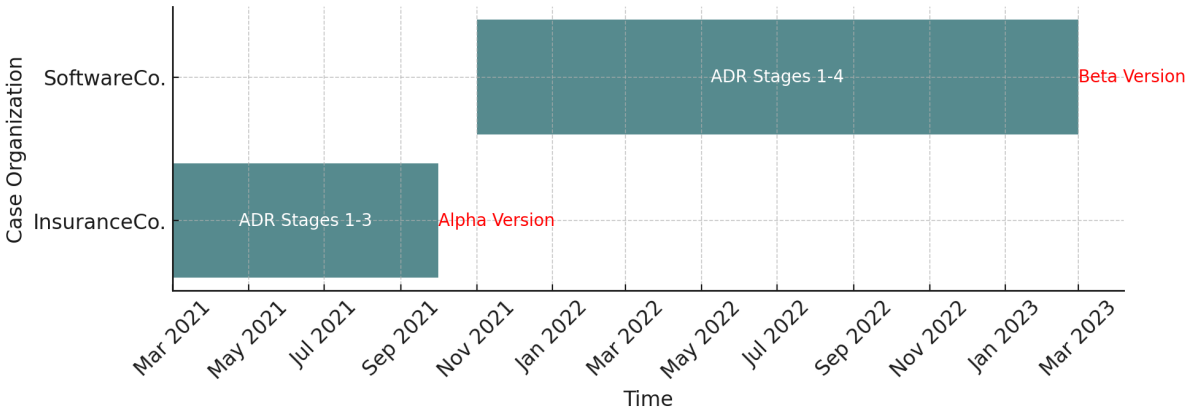


Figure 2: Timeline and Activities of the ADR Project at InsuranceCo. and SoftwareCo.

4.3. Method to Select, Operate, and Scale Metrics in Large-Scale Agile Software Development

To answer our first research question, we have developed a comprehensive method to support the goal-based selection, operation, and scaling of metrics at SoftwareCo. The method is facilitated by the Metric Catalog Web Application and its activities are executed by a metric expert team in collaboration with stakeholders who have or represent an information need. The following will describe the web application, method roles, and each activity.

4.3.1. Metric Catalog Web Application

The Metric Catalog Web Application is not an artifact of this ADR but was developed based on the findings of Philipp et al. [49]. It can be understood as a metric repository consisting of more than 200 predefined metrics. The main functions of the web app are searching for metric candidates and documenting new metrics with their corresponding goals. The metric documentation is done by declaring general metric attributes (e.g., title or calculation rule) and organization-specific metric attributes (e.g., owner and measurement frequency). See Appendix 1 for a detailed screenshot of the Metric Catalog web application user interface.

4.3.2. Roles

The metric expert team and the stakeholders perform the activities of the method together. Both roles are interdependent. The metric expert team requires the stakeholder's knowledge of their agile software development context. In turn, the stakeholders depend on the metric expert team since they usually lack the expertise and resources necessary to adopt metrics efficiently and effectively. Since the method applies to stakeholders with different roles, a top-down (e.g., Domain Managers) and a bottom-up (e.g., Scrum Masters) adoption of metrics is possible. Below, we describe both roles in greater detail.

- ***Stakeholders:*** A stakeholder is an employee at the case organization who has or represents an information need. Metaphorically, the stakeholders can be seen as information customers who

consume the measurements as a service provided by the metric expert team. As soon as the stakeholders can operate the metrics themselves, the metric expert team reduces its involvement and acts as an ad-hoc resource. The primary measure of the method's success is how much value the adopted metrics bring to the stakeholders regarding goal achievement. Stakeholders can be in various agile roles (e.g., Scrum Masters, Product Owners, Program Managers) across different organizational levels (e.g., Team, Program, Portfolio, Organization).

- ***Metric Expert Team:*** The metric team should consist of dedicated experts characterized by long-term availability, a deep understanding of the metric adoption as a socio-technical problem, excellent communication and collaboration skills, and expertise concerning the agile way of working. As a sparring partner to the stakeholders, the metric expert team must deeply understand each method activity and its rationale. Further, the metric expert team intends to increase the value of the method by scaling it within the organization, for instance, through sharing success stories within communities of practice (CoPs), a reoccurring event that invites stakeholders with the same interest to share their knowledge. Finally, for stakeholders with great change resistance, the metric expert team should be able to inspire stakeholders with a standard metric set typical to the role profile of the respective stakeholder. For example, within the case organizations, Scrum Masters tended to be most interested in process metrics, Product Owners and Developers in product metrics, and Program Managers in all metrics (i.e., Product, Resource, and Process). With more than 200 standard metrics, the Metric Catalog Web Application could possibly support the metric expert team in finding a relevant metric set.

4.3.3. Method Main Activities

To improve comprehensibility, we condensed the 20 activities into a simplified diagram (see Figure 3) where they are categorized into the three main activity groups metric selection, operation, and scaling. A comprehensive UML activity diagram illustrating all activities, their interrelationships, and their connections to the Metric Catalog Web Application is provided in Appendix 2.

Metric Selection:

- ***Ensure Management Commitment:*** Management commitment (e.g., Program Manager) is a prerequisite for the initial and every other iteration of the method. Management commitment includes providing necessary resources, such as sponsoring a dedicated metric expert team to serve as a valuable resource for stakeholders throughout the method. Additionally, announcing becoming a metric-driven organization as a goal can motivate stakeholders to fulfill their responsibilities related to the method activities. However, the method is discontinued if a re-evaluation deems it no longer viable and management stops its commitment.

- ***Choose metrics candidates that contribute to goal achievement****:* The connection of metrics to strategic goals ensures that only relevant metrics are introduced. Metrics contribute towards goal achievement by making progress towards a goal more transparent or as a condition to achieve a goal (i.e., metric target values). Goals can be collected from different sources, such as documents, tools, or stakeholders. After goal collection, the gathered goals must be documented in the Metric Catalog Web Application. Next, a measurement priority is determined for each goal. The measurement priority is based on two contributing factors. The first factor, the metric potential, describes the metrics potential to contribute to the achievement of its respective goal, for example by creating transparency on if the goal is likely to be reached or not. The second factor is defined by the goal priority, which describes how significant the goal is for the fulfillment of the organizational strategy. Discussions can serve to estimate the metric potential, goal priority, and measurement priority with the help of Likert values. We consider a detailed estimation procedure superfluous and potentially inefficient since the discussions proved effective at the case organization. Thereafter, following the GQM approach [23], questions and hypotheses are

formulated for the selected goals (i.e., goals with the highest measurement priorities). Finally, the Metric Catalog Web Application is used to search for metric candidates.

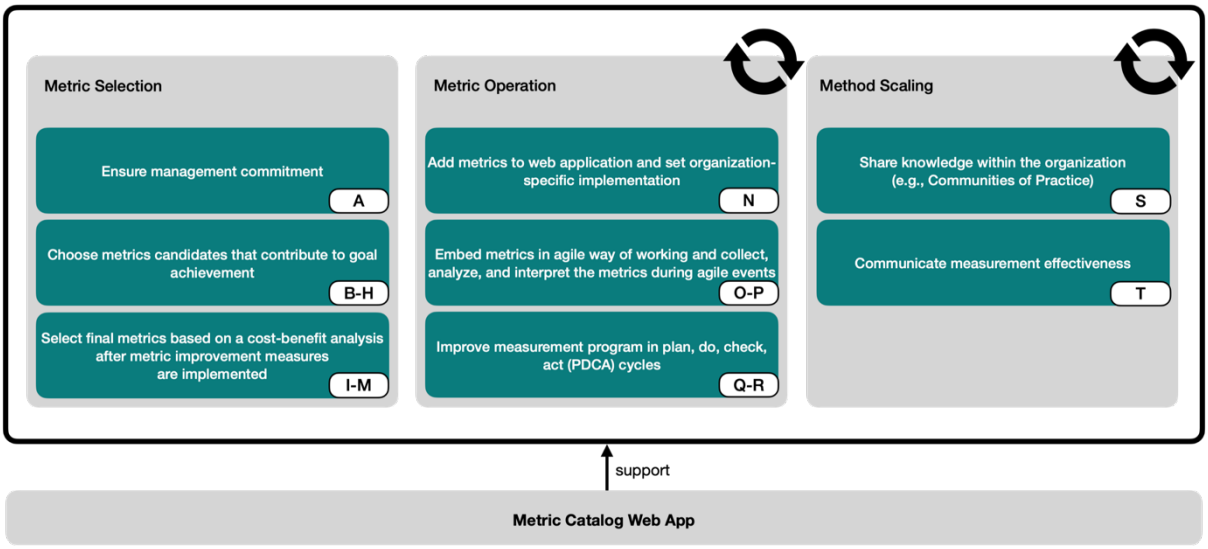


Figure 3: Method to Manage Metrics in Large-Scale Agile Software Development

- ***Select final metrics based on a cost-benefit analysis after metric improvement measures are implemented:*** For each metric candidate, a cost-benefit analysis serves as a decision basis to decide if it is selected. The benefit of a metric is determined by discussing its metric potential. On the cost side, depending on the context of the metric, multiple variables (e.g., data collection, improving data quality) are relevant. To estimate the benefits and costs, again discussions can be leveraged. During these discussions also the potential of cost improvement measures should be determined. However, since each improvement measure also has a cost, it should only be implemented if its potential for metric cost reduction outweighs its implementation cost.

Metric Operation:

The following activities are conducted regularly and aligned with the cadence of the agile events.

- ***Add metrics to the web application and set organization-specific implementation:*** The final set of metrics is added to the Metric Catalog Web Application. Therefore, the corresponding goals are assigned to the metric, and the general and organization-specific elements for each metric are filled out.

- ***Embed metrics in agile way of working and collect, analyze, and interpret the metrics during agile events:*** Metrics should be mapped to the agile event with the most suitable purpose. For example, planning metrics (e.g., velocity) are particularly suitable for the sprint planning event, whereas the sprint retrospective can be a platform to discuss the team happiness metrics. Before each event, the data collection must be finished to enable the analysis and interpretation of the data. Moreover, declaring the metric discussion as a reoccurring event agenda item can facilitate metrics to become an integral part of the agile development process.

- ***Improve measurement program in plan, do, check, act (PDCA) cycles:*** Continuously conducting PDCA cycles to improve the measurement program is necessary to increase its value over time. Important goals for the PDCA cycles could be to increase the quality of individual metrics or to ensure the relevance of the current set of metrics.

Method Scaling:

The following activities are conducted regularly.

- ***Share knowledge within the organization (e.g., Communities of Practice):*** Another option to increase the value of the measurement program lies in scaling it within the organization. Moreover, scaling the method could lead to economies of scale dynamics as introducing new metrics leads to lower long-run average costs (e.g., the metric expert team becomes more efficient). Scaling is accomplished by attracting new stakeholders to participate in the measurement program. An effective tactic to advertise the measurement program lies in sharing how other stakeholders benefited from the metrics (e.g., success stories). Communities of Practice (e.g., Scum Master CoP) could serve as a platform to efficiently communicate to many potential stakeholders.

- ***Communicate measurement effectiveness:*** Reporting on the measurement program's effectiveness serves as a basis for the management to decide if they continue or withdraw their commitment. Therefore, collecting and sharing qualitative and quantitative feedback on how valuable the stakeholders perceive the measurement program should be done regularly.

5. Reflection, Learning, and Formalization

After we ended the second stage, we successfully designed a method to support stakeholders at SoftwareCo. Limited by the nature of ADR, this method is fitted to the context of SoftwareCo. since we based the last design decisions on the context-specific requirements of this target environment. However, since our method is designed to generalize to different stakeholders within SoftwareCo. and relies on objects (e.g., goals and stakeholders) common to most organizations, we suspect some level of generalizability is already included in the current method design. Nevertheless, concerning the Metric Catalog Web Application, it is unclear if other organizations would entrust an externally deployed application with their data. A possible solution to this problem would be to host the application on servers controlled by the respective organization or use the Metric Catalog Web Application for metrics selection, not documentation. Under these considerations, we assume the method can already be applied to the broader class of problems of establishing measurement programs in large-scale agile software development organizations. To answer our second research question, we present ten design principles (ADR Principle 7) below as the output of our continuous reflection on our learnings. These design principles can potentially guide the design of similar methods for other organizations.

- ***Design Principle 1: Ensure management commitment:*** Management must provide the resources necessary to implement the method. Moreover, management should set and communicate becoming a metric-driven organization as a goal to motivate stakeholders to fulfill the responsibilities along the method activities.

- ***Design Principle 2: Establish a metric expert team who views stakeholders as customers:*** The metric expert team should view stakeholders as customers and measurements as a service. Further, the metric expert team should reduce involvement when the stakeholders can operate metrics alone and only act as ad hoc resource.

- ***Design Principle 3: Build stakeholder understanding:*** The metric expert team must ensure that the stakeholders understand the metrics they consume. Stakeholder understanding can be accomplished by involving the stakeholders in setting goals, questions, and hypotheses.

- ***Design Principle 4: Align metrics with agile software development goals****:* Metrics must be aligned with agile software development goals derived from the organizational strategy to ensure alignment.

- ***Design Principle 5: Conduct a cost-benefit analysis to select metrics:*** Metrics with the best cost-benefit ratios should be selected first. On the cost side, we learned that ensuring information quality is a primary driver; on the benefit side, the measurement priority can serve as a decision basis. In addition, the cost-benefit ratio may be enhanced by implementing improvement measures (e.g., improving information quality through defining Jira best practices).

- ***Design Principle 6: Provide a metric repository:*** Finding metric candidates and maintaining the selected set of metrics with a software solution can increase the method's efficiency. Search efficiency can be improved by providing search and filter functionalities. Maintainability is enhanced by typical create, read, update, and delete functionalities for the metrics attributes.

- ***Design Principle 7: Align metric operation with the agile way of working:*** Introducing the measurement program should not lead to losing the benefits of agile methods because the agile way of working is interrupted. In particular, the measurement program must be aligned with the agile way of working (e.g., development cadence, agile events, and agile roles).

- ***Design Principle 8: Maintain and improve the measurement program:*** The measurement program must be continuously maintained and improved. In essence, the quality and relevance of the selected metrics must be ensured continuously. Relevance and quality are critical determinants for the measurement program to provide value to the stakeholders.

- ***Design Principle 9: Scale the measurement program:*** Only maintaining and improving the measurement program limits its value to the current stakeholders. Thus, scaling the measurement program can serve as an option to increase its value since more stakeholders can profit from it. Moreover, scaling the measurement program within the organization can lead to economy of scale dynamics, reducing costs for introducing additional metrics.

- ***Design Principle 10: View the measurement program as an investment:*** The measurement program should only be continued if its future value justifies the investments necessary to run it.

6. Discussion

In response to our first research question, we developed a method that proved effective in supporting stakeholders at SoftwareCo. to manage metrics in their large-scale agile software development organization. Our method focuses on activities to select, operate, and scale metrics. The findings related to SoftwareCo. provide researchers and practitioners with an example of a successful measurement program embedded in a scaling agile environment. Thus, our study contributes to the knowledge of the inner workings of large agile software development organizations.

Moreover, to tackle our second research question, we derived general design principles from our learnings. These principles are relevant for researchers and practitioners since they can potentially help in developing measurement programs in other large-scale agile software development contexts. Moreover, we contribute to the existing body of knowledge by proving that elements of measurement programs designed for traditional software engineering can also be effective in large-scale agile software development. For instance, we build on the GQM approach proposed by Basili et al. [23], include activities to improve the measurement program (i.e., PDCA cycles), prioritize goals, conduct a cost-benefit analysis, and propose setting measurement as a goal.

However, we also contribute to existing research by presenting elements unique to our method. First and foremost, our method aligns the metric operation with the agile software development process. For instance, metrics are mapped to agile events, stakeholders can be of various agile roles, and the PDCA cycles are integrated into the agile software development workflow (c.f., Design Principle 7). Further, unique to our method, the responsibilities of the measurement experts in our method are extended (c.f., Design Principle 9) since they are not only accompanying the metric selection but also the operation and scaling (c.f., Tahir et al. [8] on measurement programs).

Moreover, our method also focuses on scaling the measurement program in the organization (c.f., Design Principle 9), whereas measurement programs designed for traditional software engineering rather focus on sustainability and reusability [8]. Another unique characteristic of our method is determined by the Metric Catalog Web application, which provides a standard set of metrics and corresponding goals for large-scale agile software development (c.f., Design Principle 6).

Concerning the limitations of our study, using ADR inevitably exposes our results to organizational bias. Only two organizations were involved in building our method, and its final design is particularly suited for the environment of SoftwareCo. In addition, each method step may not necessarily be fully developed and likely holds potential for further enhancements. Moreover, the proposed design principles are likely incomplete, and building measurement programs in other target environments would likely result in adding or adapting the design principles.

7. Conclusion

By combining ADR with SME and applying it within two large agile organizations consecutively, we developed a method that guides practitioners through activities to effectively select, operate, and scale metrics within SoftwareCo. This is accomplished by combining elements for measurement programs designed for traditional software development (e.g., Goal-orientation and tool support) with elements unique to our method (e.g., metric scaling activities and alignment with agile software development). Moreover, our method is supported by a metric management team that works closely with stakeholders during the method activities.

In addition, the Metric Catalog Web Application provides support to facilitate metrics selection and operation. Based on our learnings at the case organizations, we provide ten design principles whose applicability may go beyond the case study context. This study is relevant to researchers and practitioners, as the results provide deep insights into an effective measurement program in a large agile organization and can be used to design future measurement programs.

An interesting future research direction is to test the viability of the proposed design principles in other large-scale agile organizations. Moreover, extending the Metric Catalog Web Application to support more activities could result in a more cost-effective measurement program and is, therefore, an interesting future research direction.

8. References

[1] R. L. Nord, I. Ozkaya, and P. Kruchten. Agile in Distress: Architecture to the Rescue. in: Proceedings of the 15th International Conference on Agile Software Development (XP 2014). Rome, Italy. Springer. 2014. p. 43-57. doi: 10.1007/3-540-45017-3_9

[2] P. Kruchten, Contextualizing Agile Software Development, Journal of Software: Evolution and Process. 25.4 (2013) 351-361. doi: 10.1002/smr.572

[3] K. Dikert, M. Paasivaara, and C. Lassenius, Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review, Journal of Systems and Software. 119. (2016) 87-108. doi: 10.1016/j.jss.2016.06.013

[4] Digital.ai, 14th Annual State of Agile Report, 2020; URL: https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf.

[5] Ö. Uludağ, P. Philipp, A. Putta, M. Paasivaara, C. Lassenius, and F. Matthes, Revealing the State of the Art of Large-Scale Agile Development Research: A Systematic Mapping Study, Journal of Systems and Software. 194. (2022). doi: 10.1016/j.jss.2022.111473

[6] N. E. Fenton and M. Neil. Software metrics: roadmap. in: Proceedings of the Conference on the Future of Software Engineering. Limerick, Ireland. ACM. 2000. p. 357-370.

[7] N. Fenton and J. Bieman, Software metrics: a rigorous and practical approach. 3 ed., CRC press, Boca Raton, Florida, USA, 2014.

[8] T. Tahir, G. Rasool, and C. Gencel, A Systematic Literature Review on Software Measurement Programs, Information and Software Technology. 73. (2016) 101-121. doi: 10.1016/j.infsof.2016.01.014

[9] T. Dingsøyr and N. B. Moe. Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda. in: Proceedings of the 15th International Conference on Agile Software Development (XP 2014). Rome, Italy. Springer. 2014. p. 1-8. doi: 10.1007/978-3-319-14358-3
[10] D. Hartmann and R. Dymond. Appropriate agile measurement: using metrics and diagnostics to deliver business value. in: AGILE 2006 (AGILE'06). Minneapolis, Minnesota, USA. IEEE. 2006. p. 126–134. doi: 10.1109/AGILE.2006.17
[11] N. Oza and M. Korkala. Lessons learned in implementing agile software development metrics. in: UK Academy for Information Systems Conference (UKAIS). Oxford, UK. AIS. 2012.
[12] A. Poligadu and R. K. Moloo, An innovative measurement programme for agile governance, International Journal of Agile Systems and Management. 7.1 (2014) 26-60. doi: 10.1504/IJASM.2014.059153
[13] P. Philipp, F. Tobisch, and F. Matthes. Challenges and Success Factors for Metrics in Large-Scale Agile Development. in: Proceedings of the 28th American Conference on Information Systems (AMCIS). Minneapolis, Minnesota, USA. AIS. 2022.
[14] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, Action Design Research, MIS Quarterly. 35.1 (2011) 37-56. doi: 10.2307/23043488
[15] M. Kütz, Kennzahlen in der IT: Werkzeuge für Controlling und Management. 4 ed., dpunkt. verlag, Heidelberg, 2011.
[16] A. D. Neely, C. Adams, and M. Kennerley, The Performance Prism: The Scorecard for Measuring and Managing Business Success. Prentice Hall Financial Times London, London, 2002.
[17] E. E. Mills. Software Metrics. Carnegie-Mellon University Pittsburgh Software Engineering Institute, Fort Belvoir, Virginia, USA, 1988.
[18] R. S. Kaplan and D. P. Norton, The Balanced Scorecard: Measures that Drive Performance, Harvard Business Review. 70.1 (1992) 71-79.
[19] E. Kupiainen, M. V. Mäntylä, and J. Itkonen, Using Metrics in Agile and Lean Software Development – A Systematic Literature Review of Industrial Studies, Information and Software Technology. 62. (2015) 143-163. doi: 10.1016/j.infsof.2015.02.005
[20] P. Philipp, F. Tobisch, and F. Matthes. Investigating the Adoption of Metrics in Large-Scale Agile Software Development. in: Proceedings of the Pacific Asia Conference on Information Systems (PACIS). Virtual Event. AIS. 2022.
[21] ISO, International Standard - Software Engineering - Software Measurement Process, ISO/IEC 15939, 2002.
[22] V. R. Basili. Software Modeling and Measurement: The Goal/Question/Metric Paradigm. Maryland, USA, 1992.
[23] V. R. Basili, G. Caldiera, and H. D. Rombach, The Goal Question Metric Approach. Encyclopedia of Software Engineering. John Wiley and Sons. 528-532, 1994.
[24] V. R. Basili and D. M. Weiss, A Methodology for Collecting Valid Software Engineering Data, IEEE Transactions on Software Engineering. SE-10.6 (1984) 728–738. doi: 10.1109/TSE.1984.5010301
[25] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Munch, D. Rombach, and A. Trendowicz, Linking Software Development and Business Strategy Through Measurement, IEEE Computer. 43.4 (2010) 57-65. doi: 10.1109/MC.2010.108
[26] Ł. Cyra and J. Górski. Extending GQM by argument structures. in: Proceedings of the 2nd IFIP Central and East European Conference on Software Engineering Techniques. Poznań, Poland. Springer. 2008. p. 26–39. doi: 10.1007/978-3-540-85279-7_3
[27] M. Staron, W. Meding, M. Staron, and W. Meding, *Maintaining and Evolving Measurement Programs*, in *Software Development Measurement Programs: Development, Management and Evolution*. 2018, Springer Cham. p. 225-250.
[28] M. Staron, W. Meding, and C. Nilsson, A framework for developing measurement systems and its industrial evaluation, Information and Software Technology. 51.4 (2009) 721-737. doi: 10.1016/j.infsof.2008.10.001
[29] M. P. Boerman, Z. Lubsen, D. A. Tamburri, and J. Visser. Measuring and Monitoring Agile Development Status. in: Proceedings of the 6th International Workshop on Emerging Trends in Software Metrics. Florence, Italy. IEEE. 2015. p. 54-62. doi: 10.1109/WETSoM.2015.15

[30] P. Ram, P. Rodriguez, and M. Oivo. Software Process Measurement and Related Challenges in Agile Software Development: A Multiple Case Study. in: Proceedings of the 19th International Conference on Product-Focused Software Process Improvement (PROFES 2018). Wolfsburg, Germany. Springer. 2018. p. 272-287. doi: 10.1007/978-3-030-03673-7_20
[31] K. Kārkliņa and R. Pirta, Quality Metrics in Agile Software Development Projects, Information Technology & Management Science. 21. (2018) 54-59. doi: 10.7250/itms-2018-0008
[32] A. M. Aldahmash and A. Gravell. Measuring Success in Agile Software Development Projects: a GQM Approach. in: Proceedings of the 13th International Conference on Software Engineering Advances (ICSEA). Nice, France. IARIA. 2018. p. 38-43.
[33] J. Heidenberg, M. Weijola, K. Mikkonen, and I. Porres. A Metrics Model to Measure the Impact of an Agile Transformation in Large Software Development Organizations. in: Proceedings of the 14th International Conference (XP 2013). Vienna, Austria. Springer. 2013. p. 165-179. doi: 10.1007/978-3-642-38314-4_12
[34] P. Ram, P. Rodriguez, M. Oivo, and S. Martínez-Fernández. Success Factors for Effective Process Metrics Operationalization in Agile Software Development: A Multiple Case Study. in: Proceedings of the International Conference on Software and System Processes (ICSSP 2019). Montreal, Québec, Canada. IEEE. 2019. p. 14-23. doi: 10.1109/ICSSP.2019.00013
[35] W. Meding, M. Staron, and O. Söder, MeTeaM—A method for characterizing mature software metrics teams, Journal of Systems and Software. 180. (2021) 111006. doi: 10.1016/j.jss.2021.111006
[36] A. Hevner, S. March, J. Park, and S. Ram, Design science in information systems research., MIS Quarterly. 28.1 (2004) 75–105.
[37] J. Iivari, Distinguishing and contrasting two strategies for design science research, European Journal of Information Systems. 24. (2015) 107-115. doi: 10.1057/ejis.2013.35
[38] M. B. Miles, A. M. Huberman, and J. Saldaña, Qualitative data analysis: A methods sourcebook. 3 ed., SAGE Publications, Inc., 2014.
[39] J. Ralyté, R. Deneckère, and C. Rolland. Towards a Generic Model for Situational Method Engineering. in: Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAiSE 2003). Klagenfurt/Velden, Austria. Springer. 2003. p. 95-110. doi: 10.1007/3-540-45017-3_9
[40] G. Fridgen, J. Lockl, S. Radszuwill, A. Rieger, A. Schweizer, and N. Urbach. A Solution in Search of a Problem: A Method for the Development of Blockchain Use Cases. in: Proceedings of the 24th American Conference on Information Systems (AMCIS). New Orleans, Louisiana, USA. AIS. 2018. p. 1-11.
[41] C. Braun, F. Wortmann, M. Hafner, and R. Winter. Method construction-a core approach to organizational engineering. in: Proceedings of the 2005 ACM symposium on Applied computing. 2005. p. 1295-1299. doi: 10.1145/1066677.1066971
[42] T. A. Gutzwiller, Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen. Vol. 54. Springer-Verlag, 2013.
[43] Scrum Inc., The Official Scrum@Scale Guide, 2021; URL: https://www.scrumatscale.com/scrum-at-scale-guide/.
[44] Scaled Agile Inc., SAFe5 for Lean Enterprises, 2023; URL: https://www.scaledagileframework.com.
[45] K. Schwaber, The Enterprise and Scrum. Microsoft Press, Redmond, Washington, USA, 2007.
[46] C. Gencel, K. Petersen, A. A. Mughal, and M. I. Iqbal, A Decision Support Framework for Metrics Selection in Goal-based Measurement Programs: GQM-DSFMS, Journal of Systems and Software. 86.12 (2013) 3091-3108. doi: 10.1016/j.jss.2013.07.022
[47] P. Berander and P. Jönsson. A Goal Question Metric Based Approach for Efficient Measurement Framework Eefinition. in: Proceedings of the International Symposium on Empirical Software Engineering (ISESE'06). Rio de Janeiro, Brazil. ACM. 2006. p. 316-325. doi: 10.1145/1159733.1159781
[48] A. Boyd, The Goals, Questions, Indicators, Measures (GQIM) Approach to the Measurement of Customer Satisfaction with E-Commerce Web Sites, Aslib Proceedings. 54.3 (2002) 177-187. doi: 10.1108/00012530210441728

[49] P. Philipp, F. Tobisch, L. Menzel, and F. Matthes. Toward a Metric Catalog for Large-Scale Agile Development. in: Proceedings of the 28th American Conference on Information Systems (AMCIS). Minneapolis, Minnesota, USA. AIS. 2022.

Appendix 1

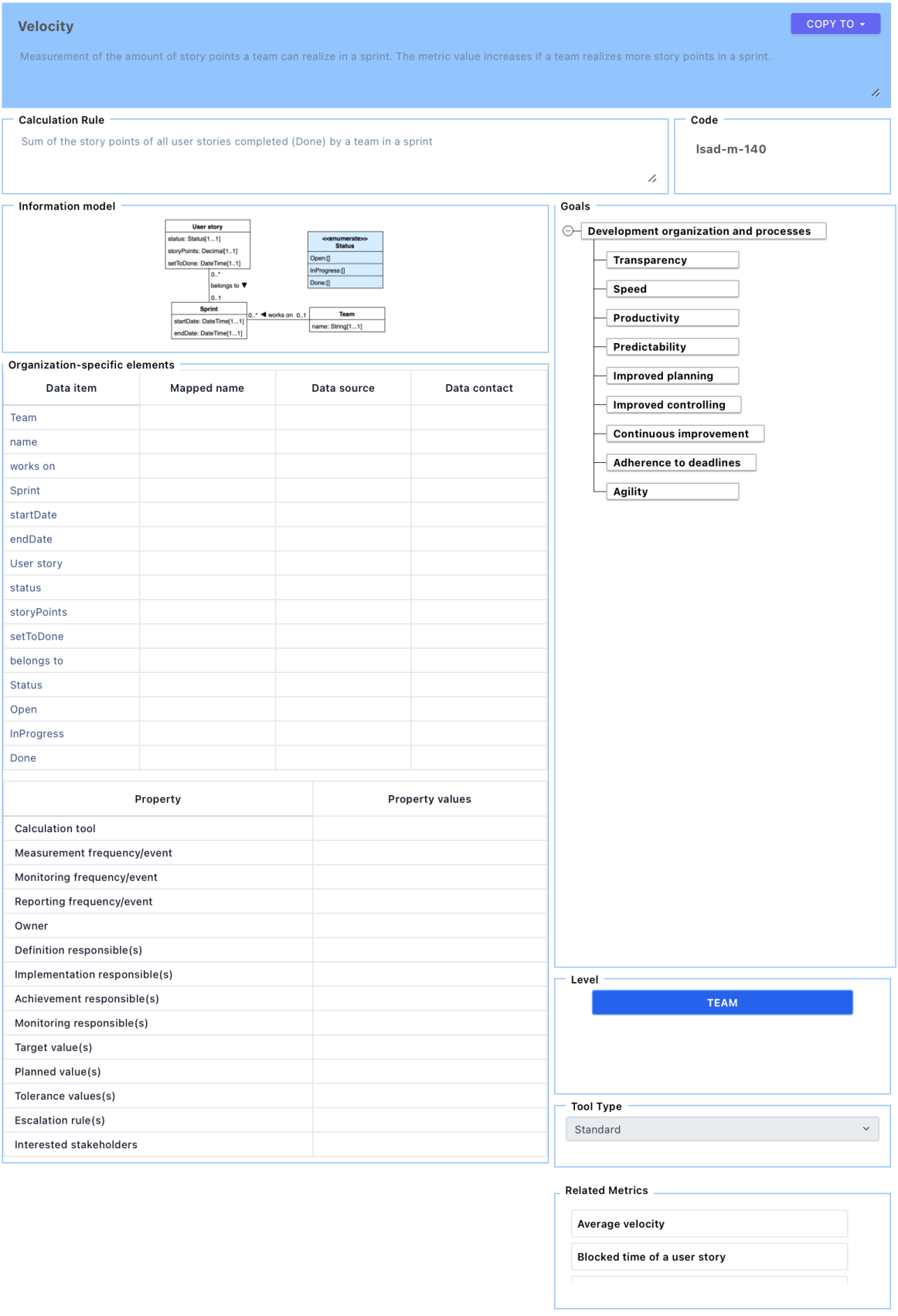


Figure 4: Metric Catalog Web Application

Appendix 2

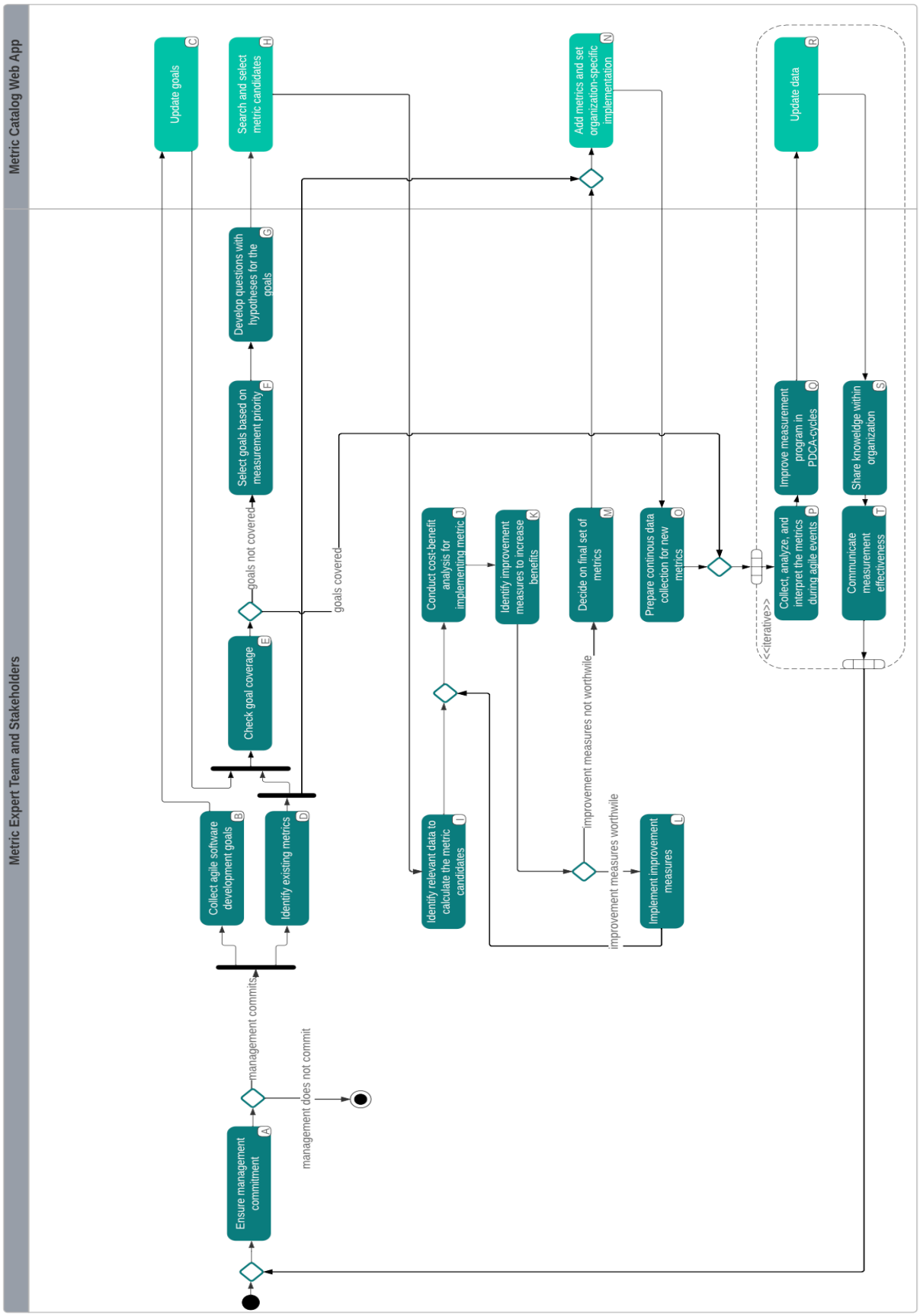


Figure 5: Comprehensive Method Activity Diagram

Investigating the Establishment of Goals in Large-Scale Agile Development

Completed Research Paper

Pascal Philipp
Chair of Software Engineering for Business Information Systems (sebis), Department of Informatics, Technical University of Munich
Boltzmannstraße 3, 85748 Garching, Germany
pascal.philipp@tum.de

Moritz Schüll
FIM Research Center, Branch Business & Information Systems Engineering of the Fraunhofer FIT
Wittelsbacherring 10, 95444 Bayreuth, Germany
moritz.schuell@fit.fraunhofer.de

Florian Matthes
Chair of Software Engineering for Business Information Systems (sebis), Department of Informatics, Technical University of Munich
Boltzmannstraße 3, 85748 Garching, Germany
matthes@tum.de

Abstract

Maintaining competitiveness in future business environments requires agile organizations to implement systematic changes such as adapting existing goal-setting practices. Such changes can only be accomplished by taking a holistic view of the development organization. In large-scale agile development, the whole development organization is considered, and some scaling agile frameworks support practitioners to establish goals by providing recommendations for goal-setting practices. However, so far, only a limited amount of research investigating the establishment of goals in scaling agile environments exists. Against this backdrop, we present a case study to explore how goals are set and documented within eight programs at a large German automobile manufacturer. Moreover, we identify and categorize goals, present goal-setting challenges encountered by the case organization, and formulate seven mitigation propositions to address these challenges. Finally, we evaluate these mitigation propositions and discuss the evaluation results by incorporating the qualitative feedback provided by the interviewees.

Keywords: Agile software development, large-scale agile, goals, goal-setting challenges

Introduction

For decades organizational thinking and practice revolved around questions of stability (Orlikowski 1996). Today however, organizations face an altered economic, technological, and political world in which reacting

to change becomes more important (Orlikowski 1996). Business agility aims at effectively managing unpredictable internal and external changes (Van Oosterhout et al. 2006). By implementing an agile IT and process architecture, organizations can enable business agility (Van Oosterhout et al. 2006). Accordingly, agile software development plays an important role for many organizations. Hitherto, several agile software development methods (e.g., eXtreme programming (XP) and Scrum) have been proposed (Dingsøyr et al. 2012). They adhere to varying degrees to the values and principles of the agile manifesto (Dingsøyr et al. 2012). The benefits of adopting agile methods are manifold, including resilience in altering environments, increased responsiveness to volatile customer requirements, and flexibility (Digital.ai 2020; Kumar and Bhatia 2012). So far, agile methods are widely adopted (Digital.ai 2020). Agile software development methods proved successful in contexts similar to the "agile sweet spot", characterized by a stable underlying architecture, co-located teams of 5-12 members, medium to low system criticality, and frequent deliveries (Nord et al. 2014). Applying agile methods in contexts departing from the agile sweet spot without or with little adaption may lead to failure (Kruchten 2013). Challenges that may occur when scaling agile methods to larger settings are, for instance, communicating with all concerned parties (Dikert et al. 2016; Talby and Dubinsky 2009; Uludağ et al. 2018) and coordinating multiple teams and stakeholders (Brown et al. 2013; Dikert et al. 2016; Uludağ et al. 2018). Multiple scaling agile frameworks, such as the Scaled Agile Framework (SAFe) (Scaled Agile Inc. 2022) and Large Enterprise Scrum (LeSS) (Larman and Vodde 2016), have been developed to support organizations in scaling agile methods.

Already Peter Drucker has emphasized the importance of management by objectives and directing all managers' vision and efforts toward a common goal. The well-known goal-setting theory (Locke and Latham 1990) originated from industrial/organizational psychology and has a high internal and external validity (Locke and Latham 2006). According to Kettunen and Laanti (2017), implementing systematic changes such as adapting the goal-setting practices becomes increasingly important for organizations to remain competitive in an environment where agility and sustainability are watchwords and software becomes the basis of products and services. Further, they emphasize that such systematic changes are complex to achieve but can be managed by taking a holistic competence- and resource-based view (Kettunen and Laanti 2017). Scaling agile frameworks like SAFe help organizations scale agile practices up to the Portfolio Level (Scaled Agile Inc. 2022), taking a holistic view of the organization. SAFe describes goal-setting mechanisms by laying out a goal-hierarchy and incorporating the SMART technique (Doran 1981) or assigning the business value to Program Increment (PI) objectives (Scaled Agile Inc. 2022). Also, in agile software development, goals are a topic of consideration. For example, in the Scrum framework, the product goal describes the future state of the product and serves as a basis for deriving Product Backlog Items (Schwaber and Sutherland 2020). However, so far, only a limited amount of research investigating the establishment of goals in the domain of large-scale agile development exists. Against this backdrop, we conducted a case study investigating how goals are established within eight programs at a large German automobile manufacturer. We describe how the goals are established, present the identified goals and encountered challenges and make mitigation propositions to address the challenges. Therefore, we formulated the following research questions (RQs) to guide this study:

RQ1: How are goals set and documented at the case organization?

RQ2: What goals have been reported at the case organization?

RQ3: What goal-setting challenges have been reported at the case organization?

RQ4: How can the goal-setting challenges of the case organization be addressed?

Related Work

Definition and Categorization of Goals

We aligned our working definition of goals with the proposition of the Goal-based Requirements Analysis Method (GBRAM) (Anton 1997) as described by Regev and Wegmann (2005): "Goals are high-level objectives of the business, organization, or system. They express the rationale for proposed systems and guide decisions at various levels within the enterprise." We identified several studies from related domains proposing categorization schemas for goals. Often these studies categorize goals among the two dimensions of goal content and organizational level. Bateman et al. (2002) developed a taxonomy of managerial goals for general management. Regarding goal content, they propose to assign goals to the categories Personal,

Financial, Customer, Market, People, Operations, Product, Organization, Competitive, and Strategy Making. In addition, they suggest the categories Ultimate, Enterprise, Strategic, Project, and Process concerning the organizational level. In the area of software requirements engineering Van Lamsweerde (2001) proposes splitting the goals into functional and non-functional requirements and differentiating between the two organizational levels high-level strategic and low-level technical. Basili et al. (1994; 2010) propose to assign goals to the categories Product, Process, and Resource regarding the goal content in the domain of software development. They suggest allocating goals among the levels Business, Software Development, and Project-Specific for the organizational level. The software value map introduces the four goal content categories innovation and learning, financial, customer, and internal business for the software development domain (Khurum et al. 2013). Along the categories of the software value map, Korpivaara et al. (2021) provide examples of performance dimensions for agile software development organizations (e.g., responsiveness to change, use of agile tools). In the large-scale agile development domain, SAFe (Scaled Agile Inc. 2022) does not propose assigning goals among the goal content dimension but distributing goals among the scaling levels Solution, Program, Team, and Iteration. Korpivaara et al. (2021) identify the seven performance objectives productivity, time to market, quality, continuous improvement, employee engagement, customer satisfaction, and alignment for scaling agile development environments. Also, dimensions to examine the success of agile transformations were an area of investigation. Korhonen (2013) identified the goals of increased visibility and capability, improved quality, and staff motivation set by the management to measure the success of an agile transformation. Likewise, Digital.ai (2019) presents dimensions of how success is measured in agile transformations. Customer satisfaction, business value, and on-time delivery are the most occurring dimensions. Furthermore, Horlach et al. (2019) present four design goals for an effective agile portfolio management system and six principles to support achieving the goals. The design goals are a customer-value-based portfolio management process, time-efficient portfolio elicitation and management process, efficient setup of allocation processes, and continuous alignment between business and IT in the portfolio management.

Goal-Setting Approaches in Agile and Large-scale Agile Software Development

In the domain of software development, the Goal Question Metric (GQM) Approach (Basili et al. 1994) is well-known for supporting practitioners in setting goals and corresponding measurements to track goal achievement. Basili et al. (2010) extended the GQM Approach to ensure business alignment. Neither the GQM Approach nor its extension GQM+Strategies incorporates requirements emerging from agile or large-scale agile development, such as adapting goals to internal and external changes. Lappi et al. (2018) argue that goal-setting in agile projects often includes close cooperation with customers to foster a common understanding of the project goals and product vision. They identify the Agile Team and the customers as the most important participants during goal-setting (Lappi et al. 2018). Further, they find the product vision a frequently used goal-setting concept because it does not require fixing goals upfront. Instead, it allows goals to emerge and change continuously (Lappi et al. 2018). Accordingly, they describe goal-setting as a continuous, iterative process instead of a separate upfront step. Moreover, Lappi et al. (2018) highlight that the alignment of project goals with the organizational strategy is discussed scarcely in the literature. A challenge related to joint performance goals based on contracts, Lappi et al. (2018) highlight problems regarding lacking commitment and conflicts over project objectives. Also, achieving relevance and sustainability is named a challenge resulting from unknown or misunderstood project objectives. Moe et al. (2019) investigate teams' goals and collective goals of programs which can be broken down into a goal hierarchy. They find that shared goals "[…] are often set by management without involving the teams, the goals are often equal to deliverables and deadlines, and team members are not always sure what the goals are". Therefore, they stress the importance of team participation during goal definition (Moe et al. 2019). Moreover, Moe et al. (2019) identify two barriers to goal-setting. Their findings show that the teams often do not interpret goals as goals. For example, frequently, teams understand goals as delivery deadlines. Moreover, they argue that a mismatch between the understanding of teams and management is the result of a missing "arena" allowing the definition of shared goals and involvement. Dreesen et al. (2020) take a control theoretical perspective to approaching agile software development. Since they defined control as "any process in which a person or group of persons or organization of persons determines […] what another person or group or organization will do" goal-setting can be interpreted as a control process where goals are not self-assigned. Korpivaara et al. (2021) find that organizations primarily focus on customer satisfaction and financial value on a higher unit level. They argue that these unit level objectives are driven by the organizational strategy and not by agile methodologies. On the program and team level, they show

that some objectives are broken down from higher level objectives. Further, on these lower levels the focus of the objectives lies primarily on efficiency and how to achieve these objectives rather than on what objectives should be set. Moreover, Berntzen et al. (2019) argue that goals in scaling agile environments can be broken down into a goal hierarchy which allows teams to have their own goals while contributing to higher-level goals. Nyrud and Stray (2017) highlight that the demo event provides an arena for creating common expectations and understanding the finished product. Vedal et al. (2021) found that Objective and Key Results (OKRs) can serve as a mechanism for managing dependencies in scaling agile environments. Further, Vedal et al. (2021) describe OKR workshops to set the direction and link the high-level objectives to the teams. During the workshop, each team was encouraged to think about the OKRs of the other teams to understand the dependencies between teams (Vedal et al. 2021). Berntzen et al. (2021), who conducted research in the same case organization, described that the workshops were conducted quarterly, and managers, team leaders, product owners, and program architects participated in defining team-specific objectives and key results iteratively. To share the progress of each team towards their OKRs across all teams, the organization used an OKR tracker (Vedal et al. 2021). According to Stray et al. (2022), who extended the research of Vedal et al. (2021), the teams were trained on the OKR topic roughly once a year. Since virtual meetings served as the medium for performing the workshops, problems regarding communication and commitment arose, and the teams struggled while defining objectives.

Concerning RQ1, current literature investigates how goal-setting is conducted in agile (Dreesen et al. 2020; Lappi et al. 2018) and large-scale agile development (Berntzen et al. 2021; Korpivaara et al. 2021; Moe et al. 2019; Nyrud and Stray 2017; Stray et al. 2022; Vedal et al. 2021). However, to the best of our knowledge, no study offers an in-depth investigation of the interplay of relevant stakeholders, events, goal-setting approaches, and documentation techniques in large-scale agile development. Regarding RQ2, contemporary literature provides high-level performance objectives and goal categorization schemas but lacks a comprehensive overview of large-scale agile development goals and their assignment to the different scaling levels. Occasionally, some studies highlight challenges occurring during goal-setting, but a thorough investigation of challenges (RQ3) and propositions to mitigate goal-setting challenges (RQ4) is lacking. Therefore, we like to fill those gaps by answering our research questions.

Research Methodology

Study Design and Overview

We conducted a single-case embedded study and analyzed the collected data to answer our research questions. The chosen research design is appropriate since case studies are means to explore contemporary phenomena within their real-life context, like goal-establishment in large-scale agile development. To ensure a rigorous research design, we followed the guidelines and recommended steps of Runeson and Höst (2009). The case selection was intentional, and the selection purpose was to identify a case typical for large-scale agile software development. Large-scale agile development either refers to environments where multiple agile teams work together on a software product (Dingsøyr and Moe 2014; Dumitriu et al. 2019) or to organizations in which agile methods are applied across the entire organization, including higher organizational levels (Dingsøyr and Moe 2014; Dumitriu et al. 2019). The case organization is typical for large-scale agile development, since in each unit of analysis within the case organization, multiple agile teams work together on software products. Also, the first sampling of interviewees was intentional (Runeson and Höst 2009). Subsequently, interviewees part of the intentional sampling recommended further interview partners (i.e., snowball sampling). Thus, we conducted twelve interviews with experts characterized by a broad spectrum of job roles, resulting in a large "variety of voices" covering numerous viewpoints (Myers and Newman 2007). Table 1 provides an overview of the interviewees.

Data Collection

We conducted our case study between March and September 2021. Before each interview, a preliminary meeting with each interviewee ensured a common understanding and definition of relevant concepts, such as goals, scaling levels or large-scale agile development roles, occurring in the interviews. The semi-structured interviews followed the same outline. We divided the interviews into two consecutive phases to investigate how the case organization establishes goals. First, we asked questions regarding the personal background of the interviewee to collect data regarding each interviewee's large-scale agile development

experience and current role within the agile setup. Moreover, we examined the development organization. Second, we asked questions exploring the current goals of each interviewee's development organization and regarding the goal-definition process. The questions of the second section were open-ended, allowing interviewees to detail the implemented processes. We conducted all interviews via videotelephony. All except for one interview were audio-recorded and subsequently transcribed. Additionally, we included relevant files (e.g., documents, pages from the corporate wiki, backlogs, and presentation slides) shared by our interview partners to facilitate the triangulation of data sources. This study only presents a subset of the overall collected data. We plan to publish the remaining evidence in a separate publication.

Data Analysis

The analysis and coding of the collected data was performed based on the guidelines by Miles et al. (2013). We combined deductive and inductive coding and performed two cycles of coding. During the first cycle, we applied a descriptive coding technique. Accordingly, significant chunks of data were summarized with codes in form of a short phrase or a word. As a result, an initial inventory of topics emerged which served as basis for the second cycle to reveal patterns among the whole data set. To create the codes, we used an integrated approach. Therefore, a provisional list of codes was created deductively considering the general interview structure and relevant concepts. The individual codes emerged inductively during coding and analysis reflecting the encountered concepts and patterns in the data. During the second cycle we used the inventory of topics created in the first cycle. We grouped the recurring and overlapping patterns using pattern codes The second cycle built on the inventory of topics created in the first cycle. Recurring, overlapping patterns across the different data were grouped using pattern codes (Miles et al. 2013).

No.	Alias	Role	LSAD experience	Program	No.	Alias	Role	LSAD experience	Program
1	AM1	Agile Master	3—5 years	A1	7	AM3	Agile Master	3—5 years	B3
2	PO1	Product Owner	3—5 years	B1	8	STE1	Solution Train Engineer	3—5 years	C1
3	BE1	Business Expert	6—10 years	B2	9	LM1	Line Manager	6—10 years	C2
4	AM2	Agile Master	3—5 years	B2	10	AM4	Agile Master	3—5 years	C2
5	PO2	Product Owner	3—5 years	B2	11	AM5	Agile Master	11—15 years	D1
6	PO3	Product Owner	3—5 years	B2	12	DEV1	Developer, Software Architect	1—2 years	D2

Table 1. Overview of the Interview Participants

Results

Context

The case organization is the IT department of a large German car manufacturer. The IT department focuses on four consecutive processes (A-D), representing major parts of the overall value chain. Each process has its own value proposition and may serve different customers. The organizational structure is standardized but independent of a particular scaling agile framework. Figure 1 illustrates the scaling levels and corresponding responsibilities in the case organization.

We mapped the terminology of the case organization to the categorization of levels in the context of large-scale agile software development inspired by Korpivaara et al. (2021) and Stettina and Schoemaker (2018). The Domain (i.e., Portfolio level) can include one or multiple products, and a Product (i.e., Program level) can consist of one or multiple Sub-Products. The Sub-Product (i.e., Team level) can be developed by one or multiple teams. Whereas agile software development only considers the team level (i.e., a single agile team), in the case organization, large-scale agile development is conducted since each process (A-D) contains at least one program consisting of multiple agile teams. At all scaling levels, an Agile Master aims to facilitate continuous improvement of the working methodologies. This role is similar to the Scrum Master but does not depend on Scrum. Product Owners are present at all levels and are in control of work content and task

prioritization. At last, the Line Manager, representing the disciplinary leadership, takes responsibility for people and resource management. Table 2 describes the major processes (A-D) in more detail.

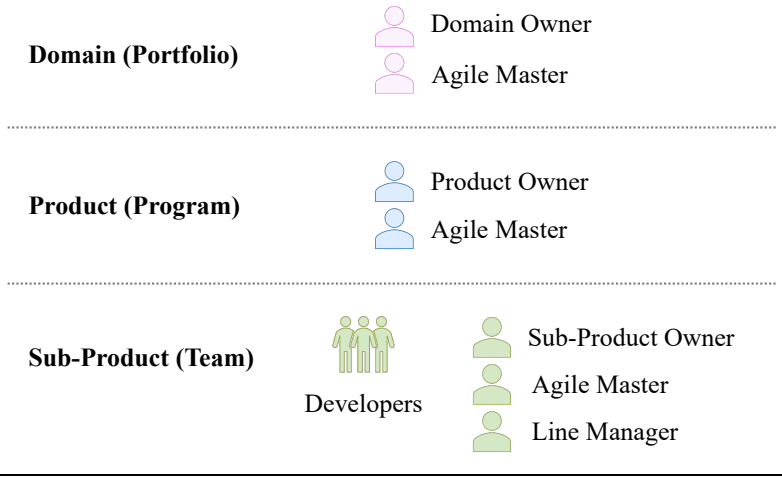


Figure 1. Overview of the Scaling Levels at the Case Organization; the Notation is Based on the One used by Stettina and Schoemaker (2018)

Process	Description
A	This process is the first step in the process chain. It comprises all activities required, from the product idea to a concrete offering presented to the customer. The focus of program A1 lies in research and development regarding autonomous driving. The program structure is aligned with the LeSS framework. The program consists of 100 agile teams working from six countries. Approximately 1,200 internal and 1200 external employees work within the program.
B	The second step, Process B, covers all activities from making an offer till receiving an order from a customer. Program B1 is concerned with software and processes for handling logistics and sales of used vehicles. Within the program, no scaling agile framework is applied. Instead, the program structure relies on the organizational structure's vertical scaling. The program comprises five agile teams, 40 internal employees, and an equal number of external employees. The teams are operating from five countries. Program B2 develops vehicle connectivity software and services, focusing mainly on off-car components. The program implements SAFe and comprises roughly 300 internal and 1,000 external employees separated into about 80 agile teams working from six countries. The focus of Program B3 lies in the development of software required by car dealers for their business operations. The software is integrated with the car manufacturers systems and processes; the development organization adopted LeSS as a framework. Two hundred fifty internal and external employees work in 22 Agile teams from four countries.
C	The next step, Process C, includes the activities from receiving a customer order to delivering the service or product to the customer. Program C1 works on replacing and renewing software systems that support the planning and ordering processes in the departments of the car manufacturer. The program implements SAFe. One hundred twenty people among 15 agile teams are working in the program from two countries. Program C2 is developing software for the shop floors within the production plants and software supporting the logistics of car parts. One thousand people are working in the program, and a combination of practices from SAFe and LeSS is applied. The number of teams was unknown by the interviewees. The teams were working from five different countries.
D	Finally, within Process D, all the activities needed to provide the financial services (i.e., car financing and leasing) are included. Program D1 belongs to a subsidiary company of the car manufacturer. The subsidiary is responsible for mobility leasing and fleet services. D1 develops a software product that aims to unify and standardize leasing processes targeting multiple mobility offerings internationally. D1 implements LeSS and has 120 employees. These employees are assigned to 17 agile teams operating from two countries. The work of program D2 is determined by the regulations the financial department must adhere to. Therefore, D2 is responsible for implementing and maintaining privileged access management comprising the organization's IT systems. D2 does not implement a particular scaling agile framework. Fifty people across seven teams from four different countries are working in D2.

Table 2. Overview of Major Processes (A-D) at the Case Organization

Goal-Setting Approaches

(1) Goal management process (GMP): Across the entire case organization a GMP is adopted to break down strategic goals throughout all scaling levels. The GMP is mandatory for all Domains and independent of the applied agile methodologies and frameworks. Since the GMP was mentioned by multiple interviewees (STE1, BE1, AM2, LM1, AM4, PO2) it seems to be highly relevant at the case organization. On the highest level, the strategic goals are defined by the Board of Directors and Strategic Management Circles with the help of a Balanced Scorecard. A Strategic Management Circle can be described as a focus group responsible for domain-specific questions not relevant to the entire board and consists of representatives from the Board of Directors and the management level directly below. For instance, in the Strategic Management Circle for the IT department, the CEO, COO, and CIO from the Board of Directors meet the heads of the IT Department to discuss IT-related questions relevant to strategic decision-making. Breaking down the goals takes place across the Product Owner structure (see Figure 1). According to PO1 and BE1, workshops are means for Product Owners to meet sub-level Product Owners to decide on sub-goals and on the distribution of subgoals among the sub-level Product Owners. This logic applies to each scaling level (i.e., Domain, Product, and Sub-Product). PO1 and BE1 stated that the GMP also allows bottom-up input from Agile Teams, which is incorporated during goal-setting mainly to clarify how goals are achieved. In contrast, the focus of the top-down process rather lies on what should be achieved. The workshops are conducted iteratively to ensure the alignment between top-down goals with bottom-up input. Before the case organization adopted agile methodologies, the GMP has been performed once a year. Currently, the strategic goals are set yearly and can span several years. Likewise, the Domain goals are derived from the strategic goals yearly. However, the frequency of deriving goals from the domain varies among the different products. In general, this happens every quarter (PO1, BE1, and LM1). On the sub-product level, deriving goals depends on the development cadence of the Agile Teams. For example, if Scrum is used on the sub-product level, deriving sprint goals might rely on the sprint length. The case organization terms the cadence for deriving goals as Domain or Product Cycle. Figure 2 shows how program C2 implements the GMP approach.

(2) Dual-track agile goal-setting process: Although adhering to the GMP is mandatory, the concrete goal-setting practices vary across the case organization. For example, one interviewee (AM5) described a dual-track agile goal-setting process implemented in program D1. This process defines how program D1 implements goal-setting within the boundaries of the GMP. Therefore, the process intends to define the next quarter's program goals and break down the defined goals into iteration goals. During the entire process, the participation of the Agile Teams is ensured. Particularly, regarding product-related goals, the input of the Agile Teams is relevant. During the goal collection, higher-level GMP goals as well as goals gathered from the Agile Teams, customers, partners, or other sources are included. Subsequently, all collected goals are aligned by the Product Owners. If a goal is observed as worthwhile to pursue, corresponding Sagas are derived and documented in the Product Backlog. Subsequently, Agile Teams conduct an initial research and ideation to identify potential solution directions for the Saga(s). If necessary, a prototype is created for a better understanding of the problem. The results of the research phase and the potential solution directions are documented as Epics within the Program Backlog. Together with the Product Owners, the Agile teams conduct an initial effort estimation for the Epics. Finally, before launching the next quarter, a "*Vision, Roadmap, and Direction (VRD) Focus Day*" is conducted. During this event which is led by the Product Owner, the stakeholders collaboratively prioritize and plan the Epics for the next quarter based on their goals. Subsequently, the Agile Teams define and refine their own Sprint Goals aligned with the Epics for the upcoming quarter. In comparison to the GMP implementation of program C2, the dual-track approach does not use OKRs and involves more planning and exploration.

Goal Definition and Documentation Techniques

(1) Pre-defined templates for goal documentation (GMP Sheets): To document the goals of the GMP, GMP Sheets must be used at the Domain level. Multiple interviewees (PO1, BE1, AM2, and LM1) described and showed the GMP Sheets. The GMP Sheets are used for goal definition, review, and reporting. The Product Owner at the respective scaling level is responsible for filling out the GMP Sheet. It is not mandatory to use the GMP Sheets on the scaling levels below the Domain level, and their adoption varies between the programs. Nevertheless, for the sake of consistency, some programs (e.g., B1) use the GMP Sheets on all scaling levels. The templates differentiate between performance- and change goals. The performance goals

represent operational targets and must be quantitative and measurable. Change goals represent changes to be implemented within the case organization to fulfill the overall strategy. The organizational guidelines recommend defining at most five goals at each scaling level. The GMP Sheet suggests documenting each goal's name, summary, and target date. For quantitative goals, the target value is also documented. The templates are usually documented with PowerPoint slides (BE1, AM5) or public accessible Confluence pages (PO1, AM5).

(2) Backlogs with a linked chain of sub-goals: Backlogs are used within each investigated program at the case organization. Usually, the Backlogs are managed with the help of tools like Jira or CodeBeamer. According to several experts (STE1, AM1, BE1, AM2, LM1, AM4, AM5, PO2, PO3), establishing a linked chain of sub-goals across the Backlogs of scaling levels is crucial. The highest level Backlog Items, which are usually coined Saga, occur on the Domain level. Sagas contribute to long-term Domain goals. The middle-level Backlog Items appear on the next lower level (i.e., Product level). These items are often called Epics. Each Epic must be linked to the Saga to which it contributes. The same logic holds true on the Sub-Product level. Each User Story must be associated with its corresponding Epic. This procedure not only supports breaking down goals, but also increases the visibility how work contributes to higher-level goals as described by a Line Manager (LM1): *"So, in this approach in this Domain it's pretty straight forward and you can kind of draw a red line directly from the targets from the Board of Directors across all of the hierarchies […]"*. Requirements contributing to goals on different levels were documented in the same Backlog or multiple times in the Backlogs on the respective level. Interviewee AM4 explained that an additional Backlog is used in program C2 to manage organizational development goals and goals elicited in Retrospectives. Moreover, according to AM4, in program B2, each Backlog Item is assigned to a goal category by using the labels Legal, Security & Compliance; Customer Functionality; Stability and Quality; and Fit for Future. The labels improve the Backlog structure and enable filtering.

(3) SMART technique: The SMART technique is commonly used in the case organization and is mandatory within the GMP. The technique recommends formulating each goal definition so that it is specific, measurable, achievable, relevant, and time-bound. Several interviewees (STE1, BE1, and AM5) stated to use this technique.

(4) Objectives and Key Results (OKRs): We identified OKRs within four programs (C1, C2, B2, D2). Figure 2 illustrates where OKRs occur in the program C2. Since OKRs are documented inside the organigram by tools, they are visible across all domains to all employees using these tools. Each Domain can decide how OKRs are implemented respectively. Interviewee LM1 described the OKR approach implemented in C2 in detail. The Domain targets are used as basis to derive quantitative objectives each quarter. The Domain Owner as well as important external stakeholders are involved during the definition of objectives. But this definition does not happen exclusively top-down as LM1 remarked: *"So, it's not like the Domain Owners together with the stakeholders define all Objectives and all Key Results by themselves. […] That's a big part, to be honest, of the daily work. But there is still room for topics coming up from the teams. […] They are able to bring those topics up for discussion as some kind of bottom-up input. So, the OKR process is designed in a way that both is possible"*. However, bottom-up input is only considered up to a certain level as stated by LM1: *"[…] to the main department. But not above, probably"*. Once, the objectives are defined quantitative Key Results are defined for the next quarter. The Product Owners are responsible to define Product OKRs based on the Domain OKRs. On Domain and Product level at most five Objectives and four Key Results per Objective are allowed. Further, each Key Result must include a Key Performance Indicator (KPI). Backlog Items are linked to Key Results to quantitatively evaluate how many Backlog Items are finished. Once all linked Backlog Items are finished, the respective Key Result is achieved. As soon as all Key Results are finished the Objective is achieved. According to LM1 the reason for this approach is: *"In this Domain we try to assure with this approach that the yearly targets get implemented on a constant basis and the progress is transparent"*.

Identified Goals[1]

In total, we identified 51 goals at the case organization. We categorized these goals among the dimensions of goal content and scaling level. Both dimensions have already been used in general management (Bateman et al. 2002), software development (Basili et al. 1994; Basili et al. 2010), and requirements engineering for

[1] Goals in Large-Scale Agile Software Development: https://bit.ly/3z5knEo

software development (Van Lamsweerde 2001). We adapted both dimensions to fit our goal data. Regarding the goal content, we chose the types of Product, Process, and Resource proposed by Basili et al. (1994) and added two additional types (i.e., Strategic and Legal, Security & Compliance). We used the levels shown in Figure 1 for the scaling level and added the enterprise and line management organization types to represent our goal data better. Table 3 shows the goal content dimension with its corresponding types. Some of the identified goals were assigned to more than one type within the dimensions.

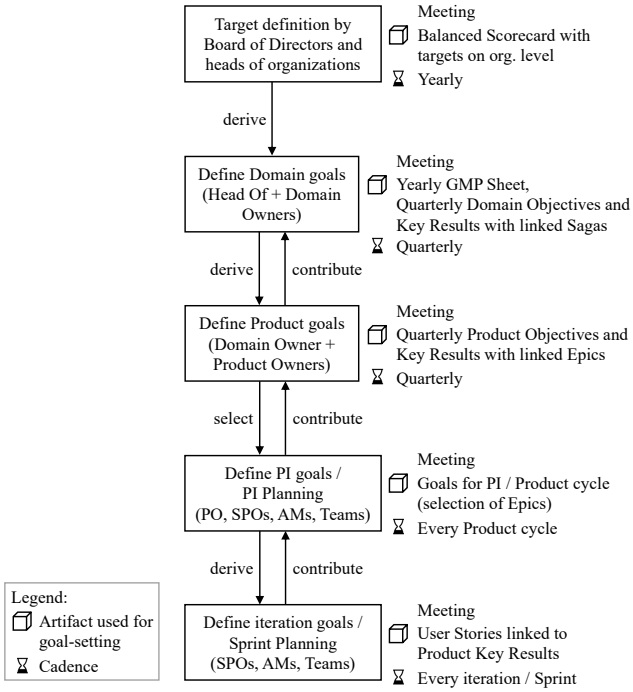


Figure 2. GMP Goal-Setting as Implemented in Program C2

Goal Content	Goal Content Types
Product Goals	Goals related to artifacts, documents or deliverables produced during the lifecycle of the offered product or service. Based on Product goals from GQM (Basili et al. 1994).
Process Goals	Goals related to internal work processes of the organization, typically associated with time. Based on Process goals from GQM (Basili et al. 1994).
Resource Goals	Goals related to resources of the organization that are used by processes for product development. Based on Process goals from GQM (Basili et al. 1994).
Strategic Goals	Long-term goals, usually relevant to multiple programs and products, that steer decisions and overall direction of the organization.
Legal, Security & Compliance Goals	Obligatory goals that must be attained; often set / defined by entities outside of the organization.

Table 3. Overview of Goal Categorization

Regarding the goal content, most of the identified goals were categorized as process goals (22 goals) and product goals (15 goals). Further, we categorized seven goals as strategic goals, four as resource goals, and three as legal, security, & compliance goals. Process and product goals occur roughly equally often on Domain and Product level. Regarding scaling level, most goals occurred on the product level (26 goals) followed by the domain level (21 goals). On the sub-product level, only one goal was identified. Moreover, we identified seven goals relevant for the entire enterprise and two goals relevant for Line Management.

Goal-Setting Challenges

We were also curious about encountered challenges during goal-setting in the case organization. In total we identified eleven challenges (see Table 4). The challenges were assigned to the constructs Interdependencies, Reporting, Adaption, Communication, Commitment, and Lack of investment. Most challenges occurred regarding Communication and Interdependencies. The most mentioned challenges related to goal-setting in the case organization are external dependencies limiting autonomous goal-setting, prioritization conflicts between goals, and management control that limits team autonomy.

ID	Construct	Name	Description	Interviewees
C1	Interdependencies	External dependencies limiting autonomous goal-setting	External dependencies like stakeholder commitment to goals, dependencies on other products or teams make it hard to define clear goals that can be achieved autonomously by the Product / Domain.	PO1, STE1, LM1, PO3
C2	Interdependencies	Prioritization conflicts between goals	Balancing different needs of stakeholders.	LM1, AM1, AM4, AM5
C3	Reporting	No goals for individual employees	Reporting on individual employees is not allowed by workers union or too resource- intensive in large programs.	PO1, STE1
C4	Adaption	Management control limits team autonomy	Fixed yearly / quarterly goals limit autonomy of Product Owners and Agile Teams. Reporting demanded by management is perceived as intrusive by Agile Teams.	AM4, AM5, STE1, DEV1, PO2
C5	Communication	Unclear goals from higher levels	Goals received from higher org. levels are not clearly defined and explained to all stakeholders.	AM2
C6	Communication	Define current state and target state for qualitative goals	For qualitative, non-measurable goals it is often hard to clearly define the current state and the target state to be achieved.	AM2
C7	Commitment	Missing attachment of teams to goals	Agile Teams lack attachment and commitment to goals they did not define themselves.	AM4
C8	Adaption	Too rigid fixation on goals	Focusing on goals causes lack of appreciation for necessary routine / operational work.	AM3, LM1
C9	Interdependencies	External contracts limiting Feature Team working model	External contractors are not allowed to operate as cross-functional Feature Teams. Instead, they own specific components only.	AM3
C10	Communication	Missing link between organizational goals and realization	Organizational goals often do not clearly define what is to be done and implemented to achieve the goal.	AM3
C11	Lack of investment	Resource constraints for goal-setting and reporting	The goal-setting and reporting processes consume a lot of resources.	BE1, PO3, DEV1

Table 4. Identified Goal-Setting Challenges

A recurring goal-setting challenge is *prioritization conflicts between different goals (C2)*. Prioritization conflicts emerge when goals are set by different stakeholders or the program itself. Further, such conflicts can be caused by rules that limit the number of allowed goals (e.g., OKR approach of program C1). Another recurring challenge of the case organization is the *limitation of team autonomy by management control (C4)*. Fixed goals set by higher-level management can limit team autonomy. Interviewee AM4 experienced this challenge as follows:

"I am struggling with the advantages of agility on the team level sometimes, because we tell the teams what they have to do within one quarter, and then we break it down into four Sprints of three weeks. And actually the four Sprints of three weeks they are predefined completely because you know what you have to do. So, agility from one Sprint to another is not really necessary."

Predefining the work of the upcoming sprints also leads to a reduction of the (Sub-)Product Owners autonomy, since their ability to prioritize their Backlog is restricted. Predetermined goals are also a result of stakeholder requirements that are bound to deadlines and effort predictions.

Propositions to Mitigate Goal-Setting Challenges

We reviewed academic literature and analyzed our interview data to identify a set of mitigation propositions that might be suitable to address the presented goal-setting challenges. During the analysis, we consolidated similar mitigation propositions and removed duplicates. In sum, we identified seven mitigation propositions (see Table 5).

(M1) Goal-setting responsibility should be shared among actors to facilitate collaborative goal-setting practices: We propose that all large-scale agile development roles (i.e., Agile Teams, Agile Masters, Product Owners, and Line Managers) should be allowed to contribute to and propose goals of any type. We argue that this proposition is a precondition allowing collaborative goal-setting activities and practices. Therefore, the proposition addresses the challenges of missing attachment of Agile teams to goals (C7) and receiving unclear goals from higher levels (C5). Furthermore, the proposition is backed by literature findings highlighting the importance of shared goals and collaborative goal-setting (Berntzen et al. 2019; Moe et al. 2019). Similarly, Schnabel and Pizka (2006) also emphasize collaborative goal-setting in their process. Also, the classical goal-setting theory postulates better group performance because of setting goals collaboratively (Locke and Latham 2006). Adhering to this mitigation proposition can address the challenges C5 and C7, and therefore, Communication and Commitment can be improved.

(M2) Goal definition practices (e.g., SMART, OKRs, GQM) ensure clear understanding of goals: The program should agree upon goal definition practices and establish a Definition of Ready (DoR) (cf. Power (2014)) for goals like the Definition of Done (DoD) for Backlog Items in Scrum. In our study, we identified OKRs (LM1 and DEV1), the SMART technique (B2, C1, and D1), and program press releases (AM5) which describe the situation that is expected in the program D1 once a specific goal will be achieved. In addition, we argue that goal definition practices can address the challenges of unclear goals (C5) and challenges related to defining the current state and target state for qualitative goals (C6). Because realizing this mitigation proposition can help overcome the challenges C5 and C6, Communication can be improved.

(M3) A linked chain of sub-goals across scaling levels should be established to facilitate transparency: We propose to link each goal or Backlog Item to the goals on the next higher scaling level. Thus, the challenge of a missing link between organizational goals and realization (C10) can be addressed. Furthermore, according to Berntzen et al. (2019), breaking down goals into a hierarchy might facilitate coordination. By realizing this mitigation proposition the challenge C10 can be addressed, and Communication facilitated.

(M4) Goals should be documented publicly for all actors and stakeholders to facilitate transparency: We already described the GMP Sheet as documentation practice within the case organization which implements our proposition. Another documentation practice found in the case organization is the corporate wiki based on Confluence. It is also used for goal documentation in multiple programs (B2, B3, C1, C2, D1, D2). We argue that the challenge of unclear goals (C5) is addressed by documenting goals publicly. Since implementing this mitigation proposition can address challenge C5, Communication can be improved.

(M5) Goals from external stakeholders should be broken top-down along the Product Owner hierarchy to ensure consideration of dependencies and coordination: We argue that this proposition can address the challenge of external dependencies limiting autonomous goal-setting of teams (C1). The proposition is aligned with the Berntzen et al. (2019) finding that the Product Owner plays an important role in coordination in large-scale agile development. Further, deriving sub-goals from higher-level goals enables teams to adhere to higher-level dependencies implicitly. Fulfilling this mitigation proposition can counteract the challenges C1 and C2, and therefore, problems regarding Interdependencies can be solved.

(M6) Definition, prioritization, and communication of middle- to lower-level goals should involve Agile Teams to ensure consideration of technical aspects and acceptance of goals by the teams: We argue, by implementing the proposition the challenge of missing attachment of teams and goals (C7) can be addressed. The concept of "top-down thinking and bottom-up acting" by Schnabel and Pizka (2006) is similar to the combination of M5 and M6. However, whereas Schnabel and Pizka (2006) only differentiate between stakeholders ("top") and developers ("bottom"), we consider multiple organizational scaling levels and roles specific to large-scale agile development. Moreover, conversely to Schnabel and Pizka (2006), we emphasize that Agile Teams should not only act on given goals ("bottom-up acting") but also be actively involved in defining goals ("thinking"). Implementing this proposition can tackle the challenges C2, C4, and C7. Consequently, Interdependencies, Adaption, and Commitment can be improved.

(M7) All goals should be maintained in Backlogs to facilitate clear understanding and transparency: According to Schwaber and Sutherland (2020) only artifacts that are documented transparently can enable transparent decisions and goals. A common practice for documenting goals is the Backlog used by all programs of the case study. Further, Backlogs are recommended by several scaling agile frameworks (e.g., (Larman and Vodde 2016; Scaled Agile Inc. 2022; Schwaber 2018)). We argue that using a Backlog can address the challenge C5 of goals from higher organizational levels that are not clearly defined and explained to all stakeholders. Since implementing this mitigation proposition can address challenge C5, Communication can be improved.

ID	Description	Addresses	Explanation
M1	Goal-setting responsibility should be shared among actors, to facilitate collaborative goal-setting practices	C5, C7	All actors and stakeholders can define goals of any kind (but not prioritize them); Based on statements by LM1, AM5; Literature finds shared, collaborative goal-setting facilitates group performance (Locke and Latham 2006) and coordination in LSAD (Berntzen et al. 2019; Moe et al. 2019; Schnabel and Pizka 2006)
M2	Goal definition practices (e.g., SMART, OKRs, GQM) ensure clear understanding of goals	C5, C6	SMART is used in programs B1, B2, C1, D1; OKRs is used by programs B2, C1, C2, D2; The used goal definition technique should be documented in the Definition of Ready document
M3	A linked chain of sub-goals across scaling levels should be established to facilitate transparency	C10	Link team goals to program goals, and program goals to portfolio goals; Commonly implemented with linked Backlog Items, using Stories (team), Epics (program), Sagas (portfolio); Jira or similar solutions are used by programs A1, B2, C1, C2, D1; Berntzen et al. (2019) make a similar observation
M4	Goals should be documented publicly for all actors and stakeholders to facilitate transparency	C5	Usage of public documentation, e.g., Wiki-pages, allows everyone to access the most recent goals (AM2, AM3, AM5, PO3, DEV1, LM1, STE1); Ensures transparency and mitigates unclear goals
M5	Goals from external stakeholders should be broken top-down along the Product Owner hierarchy to ensure consideration of dependencies and coordination	C1, C2	Breaking down goals for the next lower level should be done collaboratively with Product Owners from both levels; e.g., via regular workshops according to development cycles similar to GMP process at case org.; Influenced by findings on importance of POs for coordination by Berntzen et al. (2019)
M6	Definition, prioritization, and communication of middle- to lower-level goals should involve Agile Teams, to ensure consideration of technical aspects and acceptance of goals by the teams	C2, C4, C7	Agile Teams should always be involved in the 'how' of goals (AM2); E.g., via participation in workshops for breaking down goals in addition to participation in Refinements and Plannings on team-level; Murphy and Cormican also suggest to involve developers (Murphy and Cormican 2015)
M7	All goals should be maintained in Backlogs to facilitate clear understanding and transparency	C5	Goals of all types are documented and maintained in Backlogs by programs A1, B1, B2, B3, C1, C2, D1, and D2.

Table 5. Identified Mitigation Propositions

Evaluation of Mitigation Propositions

We conducted eleven semi-structured interviews to evaluate our mitigation propositions. Our evaluation goals were to assess to which degree practitioners agree with our propositions and collect qualitative feedback. Table 6 provides an overview of the evaluation participants.

Again, the semi-structured interviews began with gathering information on the interviewees' role and large-scale agile development experience. Subsequently, we asked the interviewees to which degree they agree with our proposition using a five-point Likert scale (Likert 1932). For each proposition, the interviewees could provide further thoughts. We conducted the coding of the qualitative data consistent with the coding procedure of our case study. Figure 3 depicts an overview of the evaluation results. The experts agreed with most of the propositions. Proposition M4 received the best evaluation results and M7 the worst. There was no clear consent among the experts regarding M5 and M7. Subsequently, we discuss the evaluations results by incorporating the qualitative feedback provided by the interviewees.

(M1) Goal-setting responsibility should be shared among actors to facilitate collaborative goal-setting practices: Multiple interviewees highlighted the importance of combining top-down with bottom-up interaction during goal-setting (LM1, BE1, PO4). According to BE1 and LM1, no single person has sufficient knowledge to set goals independently. PO5 proposed that teams should contribute to goal-setting with their technical expertise and not exclusively focus on achieving goals set on higher levels. As reported by DEV1

and PO1, shared goal-setting can be a means to increase team buy-in and transparency of goals. Interviewee AM7 marked shared goal-setting as a prerequisite for teams to achieve autonomy.

(M2) Goal definition practices (e.g., SMART, OKRs, GQM) ensure a clear understanding of goals: The interviewees did not advocate a specific goal-setting practice but emphasized that within a program, one goal-setting method should be chosen and standardized (LM1, BE1, PO4, PO5). Several benefits were highlighted for using a single agreed-upon goals-setting practice within a program. First, the goal definition overhead is reduced since all program members know how to define goals (LM1). Second, goals are aligned horizontally and vertically (DEV1), and third, trends regarding goal definition can be identified early because the goal definition is more consistent (LM1). According to AM2 and AM7, goal definition should be based on a strong theoretical background to avoid over customization of goal-setting practices.

(M3) A linked chain of sub-goals across scaling levels should be established to facilitate transparency: According to LM1 and AM7, a linkage of sub-goals to higher-level goals is necessary to ensure strategy implementation. Further, it facilitates understanding why the defined goals are pursued (BE1, PO4) and warrants that each goal is aligned with a higher-level objective (PO6). Nevertheless, while the interviewees generally agree that a linkage should be mandatory, some goals may exist that cannot be linked to a higher-level objective (AM2, LM1, PO3, PO5, PO6). One example is technical changes to the software product, which might not contribute to any higher-level goal (e.g., a version upgrade to an existing database) but was classified as important by the agile team (AM2).

(M4) Goals should be documented publicly for all actors and stakeholders to facilitate transparency: Documenting goals publicly leads to several benefits. First, since it helps to identify who is working on a similar objective (AM7, LM1), redundant work can be reduced (AM7, LM1). Moreover, it enhances the alignment of stakeholder expectations (PO4) and can reveal if stakeholder expectations do not match goals (AM2). Finally, transparency is increased because it becomes clear why employees act the way they do (DEV1, PO1).

(M5) Goals from external stakeholders should be broken top-down along the Product Owner hierarchy to ensure consideration of dependencies and coordination: According to DEV1, PO3, and PO4, top-down distribution of external goals along the Product Owner hierarchy ensures that goals are adequately defined and distributed. Nevertheless, the evaluation results show that no clear consent regarding M5 exists. Since propositions M5 and M6 are closely tied to each other, we discussed them together during the evaluation. Most evaluation participants assigned M6 higher importance than M5 within scaling agile environments. As reported by AM2, depending on how specific a goal is, it should be determined whether the Product Owners should break down goals or not. Due to the evaluation results, we suggest modifying M5. The top-down process should be applied only when stakeholders come up with broad, rather unspecific goals that cannot be directly allocated to individual teams and entail dependencies and a need for coordination.

(M6) Definition, prioritization, and communication of middle- to lower-level goals should involve Agile Teams, to ensure consideration of technical aspects and acceptance of goals by the teams: According to PO1 and PO3 teams must be involved since they possess the technical expertise that is often missing at higher organizational levels. BE1 and PO5 point out that goals should never be assigned to stakeholders without the involvement of Agile Teams. Although involving Agile Teams in goal-setting is necessary, the prioritization of goals lies in the Product Owner's responsibility (AM2, LM1). Therefore, Agile Teams should not be accountable for goal-setting but involved in the process (LM1).

(M7) All goals should be maintained in Backlogs to facilitate clear understanding and transparency: We observed two standpoints regarding the usage of Backlogs. DEV1 and PO6 argued that the Backlog is a central artifact in their development where all goals and requirements should be stored. Other interviewees (AM2, PO4, PO5) recommended strictly separating goals and requirements. However, LM1, BE1, and PO5 suggested that it also depends on the definition of the Backlog if requirements and goals are documented together. The interviewees from both standpoints agreed on documenting goals digitally (AM2, AM7, PO1, PO4) and establishing a connection between the development Backlogs and organizational goals (AM2, AM6, LM1, BE1, PO1, PO4).

No.	Alias	Role	LSAD experience	No.	Alias	Role	LSAD experience
1	PO1	Product Owner	3 - 5 years	7	PO6	Area Product Owner	3 - 5 years
2	BE1	Business Expert	6 - 10 years	8	LM1	Line Manager	6 - 10 years
3	AM2	Agile Master	3 - 5 years	9	AM7	Agile Master	6 - 10 years
4	PO4	Product Owner	3 - 5 years	10	DEV1	Developer, Software Architect	1 - 2 years
5	PO5	Product Owner	6 - 10 years	11	PO3	Product Owner	3 - 5 years
6	AM6	Agile Master	6 - 10 years				

Table 6. Evaluation Participants

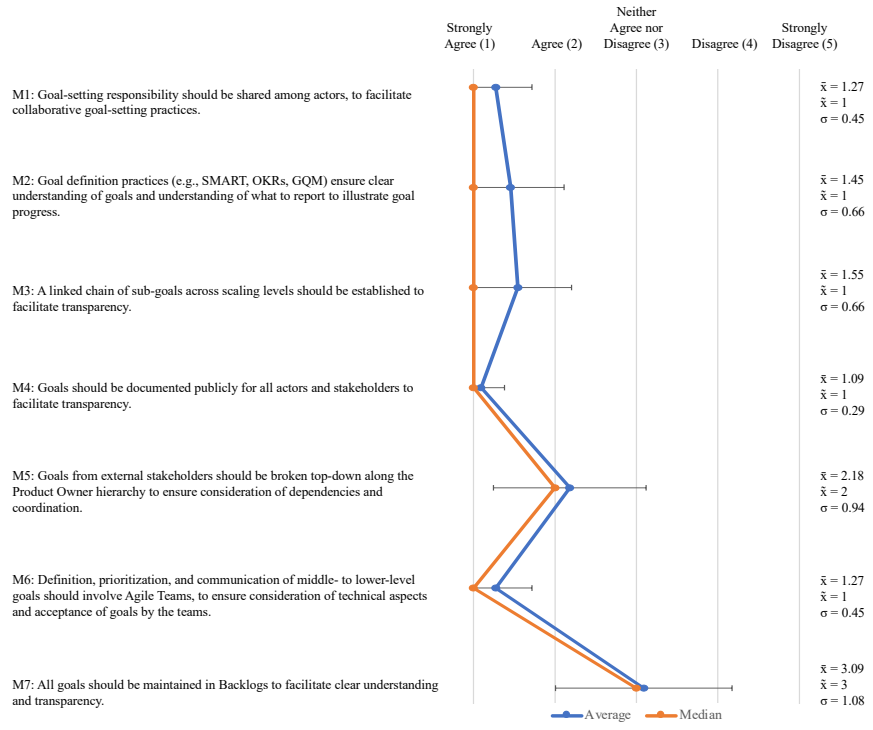


Figure 3. Evaluation Results of the Mitigation Propositions

Discussion

Key Results

Three key findings emerge from this case study. First, in large-scale agile organizations, involving the agile teams during goal-setting becomes more important. Current literature states that, the bottom-up perspective of agile teams rather focuses on how goals are achieved or implemented while defining what goals should be pursued is often the responsibility of Product Owners and management taking a top-down perspective (Korpivaara et al. 2021; Schnabel and Pizka 2006). Our study shows that in large-scale agile development this distinction may not be suitable anymore. In particular, the evaluation results of M5 and

M6 support this claim. We observed that the practitioners across the case organization need to involve the team in all aspects of goal-setting on the team and program level to cope with the challenges of rising product complexity and scope. Second, most identified goals belong to the portfolio and program level. We documented 21 goals on the portfolio level and 26 goals on the program level. Further, we observed only one goal on the team level and seven goals associated with the entire enterprise. These findings may indicate that in large-scale agile organizations, the relevance of goal-setting is higher in the middle levels (i.e., portfolio and program level) than on the highest (i.e., enterprise level) and lowest level (i.e., team level). Moreover, our findings show that process and product goals are roughly equally set on portfolio and program level. This observation contradicts Korpivaara et al. (2021), who show that process efficiency objectives are more often set at the program level than higher levels in the organization. Third, in practice the distinction between goals and requirements is inconsistent. This is supported by the insights gained during the evaluation of M7, where we observed two standpoints on whether Backlogs Items can be goals or not. We recommend that organizations should try to clearly document their goals and outline for each requirement to which goals it contributes. These suggestions can be achieved by several of our propositions (e.g., M3, M4).

Limitations

To evaluate possible validity threats, we used the assessment scheme of Runeson and Höst (2009). To address threats to construct validity, we used multiple data sources and established a chain of evidence. Moreover, our insights were gathered from interviewees with different roles and experiences in software development. To address potential threats to internal validity, we conducted a preparation meeting before each interview to ensure a common understanding of concepts and terms. We described our coding system in detail to counteract potential threats to reliability. Nevertheless, while we did a cross-case analysis between the programs, our findings are specific to the case organization. We countermeasure the threat of external validity by outlining how our results relate to the interviewees of the case organization.

Conclusion

Our research was motivated by the lack of empirical studies on establishing goals in large-scale agile software development. Therefore, we conducted a single case-embedded study at a large German car manufacturer to identify goal-setting practices, established goals, and encountered goal-setting challenges. Moreover, we presented mitigation propositions and evaluated them. Within the case organization the GMP, a mandatory and standardized process to manage goals, is implemented. Apart from the GMP, goal-setting and documentation techniques vary between the Programs. In total, we identified 51 goals. System stability and legal compliance are the most common goals. Most of the remaining goals are specific to their programs. Moreover, most goals are either process or product goals, and mostly set at portfolio and program level. The most mentioned goal-setting challenges are external dependencies limiting autonomous goal-setting, prioritization conflicts between goals, and management control limiting team autonomy. In general, the interviewees agreed with most of our presented mitigation propositions. Regarding M5 and M7 there was no clear consent among the evaluation participants. Further research could validate this study's findings and test their applicability in other organizations. For instance, additional interviews could be conducted to confirm and rank the identified challenges or to investigate the effectiveness of the reported goal-setting practices. Moreover, we propose investigating how goal establishment in large-scale agile software development can be supported with metrics.

References

Anton, A. I. 1997. "Goal Identification and Refinement in the Specification of Software-Based Information Systems." Georgia Institute of Technology.

Basili, V. R., Caldiera, G., and Rombach, H. D. 1994. "The Goal Question Metric Approach," in: *Encyclopedia of software engineering*. pp. 528-532.

Basili, V. R., Lindvall, M., Regardie, M., Seaman, C., Heidrich, J., Münch, J., Rombach, D., and Trendowicz, A. 2010. "Linking Software Development and Business Strategy through Measurement," *Computer* (43:4), pp. 57-65.

Bateman, T. S., O'Neill, H., and Kenworthy-U'Ren, A. 2002. "A Hierarchical Taxonomy of Top Managers' Goals," *Journal of Applied Psychology* (87:6), pp. 1134-1148.

Berntzen, M., Moe, N. B., and Stray, V. 2019. "The Product Owner in Large-Scale Agile: An Empirical Study through the Lens of Relational Coordination Theory," *Proceedings of the 20th International Conference on Agile Software Development (XP 2019),* P. Kruchten, S. Fraser and F.o. Coallier (eds.), Montreal, QC, Canada: Springer, Cham, pp. 121-136.

Berntzen, M., Stray, V., and Moe, N. B. 2021. "Coordination Strategies: Managing Inter-Team Coordination Challenges in Large-Scale Agile," *Proceedings of the 22nd International Conference on Agile Software Development (XP 2021),* P. Gregory and P. Kruchten (eds.), Virtual Event: Springer, Cham, pp. 140-156.

Brown, A. W., Ambler, S., and Royce, W. 2013. "Agility at Scale: Economic Governance, Measured Improvement, and Disciplined Delivery," *Proceedings of the 35th International Conference on Software Engineering (ICSE)*, San Francisco, CA, USA: IEEE, pp. 873-881.

Digital.ai. 2019. "13th Annual of State of Agile Report." Retrieved 10. September, 2019, from https://www.stateofagile.com/#ufh-c-473508-state-of-agile-report

Digital.ai. 2020. "14th Annual of State of Agile Report." Retrieved 29. January, 2022, from https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf

Dikert, K., Paasivaara, M., and Lassenius, C. 2016. "Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review," *Journal of Systems and Software* (119), pp. 87-108.

Dingsøyr, T., and Moe, N. B. 2014. "Towards Principles of Large-Scale Agile Development," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen (eds.), Rome, Italy: Springer, Cham, pp. 1-8.

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B. 2012. "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," *Journal of Systems and Software* (85:6), pp. 1213-1221.

Doran, G. T. 1981. "There's a S.M.A.R.T. Way to Write Management's Goals and Objectives," *Management Review* (70:11), pp. 35-36.

Dreesen, T., Diegmann, P., and Rosenkranz, C. 2020. "The Impact of Modes, Styles, and Congruence of Control on Agile Teams: Insights from a Multiple Case Study," *Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS)*, Grand Wailea, Maui, Hawaii, USA, pp. 6247-6256.

Dumitriu, F., Meşniţă, G., and Radu, L.-D. 2019. "Challenges and Solutions of Applying Large-Scale Agile at Organizational Level," *Informatica Economica* (23:3), pp. 61-71.

Horlach, B., Schirmer, I., and Drews, P. 2019. "Agile Portfolio Management: Design Goals and Principles," *Proceedings of the 27th European Conference on Information Systems (ECIS2019)*, Stockholm-Uppsala, Sweden.

Kettunen, P., and Laanti, M. 2017. "Future Software Organizations - Agile Goals and Roles," *European Journal of Futures Research* (5:16), pp. 1-15.

Khurum, M., Gorschek, T., and Wilson, M. 2013. "The Software Value Map - an Exhaustive Collection of Value Aspects for the Development of Software Intensive Products," *Journal of Software: Evolution and Process* (25:7), pp. 711-741.

Korhonen, K. 2013. "Evaluating the Impact of an Agile Transformation: A Longitudinal Case Study in a Distributed Context," *Software Quality Journal* (21), pp. 599-624.

Korpivaara, I., Tuunanen, T., and Seppänen, V. 2021. "Performance Measurement in Scaled Agile Organizations," *Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS)*, Kauai, Hawaii, USA, pp. 6912-6921.

Kruchten, P. 2013. "Contextualizing Agile Software Development," *Journal of Software: Evolution and Process* (25:4), pp. 351-361.

Kumar, G., and Bhatia, P. K. 2012. "Impact of Agile Methodology on Software Development Process," *International Journal of Computer Technology and Electronics Engineering (IJCTEE)* (2:4), pp. 46-50.

Lappi, T., Karvonen, T., Lwakatare, L. E., Aaltonen, K., and Kuvaja, P. 2018. "Toward an Improved Understanding of Agile Project Governance: A Systematic Literature Review," *Project Management Journal* (49:6), pp. 39-63.

Larman, C., and Vodde, B. 2016. *Large-Scale Scrum: More with Less*. Boston, MA: Addison-Wesley Professional.

Likert, R. 1932. "A Technique for the Measurement of Attitudes," in: *Archives of psychology*.

Locke, E. A., and Latham, G. P. 1990. *A Theory of Goal Setting & Task Performance*. Prentice-Hall, Inc.

Locke, E. A., and Latham, G. P. 2006. "New Directions in Goal-Setting Theory," *Current Directions in Psychological Science* (15:5), pp. 265-268.
Miles, M. B., Huberman, A. M., and Saldaña, J. 2013. *Qualitative Data Analysis. A Methods Sourcebook*, (3 ed.). Los Angeles, USA: SAGE Publications.
Moe, N. B., Dahl, B. H., Stray, V., Karlsen, L. S., and Schjødt-Osmo, S. 2019. "Team Autonomy in Large-Scale Agile," *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS)*, Grand Wailea, Maui, Hawaii, USA, pp. 6997-7006.
Murphy, T., and Cormican, K. 2015. "Towards Holistic Goal Centered Performance Management in Software Development: Lessons from a Best Practice Analysis," *International Journal of Information Systems and Project Management* (3:4), pp. 23-36.
Myers, M. D., and Newman, M. 2007. "The Qualitative Interview in Is Research: Examining the Craft," *Information and Organization* (17:1), pp. 2-26.
Nord, R. L., Ozkaya, I., and Kruchten, P. 2014. "Agile in Distress: Architecture to the Rescue," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen (eds.), Rome, Italy: Springer, Cham, pp. 43-57.
Nyrud, H., and Stray, V. 2017. "Inter-Team Coordination Mechanisms in Large-Scale Agile," *Proceedings of the XP2017 Scientific Workshops*, Cologne, Germany: ACM, pp. 1-6.
Orlikowski, W. J. 1996. "Improvising Organizational Transformation over Time: A Situated Change Perspective," *Information systems research* (7:1), pp. 63-92.
Power, K. 2014. "Definition of Ready: An Experience Report from Teams at Cisco," *Proceedings of the 15th International Conference on Agile Software Development (XP 2014),* G. Cantone and M. Marchesi (eds.), Rome, Italy: Springer, Cham, pp. 312-319.
Regev, G., and Wegmann, A. 2005. "Where Do Goals Come From: The Underlying Principles of Goal-Oriented Requirement Engineering," *Proceedings of the 13th International Conference on Requirements Engineering (RE'05)*, Paris, France: IEEE, pp. 353-362.
Runeson, P., and Höst, M. 2009. "Guidelines for Conducting and Reporting Case Study Research in Software Engineering," *Empirical Software Engineering* (14), pp. 131-164.
Scaled Agile Inc. 2022. "Safe 5." Retrieved 13. March, 2021, from https://www.scaledagileframework.com
Schnabel, I., and Pizka, M. 2006. "Goal-Driven Software Development," *30th Annual IEEE/NASA Software Engineering Workshop*, Columbia, MD, USA: IEEE, pp. 59-65.
Schwaber, K. 2018. "The Nexus Guide." Retrieved 11. November, 2021, from https://www.scrum.org/resources/nexus-guide
Schwaber, K., and Sutherland, J. 2020. "The 2020 Scrum Guide." Retrieved 11. November, 2021, from https://scrumguides.org/scrum-guide.html
Stettina, C. J., and Schoemaker, L. 2018. "Reporting in Agile Portfolio Management: Routines, Metrics and Artefacts to Maintain an Effective Oversight," *Proceedings of the 19th International Conference on Agile Software Development (XP 2018),* J. Garbajosa, X. Wang and A. Aguiar (eds.), Porto, Protugal: Springer, Cham, pp. 199-215.
Stray, V., Moe, N. B., Vedal, H., and Berntzen, M. 2022. "Using Objectives and Key Results (Okrs) and Slack: A Case Study of Coordination in Large-Scale Distributed Agile," *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS)*, Virtual Event.
Talby, D., and Dubinsky, Y. 2009. "Governance of an Agile Software Project," *Proceedings of the ICSE Workshop on Software Development Governance*, Vancouver, BC, Canada: IEEE, pp. 40-45.
Uludağ, Ö., Kleehaus, M., Caprano, C., and Matthes, F. 2018. "Identifying and Structuring Challenges in Large-Scale Agile Development Based on a Structured Literature Review," *Proceedings of the 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, Stockholm, Sweden: IEEE, pp. 191-197.
Van Lamsweerde, A. 2001. "Goal-Oriented Requirements Engineering: A Guided Tour," *Proceedings of the 5th International Symposium on Requirements Engineering*, Toronto, ON, Canada: IEEE, pp. 249-262.
Van Oosterhout, M., Waarts, E., and van Hillegersberg, J. 2006. "Change Factors Requiring Agility and Implications for It," *European Journal of Information Systems* (15:2), pp. 132-145.
Vedal, H., Stray, V., Berntzen, M., and Moe, N. B. 2021. "Managing Dependencies in Large-Scale Agile," *Proceedings of the 22nd International Conference on Agile Software Development (XP 2021),* P. Gregory and P. Kruchten (eds.), Virtual Event: Springer, Cham, pp. 52-61.

Council Policy Manual
Association for Information Systems
(Adopted: June 8, 2014; revisions through September 2021)

Table of Contents

Contents

Part 1: Introduction and Administration

This Council Policy Manual (CPM) contains the current policies adopted by Council of the Association for Information Systems since the initial approval of the CPM on June 8, 2014.

1.1 Purpose

The purpose of the Council Policy Manual is to:

a. Collect all policies of the association in one place.

b. Provide for efficient orientation of new Council members.

c. Eliminate redundant or conflicting policies.

d. Simplify the review of current policy when considering new issues.

e. Provide clear, proactive policies to guide the Executive Director and staff.

1.2 Consistency

Each policy in the Council Policy Manual is expected to be consistent with the law, the articles of incorporation, and the bylaws of AIS, all of which have precedence over these policies. All policies shall be included or referred to in this document.

1.3 Development of Procedures

The Executive Director is responsible for developing organizational and administrative procedures that are consistent with the Council Policy Manual.

1.4 Transition

Whether adopted in part or as a complete document, upon approval by Council, any policy is deemed to supersede any past policy unless a prior Council resolution or contract obligates AIS with regard to the specific matter. Actual or apparent conflicts that arise between the Council Policy Manual and other policies or Council resolutions shall be

resolved by the Executive Committee. In all cases, the Bylaws shall be the primary governing document and shall supersede any policies contained in the Council Policy Manual, or other AIS documents. The Council Policy Manual shall be considered the
secondary governing document and shall supersede policies, procedures, or guidelines contained in other AIS documents.

1.5 Amendments

These policies should be reviewed and refined as needed. Amendments to these policies may only be made by Council, meeting in official session and in accordance with quorum requirements outlined in the Bylaws. Proposed amendments may be submitted by any Council member. In most cases, proposed amendments shall be reviewed by the appropriate committee before being presented to Council for action. Whenever amendments are adopted, the new policies shall be dated and immediately made available to Council and staff by the Council Secretary or designee.

1.6 Oversight

Oversight of the Council Policy Manual is the responsibility of the assigned committee or the full Council, as determined by the President. Specific committees shall have primary responsibility for reviewing specific policies or portions of the Council Policy Manual, drafting revisions as appropriate, and submitting proposed revisions to the full Council for approval.

1.7 Maintenance

The Executive Director shall ensure the proper recording and publishing of all policies. The Executive Director, upon receipt from Council Secretary of approved policy amendments, shall maintain the Council Policy Manual and provide updates to Council whenever changes occur or upon request. As appropriate, legal counsel may be sought to review these policies to ensure compliance with the law.

Part 2: Guiding Principles

2.1 Mission (revised 12/2010)

The Association for Information Systems (AIS) serves society through the advancement of knowledge and the promotion of excellence in the practice and study of information systems. AIS is the premier professional association for individuals and organizations who lead the research, teaching, practice, and study of information systems worldwide.

2.2 Strategic Goals (revised 12/2010)

The strategic goals of AIS are to:

a. Promote AIS as a global leader for excellence in information systems research, practice, and education.

b. Position information systems as a leading profession in the service of society.

c. Lead and promote excellence in information systems education.

d. Lead and promote excellence in information systems scholarship.

e. Cultivate a community by providing services and products to meet the diverse needs of members and related communities.

Part 3: Council Structure and Function

3.1 Governing Style

Council will approach its task with a style that emphasizes outward vision rather than an internal preoccupation, encouragement of diversity in viewpoints, strategic leadership more than administrative detail, clear distinction of Council and staff roles, and proactivity rather than reactivity. In this spirit, Council will:

a. Enforce upon itself and its members whatever discipline is needed. Discipline shall apply to matters such as attendance, respect for roles, speaking with one voice, and self-policing of any tendency to stray from the governance structure and processes adopted in these policies.

b. Be accountable to its members for competent, conscientious, and effective accomplishment of its obligations as a body. It will allow no officer, individual, or committee to usurp this role or hinder this commitment.

c. Monitor and regularly discuss Council's own processes and performance, seeking to ensure the continuity of its governance function by proper selection, orientation, training, and evaluation of council members.

d. Be an initiator of policy, not merely a reactor to staff initiatives. Council, not staff, will be responsible for Council performance.

3.2 Council Job Description

In addition to those duties outlined in the Bylaws, the job of Council is to lead AIS toward its stated mission and established goals. To perform its job, Council shall:

a. Determine the mission, vision, values, strategies, and major goals, and hold the Executive Director accountable for developing a staff strategic plan, in accordance with these policies, to accomplish these goals.

b. Determine the parameters within which the Executive Director is expected to achieve the goals.

c. Monitor the performance of AIS relative to the achievement of the goals.

d. Maintain and constantly improve the policies of Council as outlined in this Council Policy Manual.

e. Select, fairly compensate, evaluate annually, and if necessary, terminate the Executive Director.

f. Ensure financial solvency and integrity through policies and behavior.

g. Require periodic financial and other external audits to ensure compliance with the law and appropriate financial practices.

h. Evaluate and constantly improve Council's performance as the governing board and set expectations for Council members' involvement as volunteers.

3.3 Council Meetings

3.3.1 Council Election Do's and Don'ts

As nominees prepare for their AIS Council campaign, please keep in mind the following Do's and Don'ts during the annual AIS Council Elections.

SOME DO'S

- Do follow AIS Election Bylaws #7 Nominations and Elections
  - Regarding the Frequency (7.1)
  - Nominating Committee (7.2)
  - Nominating Procedure (7.3)
  - Balloting and the Announcement of Results (7.4)
  - Supervisory Responsibility for Elections (7.5)
- Do remain respectful, professional, dignified while communicating about the candidates and the elections, especially with social media posts.
- Do encourage the review of Candidate's Background, Experiences and Platform statements in the election ballet. All candidates are provided equal space, equal access, equal time on the election ballet.
- Do accept that all members of AIS Office Staff should remain unbiased and neutral in all public settings during the annual election voting window.
- Do accept that AIS Council Members should remain unbiased and neutral in public settings during the annual election voting window.
- Candidates may use AISWorld Listserv for their own campaign messages, but this should be limited to 2 messages per candidate per Election season.

SIGs, Chapter Leaders, AIS Members can express their support of a candidate(s), as long as it is done in a professional and respectful manner, this includes (but is not limited to) reminding people they are not speaking on behalf of everyone in their SIG / Chapter.

SOME DON'TS

- Do not use the AIS Office eMail list(s) nor the AIS Office social media platforms to lobby / campaign for a specific candidate.
- Do not use the AISNet.org nor the AIS eLibrary to lobby / campaign for a specific candidate.
- Do not publicly advocate against candidate(s). Rather, advocate for candidates, remain respectful, professional, dignified while communicating about the candidates and the elections, especially with social media posts
- Do not conduct public debates nor public forums with a subset of candidates for particular position(s). Please recall that candidates are private individuals with fulltime Academic careers who are volunteering to serve on AIS Council.

3.3.2 Active Participation in Council Meetings (adapted 3/2021)

As a current (or prospective) AIS council member, you will be involved with many facets of the Association. The following are a few of the responsibilities to keep in mind as you consider your decision to accept the nomination for your position on Council.

Meetings:
Participation in all four of the regularly scheduled council meetings during elected term, travelling twice a year, is required. For example:

Council Meeting Dates and Locations:
September – Virtual Council Meeting (exact date TBD)
December – Council Meeting- (in person in conjunction with ICIS)
March – Virtual Council Meeting (exact date TBD)
June – Council Meeting- (exact date and location TBD)

Committees:
Council members serve on various committees as determined by the elected position. Each committee has an assigned AIS Staff Liaison. For more information of each committee please reference the links below.

President-Elect
LEO Committee
Fellow Committee
Executive Committee
Nominating Committee
Finance Committee

VP of Conferences and Meetings
Conference Committee

VP of Education
Education Committee

VP of Publications
Publications Committee

VP of SIGs, Chapters and Colleges
Communities Committee

Region 1, 2 and 3 Reps
Finance Committee

VP of Membership
Membership Committee

Communication:
Develop regular channels of communication with council members, committee members and AIS staff liaison. Encourage participation of AIS members on appropriate committees. AIS Council operates on US Eastern Standard Time. Virtual Council meeting are typically scheduled to begin at 9am US EST.

Policies, Objectives, and Plans:
Abide by and become familiar with all bylaws and policies governing the operation of the association.

AIS Bylaws
Council Policy Manual

3.4 Advisory Groups and Task Forces

To increase its knowledge base and depth of available expertise, the Council supports the use of groups or task forces of qualified advisers. The term "task force" refers to any group appointed by a Council member, upon approval of the President, to assist in carrying out various time-limited goals and responsibilities. Although any Council member may form a task force, s/he shall notify the Council of its formation, purpose, and membership within ten days of its approval. The Executive Director may assign a staff member to serve advisory groups and task forces.

3.5 Council Members' Code of Conduct

Council expects of itself and its members ethical and businesslike conduct. Council members must present themselves and conduct business in a manner that reflects positively on AIS, its mission, and its members.

3.5.1 Conflict of Interest (adopted 11/2014)

A. Purpose
The purpose of the conflict-of-interest policy is to protect AIS' interest when it is contemplating entering into a transaction or arrangement that might benefit the private interest of an employee, contractor, officer or Council member of AIS or might result in a possible excess benefit transaction.

B. Definitions

1. Interested Person: Any employee, contractor, Council member, officer, or member of a committee with Council delegated powers, who has a direct or indirect financial, academic, or programmatic interest, as defined below, is an interested person.

2. Financial Interest: A person has a financial interest if the person has, directly or indirectly, through business, investment, or family:

    a. An ownership or investment interest in any entity with which AIS has a transaction or arrangement,

    b. A compensation arrangement with AIS or with any entity or individual with which AIS has a transaction or arrangement, or

    c. A potential ownership or investment interest in, or compensation arrangement with, any entity or individual with which AIS is negotiating a transaction or arrangement. Compensation includes direct and indirect remuneration as well as gifts or favors that are not insubstantial. A financial interest is not necessarily a conflict of interest. Under Procedures, Item 2 of this policy, a person who has a financial interest may have a conflict of interest only if Council or the appropriate committee decides that a conflict of interest exists.

3. Academic or Programmatic Interest: A person has an academic or programmatic interest if the person has, directly or indirectly, leadership or other intensive interests that might give rise to substantial time commitments outside of AIS. The following are examples of academic or programmatic interests that can give rise to conflict:

    a. Edit, control, or otherwise advance the mission of a journal or publication owned by AIS while also advancing the financial interest of non-AIS owned entities,

    b. Sponsor, manage, advise, or otherwise promote a conference, meeting, or assembly that is owned by AIS while also advancing the financial interest of non-AIS owned entities.

C. Procedures

1. In connection with any actual or possible conflict of interest, an interested person must disclose the existence of a conflict of interest and be given the opportunity to disclose all material facts to the directors and members of committees with Council delegated powers to consider the proposed transaction or arrangement.

2. After disclosure of the conflict of interest and all material facts, and after any discussion with the interested person, he/she shall leave the Council or committee meeting while the determination of a conflict of interest is discussed and voted upon. The remaining Council or committee members shall decide if a conflict of interest exists.

3. The chairperson of Council or the committee shall, if appropriate, appoint a disinterested person or committee to investigate alternatives to the proposed transaction or arrangement.

4. After exercising due diligence, Council or the committee shall determine whether AIS can obtain with reasonable efforts a more advantageous transaction or arrangement from a person or entity that would not give rise to a conflict of interest.

5. If a more advantageous transaction or arrangement is not reasonably possible under circumstances not producing a conflict of interest, Council or the committee shall determine by a majority vote of the disinterested Council members whether the transaction or arrangement is in AIS' best interest, for its own benefit, and whether it is fair and reasonable, then make its decision as to whether to enter into the transaction or arrangement.

6. If Council or the committee has reasonable cause to believe an individual has failed to disclose actual or possible conflicts of interest, it shall inform the individual of the basis for such belief and afford the member an opportunity to explain the alleged failure to disclose.

7. If, after hearing the individual's response and after making further investigation as warranted by the circumstances, Council or the committee determines the individual has failed to disclose an actual or possible conflict of interest, it shall take appropriate disciplinary and corrective action.

D. Compliance

1. The minutes of the Council and all committee meetings with Council-delegated powers shall contain:

    a. The names of the persons who disclosed or otherwise were found to have an actual or possible conflict of interest, the nature of the conflict of interest, any action taken to determine whether a conflict of interest was present, and the Council's or committee's decision as to whether a conflict of interest in fact existed.

    b. The names of the persons who were present for discussions and votes relating to the transaction or arrangement, whether or not any alternatives to the proposed transaction or arrangement were considered, and a record of any votes taken in connection with the proceedings.

2. A voting member of Council who receives compensation, directly or indirectly, from AIS for services is precluded from voting on matters pertaining to that member's compensation.

3. Each Council member, officer, employee, contractor, and member of a committee with Council-delegated powers shall annually sign a statement (see Addendum for form) which affirms that he/she:

    a. Has received a copy of the conflicts of interest policy,
    b. Has read and understands the policy, and
    c. Has agreed to comply with the policy.

    4. The signed Conflict of Interest statement shall be kept on file in the AIS office.

Part 4: Board/Staff Relationship

4.1 Delegation to the Executive Director

While the Council's job is generally focused on establishing policies and strategy, implementation and procedure development are delegated to the Executive Director as established in the bylaws.

a. All Council authority delegated to staff is delegated through the Executive Director, as established in the Bylaws, so that all authority and accountability

of staff, as far as Council is concerned, is considered to be the authority and accountability of the Executive Director.

b. Guiding Principles policies (Part 2) guide the Executive Director on the goals of the organization. Operational Parameters policies (Part 5) define the acceptable boundaries of prudence and ethics within which the Executive Director is expected to operate. The Executive Director is authorized to establish procedures, make all decisions, take all actions, and develop all activities as long as they are consistent with any reasonable interpretation of Council's policies as contained in this Council Policy Manual and AIS Bylaws.

c. Council may change its policies during any meeting, thereby shifting the boundary between Council and Executive Director domains. Consequently, Council may change the latitude of choice given to the Executive Director, but so long as any particular delegation is in place, Council and its members will respect and support the Executive Director's choices.

4.2 Executive Director Job Description

The Executive Director's job contributions can be stated as performance in two areas: (1) accomplishment of the major organizational goals as stated in Policy 2.2; and (2) AIS operations within the boundaries of prudence and ethics established in Council policies on Operational Parameters (Part 5). (Job description located in Addendum.)

4.3 Communication and Counsel

With respect to providing information and advice to Council, the Executive Director shall keep Council informed about matters essential to carrying out its governance duties. Accordingly, the Executive Director shall:

a. Inform Council of relevant trends, anticipated adverse media coverage, and material external and internal changes, and particularly changes in the assumptions upon which any Council policy has been previously established, always presenting information in a clear and concise format.

b. Communicate with Council as a whole except when fulfilling reasonable individual requests for information or responding to officers or committees as established or assigned by Council.

c. Report immediately any actual or anticipated material noncompliance with a policy of Council, along with suggested changes.

Part 5: Operations

5.1 General Guidelines

Council expects that AIS members, officers, and staff will always act in an ethical and prudent manner as provided in the bylaws in regard to the operation of the organization.

5.2 Financial

5.2.1 Reserves and Investment (adopted 06/2004; revised 06/2008; revised 05/2015)

A. Purpose

The purpose of the Reserves and Investment Policy is to provide guidelines for the prudent management of AIS funds to meet the financial objectives outlined in the Bylaws.

B. Procedures

1. Council has the authority and responsibility to manage the affairs of the corporation and shall have supervision, control, and direction over the assets and investment policies of the organization.
    a) AIS Council approval is needed to set aside net assets into an endowment or reserve fund. Pursuant to Bylaw 15.4 Endowment and Reserve Funds.

2. The AIS Executive Committee recommends investment policies to Council for approval.

3. The AIS staff and contractors have authority and responsibility to implement, oversee, and report on investments managed according to approved policies.

4. Funds will be invested to meet the following objectives.

    a. AIS intends to keep in cash reserves an amount equal to no less than six (6) months of average operating expenses. AIS intends to keep in total reserves, including cash and cash equivalents, an amount equal to no less than twelve (12) months of average operating expenses. (rev.03/15)

    b. AIS intends for reserve funds to earn a reasonable return with a conservative level of risk.

c. AIS intends for at least 50% of reserve funds to be liquid and available if needed for operations.

d. AIS intends for reserve funds to be invested with an allocation target of 65% fixed income and 35% equity.

e. Of the equity portion, 30% shall be in the 500 Index, 35% shall be in the mid-cap index, and 35% shall be in the small-cap index.

f. The fixed or liquid portion of the portfolio is expected to have an annual yield that meets or exceeds the Vanguard money market annual return when compared to the same time period.

g. The equity portion of the portfolio is expected to have an annual yield that meets or exceeds the S&P 500 annual return when compared to the same time period.

5. Funds will be invested according to the following parameters.

   a. Approved investment options include:

      1. Fixed income instruments such as CD's, T-bills, and money market accounts
      2. Bond mutual funds with a credit rating of AA or higher
      3. Equity mutual funds with a Morningstar rating of 4-stars or higher.

   b. No investments shall be made in the following: futures, options, margin accounts, private loans, private mortgages, individual stocks, individual options contracts, commodities, or commodity contracts

C. Compliance

1. AIS Council shall be provided with quarterly updates to include a comprehensive profit and loss statement and balance sheet. The AIS Executive Committee shall be provided with this update monthly. (rev. 3/15)

2. If reserves drop to less than four (4) months operating expenses, AIS Council must be notified and AIS Council must then approve any future withdrawals from reserves.

3. AIS Council shall be provided updates from the AIS Treasurer of the association's compliance with AIS Financial Reserve Policies as noted in

5.2.1 Reserves and Investment twice a year during the face to face AIS Council meetings. (rev. 3/15)

5.2.2. Council Travel Reimbursement Policy (adopted 09/2014)

A. Purpose

It is the policy of AIS to reimburse the reasonable expenses of members of council (or designee) for official, approved AIS business travel and meetings as provided for in the AIS budget and/or by the President. The amount of reimbursement may change from year-to-year based on the resources available and budgeted. When incurring business expense, Council members are expected to:

a. Exercise discretion and good business judgment with respect to those expenses.

b. Be cost conscious and use the lowest options available.

c. Provide proof of purchase (receipts) for all expenses. (Receipts are required for all expenditures billed directly to the individual, such as airfare and hotel charges.)

B. Reimbursement Parameters

1. Transportation:

    a. Airfare: Reimbursement for economy class (21 day advanced-purchase lowest available) airfare will be provided. General air travel reservations should be made as far in advance as possible in order to take advantage of reduced fares. Individuals may not deliberately patronize a single airline to accumulate frequent flyer miles if less expensive comparable tickets are available on another airline.

    b. Train: Reimbursement for economy class train fare will be provided.

    c. Automobile: Reimbursement for mileage (per the IRS rate) or car rental will be provided.

2. Lodging: Reimbursement for hotel accommodations (selected by AIS) will be provided.

3. Meals: Reimbursement for meal expenses related to travel to/from/during the event will be provided based on current IRS per diem rates.

4. Ground Transportation: Council members are expected to use the most economical ground transportation available.

5. Other Miscellaneous Expenses: Reimbursement for any additional travel expenses such as Wi-Fi, photocopies, and telephone will be provided.

6. The maximum annual amount permitted by council members is $1,250 unless otherwise budgeted in any given year or approved by the President. (rev. 12/14)

7. Council members need to use the AIS Expense Reimbursement Form (see attached).

8. Expense reimbursement requests must be received by the AIS Office no later than 60 days after the date the travel is completed. Expense reimbursement requests received after 60 days from the completion of travel will be denied and payment will not be made. (Please note: The rationale for Parameter 8 is to better insure AIS's compliance with IRS publication 535, Chapter 11 and the reporting of taxable income.) (rev. 12/14)

5.3 Programs

5.3.1 Award Programs

5.3.1.1 Distinguished Member Memorial Award (adopted 06/2006; revised 12/2006; revised 12/2015; revised 03/2016; revised 06/2019)

A. Purpose
The Distinguished Member Memorial Award, established by AIS Council in 2006, honors members of the AIS community who have made significant contributions to the field of information systems but, unfortunately, died before they could be considered for an AIS Fellow or AIS LEO Award.

B. Procedure

1. The Distinguished Member Memorial Award is given to an individual who:

   a. died during the previous calendar year;

   b. was a member of AIS for at least five (5) years;

   c. is generally considered to have advanced the field through research or service; and

   d. is not a LEO Award winner or AIS Fellow.

2. The Distinguished Member Memorial Award committee is comprised of (4) members: AIS President-Elect (who acts as Chair), AIS Immediate Past-President, one LEO awardee and one AIS Fellow as established in the Bylaws.

3. There is no requirement for the committee to make an annual award but rather it may make one or more awards when the stated criteria are satisfied.

4. Each year in which an award is made, the award is announced by the President of AIS in a letter to the deceased's family, in an email to all AIS members, and by announcement on the AIS website home page.

5. A short synopsis of the individual's career (and photo if available) will be published as appropriate.

5.3.1.2 LEO Award (adopted 06/1999; revised 12/2005; revised 12/2015; revised 12/2016; revised 9/2021)

A. Purpose

The LEO Award, established by the AIS Council in 1999, and named for one of the world's first commercial applications of computing (The Lyons Electronic Office), recognizes truly outstanding individuals in the field of information systems.

B. Procedure

1. All recipients of the LEO Award are outstanding scholars or practitioners who have made a global impact on the field of information systems. Qualified candidates

   a. are involved on the local level in the field of information systems;

   b. act as role models to colleagues and students;

   c. are well regarded by their peers as well as those outside of the field for their professional work and personal integrity; and

   d. are not required to be members of AIS.

2. The LEO Award cannot be given posthumously.

3. The LEO Award Committee shall be formed by July 1 of each year and is comprised of AIS members as established in the Bylaws.

4. By September 1 of each year, the Chairperson of the AIS LEO Committee will issue a widespread call for nominations to AIS membership. A nomination must include both the name of the nominee and the basis for the nomination, and candidates may not be self-nominated. Nominations will close on October 7. Nominations will be valid for three (3) years.

5. The LEO Award Committee will determine the recipients of the award by October 15 of each year.

6. The Chairperson of the LEO Award Committee will notify, in confidence, those colleagues who have been chosen for the award. In addition, the Chairperson will notify, in confidence, members of the subcommittee of AIS Council of the outcome of their deliberations.

7. The Chairperson of the LEO Awards Committee will present the recipients with suitable recognition at a ceremony held at ICIS.

8. The names of recipients will also be published as appropriate.

9. LEO recipients will be granted an honorary Lifetime Membership in AIS.

10. When appropriate, there may be additional forms of recognition.

11. Nominations may be submitted using the online nomination form.

5.3.1.3 AIS Fellows (adopted 06/1999; revised 12/2005; revised 12/2015; revised 9/2021)

A. Purpose

The AIS Fellow award, which was established by the Council in 1999, recognizes individuals who have made outstanding research, teaching, and/or service contributions to the field of information systems.

B. Procedures

1. Qualified AIS Fellow candidates:

    a. are expected to have made significant global contributions to the discipline as well as outstanding local contributions in the context of their country and region;
    b. are expected to be role models to colleagues and students within the discipline;

    c. garner the respect of individuals from outside the discipline; and

    d. should be esteemed for their high levels of professional and personal integrity.

2. The AIS Fellows Committee shall be formed by July 1 of each year and is comprised of AIS members at outlined in the Bylaws.

3. By September 1 of each year, the Chairperson of the AIS Fellows Committee will issue a widespread call for nominations to AIS Fellow. A nomination must include both the name of the nominee and the basis for the nomination, and candidates may not be self-nominated. Nominations will close on October 7.

4. The AIS Fellow Committee will determine the recipients of the award by October 15 of each year.

5. Nominations may be submitted using the online nomination form.

6. No more than eight (8) fellows may be appointed across the organization, with a minimum of one (1) per AIS region, but no more than four (4) per AIS region. Nominations will be valid for three (3) years.

7. AIS Fellows must be current members of AIS and must remain members until they retire from full-time work.

8. In electing fellows, the AIS Fellows Committee will consider the nominations that they receive. In addition, members of the committee may consider other colleagues whom they believe are deserving of the award.

9. The chairperson of the committee will notify those colleagues who have been chosen for the award. In addition, the chairperson will notify the nominators of the outcome of their nominations.

10. The committee's deliberations will be confidential to the members of the committee. The committee is not obligated to make its deliberations available to subsequent AIS Fellows Committees. Members shall not enter into correspondence in relation to their deliberations and decisions.

11. In any year, the committee may choose to elect no fellows. In the event of a tie among the six-member committee, the chairperson of the committee may cast the deciding vote.

12. New AIS Fellows will be inducted by the chairperson of the AIS Fellows Committee at a ceremony held at the International Conference on Information Systems. New Fellows will be presented with suitable recognition. Fellows are also entitled to use the letters "FAIS" after their name.

13. From time to time, the committee may decide that additional forms of recognition of fellows are also appropriate.

5.3.1.4 Doctoral Student Service Award (adopted 03/2014; revised 12/2015; revised 9/2021)

A. Purpose

The Doctoral Student Service Award, established by AIS Council in 2014, recognizes volunteer contributions made by doctoral students toward the success of AIS conferences, journals, and programs.

B. Procedures

1. The Doctoral Student Service Award is given to individuals who:
    a. Made exceptional volunteer contributions to one or more of the following programs.
        i. AIS Conferences, AMCIS or ICIS
        ii. AIS affiliated conferences of ECIS or PACIS
        iii. AIS Publications
        iv. AIS SIGs, Chapters, Colleges, or Student Chapters.

    b. Made contributions over and above typical student volunteer work during the academic year preceding that of the nomination year.

    c. Were doctoral students at the time of making the contributions but may have graduated at the time of award receipt.

    d. Are members of AIS, both at the time of service and at the time of recognition.

    e. Have not received this award previously.

2. The Doctoral Student Service Award committee shall be formed by July 1 of each year and shall consist of a chairperson and four members appointed by the Vice President of Member Services and Chapters. Committee members shall serve for no more than two years, with rotating terms. The committee shall include at least one member from each of the three AIS regions. All committee members must be AIS academic members in good standing.

3. By September 1 of each year, the committee chairperson will issue a wide-spread call for nominations. Nominations must be submitted using the online nomination form by October 7. Self-nominations will not be accepted.

4. The Doctoral Student Service Award Committee will determine the recipients of the award by October 15 of each year.

5. Award recipients will be presented with a plaque at the ICIS conference and listed on the AIS website. When appropriate, there may be additional forms of recognition.

6. There is no limit to the number of recipients who may receive the award in a given year.

5.3.1.5 Early Career Award (adopted 09/2014; revised 12/2015, revised 02/2018; revised 9/2021)

A. Purpose

The Early Career Award, established by AIS Council in 2014, recognizes individuals in the early stages of their careers who have already made outstanding research, teaching, and/or service contributions to the field of information systems.

B. Procedures

1. Qualified AIS Early Career Award candidates:
   a. at the time of nomination should:
      i. have received their doctoral degrees no more than seven (7) years prior to nomination; OR
      ii. be within the first seven (7) consecutive years of their academic career if they do not have a doctoral degree;

   b. have made global contributions to the discipline as well as local contributions in the context of their country or region;

   c. are role models to colleagues and students within the discipline;

   d. garner the respect of individuals from outside the discipline;

   e. are esteemed for their high levels of professional and personal integrity;

   f. have not received the AIS Early Career Award previously; and

   g. have been an AIS member for at least two (2) consecutive years and are current members of AIS at the time of nomination and receipt of recognition.

2. The AIS Early Career Award Committee shall be formed by July 1 of each year and shall consist of a chairperson and four members appointed by the Vice President of Member Services and Chapters. Committee

members shall serve for no more than two years, with rotating terms. The committee shall include at least one member from each of the three AIS regions. All committee members must be AIS academic members in good standing.

3. By September 1 of each year, the committee chairperson will issue a widespread call for nominations. Nominations may be submitted using the online nomination form by October 7.

4. The AIS Early Career Award Committee will determine the recipients of the award by October 15 of each year.

5. The committee's deliberations will be confidential to the members of the committee. The committee is not obligated to make its deliberations available to subsequent AIS Early Career Awards Committees. Committee members shall not enter into correspondence in relation to their deliberations and decisions.

6. A maximum of six (6) Early Career awards will be granted, with at least one (1) per AIS region.

7. Award recipients will be presented with a plaque at ICIS and listed on the AIS website. When appropriate, there may be additional forms of recognition.

5.3.1.6 Sandra Slaughter Service Award (adopted 12/2014; renamed the Sandra Slaughter Service Award 05/2015; revised 12/2015; revised 9/2021)

A. Purpose

The Sandra Slaughter Service Award, established by AIS Council in 2014, recognizes longstanding members who have provided leadership within the Association, particularly through such activities as participating in the SIGs/chapters/colleges, strengthening the conferences, and participating in AIS-sponsored journals.

B. Procedures

1. Qualified Sandra Slaughter Service Award candidates:
   a. have been AIS members for at least 3 years;

b. have not received the Sandra Slaughter Service Award within the five (5) years prior to nomination; and

c. have made significant contributions to the Association through activities such as (but not limited to) the following:
   a. Council (past; sitting Council members are not eligible to receive this award)
   b. Association committees
   c. Journal editorial boards
   d. Conference involvement
   e. SIGs, Chapters, or Colleges

2. The Sandra Slaughter Service Award Committee shall be formed by July 1 of each year and shall consist of a chairperson and four members appointed by the Vice President of Member Services and Chapters. Committee members shall serve for no more than two years, with rotating terms. The committee shall include at least one member from each of the three AIS regions. All committee members must be AIS academic members in good standing.

3. By September 1 of each year, the committee chairperson will issue a widespread call for nominations. Nominations may be submitted using the online nomination form by October 7. Self-nominations will be accepted.

4. The Sandra Slaughter Service Award Committee will determine the recipients of the award by October 15 of each year.

5. The committee's deliberations will be confidential to the members of the committee. The committee is not obligated to make its deliberations available to subsequent Sandra Slaughter Service Awards Committees. Committee members shall not enter into correspondence in relation to their deliberations and decisions.

6. A maximum of six (6) Sandra Slaughter Service Awards will be granted, with at least one (1) per AIS region.

7. Award recipients will be presented with a plaque at ICIS and listed on the AIS website. When appropriate, there may be additional forms of recognition.

5.3.1.7 AIS Outreach Publications Award (adopted 03/2015; revised 03/2017; revised 07/2018; revised 09/2021)

A. Purpose

The AIS Outreach Publication Award, established by AIS Council in 2015, recognizes members who successfully transfer research to practitioner audiences in practice-based publications. The publication can be a book in any medium in the public domain or an impactful online or offline publication other than a journal article or a report that serves as the primary basis for a journal article. The book or other practice-based publication selected for the award must be publicly available with evidence that it has influenced practitioner or decision makers in the public or private sectors.

B. Procedures

a. Qualified AIS Outreach Practice Publication Award candidates must:
   i. Be a member of AIS
   ii. May not have received the AIS Outreach Practice Publication Award within the five (5) years prior to the nomination
   iii. Have made significant contribution to the practitioner audience in practice-based publications through research

b. The AIS Outreach Practice Publication Award Committee shall be formed by July 1 of each year and shall consist of a chairperson and four members appointed by the Vice President of Publications. Committee members shall serve for no more than two years, with rotating terms. The committee shall include at least one member from each of the three AIS regions. All committee members must be AIS academic members in good standing.

c. By September 1 of each year, the committee chair will issue a widespread call for nominations. Nominations may be submitted using the online nomination form by October 7. Neither nominations nor materials supporting a nomination can be obtained

from organizations (i.e., one or more of an organization's members) either funding the research on which the publication is based or actively/passively collaborating in this research as a source of data.

d. The AIS Outreach Publications Awards Award Committee will determine the recipients of the award by October 15 of each year.

e. The committee's deliberations will be confidential to the members of the committee. The committee is not obligated to make its deliberations available to subsequent AIS Outreach Publications Awards Committees. Committee members shall not enter into correspondence in relation to their deliberations and decisions.

f. A maximum of six (6) AIS Outreach Publication Awards will be granted.

g. Award recipients will be presented with a plaque at ICIS and listed on the AIS website. When appropriate, there may be additional forms of recognition.

5.3.1.8 AIS Community Leadership Award (adopted 03/2015); renamed AIS Leadership Excellence Award (12/2016) (revised 09/2021)

A. Purpose

The AIS Leadership Excellence Award, established by AIS Council in 2015, recognizes a distinguish industry or external community professional for their contributions to the field of IS. The recipient will be chosen from candidates in the region of the current year's International Conference on Information Systems (ICIS). The recipient will be selected for their leadership and have a high reputation for the development, use, and/or application of IS in the private or public sector in the region. The scope of impact of the selected individual must at the minimum be of national (or international) level.

B. Procedures

a. Qualified AIS Leadership Excellence Award candidates must:
   i. Be from the region of the current year's International Conference on Information Systems (ICIS).

   ii. May not have received the AIS Leadership Excellence Award within the five (5) years prior to the nomination.

   iii. Have proven leadership and a high reputation for the development use and/or application of IS in the private or public sector in the region.

   iv. Impact the field of IS at the national or international level.

b. The AIS Leadership Excellence Award Committee shall be formed by July 1 of each year and shall consist of a chairperson and four members, as appointed by the Vice President of Communications. Committee members shall serve for no more than two years, with rotating terms. The committee shall include at least one member from each of the three AIS regions. All committee members must be AIS academic members in good standing.

c. By September 1 of each year, the committee chair will issue a widespread call for nominations. Nominations may be submitted using the online nomination form by October 7.

d. The AIS Leadership Excellence Award Committee will determine the recipients of the award by October 15 of each year.

e. The committee's deliberations will be confidential to the members of the committee. The committee is not obligated to make its deliberations available to subsequent AIS Leadership Excellence Award Committees. Committee members shall not enter into correspondence in relation to their deliberations and decisions.

f. A maximum of six (6) AIS Leadership Excellence Award will be granted.

g. Award recipients will be presented with a plaque at ICIS and listed on the AIS website. When appropriate, there may be additional forms of recognition.

5.3.1.9 AIS Distinguished Member Designation (adopted 03/2019)

A. Purpose

The AIS Distinguished Member Designations of Distinguished Member and Distinguished Member - Cum Laude was established by AIS Council in 2019. The designations honors members of the AIS community who have made significant contributions to the field of information systems as AIS members.

B. Procedure

6. The AIS Distinguished Member Designations is given to an individual who:

Distinguished Members will at a minimum have:

a. Continuous AIS membership for 5 years.
b. Significant role in AIS or AIS related activity (e.g., SIG, Chapter, Conference, Journal, and so on).
c. A publication in an AIS or AIS affiliated journal.

Distinguished Member – Cum Laude will at a minimum be:

a. Continuous AIS membership for 10 years
b. Leadership role in AIS or AIS related activity (e.g., SIG, Chapter, Conference, Journal, and so on).
c. Four publications in an AIS or AIS affiliated journal.

2. AIS members can self-nominate or be nominated for the new member designation and the VP, Membership and Chapters will make the final selections based on information provided.

C. Recognition

- Recognition on the AIS website.
- Designation on AIS membership card.
- A print quality certificate for tenure packets.
- A physical token (lapel pins as being the contingency.

5.3.1.10 AIS Impact Award (adopted 12/2020)

A. Purpose

The AIS Impact Award recognizes the impact of information systems research beyond academia. Impactful research can be a powerful individual motivator. Seeing that one's own research delivers societal value and helps business succeed is rewarding in its own right. Within and beyond our own community, multiple initiatives seek to encourage us to pursue research with societal value. Increasingly, however, impact is also a hallmark of research excellence in the view of accreditation, evaluation, and funding programs, making impact and individuals with impactful research a valuable resource to any institution.

B. Procedure

There is a two-step process for nominations. In the first step, a short nomination is required providing the following information:

- the name(s) of the person or set of collaborators responsible for the contribution
- a brief summary of the basis of impact (e.g. construct, model, method, software or other form of instantiation, etc.)
- a description of the societal/practitioner population that has been impacted by this research and what kind of impact achieved in this population the nature of the contribution made by the person or group being nominated reference(s) to the underpinning peer-reviewed research.

In the second step, the AIS Impact Award Committee will determine a short list of potential awardees by October 7. Shortlisted candidates will be asked to provide

additional information by November 1. This includes documentation that a contribution is widely used. Additional information will be requested to better understand the precise nature of an individual's contribution and the impact, including quantitative measures, testimonials, or other forms of evidence from benefactors of this research.

5.4 Publications

5.4.1 Copyright

A. Purpose

The purpose of the copyright policy is to protect the interests of both AIS and authors by defining ownership of scholarly work published by AIS and identifying the rights of authors. (Note: Publication of work in an AIS journal and publication of work as part of an AIS conference proceeding serve different purposes. Ownership of the material and the rights of both AIS and authors differ for journals and conference proceedings.)

B. Journal Copyrights

1. Authors are required to hold exclusive copyright of material submitted for publication in an AIS journal and to warrant that the work is not an infringement of any existing copyright, proprietary right, invasion of privacy or libel and will hold harmless AIS from any damages, expenses, and costs against any breach of such warranty.

2. Authors are required to transfer limited copyright for material published in an AIS journal to AIS. This limited copyright allows AIS to grant republication rights, without charge and without restriction, for any academic use as long as there is full acknowledgement of the original source of the work.

   Each republished journal article shall contain the following statement regarding the copyright of material.

   > *Copyright (date), by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to*

republish, to post on servers for commercial use, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712, Attn: Reprints, or via e-mail from: publications@aisnet.org

3. Republication, in whole or in part, for commercial purposes will be granted for a fee. The fee for each AIS published journal shall be established by Council.

4. The following rights are reserved by the author of articles published in AIS journals:

   a. The right to use, free of charge, all or part of the published material in future works of their own, such as books, lectures, and conference presentations, provided reference is given to the original AIS publication.

   b. The right to include a copy of the material on the author's web page provided reference is given to the original AIS publication, the author states that AIS owns limited copyright of the material and that use for commercial purposes is not allowed. The author version published on the web site must be identical to the final version published by AIS and include a link to the appropriate AIS journal/proceedings.

   c. The right to use the material for internal training or teaching purposes by the author's employer.

   d. The right to share material with colleagues for their research.

   e. The right to use the material in subsequent compilations of the author's work.

   f. The right to grant permission to include the material in a thesis or dissertation written by a student at an academic institution.

   g. The right to reuse portions or extracts of the material in other works by the author.

   h. The right to use the material in other derivative works by the author.

C. Conference Proceedings Copyrights

1. Authors are required to hold exclusive copyright of material submitted for publication in an AIS conference proceedings and to warrant that the work

is not an infringement of any existing copyright, proprietary right, invasion of privacy or libel and will hold harmless AIS from any damages, expenses, and costs against any breach of such warranty.

2. Authors retain copyright for material published as part of AIS conference proceedings. Submission of a paper to the conference represents the author's agreement to allow AIS a nonexclusive license to publish the paper in any written or electronic format for distribution to all interested parties in perpetuity with or without compensation to AIS and without compensation to the author.

D. Paper Acceptance – Conference Attendance (adopted 12/2009)

1. Authors of papers accepted for presentation at an AIS conference are required to have at least one author register for and attend the conference. In the case of papers presented by a panel, all panel presenters are required to register for and attend the conference. Failure to comply with this policy may result (at the discretion of the conference program chairperson/committee) in removal of the paper from the conference proceedings.

E. Indemnification for Article Submission

1. By submitting a manuscript to an AIS journal or conference, the author is required to acknowledge the following:

   a. AIS may, at its discretion, upload the article to any verification software, database, or other technology which may be used from time to time to verify the authenticity of the work.

   b. If the work is found or suspected to infringe copyright of third parties, AIS may investigate the full circumstances of the case. This may involve forwarding the submitted work to the actual, or suspected copyright owners or any other individual or entity AIS deems appropriate for review.

5.4.2 Creation of New Publications (adopted 12/2006)

A. Purpose

The purpose of this policy is to establish guidelines to govern new AIS journals to ensure these new publications are consistent with the mission of AIS and fit well with the existing suite of journals.

B. Procedures

1. To propose a new AIS journal, the following information should be submitted to the VP of Publications.

    a. The proposed name of the journal, which shall include “AIS” as part of the name.

    b. The names of six (6) AIS members proposing the new journal and the endorsement of at least one AIS SIG or Chapter

    c. The mission of the journal, the audience for the journal, and the types of IS topics that would best fit with the journal

    d. How the journal complements/extends the suite of current AIS journals and affiliated journals

    e. The journal’s language(s) of publication and its proposed publication frequency

    f. The journal’s publication format(s) - electronic, print, or both

    g. A list of at least two (2) proposed editors-in-chief and five (5) proposed senior editors, including bios and statements indicating they would be interested in serving on the editorial board

    h. A plan and timetable to gain ISI indexing of the journal

    i. A three (3) year budget, including revenues (such as fund-raising), in-kind contributions, and expenses

2. The VP of Publications will review the proposal with the AIS Publications Committee, and then, upon satisfactory review by the Committee, send it to Council for approval.

C. Compliance

1. If approved, new AIS journals will be required to:

   a. Maintain the same "look and feel" as other AIS journals.

   b. Work with designated staff to develop and use AIS submission and review systems to ensure reliable journal access for AIS.

   c. Maintain international representation on the editorial board.

   d. Use double-blind reviews.

   e. Submit annual progress reports and annual budgets.

   f. Select editors-in-chief by following AIS bylaws for selection.

   g. In consultation with the AIS Publications Committee, create an advisory board/council.

   h. Keep records in support of ISI indexing requirements.

2. The editor-in-chief will be a member of the AIS Publications Committee.

3. A new journal will be evaluated after three (3) years by the AIS Council to determine if it will continue as an AIS publication.

5.4.3 Scholarly Misconduct (adopted 10/2003; revised 10/2013, 06/2014)

A. Purpose

The purpose of the Scholarly Misconduct policy is to provide guidelines for investigating and adjudicating claims of plagiarism and other scholarly misconduct. This policy embodies the principle that every effort should be made to resolve disputes without resort to formal investigation. Moreover, it is not reasonable to assume that all researchers are informed, or even agree, as to what constitutes acceptable professional practice. Even experienced scholars

are sometimes in dispute over what constitutes ethical behavior. Thus, while sanctions may be appropriate in some cases, counseling and training may be appropriate in others.

Misconduct occurs in varying degrees of severity and in many different forms. Each case of alleged member misconduct is expected to have unique aspects and it is impossible to provide a “one size fits all” procedure. This policy is perhaps most suitable for cases where documentary evidence can be obtained to establish authorship or other misconduct with a high degree of certainty. In general, the process should keep all parties informed and allow a respondent to take responsibility before a more serious step in the process occurs. The process should be adapted to fit each individual case while preserving the important principles of the policy.

In general, AIS members are expected to comply with the AIS Code of Research Conduct (AIS Administrative Bulletin 2014.0224.01) which provides a basis for ethical scholarly practice. The AIS Research Conduct Committee regards the Code of Research Conduct not merely as a set of narrow rules upon which complaints of scholarly misconduct may be based, but rather as a source of guidance for the scholarly community.

Editors of non-AIS publications are free to adopt in whole or in part the AIS Code of Research Conduct and the procedures described in this policy. However these processes, guidelines, or such “rules” as they may imply, are not intended to apply to, or otherwise inappropriately interfere with, non-AIS publications.

B. Definitions

1. The AIS “Research Conduct Committee (RCC)” is comprised of AIS members as outlined in the Bylaws.

2. "Scholarly misconduct" is the term we use to describe behaviors by researchers that contravene the AIS Code of Research Conduct. While plagiarism is the form of scholarly misconduct that receives the most attention, there are other forms of scholarly misconduct, such as the fabrication or falsification of data, research procedures or data analysis, or the exercise of power or bias in scholarly activities.

3. "Plagiarism" is:
    a. the use of another person’s writings, information, ideas, concepts or data without that person’s permission;

b. misrepresentation to an editor or conference program chairperson about the originality of the submitted work (often called "self-plagiarism").

4. The term “editor” refers to the Editor-in-Chief of an AIS publication or a program chairperson of an AIS conference who has been selected according to guidelines provided in the Bylaws.

5. The term "complainant" refers to the person or persons submitting a complaint alleging scholarly misconduct. The term refers to the role in the procedure, not the validity of the complaint.

6. The term "respondent" refers to the person or persons alleged to have contravened the AIS Code of Conduct. The term refers to the role in the procedure, not the guilt of the individual.

7. The term "offender" refers to the person or persons determined to have contravened the AIS Code of Conduct.

8. The term "victim" refers to the person or persons who are harmed as a result of the offender's scholarly misconduct.

C. Procedures – Initial Steps

1. Complainants may be any observant scholar, an editor or reviewer, or an author or scholar that believes they have been wronged by another. Complainants may submit their complaints to the appropriate AIS editor or to the AIS President. They may not submit allegations directly to the RCC.

2. Allegations of scholarly misconduct submitted to an AIS editor shall first be investigated by the editor.

3. The editor should undertake an initial investigation to ascertain if the complaint is valid.

4. If, in the opinion of the editor, the complaint is deemed not valid (for instance if the reviewer has mistakenly identified plagiarism, or if an author has uploaded an incorrect file, or if an author has blinded mention of his/her own future work so as to preserve anonymity), then the editor should take those simple steps that will remedy the situation (inform the reviewer, upload the correct file) and then dismiss the complaint. The editor should notify the AIS

President if the editor foresees any need to establish a record of a pattern of behavior.

5. If, in the opinion of the editor, the complaint may be valid, then the editor must escalate the matter to the AIS President, who may then turn it over to the RCC. All such allegations must be routed via the AIS President who alone has the authority to refer them to the RCC. In such a case, the editor should not attempt to contact the author or other respondent to ask for explanations.

6. The editor should not ask the eLibrary manager to remove a paper from the eLibrary at any stage of the process or under any circumstances.

D. Procedure –Consideration of Evidence by the RCC

1. The RCC shall only consider cases assigned by the AIS President.

2. Editors or publishers of non-AIS publications may apply to the AIS President on a case-by-case basis for the adjudication by the RCC of allegations of scholarly misconduct in their own publications.

3. Once a case has been assigned to the RCC and the names of the individuals involved in the case identified, each RCC member should consider if he/she has any conflict of interest with any of these individuals. If such a conflict of interest is identified, the member with that conflict is professionally obligated to declare it and to recuse him/herself from the RCC for the duration of the specific case where the conflict applies. The member will then need to be replaced by another member, who will be appointed by the AIS President.

4. If the AIS President has a conflict of interest, then the same rules of recusal apply, and another individual who does not have such a conflict of interest will need to serve as proxy for the AIS President throughout the course of the investigation and the reporting of outcomes.

5. The RCC begins by deciding if sufficient evidence exists to undertake a formal investigation. All evidence gathered by the editor, complainant, and respondent will be examined by the RCC. The RCC may also request additional evidence from either the complainant or the respondent or obtain additional evidence from any other parties (e.g., the superiors of the complainant and respondent) through other channels, as it deems necessary.

6. The RCC should be able to answer each of the following questions in the affirmative to justify further inquiry.
    a. Has sufficient evidence been submitted to conclude that plagiarism or other scholarly misconduct may have occurred?

    b. Is the alleged misconduct sufficiently serious to justify pursuing?

    c. Is there a reasonable prospect of resolution?

    Unless all of these questions can be answered in the affirmative, or unless there are other compelling circumstances, the RCC shall inform the AIS President that no further action will be taken unless more compelling evidence becomes available.

7. If the above questions are answered in the affirmative, the RCC shall notify the respondent that if exculpatory evidence is not provided promptly, the RCC will proceed to investigate the case and will reach a conclusion independently. This conclusion will be reported to the AIS President. If an individual admits misconduct at any stage, the RCC may immediately recommend appropriate actions for ratification by the AIS Executive Committee.

E. Compliance – Sanctions and Redress

1. Following completion of an investigation of alleged scholarly misconduct by the RCC, recommended penalties and actions to be taken will be identified for the AIS Executive Committee to review. The AIS President may communicate these outcomes to the complainant, the respondent and a senior administrator (Dean or above) at the institutions of both the complainant and the respondent, including (if appropriate) a specific response to each allegation made. Further the actions (if any) to be taken as a result of the investigation and its confirmation by the AIS Executive Committee should be effected as soon as is practically possible. Once they have been effected, the complainant and respondent should be informed of the situation, again as soon as practically possible.

2. The nature of the penalties and actions taken by the AIS Executive Committee will depend on the precise circumstances of the case. However, it is important that the injustice done to the victim be redressed and be seen to be redressed.

3. A finding of misconduct by an AIS member that pertains to a scholarly work published in a non-AIS publication may result in sanctions against such individuals as determined by the AIS Executive Committee.

4. A finding of misconduct by an AIS member or non-member that pertains to a scholarly work published by AIS may result in changes to the publication.
    a. Where scholarly misconduct is demonstrated to have been perpetrated and where the offender is a single author, the entire article is to be removed.

    b. Where multiple authors are involved, these authors will need to be consulted as well, since any action taken may unduly and unreasonably affect them and their careers.

    c. Ideally, it will be possible to remove only the offending material, without retracting the whole article. If this occurs, then the offender's name is to be removed from the list of authors of the article, and a retraction notice that details the upheld allegations, the names of the victim(s) and offender(s) and a brief summary of the actions taken should be

       posted to explain why the changes have occurred. This notice is to be appended as a footnote to the article, which will remain in the eLibrary.

    d. If the complete article is retracted, then a short note can replace the article to explain the retraction. Where a retraction occurs because of plagiarism, a link (hyperlink where possible, otherwise a text note) to the correct original publication that was plagiarized is to be provided.

    e. If the original material was unpublished, then a note that describes its provenance can be included instead.

4. In cases where scholarly misconduct has been established, sanctions may include, but are not limited to:
    a. The AIS President issuing a warning letter, with or without copies to the offender's academic and administrative superiors;
    b. Barring the offender from submitting papers to any of the AIS journals or proceedings;

    c. Barring the offender from registering for any of the AIS official conferences; or

d. Expelling the offender from membership in the AIS for a stated period of time.

5. If significant scholarly misconduct is determined to have occurred in connection with a research project funded by an agency (local, regional, national, or international), the AIS President must also inform the agency.

5.4.4 Conference and Journal Publishing (adopted 03/2016)

A. Purpose

One important component of the Mission of the Association for Information Systems (AIS) is the advancement of knowledge in the service of society. This mission is not served by constraining the dissemination of research contributions. AIS therefore encourages authors of papers published at AIS conferences to submit them for publication in AIS journals bearing in mind that most journals expect a more substantial contribution than most conferences. Specifically, it is the policy of AIS that:

1) *A paper published at an AIS conference may be submitted to a journal, even without change; and*

2) *A paper submitted to an AIS journal cannot be rejected only because an earlier version of the paper was previously published at a conference.*

Please note that this policy applies directly to AIS conferences and journals (see http://aisnet.org/?AISeLibrary). Conferences and journals published by other organizations may have different policies which must be respected. Further, authors are not permitted to submit their AIS conference papers to multiple journals, since this explicitly violates the AIS Code of Research Conduct.

5.5 Conferences (future development)

5.6 Advancement

5.6.1 Online Social Networks (adopted 03/2011; revised 06/2013)

A. Purpose

The purpose of the online social networks policy is to protect the identity and integrity of AIS while promoting the appropriate use of social networks to advance the purposes of AIS. Social networks operate through online services to help build social relations between people with common interests, activities, or other connections. Social networks may appear from time-to-time that are relevant to the purposes of AIS. These online venues offer an opportunity to share knowledge and information among AIS members, to increase public awareness of AIS, and to attract new members to AIS.

B. Definitions

1. Branded AIS Online Social Networks use the name, “Association for Information Systems” and the AIS logo. This usage requires written approval by the AIS Executive Director. The purpose of AIS Branded Online Social Networks is to be a medium for IS-related social interactions of parties interested in topics related to, and relevant for, members of the IS academic community.

2. An AIS Department Editor is an AIS member who is participating in a relevant social network and appointed by the VP of Communications to assume a liaison role in the network. Where no such AIS members are available, the VP of Communications will, in cooperation with related SIG officers, seek an AIS member to join this network and serve as liaison under appointment as the AIS Department Editor in the social network topic. Department editors may also be recommended by other AIS Vice Presidents to the VP of Communications. The VP of Communications is responsible to AIS Council for governance of the collective body of all AIS Department Editors.

C. Procedures

1. The VP of Communications will routinely encourage officers, council members, SIG officers, and AIS members to provide information about relevant social networks. The VP of Communications will annually deliver a report to AIS council providing a summary and evaluation of the activities of the collective body of AIS Department Editors.

2. When the AIS name and/or logo is approved for use, the Executive Director will notify the VP of Communications, who will then appoint a Department Editor to be designated by the social networking host organization (e.g., Facebook, LinkedIn) as the nominal owner of the social network. This

Department Editor will be responsible for monitoring and controlling the social network in keeping with this policy.

3. Department Editors with ownership or liaison roles in networks have the responsibility to immediately represent AIS when opportunities arise to enhance AIS' reputation and standing in the relevant social community. Department Editors are encouraged to immediately inform the VP of Communications, or if appropriate the President of AIS, of these opportunities, in an effort to enable further enhancements. A swift response in the case of misinformation is particularly important in promoting AIS to social networking communities.

D. Compliance

1. AIS Branded Online Social Networks should be "owned" officially by an AIS Department Editor.

2. AIS Branded Online Social Networks should be used as a forum for discussions that are IS-related and of significant importance to a large majority of the participants.

3. AIS Branded Online Social Networks should encourage thoughtful enrollment and participation by AIS members and potential AIS members.

4. AIS members may use AIS Branded Online Social Networks to speak for themselves individually.

5. Misrepresentations about AIS made by media or other social networking users should be corrected only by an AIS officer in an official capacity.

6. Members using AIS Branded Online Social Networks should avoid:

    a. Use to advertise events, items, or services that are marketed to generate profit, without advance written permission of the AIS Executive Director;

    b. Use for student exercises;

    c. Use for repeated distribution of the same information;

    d. Communication that is confidential, in violation of any copyright law, defamatory, or in violation of any other law;

e. Communication that is in bad taste, inflammatory, offensive, or political in content, tone, or implication to any member of the diverse, global, and multi-cultural AIS community;

f. Distribution of intellectual property without the owner's permission.

7. AIS reserves the right to temporarily or permanently block any AIS Branded Online Social Network participant who fails to abide by these conditions, and also to remove any and all communications from the network archives with or without cause.

5.7 Audit and Compliance

5.7.1 Privacy Policy (adopted 06/2009; revised 06/2018)

A. Purpose

This policy declares the Undertakings by AIS in relation to its handling of your personal data, as those terms are defined in the definitions section below. AIS handles personal data for the following purposes:

1. *Membership data*: All data-items are supplied by the member and can be viewed through the membership renewal web-forms. Credit-card details are retained only as long as they are needed to complete the payment transaction, with only partial details held long-term, in a log-file. Access to membership data is protected by password and is permitted only by the member and relevant AIS staff, contractors and officers.

2. *Officer data*: The data-items are limited to name, affiliation and contact-points, as provided by the officer.

3. *Candidate data*: The data is limited to that provided by candidates for AIS offices and is deleted after the election process is complete.

4. *Site-visitor*: The data is limited to that provided by visitors and/or disclosed by the visitor's client-software in the normal course of operation of services such as the Web.

5. *Website contributor data*: The data is limited to that provided by contributors of AIS resource-pages.

6. *Author data*: The data is limited to that provided by authors to AIS publications such as JAIS, CAIS, RELCASI, sprouts, ICIS and AMCIS, and to AIS-affiliated journals and conferences contained in the AIS Electronic Library.

7. *Editor and reviewer data*: The data is limited to that provided to AIS publications by editors and reviewers and stored in the relevant manuscript management tool and access is limited to people performing relevant functions.

8. *Employee and contractor data*: The data-items are limited to that relevant to the relationship. Disclosures are limited to name, position description and contact-point, as displayed in the AIS staff directory.

9. *Data Controller*: AIS processes personal data both as a processor and as a controller. The AIS entity which you as a member entered an agreement with when using the AIS platform, will be the controller for user data, as outlined in the 'collection of user data' section.

10. *Data Contact information for Data Controller*: Contact the Data Controller at any time to:
    - Request access to information that AIS has about you
    - Correct any information AIS might have about you
    - Delete information AIS has about you

    If you have additional questions about AIS's collection and storage of data, please contact us at membership@aisnet.org

11. Data Protection Officer: AIS has a Data Protection Officer, responsible for matters relating to privacy and data protection. This Data Protection Officer can be reached at webmater@aisnet.org

B. Definitions

1. "AIS" means the Association for Information Systems Inc., incorporated in Illinois USA as a non-profit organization, and which can be contacted here http://aisnet.org

2. "Undertaking" means an enforceable obligation that arises from a statement made, or an assurance or commitment given. In each case where this Privacy

   Policy Statement says that "AIS undertakes" to do something, that statement gives rise to an Undertaking.

3. “Your data” means data that is capable of being associated with you, whether or not it includes an explicit identifier such as your name or customer number. In particular, it encompasses all data that AIS is capable of correlating with you, using such means as server-logs and cookie-contents. “Your data” does not refer to data that cannot or can no longer be associated with you. This includes aggregated data that does not and cannot identify the individuals whose data are included in the aggregation.

4. “Consent” means your concurrence with an action to be taken by AIS. Consent may be express or implied, but in either case must be informed and freely given.

5. “Jurisdiction” means the sphere of authority within which relevant legal powers may be exercised, in particular within which a particular court has authority. AIS is incorporated in Illinois USA. Its primary operations are in Georgia USA. Its officers and agents are in many locations around the world, and hence AIS may under various circumstances be subject to laws in a wide variety of jurisdictions.

C. Procedures – Data Collection

1. AIS undertakes to collect your data by means that are fair, legal, and transparent.

2. If you visit AIS's website, your web-browser automatically discloses, and AIS's web-server automatically logs, the following information: the date and time, the IP address from which you issued the request, the type of browser and operating system you are using, the URL of any page that referred you to the page, the URL you requested, and whether your request was successful. This data may or may not be sufficient to identify you.

3. Any additional data that you provide, e.g. in a web-form, may also be logged. This data may or may not be sufficient to identify you.

4. Any additional data that your web-browser automatically provides may also be logged. This will be the case, for example, if your browser has previously been requested to store data on your computer in 'cookies' and submits them each time you request a web-page within a particular domain (in particular, aisnet.org). This data may or may not be sufficient to identify you.

5. AIS uses cookies when a visitor arrives on the web-site, and when a user logs on to sections of the AIS website that require their username and password, such as the Directory and the Digital Library. Most of the cookie-data is encrypted and contains no sensitive data such as passwords. All cookies are set for session-only, and hence browsers should delete them at the next opportunity, such as the next quit and re-start.

6. If you disclose personal data to AIS in conjunction with an identifier such as your name or your credit-card details, AIS will collect your data. Moreover, any data that becomes available to AIS through any of the means described in the preceding paragraphs may be able to be associated with that identifier, and hence become your data.

7. AIS undertakes to collect your data from you and not from other parties. Where AIS collects your data from sources other than you, it undertakes to do so only by legal mean, to do so only with your consent, and to declare to you what sources it uses and under what circumstances.

8. AIS will list acceptable uses of your data in its Privacy Statement, which may be amended from time to time.

D. Procedures – Data Use

1. AIS undertakes to store your data in a manner that ensures security against unauthorized access, alteration or deletion, at a level commensurate with its sensitivity.
2. AIS undertakes to transmit your data in a manner that ensures security against unauthorized access, alteration or deletion, at a level commensurate with its sensitivity.

3. AIS undertakes to apply protections for your data at a level consistent with the OECD Guidelines on Privacy and on Security, even though your data may be stored in or transmitted to a jurisdiction whose legal requirements are lower than that.

4. AIS undertakes to implement appropriate measures to ensure security of your data against inappropriate behavior by AIS's staff-members, contractors and officers. These include:

    a. training for staff in relation to privacy;

b. access control, to limit access to your data to those staff, contractors and officers who have legitimate reasons to access it;

c. logs of changes to data;

d. reminders to staff and contractors from time to time about the importance of data privacy, and the consequences of inappropriate behavior;

e. declaration of appropriately strong sanctions that are to be applied in the event of inappropriate behavior;

f. clear communication of policies and sanctions to staff; and

g. processes to investigate and to impose sanctions.

5. Data Breach Response Policy - Any individual who suspects a theft, breach or exposure of AIS protected data has occurred must immediately provide a description of the event to webmaster@aisnet.org. AIS will investigate all reported thefts, data breaches and exposures to confirm if an incident has occurred. If an incident has occurred all affected users will be notified immediately of the scope of incident, steps to correct the situation, and any further actions to help serve AIS members in protecting their information.

E. Procedures – Disclosure

1. Disclosure refers to making your data available to any party other than AIS and you. The term disclosure may include many different conditions of data transfer, including selling, renting, trading, sharing and giving.

2. AIS undertakes to disclose your data only under the following circumstances:

a. for purposes for which we have your Consent, including purposes that are initially or subsequently agreed between you and AIS, purposes directly implied by the agreed purposes, and at your request;

b. for such additional purposes as are required by law, such as a provision of a statute, or a court order such as a search warrant or subpoena. In these circumstances, AIS will take any reasonable steps available to it to communicate to you that the disclosure has occurred, unless it is precluded from doing so by law;

c. for such additional purposes as are permitted by law (e.g. the reporting of suspected breach of the criminal law to a law enforcement agency; and in an emergency, where AIS believes on reasonable grounds that the disclosure of your data will materially assist in the protection of the life or health of some person), provided that AIS will apply due diligence to ensure that the exercise of the permission is justifiable.

3. Where your data is disclosed to an outsourced service-provider (e.g. to a company that processes credit-card transactions), AIS undertakes to make

   reasonable endeavors to exercise control over compliance by its service-provider with the terms of this Privacy Policy Statement.

4. AIS undertakes to disclose your data only if it has demonstrable relevance to the particular use to which it is being put, and to disclose only such of your data as is necessary in the particular circumstances.

5. AIS undertakes to disclose your data in such a manner as to take into account the possibility that it is not of sufficient quality for the purpose, e.g. because it is inaccurate, out-of-date, incomplete, or out-of-context.

F. Procedures – Data Retention and Destruction

1. Subject to the qualifications immediately below, AIS undertakes:

   a. to retain your data only as long as AIS reasonably believes it is consistent with its purpose; and

   b. to destroy your data when AIS reasonably believes its purpose has expired, and to do so in such a manner that your data is not subsequently capable of being recovered.

2. This Undertaking is qualified as follows:

   a. when your data falls due for destruction, it may be retained for a period beyond its expiry of purpose, until the next regular deletion cycle;

   b. your data may be retained in AIS's logs, backups and audit trails within short-term retention cycles that are devised to protect the company's operations. In such cases, your data will be destroyed in accordance with those cycles;

c. in some circumstances, your data may be retained in an archive. An archive may be internal-only and accessible only by staff, contractors and officers; or it may be publicly available, as is the case with data relating to previous AIS officers;

d. your data may be retained beyond the expiry of its purpose if that is required by law, such as a provision of a statute, or a court order such as a search warrant or subpoena, or a warning by a law enforcement agency that delivery of a court order is imminent. In these circumstances, AIS:

    1. will take any reasonable steps available to it to communicate to you that your data is being retained, unless it is precluded from doing so by law; and

    2. will only retain your data while that provision is current, and will then destroy your data;

    3. your data may be retained beyond the expiry of its purpose if such retention is authorized by law (in particular to protect AIS's interests, e.g. if it believes on reasonable grounds that you have failed to fulfil your Undertakings to AIS or may have committed a breach of the criminal law). In these circumstances, AIS will only retain your data while that situation is current and will then destroy your data.

G. Procedures – Access by You to Your Personal Data

1. AIS undertakes to provide you with access to your data, subject to only such conditions and processes as are reasonable in the circumstances. In particular, AIS undertakes to enable access conveniently, without unreasonable delay, and without cost to you.

2. AIS undertakes to establish and operate identity authentication protections for access to your data that are appropriate to its sensitivity, but practical. This may involve some inconvenience; for example, relatively straightforward procedures may be involved in order to provide you with access through a channel that you have previously provided to AIS (such as a particular email-address), but more onerous procedures may have to be imposed if you wish to use some other channel.

3. If you request it, AIS undertakes to take reasonable steps in relation to the amendment, supplementation or deletion of your data.

4. In providing these undertakings, AIS is working on the assumptions that:

    a. you will not to seek access, amendment, supplementation or deletion for frivolous purposes, or unreasonably frequently;

    b. you accept that deletion of some categories of data may result in AIS no longer being able to provide particular services to you.

5. Access to information - AIS members have the right to request access to information AIS has on you. Contact us at membership@aisnet.org to obtain such data. You will be provided with a copy of the data we process about you. Verification of identity will be required. Requests will be fulfilled electronically within five business days.

6. Information Correction & Deletion- AIS members have the right to correct or update information. Any data that is no longer needed for the purpose of membership will be deleted at your request. Contact membership@aisnet.org for instructions on correcting and deleting information.

H. Procedures – Data Handling Practices

1. AIS undertakes to make information available to you about the manner in which AIS handles your data:

    a. in general terms, in a readily accessible manner, by means of this Privacy Policy Statement published on the AIS website;

    b. and in more specific terms, on request.

2. Where your data is disclosed to an outsourced service-provider, AIS undertakes to make information available to you, on request, about the manner in which AIS's outsourced service-provider handles your data.

3. AIS undertakes to ensure that the information provided about data-handling practices is meaningful and addresses your concerns.

4. In providing these undertakings, AIS is working on the assumptions that:

    a. you will not seek such information for frivolous purposes, or unreasonably frequently;

b. and you accept that the disclosure of excessive detail may harm the security of your data and AIS's business processes and may harm AIS's operational activities.

I. Procedures – Handling of Inquiries, General Concerns and Complaints

1. In providing these undertakings, AIS is working on the assumption that, if you have inquiries, general concerns, or complaints about any aspect of this Privacy Policy Statement, or about AIS's behavior in relation to its Undertakings, you will communicate them in the first instance to AIS only, in sufficient detail, and through a channel made available by AIS for that purpose.

2. AIS undertakes:

   a. to provide one or more channels for communications to AIS, which are convenient to users. To find these channels, please go to the AIS Contacts Page http://aisnet.org/?ContactUs;

   b. to promptly provide acknowledgement of the receipt of communications, including a copy of the communication, the date and time it was registered, and an indication of how to follow up the matter with AIS if a formal response is slow in arriving;

   c. to promptly provide a response to the communication, in an appropriate and meaningful manner.

3. In providing these undertakings, AIS is working on the assumption that you will not pursue AIS through any regulator or the media:

   a. until and unless AIS has had a reasonable opportunity to respond to the initial communication;

   b. and while AIS and you are conducting a meaningful dialogue about the matter.

J. Compliance – Enforcement

1. AIS declares that the Undertakings expressed in this Privacy Policy Statement are intended to create legal obligations, and that those obligations are intended to be enforceable under appropriate laws in appropriate jurisdictions. These may include laws relating to data

protection, privacy, fair-trading, unfair competition, the corporations law, and the criminal law.

2. In providing these undertakings, AIS is working on the assumptions that:

   a. you will not unreasonably seek enforcement until you have initiated the complaints-handling process and AIS has had the opportunity to redress the wrong; and

   b. you will seek enforcement only in a jurisdiction that is relevant to the transactions that have taken place between you and AIS, in particular the jurisdiction in which you live or in which you performed the relevant acts, and the jurisdiction in which AIS is domiciled or performed the relevant acts.

3. If you wish to discover the relevant laws in any particular jurisdiction, AIS draws your attention to the following resources:

   a. WorldLII Privacy Links (http://www.worldlii.org/catalog/273.html)

   b. Compilation of [U.S.] State and Federal Privacy Laws, Privacy Journal, Providence RI (http://www.privacyjournal.net/work1.htm)

   c. Privacy Law Sourcebook 2004, EPIC, Washington DC (http://epic.org/bookstore/pls2004/)

   d. Privacy International (https://www.privacyinternational.org/)

   e. The Australian Privacy Foundation's pages:
      1. International Instruments Relating to Privacy Law (http://www.privacy.org.au/Resources/PLawsIntl.html)
      2. Privacy Laws of Countries of the World (http://www.privacy.org.au/Resources/PLawsWorld.html)
      3. Privacy Laws of the Commonwealth of Australia (http://www.privacy.org.au/Resources/PLawsCIth.html)
      4. Privacy Laws of the States and Territories of Australia (http://www.privacy.org.au/Resources/PLawsST.html)

4. International Privacy Laws - Visitors to AIS websites from outside the United States are sending information (including personal data) to the United States where AIS servers are located. That information may then be transferred within the United States or back out of the United States to other countries

outside your country of residence, depending on the type of information and how it is stored. These countries (including the United States) may not necessarily have data protection laws as comprehensive or protective as those in your country of residence; however, our collection, storage and use of your personal data will, at all times, continue to be governed by this privacy policy.

K. Compliance – Changes to These Privacy Undertakings

1. AIS undertakes:

    a. not to change this Privacy Policy Statement in a manner that materially reduces the protections for your data;

    b. to subject proposals for material changes to this Privacy Policy Statement, or for more specific terms relating to particular services, to a process comparable with that used when making changes to the AIS By-Laws, and including consultation with members and/or with one or more appropriate representative and advocacy organizations; and

    c. where new versions of this Privacy Policy Statement are promulgated, to ensure that:

        1. the previous versions and their dates of applicability remain accessible and the differences between successive versions are visible; and

        2. to take all possible steps to prevent any organization that may take over or absorb AIS or any of its relevant assets from materially changing the terms applicable to your data in a manner that reduces the protections for your data.

5.7.2 AIS Whistleblower Policy (adopted 12/2016)

The Association for Information Systems (AIS) is committed to operating in furtherance of its tax-exempt purposes and in compliance with all applicable laws, rules and regulations, including those concerning accounting and auditing, and prohibits fraudulent practices by any of its board members, officers, employees, or volunteers. This Policy outlines a procedure for employees to report actions that an employee reasonably believes violates a law, or regulation or that constitutes fraudulent accounting or other practices. This Policy applies to

any matter which is related to AIS's business and does not relate to private acts of an individual not connected to the business of AIS.

If an employee has a reasonable belief that an employee or the organization has engaged in any action that violates any applicable law, or regulation, including those concerning accounting and auditing, or constitutes a fraudulent practice, the employee is expected to immediately report such information to the Executive Director. If the employee does not feel comfortable reporting the information to the Executive Director, he or she is expected to report the information to the President.

All reports will be followed up promptly, and an investigation conducted. In conducting its investigations, AIS will strive to keep the identity of the complaining individual as confidential as possible, while conducting an adequate review and investigation.

AIS will not retaliate against an employee in the terms and conditions of employment because that employee: (a) reports to a supervisor, to the Executive Director, the AIS leadership, or to a federal, state or local agency what the employee believes in good faith to be a violation of the law; or (b) participates in good faith in any resulting investigation or proceeding, or (c) exercises his or her rights under any state or federal law(s) or regulation(s) to pursue a claim or take legal action to protect the employee's rights.

AIS may take disciplinary action (up to and including termination) against an employee who in management's assessment has engaged in retaliatory conduct in violation of this policy.

In addition, AIS will not, with the intent to retaliate, take any action harmful to any employee who has provided to law enforcement personnel or a court truthful information relating to the commission or possible commission by AIS or any of its employees of a violation of any applicable law or regulation.

Supervisors will be trained on this policy and AIS's prohibition against retaliation in accordance with this policy.

5.7.3 AIS Document Management Policy (adopted 12/2016)

1. Policy and Purposes

This Policy represents the policy of the Association for Information Systems (the "organization") with respect to the retention and destruction of documents and

other records, both in hard copy and electronic media (which may merely be referred to as "documents" in this Policy). Purposes of the Policy include (a) retention and maintenance of documents necessary for the proper functioning of the organization as well as to comply with applicable legal requirements; (b) destruction of documents which no longer need to be retained; and (c) guidance for the Council, officers, staff, and other constituencies with respect to their responsibilities concerning document retention and destruction. Notwithstanding the foregoing, the organization reserves the right to revise or revoke this Policy at any time.

2. Administration

2.1 **Responsibilities of the Administrator**. The organization's Executive Director shall be the administrator ("Administrator") in charge of the implementation of this Policy. The Administrator's responsibilities shall include supervising and coordinating the retention and destruction of documents pursuant to this Policy and particularly the Document Retention Schedule included below. The Administrator shall also be responsible for documenting the actions taken to maintain and/or destroy organization documents and retaining such documentation. The Administrator may also modify the Document Retention Schedule from time to time as necessary to comply with law and/or to include additional or revised document categories as may be appropriate to reflect organizational policies and procedures. The Administrator is also authorized to periodically review this Policy and Policy compliance with legal counsel and to report to the Council as to compliance. The Administrator may also designate one or more assistants to aid in carrying out the Administrator's responsibilities, with the Administrator, however, retaining responsibility for administration of this Policy.

2.2 **Responsibilities of Constituencies**. This Policy also relates to the responsibilities of Council members, staff, volunteers, and outsiders with respect to maintaining and documenting the storage and destruction of the organization's documents. The organization's staff shall be familiar with this Policy, shall act in

accordance therewith, and shall assist the Administrator, as requested, in implementing it. The responsibility of volunteers with respect to this Policy shall be to produce specifically identified documents upon request of management, if the volunteer still retains such documents. In that regard, after each project in which a volunteer has been involved, or each term which the volunteer has served, it shall be the responsibility of the Administrator to confirm whatever types of documents the volunteer retained and to request any such documents which the Administrator feels will be necessary for retention by the organization

(not by the volunteer). Outsiders may include vendors or other service providers. Depending upon the sensitivity of the documents involved with the particular outsider relationship, the organization, through the Administrator, shall share this Policy with the outsider, requesting compliance. In particular instances, the Administrator may require that the contract with the outsider specify the particular responsibilities of the outsider with respect to this Policy.

3. **Suspension of Document Destruction; Compliance.** The organization becomes subject to a duty to preserve (or halt the destruction of) documents once litigation, an audit, or a government investigation is reasonably anticipated. Further, federal law imposes criminal liability (with fines and/or imprisonment for not more than 20 years) upon whomever "knowingly alters, destroys, mutilates, conceals, covers up, falsifies, or makes a false entry in any record, document, or tangible object with the intent to impede, obstruct, or influence the investigation or proper administration of any matter within the jurisdiction of any department or agency of the United States … or in relation to or contemplation of any such matter or case." Therefore, if the Administrator becomes aware that litigation, a governmental audit, or a government investigation has been initiated, or is reasonably anticipated or contemplated, the Administrator shall immediately order a halt to all document destruction under this Policy, communicating the order to all affected constituencies in writing. The Administrator may amend or rescind the order only after conferring with legal counsel. If any council member or staff member becomes aware that litigation, a governmental audit, or a government investigation has been initiated, or is reasonably anticipated or contemplated, with respect to the organization, and they are not sure whether the Administrator is aware of it, they shall make the Administrator aware of it. Failure to comply with this Policy, including, particularly, disobeying any destruction halt order, could result in possible civil or criminal sanctions. In addition, for staff, it could lead to disciplinary action including possible termination.

4. **Electronic Documents; Document Integrity.** Documents in electronic format shall be maintained just as hard copy or paper documents are, in accordance with the Document Retention Schedule. Due to the fact that the integrity of electronic documents, whether with respect to the ease of alteration or

deletion, or otherwise, may come into question, the Administrator shall attempt to establish standards for document integrity, including guidelines for handling

electronic files, backup procedures, archiving of documents, and regular checkups of the reliability of the system; provided, that such standards shall only

be implemented to the extent that they are reasonably attainable considering the resources and other priorities of the organization.

5. **Privacy.** It shall be the responsibility of the Administrator, after consultation with counsel, to determine how privacy laws will apply to the organization's documents from and with respect to employees and other constituencies; to establish reasonable procedures for compliance with such privacy laws; and to allow for their audit and review on a regular basis.

6. **Emergency Planning.** Documents shall be stored in a safe and accessible manner. Documents which are necessary for the continued operation of the organization in the case of an emergency shall be regularly duplicated or backed up and maintained in an off-site location. The Administrator shall develop reasonable procedures for document retention in the case of an emergency.

7. **Document Creation and Generation.** The Administrator shall discuss with staff the ways in which documents are created or generated. With respect to each employee or organizational function, the Administrator shall attempt to determine whether documents are created which can be easily segregated from others, so that, when it comes time to destroy (or retain) those documents, they can be easily culled from the others for disposition.

8. **Document Retention Schedule**

Document Type	**Retention Period**
Accounting and Finance	
Accounts Payable	7 years
Accounts Receivable	7 years
Annual Financial Statements and Audit Reports	Permanent
Bank Statements, Reconciliations & Deposit Slips	7 years
Canceled Checks – routine	7 years
Canceled Checks – special, such as loan repayment	Permanent
Credit Card Receipts	7 years
Employee/Business Expense Reports/Documents	7 years
General Ledger	Permanent
Interim Financial Statements	7 years
Contributions/Gifts/Grants	
Contribution Records	Permanent
Documents Evidencing Terms of Gifts	Permanent

Grant Records	7 years after end of grant period

Corporate and Exemption

Articles of Incorporation and Amendments	Permanent
Bylaws and Amendments	Permanent
Minute Books, including Board & Committee Minutes	Permanent
Annual Reports to Attorney General & Secretary of State	Permanent
Other Corporate Filings	Permanent
IRS Exemption Application (Form 1023 or 1024)	Permanent
IRS Exemption Determination Letter	Permanent
State Exemption Application (if applicable)	Permanent
State Exemption Determination Letter (if applicable)	Permanent
Licenses and Permits	Permanent
Employer Identification (EIN) Designation	Permanent

Correspondence and Internal Memoranda

Hard copy correspondence and internal memoranda relating to a particular document otherwise addressed in this Schedule should be retained for the same period as the document to which they relate.

Hard copy correspondence and internal memoranda relating to routine matters with no lasting significance	2 years
Correspondence and internal memoranda important to the organization or having lasting significance	Permanent, subject to ED review

Email to or from the organization

Electronic mail (emails) relating to a particular document otherwise addressed in this Schedule should be retained for the same period as the document to which they relate, but may be retained in hard copy form with the document to which they relate.

Emails considered important to the organization or of lasting significance should be printed and stored in a central repository.	Permanent, subject to ED review

E-mails not included in either of the above categories	1 Year

Electronically Stored Documents

Electronically stored documents (e.g., in pdf, text or other electronic format) comprising or relating to a particular document otherwise addressed in this Schedule should be retained for the same period as the document which they comprise or to which they relate, but may be retained in hard copy form (unless the electronic aspect is of significance).

Electronically stored documents considered important to the organization or of lasting significance should be printed and stored in a central repository (unless the electronic aspect is of significance).	Permanent, subject to ED review
Electronically stored documents not included in either of the above categories	2 years

Employment, Personnel and Pension

Personnel Records	10 years after employment ends
Employee contracts	10 years after termination
Retirement and pension records	Permanent

Insurance

Property, D&O, Workers' Compensation and General Liability Insurance Policies	Permanent
Insurance Claims Records	Permanent

Legal and Contracts

Contracts, related correspondence and other supporting documentation	10 years after termination
Legal correspondence	Permanent

Management and Miscellaneous

Strategic Plans	7 years after expiration

Policies and Procedures Manual	Current with revision history
Property – Real, Personal and Intellectual	
Property deeds and purchase/sale agreements	Permanent
Property Tax	Permanent
Real Property Leases	Permanent
Personal Property Leases	10 years after termination
Trademarks, Copyrights and Patents	Permanent
Tax	
Tax exemption documents & correspondences	Permanent
IRS Rulings	Permanent
Annual information returns – federal & state	Permanent
Tax Returns	Permanent

Addendum

AIS Council Policy Manual Addendum

Document	Policy Reference	Link
Conflict of Interest Disclosure Form	3.5.1	http://aisnet.org/resource/resmgr/Council_Docs/2009jnConflictOfInterestPoli.pdf
Executive Director Job Description	4.2	http://aisnet.org/resource/resmgr/Council_Docs/executive_director_job_descr.pdf

Other AIS Governing Documents

AIS Bylaws: http://ais.site-ym.com/general/custom.asp?page=AISBylaws
AIS Constitution: http://ais.site-ym.com/?AISConstitution
AIS Administrative Bulletins: http://ais.site-ym.com/?AdminBulletins

END OF DOCUMENT

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

For detailed information on the copyright, please refer to page 9.

Publishing at CEUR-WS.org (valid from 2019-08-01)

2019-07-02: This is the new procedure reflecting our adoption of CC BY 4.0 as the license for newly published proceedings. Existing volumes are not affected by the new license model. See here for our old procedure valid until end of July 2019.

This document is addressed to organizers of scientific workshops/conferences who are interested in distributing their proceedings via the Internet. CEUR-WS.org focuses on the discipline of **computer science** including information systems. Please read this document carefully to avoid unnecessary delays.

CEUR-WS.org publishes computer science workshop proceedings

A **computer science** workshop proceedings volume is characterized as follows:

- The organizers and program committee members come overwhelmingly from computer-science departments and have a degree in computer science (or a related discipline such as information systems, business informatics).
- The vast majority of papers included in the proceedings have at least one author coming from a computer science department and/or having a computer-science degree (or a related discipline such as information systems, business informatics).
- The papers in the proceedings mainly apply research methods from computer science (or a related discipline such as information systems, business informatics). It is not sufficient to use computer programs as a tool for facilitating the research methods from another discipline such as material science.
- There are typically not more than roughly 20 papers in the volume, i.e. filling 2-4 sessions for the workshop. The papers relate to a narrow set of topics listed in the call for papers of the workshop.

Open Access Policy at CEUR-WS.org

- The copyright and any similar right for the proceedings and all included material remain with the papers' authors/owners (for the individual papers) and with the proceedings editors (for the proceedings volume as a whole). The proceedings editors are responsible for the quality of the published content and its bilbliographic metadata.
- User rights are expressed by the license dedication of a volume and its contents. New volumes after summer 2019 adopt the Creative Commons Attribution (CC-BY 4.0) license. Older volumes permit the use for private and academic purposes. See the appropriate license dedication in the volume and/or the digital artefact/paper in a volume.
- CEUR-WS.org provides published proceedings and papers as 'open access'. CEUR-WS.org employs no access control.
- CEUR-WS.org has the right to adapt the index file of a proceedings file to conform to the common style of CEUR-WS.org. Adaptions to the style can lead to certain adaptions of the style of the index file of a published volume. We also reserve the right to correct errors in the index file.
- CEUR-WS.org has the right to tag published papers (e.g. the pdf file) with bibliographic details (e.g. paper URL).
- CEUR-WS.org is not obliged to publish a submission. The CEUR-WS Team decides whether or not to publish a submission on the basis of the policy and pre-conditions spelled out in this document. The CEUR-WS Team is not obliged to justify its decision.

Benefits at CEUR-WS.org

We invite organizers of computer science workshops to use the WWW site CEUR-WS.org physically located at SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS as a medium to publish their proceedings. The service is part of the activities under the umbrella of SunSITE Central Europe. The goal of CEUR-WS.org is promoting information exchange within the academic community. We aim at a high-quality service with the following characteristics:

- Thanks to the management of Sun SITE Central Europe at RWTH Aachen, the service is free of charge for organizers of scientific workshops/conferences and for the readers of their proceedings.
- CEUR-WS.org aims at a high service level. Proceedings are usually made available online within a couple of days after submission. It may also take several weeks depending on our workload and the number of errors in the submission.
- Proceedings are assigned a uniform bibliographic identifier, being its "cool" uniform resource locator (URL) http://ceur-ws.org/Vol-XXX. Papers in volumes also have cool URLs such as http://ceur-ws.org/Vol-XXX/paper1.pdf. We assign persistent identifiers (URN) to all published volumes in collaboration with Deutsche Nationalbibliothek.

Proceedings published on CEUR-WS.org are freely accessible on the SunSITE. Hence, there are no technical provisions (access authorization) to prevent un-authorized use of the proceedings contents. Being freely accessible does not mean 'public domain', however. All material remains copyrighted.

Preconditions for publishing at CEUR-WS.org

We do not formally evaluate the scientific quality of submitted volumes but expect that this is guaranteed by the editors who submit the proceedings volume. There are a few formal rules that your submission should fulfill (updated 2017-12-18):

1. **Peer review by international program committee**: Proceedings papers must be *peer-reviewed* by a program committee of well-known international scholars, who are experts in the topics presented in the proceedings. Invited papers may be included without being peer-reviewed, provided that the majority of papers in the proceedings are not invited papers. The members of the program committee and the rules for paper selection must be **publicly announced**, e.g. on the web page of the conference/workshop and/or in the preface of the proceedings. If the workshop/track is part of a larger event (conference), then there must be a **dedicated call for papers and program committee** different from the call for papers / program committee of the larger event.

2. **Minimum size**: There shall be **at least six papers** in a submitted volume. The minimum length of a regular or short paper should be **five "standard" pages** (=2500 chars per page, calculated from a sample of LNCS one-column papers). Invited papers can have less pages. The whole proceedings volume should have **at least 40 pages** excluding frontmatter. Since page counts depend on the page layout, we calculate with around 2500 characters per page (=380-400 words per page).
   We distinguish regular papers (at least 10 "standard" pages) and short papers (5-9 "standard" pages). See here for details. The index file shall use appropriate tags such as session names or tags behind the paper title to qualify the type of non-regular papers.
3. **Open submission**: Submission of papers to the workshop/conference should be open. For example, it should not be restricted to members of a certain project.
4. **Academic editor**: There is at least one person with a PhD in the list of editors. This person gives her good name for the quality of the submission.
5. **Consistent paper set**: A proceedings volume shall not be or stay published at CEUR-WS.org, if there is another proceedings publication for the same event (identified by its title plus year) but with a *different* set of papers.
6. **Single location, single time**: The workshop/conference must take place (physically) at one location for the usual duration of such an event at a sequence of consecutive days. We do not publish proceedings for virtual and on-line conferences.
7. **Use of English**: The majority of the papers in a proceedings volume shall be in English. For papers written in a language different to English follow these instructions.
8. **Focus on computer science and wide audience**: The CEUR-WS.org publishing service is a venue to publish proceedings of the workshop and conferences, where the main topic is related to computer science (incl. information systems). Due to a significant number of submissions that are not directly related to computer science or that target mostly a national audience, we reserve the right to ask the editors to provide us a so-called DBLP-footprint with the number of publications listed in DBLP.org for each author and/or PC member.
9. **No "student" proceedings**: We expect that the papers are (co-)authored by academics, usually PhD holders. PhD student workshops are a notable exception (see below). We do not support student proceedings dedicated for master/bachelor students as main authors. Regular proceedings may include a few student papers if their number is small compared to the total number of papers and if those papers are reviewed in the same way as all other papers. (This rule formally applies for submissions after 2020-04-01.)
10. **Timeliness**: The submission of a proceedings volume should be close to the event date, not more than 2 years after the event. Exceptions may be possible but need to be negotiated with us.
11. **Format** We require that authors use the new CEURART style for writing papers to be published with CEUR-WS.

Doctoral consortiums also form an exception to the open submission rule in the following sense:

- **Doctoral consortiums / PhD workshops**: Submissions for proceedings of computer science PhD workshops (=workshops targeting only PhD students) are possible as an exception to the open submission rule. The workshop has to be organized by senior researchers, usually organized co-located to a well-known computer science conference. The workshop should target all PhD students working on the topics mentioned in the call for papers, i.e. it should not be restricted to PhD students from a restricted set of organizations. The title of the workshop shall indicate its type, e.g. "PhD Workshop on Machine Learning". PhD student as authors are exempted from the "DBLP footprint" mentioned above.

CEUR-WS is mainly about workshop proceedings, since we believe that they are the place where new results tend to be presented first to an academic audience, at least in computer science. We publish conference proceedings only as a rare exception:

- **Conference proceedings**: Under rare circumstances, we may publish computer science conference proceedings if the conference series is already well-established in the international academic community as a high-quality conference. A conference proceedings typically may have a larger number of papers compared to workshop proceedings. Conference organizers are advised to contact us via email before they submit their proceedings for the first time. Note that we may regard proceedings with more than roughly 20 papers as de-facto conference proceedings. Workshops usually span only 2-4 sessions, hence proceedings with many papers may violate the definition of a workshop as we see it. We shall very likely reject proceedings submissions that have a very broad set of topics. The more focused, the better!

If your volume would violate the "minimum size" constraint, then consider to form a joint submission with another workshop that was held at the **same event** (same place, same time): See Vol-3443 for an example. In such cases, the organizers of the joint workshops are typically the editors of the joint submission and one of them should perform the procedure PUT (see below). Joint proceedings of workshops that took place at different events cannot be published with CEUR-WS.org.

CEUR-WS.org is a publication channel for workshops and conferences from the computer science and information systems domain. We may publish proceedings that have a cross-section with neighbor disciplines if they are written from a computer science and/or information systems point of view. That concerns the research methods used in the papers, not the fact whether or not computers are used for the research. The workshop/conference should have a clear focus on **specific computer science topics**.

We are quite strict on the preconditions and want to avoid unnecessary rejections. Since we check the constraints after the submission, you should **be careful with promises** to your authors that the proceedings will be published with CEUR-WS.org. However, if you follow the guidelines of this document, you can be rather sure that your request shall be accepted. You may include a phrase like "Proceedings shall be submitted to CEUR-WS.org for online publication" in the Call for Papers of your workshop/conference and on your website. Please do not use the CEUR-WS logo on your website.

Papers written in a language different than English

The majority (50% or more) of the papers in a volume must be in English. English is the de facto standard when you want to target an international audience. The submitted papers have to be written in the Latin alphabet. Spell author names in Latin characters (accents are allowed). Avoid non-Latin characters in the paper titles (in some cases special characters, e.g. for mathematical concepts, are allowed).

If you submit papers written in a language different to English, we must be able to verify the scientific character of the papers. Therefore, for papers written in a language different than English, **we require that you provide an English translation of the paper titles in the index file and additional English abstracts in the papers** (see Vol-1877 as example). This is not just to be able to verify the scientific character of a paper, but also to make the content of these papers accessible to the international scientific community. An English abstract and title allows any scientist to decide whether the contribution of a paper is relevant to his or her research. If interested, he or she then may contact the authors for more information.

How to publish

CEUR-WS.org is a publication service of RWTH Aachen. The hosting is provided by Informatik 5 at RWTH Aachen. The publishing workflow is provided by the members of the CEUR-WS Team, consisting of academics from all over the world. In CEUR-WS.org, **you** as proceedings editor are responsible for the quality of the material and its bibliographic metadata. The academic quality shall be safeguarded by a proper peer-review process. The metadata quality is a particular responsibility of yours. You submit the material to our server (see procedure PUT below). We provide you with the

publication tool, i.e., the WWW server of SunSITE.Informatik.RWTH-Aachen.DE. You do **not** need to ask for permission from us to submit your proceedings. **We get aware of a new submission when it is uploaded (procedure PUT). There is no pre-approval step.** We are free in our decision to accept your submission, but will follow the rules layed out in this documents. You, the proceedings editor, are responsible for having acquired the non-exclusive right for electronic publication for all published material (papers, images, metadata, etc.) from the copyright owners, in particular the papers' authors via the CEUR-WS **AUTHOR-AGREEMENT** form:

Author agreement variants

1. AUTHOR-AGREEMENT (NTP): Authors shall use this form if they included no copyrighted third party material in their paper text (or accompanying sources, datasets), and no material in the paper was produced with the help of Generative Artificial Intelligence tools including tools based on large language models (LLM). This is the right variant in most cases.
2. AUTHOR-AGREEMENT (TP): Authors shall use this form if they **did** include copyrighted third party material in their paper or accompanying material or they used Generative Artificial Intelligence tools to produce material in the paper. In case of third party material, they must then append a copy of the permission(s) by the third parties to use this material to the signed author agreement! In the case of material produced by Generative Artificial Intelligence tools, they must fill in and sign a declaration on which elements of the paper was produced by AI tools, see AI STATEMENT. This signed declaration must then be scanned and appended to the signed author agreement. Check ACADEMIC-ETHICS for our rules on including such material.

You, the editor, need to collect the correctly signed author agreements plus the editor agreement and pass it to us via the upload procedure. The name and year of the event in the editor and author agreements shall be the same the you plan to use as the full proceedings title used in the file index.html. This shall also be the name of the event specified in the footnote of page 1 of the papers. if the proceedings is a joint proceentings of multiple workshop (co-located with the same conference), the use as event name the full name of the joint proceeings title, e.g. "Joint Proceedings of the ELDC 2024 Workshops ABCD and DEFG".

We impose certain preconditions, in particular on the minimum size of a proceedings volume. Make sure your submissions fulfills the preconditions. We kindly ask you to include in your submission file a document that lists the members of the program committee, and specifies how many papers were submitted/accepted. You can include this information in a preface document, as usually done for printed proceedings.

View the index file as plain Unicode text rather than as HTML code. Your file index.html must at least contain the title of the material and the names/addresses of the proceedings editors (normally identical to the workshop organizers). It must use the style sheet `http://ceur-ws.org/ceur-ws.css` which defines some common layout for proceedings volumes. It may **not** contain or start executable code such as Java, JavaScript, ActiveX or any other type of executable code. Neither may it contain cookie definitions nor invisible pixels and the like.

HTML Validation

You can check the consistency/completeness of the submission directory by accessing it locally with your Web browser, and we require you to **validate** it using the W3C Validator. If you use RDFa tags, we ask you to validate your RDFa by using the W3C RDFa parser.

We also recommend that you check any links in your file. For example, if you have temporarily uploaded your file to your own homepage (from which it should be removed once published with us!), you can feed its URL into the W3C link validator.

Do not include script code (Javascript or similar) in your index.html file or paper files (eg. if papers are rendered as HTML).

Plain text editor

Do **not** use a Web page editor to produce index.html but rather a simple text editor like 'vi' or 'notepad'. Web page editors including Microsoft Word tend to produce unreadable HTML code which we want to avoid in CEUR-WS.org. CSS must be preferred over FONT tag. As of 2015, we accept characters beyond the 7-bit US-ASCII character set; however, the index.html file must be encoded as UTF-8 Unicode.

Please be careful in the preparation of the file index.html. Delays in publishing a volume are mostly due to errors in that file. The management of CEUR-WS.org reserves the right for adapting the file index.html to accommodate the common style of CEUR-WS.org and to include volume numbers and similar meta information.

Rules for papers in the proceedings

- A *regular paper* has at least 10 "standard" pages (1 standard page = 2500 characters) and an appropriate number of references. It shall contain enough substance that it can be cited in other publications. There may be exceptions from this rule for specific types of workshops/conferences but then please discuss the exception with us prior to the submission. Use the CEURTITLE/CEURAUTHOR/CEURPAGES labels for tagging these papers.
- A *short paper* is still a paper with references but has between 5-9 "standard" pages. Short papers shall also use the CEURTITLE/CEURAUTHOR/CEURPAGES labels in the index file. The index file shall clearly indicate whether a paper is a short paper or a regular-length paper. This can be done in different ways. For example, the whole volume could consist of short papers. Then, the full name of the volume could be "Short Paper Proceedings of ...". The short papers could also be grouped by the CEURSESSION element with label "Short Papers". If a short paper is grouped together with regular papers, its CEURTITLE can be followed by the string "(short paper)". Sometimes, the term "extended abstract" is also used as a synonym to "short paper".
- An *abstract* is a paper with a title, author and abstract but without a body and without references. Such papers are common for invited talks. Such abstracts can be included in the submission but should use a title that includes the word "abstract" or "extended abstract" like in "The Future of Data Mining - Abstract".
- A *poster paper* is a paper with a title, author and a very short text, i.e. less than 5 "standard" pages. Such papers are not regarded as citable contributions and should be handled like an abstract. Poster papers that are containing novel results (and are not summarizing results already published elsewhere) may be an exception. It depends on the substance of the content of the paper. Editors need to decide this together with the reviewers of the paper.
- A *preface* is typically a short text by the workshop/conference organizers that is part of the front matter of the proceedings. In most cases, a preface is not citable and thus you should not use the CEURTITLE/CEURAUTHOR/CEURPAGES tags for them. There is however an exception: if the preface is a longer introduction to the workshop/conference topic and has its own set of references, then it shall be treated similar to a regular paper. Give it a suitable title (not just "Preface") and authors (the workshop/conference organizers).
- A *pre-published paper* is a paper presented at a workshop/conference that has already been published elsewhere. Such papers should not be included in a volume submitted to CEUR-WS. Editors may however decide to include an abstract of the paper in the volume and prescribe appropriate paper titles that distinguish the abtract from the original paper and use the keyword "abstract" in the paper title. Note that a permission by the copyright holder is required to do so!

The page limit of 10 pages to distinguish regular from short papers is a rough indication. The workshop/conference organizers may impose their own

rules on page counts for regular vs. short papers. A paper with less than 10 "standard" pages should however not be classified as a regular paper.

You may want to use filenames like "paper1.pdf" for regular papers and "short1.pdf" for short papers. This is not a hard rule, however. Further material like bibliography file, author index, etc. can be included and linked in the index file but is never a paper. We no longer allow including a PDF of the "whole proceedings" since this can easily lead to inconsistencies and blows up the storage space. If you really want to distribute such a redundant PDF, then do so via your workshop/conference home page. You may include a link to that PDF in the index file.

The papers must be *original*, i.e. not published in an earlier workshop or conference or journal!

The papers in the proceedings should be in Portable Document Format (PDF). Prefer neutral filenames like `paper1.pdf` over content-carrying filenames like `SmithAndWagon.pdf`. **Strictly use ISO-compliant filenames and directory names!** For example, ISO does not allow blanks in a filename like in `"paper 1.pdf"`. Each paper shall correspond to single file in the volume.

Author/editor affiliations

The authors in the paper PDFs should give meaningful affiliations, which typically include the author's organization and country. A similar rule holds for the editors of the proceedings volume. The information shall allow readers to establish a contact. A few regions in the world have a disputed legal status. We welcome submissions also from such regions but their affiliation should then by default use the United Nations standard reference for the region. In some cases, an ambiguous reference, e.g. just the region name, may be used in agreement with the CEUR-WS.org editor. CEUR-WS.org adopts a conservative (i.e. like United Nations) position about the legal status. We may refuse to publish a submission if no agreement about representing the affiliation can be reached.

Local vs. absolute links

The links in "index.html" to the published material must be local, e.g.

```
HREF="paper1.pdf"
```

rather than absolute

```
HREF="http://www.dept.org/paper1.pdf".
```

Paper files and other items must be put in the main directory rather than sub-directories of the submission directory. This allows short URLs to the citable items of a published proceedings volume. An exception to this general rule are back links to workshop/conference home pages and home pages of scientific institutions (or research labs) organizing the workshop/conference. Moreover, back links to editor and author home pages are welcome. Please note however that such absolute links can and will become dangling when people change their affiliation! That's also a reason why putting the proceedings online at CEUR-WS.org is probably a better idea than putting it on your home page.

We advise proceedings editors to include a link to their workshop/conference web page in their index file of their CEUR-WS volume. This allows readers to easily locate further information about the workshop/conference such as the call for papers. We also recommend to include a back link from the workshop/conference page to the CEUR-WS volume.

Hints for Mac OS X users

The operating system Mac OS X uses case-insensitive file names. For example, a file name "PaperX.pdf" is equivalent to "paperx.pdf" on Mac-OS X. However, the CEUR-WS.org web site uses *case-sensitive* file names! Hence, make sure that the files names in your directory have the same capitalization as the URL links to them in your file index.html. The Mac OS X system apparently uses cryptic directories like ".DS_Store" or "__MACOSX". Please make sure that you do not include them in your submission file! Do not use the Mac ZIP utitily for producing the submission file.

How to deal with page numbers

In many cases, authors are interested that a published proceedings volume contains information about number of pages of their paper. Such data is typically used for evaluating the research output of academic staff. It can also be used to indicate the length of an article in a reference. While CEUR-WS.org **does not** require you to supply such data, you might be interested in how to deal with this.

Most volumes in CEUR-WS.org have no data about page numbers. So, providing them is an extra service. We have no specific knowledge about suitable tools to change page numbers or merge multiple pdf files into a single one. You might want to try Pdftk, Adobe Acrobat or CutePdf.

If you want to merge several pdf files into one and create a table of contents for the merged document, you may want to use the LaTeX macro definitions by Daniel A. Sadilek, originally used for producing the aggregated proccedings file Vol-324/dsml08.pdf.

CEURART style files for papers

From 2022 onwards, we require that authors use the new CEURART style for writing papers to be published with CEUR-WS. The style is available from Vol-XXX. An Overleaf page for LaTeX users is available at as template. Editors are encouraged create an adaption of this page for their authors, e.g. by filling in the name of the event and some other meta data, and then publish an Overleaf page specifically for their authors.

You can also download an offline version with the style files from http://ceur-ws.org/Vol-XXX/CEURART.zip. It also contains DOCX template files. You can choose between 1-column style and 2-column style. However, these should not be mixed in the proceedings volume. We may in the future prescribe that all submissions use the 1-column CEURART style! More information on the CEURART style can be found in our blog entry.

Title capitalization

The titles of papers should be either all use the emphasizing capitalized style or they should all use the regular English (or native language) style. It does not make a good impression if you or your authors mix the styles. Consider the following two titles

1. Preparing the submission file
   Ken Bar, Anne Foé
2. Filling an Author Agreement by Autocompletion
   Mary Doe, Peter Müstermann

The first title uses the regular capitalization of English whereas the second shows the emphasizing style. Both are possible but you should decide on which one you want to consistently apply to your proceedings volume. Some hints on correctly emphasizing titles in English are available at MusicBrainz. It would be great if the paper titles in the index uses the some capitalization as in the paper itself. This would require you to tell your authors what you expect before they submit the final version of the paper. In practice, this soft rule is frequently violated. The correct titles for the

above example in emphazizing style would be:

1. Preparing the Submission File
   Ken Bar, Anne Foé
2. Filling an Author Agreement by Autocompletion
   Mary Doe, Peter Müstermann

Non-Latin titles and names

index.html is required to be encoded as UTF-8 Unicode, which, in principle, enables you to use non-Latin scripts. However, in the case that you have non-Latin paper titles, we require that you *additionally* provide a Latin transliteration. Inside the machine-readable metadata (i.e. the `CEURTITLE` fields), there must be a Latin transliteration. It is strongly recommended to keep the non-Latin original text outside of these machine-readable tags.

For compatibility with publication databases, non-Latin *author names* must be transliterated into Latin. If an author already has an entry in a widely known publication database such as DBLP, it is strongly preferred to use the same transliteration that is also used there. If you think you are transliterating an author's name for the first time, it is strongly preferred to use a transliteration that is aligned with the English pronunciation, e.g. the BGN/PCGN romanization for Russian.

License footnote in paper PDFs

The PDFs (or other formats supported by CEUR-WS) of each paper to be published must contain a footnote on the first page that designates the paper as being published under the CC BY 4.0 license. The text shall be like

> "Copyright © JJJJ for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)."

The string JJJJ is to be replaced by the publication year of the paper within the volume. See exceptions for Crown or US government employees.

Creating the submission file

As soon as your submission directory is ready you should pack it into a single *submission file*. The submission file should be prepared by the UNIX command

```
zip -r sub_file.zip dir_name
```

where "sub_file.zip" is the name of the submission file and "dir_name" is the name of the submission directory. For example, the command

```
zip -r  ABCD19.zip ABCD19
```

creates a submission file for the material contained in the directory ABCD19. Use meaningful names for the submission file, e.g. the acronym of the workshop/conference. Under Windows, you can use packers like WinZIP, PKZIP or similar to produce the submission file. But beware: use ISO-standard file names! In particular, blank characters in file names are **not** allowed. Under Unix, you may also use the 'tar/gzip' commands to create the submission directory. Make sure that the submission file expands to a **directory** and not to files in the current directory. Subsequently, the procedure PUT (submit a proceedings volume to be published on CEUR-WS.org) is explained. This procedure shall be executed by one of the proceedings editors., i.e. by you. **You should in particular sign the EDITOR-AGREEMENT, and you should be the person mentioned in the clause "submitted by ..." at the end of the index file of the volume.** If you are one of the editors but the technical upload is done by a person who is not one of the editors, then you should use the clause "submitted by *your name, other name*" in the index file.

ZIP file size limitation: Our system allows maximum 100MB per ZIP file. In case of the AGREEMENTS ZIP file (step 5 below), make sure that authors use a reasonable resolution for the scan of the signed agreements. One scanned agreements should not use more than approx. 1MB space.

Location designation for disputed regions

There a few regions in the world that are disputed between countries and where the current executive government is not recognized by many other countries. CEUR-WS takes no position in such cases but we ask editors of volumes to follow these two rules:

1. We expect that proceedings editors advise authors to use neutral location names in their affiliation, if the corresponding region is disputed.
2. We require that proceedings editors use neutral location names in the main index.html file and in frontmatter/preface, if the event is organized in a disputed region.

B. Upload (Procedure PUT)

1. Collect and **carefully check** the signed AUTHOR-AGREEMENT from each contact author of a paper to be included in your proceedings. Only accept signed author agreement that were originally physically signed with an ink pen on paper (why?). Authors shall send an electronic scan (jpg or pdf) of their signed agreement to you. Keep a copy of these signed agreements for your own records. The scans of the signed author agreements must have filenames like AUTHOR-AGREEMENT-paperX.pdf, where paperX is the filename of the paper in the proceedings index file.
2. **Avoid the most-frequent mistakes**. Take this step very seriously!
3. Read the legal disclaimer of Sun SITE. Also read our rules on the limited persistency of volumes published at CEUR-WS.org. If you don't agree then do not proceed.
4. **Share the submission file with your authors** to give them a last chance to identify errors. Give them sufficient time (a few days) to report errors or to provide corrected versions of their papers. We do not monitor this step but assume that you take care of it. Note that after publication of your volume, you will only have two days to correct certain bibliographic errors in a volume! So, you need to do this quality check BEFORE the submission! See also procedure for requesting corrections.
5. Print the EDITOR-AGREEMENT, read it carefully, fill it out, and sign it **physically** (why?) with a pen on paper. The signature must be by an editor of the proceedings listed as editor in the index file! Make a scan of the signed EDITOR-AGREEMENT (format pdf or jpg); name the scanned file EDITOR-AGREEMENT-ABCD19.pdf (.jpg), where ABCD19 is the acronym of your workshop/conference. Put the scanned signed editor agreement **plus** scans of all signed author agreements (see step 1) into a ZIP file with name AGREEMENTS-ABCD19.zip. Note that you may have to include an AUTHOR-AGREEMENT for the preface written by the editors.
6. Go to https://submissions.ceur-ws.org/. Open a new ticket, select the ticket topic from "Help Topic" and choose "New submission". Fill any information needed and provide the two ZIP files for the proceedings and for the author agreements. You will receive a first email to acknowledge reception of the ticket. A second email will be sent to you when a team member will start processing your submission.

- If the submission site is offline, we have an alternative procedure: Put the submission file and the AGREEMENTS ZIP file in a temporary folder

on a web/ftp server of your organization (or a public folder of a cloud provider) and send us the URL of that folder via email. Make sure that we can access the two ZIP files without a password!

Use proper ISO-compliant document names (no blanks in the file names) for the submission file like ABCD19.zip and AGREEMENTS-ABCD19.zip for the scanned editor and author agreements. Only use ZIP format for the submission file and the agreements!

If the material is accepted by us you will receive a notification with the bibliographic reference of the material. Please be **reachable** via email the days following the upload of the submission file. We might have some questions or we may have to ask you to correct certain issues. If you do not get a response from us after more than 5 workdays, then send us a **reminder email**. We assign the next free volume number to your proceedings as soon as you upload the submission file and send the accompanying author and editor agreements. This is a mature procedure and creates least work on our side.
Note! The advance reservation of volume number is discontinued by 2021-04-01!

The actual date of execution of PUT is subject to local policy of the management of CEUR-WS.org and SunSITE.Informatik.RWTH-Aachen.DE.

How to correct errors in an already published volume

First of all, be very **careful** in preparing the submission file in order to avoid errors! Ask a colleague to double-check, i.e. that all paper files are included with the correct file names and that all author names and paper titles are correct. As a rule of thumb, we will reject requests to update an already published volume! There are only three exceptions:

1. Papers can be corrected within **two working days** after the publication of the proceedings at CEUR-WS.org. Only corrections of typographic errors in the bibliographic elements (author names, title, affiliation) are allowed. Providing improved paper versions is not allowed even if the original version was included due to a human error!
2. Forgotten papers can be included within **two working days** if they were referenced in the original index.html file.
3. Typographic errors and faulty links in the file index.html can be corrected within **two working days** after the publication at CEUR-WS.org.

The deadlines are very tight. Hence, you must check the correctness of the files before submitting them to us! Include the authors in the correctness check by letting them confirm that the right version of their paper is included. You cannot change them after submission!

Individual published papers cannot be removed from a volume. To submit the corrected version, please proceed as follows:

1. Create a new index.html file that is identical to the one currently on the server. Do not edit your old index.html file since it differs from the one that is on-line. The easiest way to create the identical file is to copy the source code of the file that is currently on-line (to do that press "View Page Source" or similar option on your web browser. Once you can see the source code, select all and then copy it to your new index.html file). Do not just download the file since that usually changes links.
2. Edit the index.html file
3. Create a simple text file changes-*currentdate*.txt, where *currentdate* is the current date (e.g. 2024-03-21), and in which you list all the changes you made, e.g.:

   ```
   corrected title of paper 3
   corrected name of author of paper 7
   corrected link pointing to the affiliation of the first editor
   ```

4. Afterwards, create a zip archive Corrections-Vol-XXX.zip containing the files index.html, changes-[currentdate].txt, and possibly new paper versions (see above). To create a zip archive use the procedure described above.
5. Send us your corrections in the same way as you uploaded the submission file in procedure PUT by using our ticket system. The easiest way is to attach the Corrections-Vol-XXX.zip to an answer to the last message on your current ticket for the volume. Your can also create a new ticket https://submissions.ceur-ws.org/open.php with Help Topic "Volume correction". Attach/drop the file Corrections-Vol-XXX.zip (XXX to be replaced by your volume number) to the ticket.

Corrections can only be requested by proceedings editors, not by authors of individual papers. Implementing corrections puts an extra burden on us. We provide this service in our spare time and get not paid for it. So, please take great care in removing errors before uploading the volume to CEUR-WS.org.

Top errors in submissions

The following simple mistakes occur in many submissions. Please check that your submission does not repeat them!

1. *Submissions using an outdated template for the index.html and pdf files.*
   Always use the latest template http://ceur-ws.org/Vol-XXX/index.html! All papers must be formatted with the CEURART style.
2. *Papers with incompatible license clauses.*
   This occurs when your authors use an unedited format file that assigns the copyright to another publisher, or uses a license clause that is too different from the standard CEUR-WS.org CC-BY clause.
3. *Inconsistent title capitalization.*
   Use the same rule for title capitalization for all papers listed in your index file!
4. *Index file contains HTML errors.*
   Check your index file with the validators before you submit your files!
5. *Incomplete or incorrect paper data.*
   Do not use author names with abbreviated given names such as "S. Writer". Always provide full author names like "Sarah Writer". Check the author name spelling with services like DBLP, if applicable. Also check the author names and paper title in the index file against the paper PDF.
6. *False LI tags.*
   Make sure that the LI tags must carry an ID tag that has a label identical to the filename without file extension, e.g.

   ```
   <li id="paper01"><a href="paper01.pdf">
   ```

7. *Redundant PDF files.*
   We no longer support the inclusion of redundant "full pdfs" for the whole proceedings. However, you can add a link to a zip of complete proceedings that we store on our server:

   ```
   <a href="http://sunsite.informatik.rwth-aachen.de/ftp/pub/publications/CEUR-WS/Vol-XXX.zip">Complete proceedings in one ZIP file</a>
   ```

8. *False paper classification.*
   If a paper is only an abstract then the title of the paper must indicate it. Similar with invited papers.
9. *Mistakes in the editor/author agreements.*
   Some editors/authors still copy/paste bitmaps of their signatures into the copyright agreements or sign on a touchscreen, instead of signing physically with a pen on paper. Further some authors forget to completely fill out the form.
   2020-03-28: Due to the fact that currently many are working in their home offices, we temporarily accept an alternative to signing the form by hand on paper. You can also fill in the form on the computer and place a hand-signed statement below the form and then take a photo of it, see example surrogate agreement.

The contact editor (the person who submitted the proceedings and signed the EDITOR-AGREEMENT) must be reachable via Email in the first couple of days after the submission to clarify issues with the submission.

Ethical issues

We assume that you, the proceedings editors, are fully aware of the copyright requirements as discussed further above in this document and have acquired the copyright from the authors of the papers/material published in your volume.

Under rare circumstances, it can happen that already published papers later turn out to be plagiated. We have a procedure to handle this and ask you to follow the corresponding rules in case that you become aware of such a case.

The papers and volumes published on CEUR-WS.org are freely accessible for academic and private use. Only such use is permitted!

Since authors retain their copyright, they are legally (in contrast to morally) allowed to re-publish their paper with a second publisher. While this appears lawful, it could be regarded as **self-plagiarism** or **double publication**. We strongly discourage such re-publication of papers that are already published with us. Likewise, do not publish papers with us that are already published by another publisher. Making a copy of the paper available on the author's home page or on the institutional repository is not regarded as a re-publication. In such cases, authors should include a reference/URL to the original publication at CEUR-WS.

AI-based writing assistance tools may only be used under very strict rules, in particular such tools may not bve used to create substantial parts of a paper's conceptual ideas. See our academic ethics page for details.

Re-publication vs. mirroring

Being freely accessible on the Internet does not imply an automatic right to re-publish/re-package CEUR-WS.org proceedings volumes (or parts of them) without authorization. Normally, only the proceedings editors have the right to re-publish their proceedings elsewhere, e.g. with a publisher or via a Web site. There are however unwritten rules for academic publishing that usually prohibit re-publications. The same paper should **not be published twice**, unless there are special circumstances such as re-publishing highly influential papers after a long time. Another reason could be that CEUR-WS ceases to exist (not to be expected any soon!).

When proceedings editors (or any entity with the right to re-publish) decide to re-publish their CEUR-WS.org volume elsewhere, they should make sure that the re-published version is clearly distinguishible from the CEUR-WS.org version. In particular, it may not use any of the following attributes characteristic to CEUR-WS.org:

- the CEUR-WS.org volume numbering CEUR-WS.org/Vol-1,CEUR-WS.org/Vol-2,...
- the label "CEUR" as part of the URL to the online location of the proceedings
- the ISSN number 1613-0073
- the CEUR-WS.org logo
- the precise wording of the standard phrase for the copyright indication
- the CEUR-WS.org style

Some proceedings editors want to distribute their proceedings via additional publication channels (e.g. printed copies or stored on electronic media or published on the workshop/conference web site) in parallel to the online version at CEUR-WS.org. This is fine with us as long as you (the proceedings editors) clearly distinguish it from the CEUR-WS version like indicated above for the case of re-publications. The re-published volume must then include a phrase like

> *Originally published online by CEUR Workshop Proceedings (CEUR-WS.org, ISSN 1613-0073)*

or similar on one of the introductory pages of proceedings volume published via the other channel. Such duplicate channels are sometimes convenient but be aware that it is potentially detrimental to the authenticity of your volume. The readers may get confused about where the volume was originally published.

Commercial and non-commercial publishers interested in re-publishing complete proceedings volumes or individual papers are advised to negotiate terms with the editors of the respective volume and/or the authors of the respective papers. CEUR-WS.org is not holding the copyright to the volumes or the individual papers. Hence, we are not a party in such negotiations. The only constraint from our side is that the re-publication is **clearly distinguishable** from the version published at CEUR-WS.org (see above). Since the CEUR-WS version of the paper is usually the original one, the re-published copy shall include a back reference to the original publication like

> *Originally published in A. Smith, B. Miller (eds.): Proceedings "Title", CEUR-WS.org/Vol-XXX*

CEUR-WS.org is not a re-publisher of already published proceedings or papers. If your proceedings have already been published elsewhere, then do not submit those proceedings to CEUR-WS.org. This rule applies when the proceedings were published with another recognized academic publisher, typically holding an ISSN or ISBN number.

We forbid third-party mirroring of the CEUR-WS.org web site or its parts. Mirroring is different from re-publication since it publishes a one-to-one copy via another Web site. An unauthorized mirror would likely violate the copyright of the CEUR-WS.org Team to its own contents and negatively affect the authenticity of CEUR-WS.org.

Like mirroring, we also disallow to integrate CEUR-WS.org or its parts as an *inner frame* inside another web site. While this is technically not an act of copying, it can give the impression as if the content of CEUR-WS.org were part of the other web site.

Author and editor rights

Authors of individual papers keep the copyright to their papers. Authors publish their papers as open-access licensed via the Creative Commons License Attribution 4.0 International (CC BY 4.0). To do so, authors need to include an appropriate footnote on their papers and sign the AUTHOR-AGREEMENT. The signed agreement shall be passed to the proceedings editors, who shall pass electronic copies of these agreements together with the EDITOR-AGREEMENT as part of the submission procedure.

Authors keep the right to publish a copy of the published **original** of their paper on their home page or via the **institutional repository** of their organization or via comparable repositories. The paper may also be publicly available via such sites. We kindly ask authors to specify in such cases the bibliographic details of the original publication of the paper at CEUR-WS.org. See below for example on how to reference the paper's bibliographic details.

Likewise, the author has the right to publish a **pre-print**, **post-print** or the **published version** of his CEUR-WS.org paper on his homepage, an institutional repository, or elsewhere. We advise authors to be careful about publishing versions of their paper that deviate from the originally published version because it could diminish the authenticity of the paper.

The editors confirm to us in the signed EDITOR-AGREEMENT that the authors continue to hold the copyright to their papers.

CEUR-WS.org acquires no rights to the papers in a proceedings volume, or to the volume as a whole, beyond the rights entailed by the author/editor agreements. **The editors of the volume hold the right to publish the complete volume**. The editors execute this right by submitting the volume on CEUR-WS.org. Authors do not have the right to demand the removal or update of their paper published at CEUR-WS.org, except for the cases mentioned in the author agreement. Any communication of authors about their papers shall instead be directed to the editors of their proceedings volume.

The **published version** of a paper on CEUR-WS.org may be watermarked by its URL at CEUR-WS.org and by the CEUR-WS logo. These watermarks assure the reader that this is the paper original. If a paper published on other repositories carries the CEUR-WS logo, then it must be 100% identical to the original published on CEUR-WS.org. Otherwise, it would violate our rights to the CEUR-WS logo!

How to reference papers published via CEUR-WS.org

A paper in a CEUR-WS.org volume should be referenced using its online URL, for example

```
M. Lenzerini: Description logics for schema level reasoning in databases.
Proc. of 1st Workshop KRDB'94, Saarbrücken, Germany, September 20-22, 1994,
CEUR-WS.org, online CEUR-WS.org/Vol-1/lenzerini.pdf.
```

Use CEUR-WS.org as 'publisher' for the paper even though the legal publsiher is RWTH Aachen. CEUR-WS.org is publication service run bu RWTH Aachen. A shorter form for citing a paper can be:

```
M. Lenzerini: Description logics for schema level reasoning in databases.
Proc. of 1st Workshop KRDB'94, Saarbrücken, Germany, September 20-22, 1994,
CEUR-WS.org/Vol-1/lenzerini.pdf.
```

A whole proceedings volume can be referenced by its ONLINE URL as follows:

```
Enrico Franconi, Michael Kifer (eds.): Knowledge Representation meets Databases 1999.
Proc. 6th Intl. Workshop KRDB'99, Linköping, Sweden, July 29-30, 1999,
CEUR-WS.org, online CEUR-WS.org/Vol-21.
```

We also assign a persistent identifier (URN) to proceedings volumes published at CEUR-WS.org. You can use the URN instead of the ONLINE URL like shown in this example:

```
Massimo Melucci, Stefano Mizzaro, Gabriella Pasi (eds.): Italian Information Retrieval Workshop 2010.
Proceedings of the First Italian Information Retrieval Workshop (IIR-2010), Padua, Italy,
January 27-28, 2010, CEUR-WS.org/Vol-560, urn:nbn:de:0074-560-7.
```

We will continue to assign ONLINE URLs to new volumes. The URN is an additional identification scheme. The URN of a published volume can be mapped to its online location via a URN resolver. For example, https://nbn-resolving.org/urn:nbn:de:0074-560-7 is resolved to CEUR-WS.org/Vol-560. The physical URL http://SunSITE.informatik.rwth-aachen.de/Publications/CEUR-WS/ should not be used for references! The URN is provided by Deutsche Nationalbiblothek. Individual papers do not (yet) have a URN.

Note that the ISSN number 1613-0073 identifies CEUR-WS.org as a publication series, not an individual volume within CEUR-WS.org! We recommend not to include the ISSN number in a citation of a paper that appeared in CEUR-WS.org. If you do prefer to include it, then insert the ISSN number after the label "CEUR Workshop Proceedings" or after the acronym CEUR-WS.org, not after the label of your workshop/conference or attached to your volume number. We do not issue ISBN numbers for volumes! Neither do we issue DOIs for individual papers or for volumes.

Here is how to cite a paper and a proceedings volume with BibTeX (assuming the BibLaTeX extension). We recommend to always create a separate `@proceedings` entry for the volume, as it will facilitate citing two or more papers from that volume. Note that some volumes come with a ready-to-use BibTeX file.

```
@inproceedings{LM:Greycite2013,
  title = {Twenty-Five Shades of {Greycite}: Semantics for Referencing and Preservation},
  author = {Phillip Lord and Lindsay Marshall},
  pages = {10--21},
  url = {http://ceur-ws.org/Vol-994/paper-01.pdf},
  crossref = {SePublica2013},
}
@proceedings{SePublica2013,
  booktitle = {3\textsuperscript{rd} Workshop on Semantic Publishing (SePublica)},
  year = 2013,
  editor = {Garc{\'i}a Castro, Alexander and Christoph Lange and Phillip Lord and Robert Stevens},
  number = 994,
  series = {CEUR Workshop Proceedings},
  address = {Aachen},
  issn = {1613-0073},
  url = {http://ceur-ws.org/Vol-994},
  venue = {Montpellier, France},
  eventdate = {2013-05-26},
  title = {Proceedings of the 3\textsuperscript{rd} Workshop on Semantic Publishing, {Extended} {Semantic} {Web} {Conference}}
}
```

Frequently Asked Questions

FAQ 1: Are CEUR-WS.org workshop proceedings automatically indexed by publication databases such as DBLP, Scopus, etc.?

DBLP indexes many CEUR-WS.org proceedings volumes. We do not have a formal relation with DBLP but cooperate with them to make it as easy as possible for them to index our proceedings volumes. However, there is no guarantee that *your* proceedings volume will be indexed automatically. You (as proceedings editor) are advised to contact DBLP yourself to check whether your volume shall be considered by DBLP for indexing. Other popular search engines and publication databases, e.g. Scopus, GoogleScholar etc. also crawl CEUR-WS.org or DBLP on a regular basis. We have no formal relation with any of them. If they have a wrong or missing entry on your publication, then you need to communicate with them, not with us! We have no communication channels to these indexers.

FAQ 2: Can I register a URN in advance?

We discontinued the pre-registration of volume numbers (and thus URNs) by 2021-04-01. You can check the URN of a volume number using the check digit calculation service of the German National Library. In this form, please enter `urn:nbn:de:0074-XXXX-`, where `XXXX` is a volume number.But note that we do no longer reserve volume numbers in advance. Hence, you can also not compute the URN in advance.

FAQ 3: I am an author of a paper published in a CEUR-WS.org volume. Can I put a copy of my paper in the institutional repository of my university/institute/organization? Can I publish it elsewhere? Can I include it in my cumulative PhD?

You hold the copyright to your paper. So, you are free to do so. You don't even have to ask us or the editor of your volume. Likewise, you can put a copy of your paper on your personal website. You can use the version of the paper published at CEUR-WS.org.

The second question is different. Putting a paper on an institutional repository is not a re-publication. It just is a copy of the same paper with the same bibliographic meta data. As a copyright holder, you could legally re-publish your paper also with another channel, e.g. by submitting it to another conference or a journal. This may legally be permitted, but it could establish a case of self-plagiarism or double publication. It is quite usual to extend a workshop or conference paper considerably and then submit it to a journal or a major conference. Some research communities, in particular from computer science, regard this as ethically acceptable as long as the extended version significantly exceeds the original paper and a reference to the original paper is given. We have no role in judging particular cases.

What concerns cumulative PhDs, there are no restrictions from our side. PhD students, who published their papers in a volume at CEUR-WS, retain the copyright to their papers. Kindly include a reference to the original publication at CEUR-WS. There is no need to ask us for permission.

FAQ 4: We are academic publishers and want to publish a book with papers on a certain subject. We like to include some papers published at CEUR-WS.org. Do you allow us to do so?

You need permission from the authors, not from us. We do not see a reason to re-publish a paper that is anyway freely available at CEUR-WS.org.

FAQ 5: We are the organizers of a workshop/conference. We plan to publish our proceedings with you. Is this possible?

Yes, just follow instructions at http://ceur-ws.org/HOWTOSUBMIT.html.

FAQ 6: How long does it take to publish a volume?

If your submission (ZIP file with papers and ZIP file with signed agreements) meets our requirements, it takes only a couple of days in most cases. In 2014 and 2015 there were only 7 out of more than 400 volumes whose publication took longer than one week; in most cases this was due to the initial submission not meeting the requirements. In recent years, the number of submissions has increased significantly, leading to somewhat longer cycle times for publishing a submission. Hence, give us at least one week time before reminding us of your submission.

FAQ 7: Why is a hand signature required on the AUTHOR/EDITOR-AGREEMENT?

CEUR-WS.org is committed to publishing proceedings with a long term sustainability, both from a legal and from a technical perspective, and in a way that works easily for a wide audience. At the time of this writing (2016), trustable digital signatures based on cryptography would require considerable technical experience and infrastructure both on the side of the volume editors and on our side. So far, a scan or photograph of a hand-written signature on printed paper is the only signature that meets all these requirements. All kinds of digital imagery that does not involve cryptography, such as hand signatures produced with a digital pen, are too easy to forge, and thus not acceptable. Note that we do not regard signatures made on touch-sensitive screens as valid.

2020-03-28: Due to the fact that currently many are working in their home offices, we temporarily accept an alternative to signing the form by hand on paper. You can also fill in the form on the computer and place a hand-signed statement below the form and then take a photo of it, see example surrogate agreement.

FAQ 8: We would like to start our own sub-series, what should we do?

An example of a sub-series is the AI*IA Series. If you are interested in having your own sub-series, you first need to make sure that you fulfill the following requirements:

- The sub-series can be granted only to an association or community with an internationally recognized scientific legacy. In addition, we require that this community has an organizational structure. The negotiations about the series should be lead by the president or a spokesperson, on behalf of the whole community.
- The community should organize and publish several **different** events per a year.
- There should be an associate editor from that community who takes over the responsibility for the sub-series.

If you complete all the above requirements, please contact us and we will take your request into consideration. Every request is considered and discussed in details by all the members of the CEUR-WS.org management team and the advisory board.

FAQ 9: We are a company working in the online advertisement industry. We like to make you a proposal for placing advertisement on your site. Are you interested?

No.

FAQ 10: Can we publish our paper or volume with a different license than CC-BY 4.0?

No. At this moment, we only support CC-BY 4.0 as the single license for both papers and volumes. The license clause in the paper must be as specified here.

Special cases: If an author is a Crown employee (e.g. UK government), the copyright of the paper may be with the Crown. Then, use the license clause "Crown Copyright © JJJJ. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)." for the paper. If the author is an employee of the US government, then there may be no copyright with the paper. In this case, use the license clause "No copyright. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).".

FAQ 11: Can we use the CEUR-WS logo on our (workshop) website?

No. The logo is reserved for official use at CEUR-WS.org. When advertising your workshop, you may use a phrase like "Proceedings shall be submitted to CEUR-WS.org for online publication" in your Call for Papers.

FAQ 12: Can authors use the CEUR-WS logo in the paper PDF?

This is allowed for the official version published at CEUR-WS. If a web-accessible version of the paper contains the CEUR-WS logo, then it must be identical to the version of the paper published on CEUR-WS. Authors can put the original CEUR-WS version of their paper in their institutional repository or their home page. If authors want to publish a version of their paper that deviates from the version published at CEUR-WS, then the CEUR-WS logo must be removed from the that version!

FAQ 13: Can you tell me how many times my paper was downloaded in the last year? I need this information for my academic assessment and/or for claiming a retribution from copyright claims associations.

No. We do not maintain such counters.

FAQ 14: What is the DBLP footprint? Why do we have to provide this information?

CEUR-WS supports the publication of computer-science workshops (and conferences). We sometimes receive submissions that apears to have only marginal relations to computer science. To check the relevance to computer science, we may request editors to provide us with data on how many publications authors and PC members have in the DBLP bibliography. Typically, we demand that each paper has at least one author with at least 5 papers listed in DBLP. PC members should have a higher number of papers in DBLP. The precise thresholds may change and will be specified in our request. The DBLP footprint ist just a simple textfile that editors add to the submission file that contains the requested data. Passing the DBLP footprint test does not guarantee that we shall publish a submission though. All other criteria must also be met. Note that you should not include ArXiv and other non peer-reviewed papers in your DBLP listing. Doctoral consortiums affiliated to well-known conferences are exempted from the rule on DBLP listed papers, but in such cases the program committee members should have a good representation in DBLP.

FAQ 15: Can organizations like companies be authors of a paper in CEUR-WS?

No. Only natural persons can be authors of a paper published in CEUR-WS. Likewise, only natural persons can be editors of a volume in CEUR-WS. AI programs can also not be authors, even when given human-like names.

FAQ 16: I used an AI program to improve the text and/or images in my paper. Is this permitted?

This depends on the nature of the AI program and the paper parts that were created with the help of AI, see also our Academic Ethics rules. For example, a tool to improve the contrast of an image is permitted because it does not change the content of the image. If the image was however generated by AI, you must use a sutaible caption including the phrase "generated by AI tool X". Generating text paragraphs or even whole sections by AI is not permitted. Improving the grammar or writing style by an AI tool is a borderline case. If it adds to or changes the content, then it is not allowed.

CEUR-WS.org Team
Sun SITE Central Europe Homepage
21-Apr-1995 (5-Jun-2024/MJ)