Information Systems and e-Business Management (2023) 21:571–601
https://doi.org/10.1007/s10257-023-00643-1

ORIGINAL ARTICLE

Reporting in large-scale agile organizations: insights and recommendations from a case study in software development

Moritz Schüll[1] · Peter Hofmann[1] · Pascal Philipp[2] · Nils Urbach[3]

Received: 12 April 2022 / Revised: 22 May 2023 / Accepted: 22 June 2023 /
Published online: 11 July 2023

Abstract

Application of agile software development methodologies in large-scale organizations is becoming increasingly common. However, working with multiple teams and on multiple products at the same time yields higher coordination and communication efforts compared to single-team settings for which agile methodologies have been designed originally. With the introduction of agile methodologies at scale also comes the need to be able to report progress and performance not only of individual teams but also on higher aggregation of products and portfolios. Due to faster iterations, production of intermediate work results, increased autonomy of teams, and other novel characteristics, agile methodologies are challenging existing reporting approaches in large organizations. Based on 23 interviews with 17 practitioners from a large German car manufacturing company, this case study investigates challenges with reporting in large-scale agile settings. Further, based on insights from the case study, recommendations are derived. We find that combining reporting and agile methodologies in large-scale settings is indeed challenging in practice. Our research contributes to the understanding of these challenges, and points out opportunities for future research to improve reporting in large-scale agile organizations by goal-setting and automation.

Keywords Large-scale agile · Software development · Reporting · Case study

✉ Moritz Schüll
moritz.schuell@fim-rc.de

[1] FIM Research Center for Information Management, Branch Business and Information Systems Engineering of the Fraunhofer FIT, University of Bayreuth, Wittelsbacherring 10, 95444 Bayreuth, Germany

[2] Department of Informatics, Technical University of Munich, Boltzmannstr. 3, 85748 Garching, Germany

[3] FIM Research Center for Information Management, Branch Business and Information Systems Engineering of the Fraunhofer FIT, Frankfurt University of Applied Sciences, Nibelungenplatz 1, 60318 Frankfurt am Main, Germany

1 Introduction

Combining agile software development methodologies and traditional reporting as a means of organizational oversight seems contradictory at first glance. While agile organizations typically focus on reporting related to the activity of software development, reporting to steer the organization and measure overall progress towards the organization's goals is often not considered to the same extent in the agile context (Moe et al. 2021). Yet we know from project management research that organizations, especially larger ones, that establish progress monitoring and reporting procedures are more likely to achieve their goals (Müller et al. 2008) and avoid unjustified resource bindings (Hoffmann et al. 2020). Further, balancing the agile autonomy with organizational oversight, as well as maintaining transparency across agile teams are known success factors for large-scale agile development (Edison et al. 2022; Moe et al. 2021). Thus large-scale agile organizations may strive to employ reporting procedures to ensure that the various agile software development initiatives contribute towards common overall goals, to evaluate performance, and to maintain transparency. Just recently, in an interview, the CIO of German consumer goods company Beiersdorf described the company's efforts to establish recurring reporting procedures as part of their digital and agile transformation, in order to make the performance and contribution to overall goals visible for every project (Herrmann 2022). Moreover, adequate reporting procedures might even catalyze the continuous improvement cycles (i.e., inspection and adaption cycles) that agile methodologies rely on.

However, while agile methodologies encourage the reduction of formal structures and processes (Beck et al. 2001), reporting seems to represent such a formalism. A core aspect of agile methodologies is to grant agile teams a high stake in decision-making processes and extensive autonomy in product development and work organization (Beck et al. 2001; Dingsøyr and Moe 2014; Kasauli et al. 2021). This autonomy is one of agile methodologies' success factors and challenges at the same time (Dikert et al. 2016; Moe et al. 2021). Coordination is known to be a major challenge in settings with self-managed groups and teams (Ingvaldsen and Rolfsen 2012; Dingsøyr et al. 2018). Research has also shown that coordination is often particularly challenging in large-scale agile organizations (Dikert et al. 2016; Nyrud and Stray 2017; Scheerer et al. 2014). Yet, agile frameworks often lack guidance and tools for the coordination of multiple teams (Dingsøyr et al. 2018). Reporting can be such a tool for coordination (Dingsøyr et al. 2018; Hackman 1986). However, reporting is a concept of formal control rooted in a traditional, manager-led type of organization (Hackman 1986; Dreesen et al. 2020). It seems paradoxical to combine reporting with agile methodologies in large-scale organizations to coordinate and steer the ensemble of agile products and teams, and to hope to profit from the best of both sides. Nevertheless, a combination of formal and informal control mechanisms in agile settings was found to have a positive impact on team performance (Dreesen et al. 2020). Further, maintaining oversight and transparency over many agile teams and projects is both a challenge and success factor of large-scale agile development (Edison et al. 2022).

Against this backdrop, we hypothesize that, while beneficial for organizational performance, the usage of reporting in large-scale agile organizations is non-trivial and leads to several points of friction in practice. To better understand these points of friction and, in a future step, be able to systematically develop approaches to address and mitigate these points of friction, we need a differentiated understanding of the combination challenges of reporting and agile methodologies in large-scale organizations. In this study, we seek to develop such an in-depth understanding. Therefore, we ask the following research question:

> *What are the challenges of reporting in large-scale agile organizations that should be considered in the development of a reporting approach for large-scale agile organizations?*

To answer this research question, we adopt a qualitative research approach and conduct an embedded single-case study (Yin 2014) at a large German car manufacturing company that transitioned its whole IT department to agile methodologies. The case study helps us gain in-depth insights into the challenges and approaches of practitioners in an organizational environment. We focus on understanding the challenges that practitioners are facing with regard to reporting in their large-scale agile programs. Further, in our case study, we discuss and document recommendations on how large organizations can balance agility and reporting.

We contribute to the academic discourse on large-scale agile software development by expanding the understanding of the challenges arising from the integration of large-scale agile methodologies and reporting. This novel understanding uncovers promising paths for further research at the intersection of agile methodologies and reporting in large-scale organizations. We also provide guidance for practitioners in large-scale organizations seeking to combine agility and reporting.

The remainder of this paper is structured as follows. In Sect. 2, we provide an overview of relevant literature and concepts for this study. Section 3 thoroughly describes the case study research approach. After a description of the case organization and context of the study in Sect. 4, we present our main findings in Sect. 5. Finally, the discussion of our findings is followed by the conclusion of our study in Sect. 6.

2 Background

2.1 Large-scale agile methodologies

As of today, agile methodologies have become the dominant type of software development methodologies (digital.ai 2021). While traditional methodologies consider the software development process to be of defined nature, agile methodologies commonly consider software development an empirical process (Williams and Cockburn 2003). Empirical processes require frequent feedback and adjustment to changing environments to allow for adequate reaction to unpredictable demands and requirements (Williams and Cockburn 2003). Agility is the organization's ability to sense changes and opportunities and to seize these opportunities by quickly adapting to changes (Sambamurthy et al. 2003). To achieve this, agile methodologies focus on the core concepts of incremental design and iterative product development, inspect and adapt cycles, close collaboration and communication in teams, and continuous customer involvement (Baham and Hirschheim 2022). These key characteristics of agile methodologies make them suitable for projects with unforeseeable or changing demands, which is the case for many software development projects (Williams and Cockburn 2003).

Following the publication of the Agile Manifesto in 2001 (Beck et al. 2001), over the past 20 years, agile methodologies have gained significant traction. Today, a multitude of frameworks for agile development exists, e.g., Scrum (Schwaber and Beck 2002) and XP (Beck 2000). Even though agile methodologies were originally designed for small-scale organizations (Abrahamsson et al. 2009; Rolland et al. 2016; Reifer et al. 2003), their shown benefits of increased customer value, increased flexibility, and frequent product delivery are making them increasingly popular in larger organizations (Dikert et al. 2016; Wińska and Dąbrowski 2020). Several scaling agile frameworks have emerged that focus on the application of agile methodologies in large-scale environments with multiple products and development teams (Uludağ et al. 2021). The Scaling Agile Framework (SAFe) (Leffingwell 2018) and Scrum of Scrums (Schwaber 2007) are among the most popular scaling agile frameworks in practice (digital.ai 2021).

Likewise, the research area around large-scale agile development (LSAD) has seen steadily increasing activity (Uludağ et al. 2022). Research on large-scale agile development has been increasingly based on empirical insights over the past few years (Edison et al. 2022; Baham and Hirschheim 2022). However, many of these studies have shown that adopting agile methodologies in large-scale organizations is a non-trivial endeavor (Paasivaara 2017; Boehm and Turner 2005). Literature has documented various challenges as well as success factors for large-scale agile development and the adoption of large-scale agile methodologies (Edison et al. 2022; Dikert et al. 2016; Conboy and Carroll 2019; Uludağ et al. 2018). These challenges often are independent of particular frameworks but are rather related to aspects such as common understanding of concepts (Conboy and Carroll 2019), resistance to change (Kalenda et al. 2018), maintaining the autonomy of developers and teams (Conboy and Carroll 2019), coordination and communication between teams (Kischelewski and Richter 2020; Dikert et al. 2016), or performance and progress measurement (Kalenda et al. 2018; Uludağ et al. 2022).

For the term *large-scale agile*, however, no final definition has been established yet. Research so far has proposed varying definitions (Conboy 2009; Dikert et al. 2016; Edison et al. 2022). Some definitions of large-scale agility focus on certain attributes such as project cost or project duration (Dikert et al. 2016; Uludağ et al. 2021). Limaj and Bernroider (2022) propose, that scaling agility in general refers to the process of expanding the initial application of agile concepts from isolated instances (e.g., individual teams or units) to further areas of an organization. In line with the mapping study by Uludağ et al. (2022), we define the term *large-scale agile* as environments in which either multiple agile teams are collaborating or the agile methodologies are adopted on the organizational level comprising multiple multi-team settings. We deem this definition appropriate for our study because the number of teams and organizational levels influences the complexity of reporting structures in organizations (i.e., the coordination overhead, cf. Dingsøyr et al. (2014)).

2.2 Reporting in agile settings

Research in the project and portfolio management domain has intensively studied reporting in organizations (Müller et al. 2008). Reporting in an organization can be a useful tool to identify potential for improvement, especially in larger, more complex organizations with multiple teams and projects. A shared reporting approach was found to help organizations in large-scale environments achieve their goals by channeling information from lower organizational levels to higher levels (Müller et al. 2008).

Reporting can therefore contribute to transparency and keeping an overview across multiple teams in large organizations, which is both a challenge and a success factor for large-scale agile development (Edison et al. 2022). Literature focusing on reporting in large-scale agile settings, however, is scarce (Stettina and Schoemaker 2018). Stettina and Schoemaker (2018), as the only study to the knowledge of the authors, investigated the types of reporting that are typically present in agile organizations. One of their key findings is that reporting responsibility in agile settings does significantly differ from traditional project management frameworks (such as PMBOK-Guide and PRINCE2), where a project manager usually bears the main reporting responsibility. In agile settings, Stettina and Schoemaker (2018) describe that knowledge and reporting responsibility is divided into three parts: (1) product and portfolio, (2) development, and (3) process responsibility. Further, each of these reporting responsibilities is accounted for by a different agile role. Agile teams are responsible for development reporting, Product Owners for product reporting, and Scrum Masters for process reporting. In Sect. 4, we use these insights by Stettina and Schoemaker (2018) to structure and describe the reporting approaches that we find at the case organization.

Apart from the study by Stettina and Schoemaker (2018), our literature search has not identified any other study that addresses the aforementioned research gap of how to balance reporting and agility in large-scale agile organizations.

3 Research approach

To answer the research question, we conducted an embedded single-case study (Yin 2014). We chose the case study methodology based on the following rationals, as suggested by Benbasat et al (1987) and Yin (2014). The research question is a *how* question and our research does not require behavioral control, because it seeks to study a real, productive organization applying large-scale agile software development. We are interested in active organizations, hence the focus clearly is on contemporary events. And finally, large-scale agile development is still a novel area of research lacking an established theoretical base (Uludağ et al. 2022; Dikert et al. 2016).

The case company is a large car manufacturing organization. For a detailed case description, we refer to Sect. 4. We selected this organization because we are interested in an organization that already has established large-scale agile initiatives and seeks to implement or improve reporting. This allows us to investigate the reasoning behind the selection of these routines and procedures, and to be able to gather challenges specific to combining reporting and agile methodologies in large-scale initiatives. An organization that started with agile methodologies from scratch most likely would have lacked a basis of comparison to non-agile settings.

We collected and analyzed data in two phases. After describing the rationale for our two-phased approach, we provide more details on how we collected and analyzed data in each of the two phases.

In the first phase, we conducted 12 semi-structured interviews and collected additional third-degree case data in the form of documents, pages from the corporate wiki, backlogs, and presentation slides (Runeson and Höst 2009). The interview guides are documented in Appendix 1. The objective of the first phase was to identify how the case organization is currently implementing reporting in their large-scale agile initiatives, which challenges practitioners are facing with these current approaches, and which potential solutions they had already tried or think might help to address the challenges.

In the second phase, we conducted another 11 interviews. The objective of this phase was to discuss and evaluate the challenges from the first phase. Using this two-phased approach, we sought to improve on the validity of our study by letting informants from the first phase and further practitioners review the challenges. Further, by combining data from interviews and third-degree data, we achieve data source triangulation (Yin 2014; Runeson and Höst 2009). Both phases combined, the implementation of the case study at the partner organization spanned a duration of six months. In total, we conducted 23 semi-structured interviews with 17 practitioners, following the guidelines by Runeson and Höst (2009).

3.1 First phase of the study

We conducted a short preliminary discussion with each of the interviewees to explain the background of our study and to ensure that interviewers and interviewees had a common understanding of agile methodologies and reporting. The semi-structured interviews in the first phase followed the interview guide in Appendix 1. After some general information about the interviewee, the company, and the development program, we asked the interviewees about the reporting approaches used on the different organizational levels in the case organization. Further, we asked interviewees which challenges they are facing regarding these approaches and how, in their opinion, one could address these challenges.

Interview participants for this first round were sampled from eight different large-scale agile programs at the case organization. Table 1 provides details on the interview participants. The initial sampling was intentional (Runeson and Höst 2009) and

Table 1 Participants of the first round of interviews

No.	Alias	Role	Scaling level	Large-scale agile development experience	Softw. dev. experience	Duration (h:m)
1	AM1	Agile master	Program	3–5 years	3–5 years	1:42
2	PO1	Program owner	Program	3–5 years	11–15 years	1:10
3	BE1	Business expert	Portfolio	6–10 years	6–10 years	1:29
4	AM2	Agile master	Portfolio	3–5 years	6–10 years	1:21
5	PO2	Program owner	Program	3–5 years	11–15 years	1:00
6	PO3	Program owner	Program	3–5 years	11–15 years	0:52
7	AM3	Agile master	Program	3–5 years	11–15 years	1:11
8	STE1	Solution train engineer	Portfolio	3–5 years	6–10 years	ca. 2:00 (not recorded)
9	LM1	Line manager	Team	6–10 years	6–10 years	1:15
10	AM4	Agile master	Program	3–5 years	>20 years	1:31
11	AM5	Agile master	Program	11–15 years	11–15 years	1:24
12	DEV1	Developer, software architect	Team	1–2 years	6–10 years	1:23

happened primarily via the network of Agile Masters at the case organization. The network of Agile Masters is mainly concerned with work processes and methodology improvement and is represented across all organizational scaling levels. They are also involved in the goal-setting and reporting processes, hence they served as a natural first sampling source. After the first interviews had been conducted, we made use of snowball sampling and contacted further potential interviewees who were recommended to us. Further, during and after the interviews we collected potentially relevant documents, which were provided to us by interviewees or were available on the development programs' intranet pages.

All the collected data, i.e., interview transcripts and third-degree data, were analyzed and coded using the qualitative data analysis software tool MAXQDA. The coding was primarily done by one of the authors of this paper, and regularly reviewed by and discussed with another author. The analysis and coding followed the guidelines by Miles et al. (2013). An initial immersion of the researchers was achieved by exploring and reading the whole data set (Miles et al. 2013). After this initial immersion, the first cycle of coding was conducted. In the first cycle, a descriptive coding technique was applied, assigning codes to significant chunks of data that summarize the chunk in a short phrase or word (Miles et al. 2013). Descriptive coding was chosen as it allowed us to create an initial inventory of topics that could be used as a basis for the second cycle of coding to uncover patterns across all the different data (Miles et al. 2013). The codes were created using an integrated approach (Cruzes and Dyba 2011; Miles et al. 2013). A provisional starting list of code categories was created deductively by the researchers, based on the general structure of the interviews and the categories of concepts relevant to the research project (Cruzes and Dyba 2011; Miles et al. 2013). The individual codes then emerged inductively during the process of analysis and coding, reflecting the encountered concepts and patterns in the data (Cruzes and Dyba 2011; Miles et al. 2013). This integrated approach was chosen because the categories of relevant concepts (i.e., reporting procedures, challenges, and potential solutions) were already known before data collection, but the actual concepts should emerge from the data itself (Cruzes and Dyba 2011; Miles et al. 2013). After the first cycle, the second cycle of coding was conducted (Miles et al. 2013). The second cycle builds 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). Figure 1 shows an example of the used coding system. In the first round, we created 55 different descriptive codes, of which 27 codes referred to reporting approaches, 28 codes to challenges, and 35 codes to potential solutions. We subsequently grouped them into pattern codes, respectively. Using the codes from the analysis, we then derived the textual descriptions of the challenges.

3.2 Second phase of the study

In the second phase of the case study, we then went back to the case organization to conduct another round of 11 semi-structured interviews in order to validate our identified challenges. For each of our challenges, we asked interviewees to explain to us whether they agree or disagree with our findings. Further, we asked

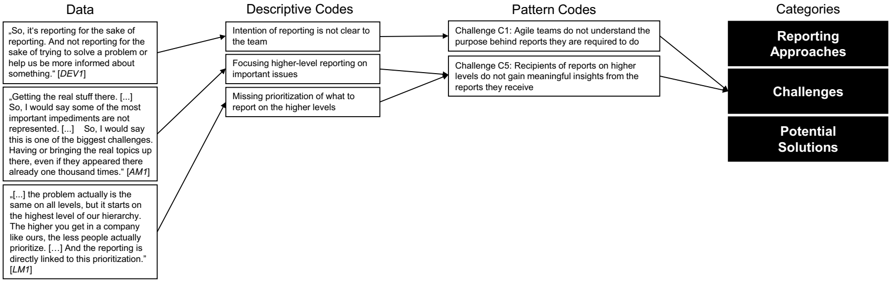
Data
Descriptive Codes
Pattern Codes
Categories
„So, it's reporting for the sake of reporting. And not reporting for the sake of trying to solve a problem or help us be more informed about something." [DEV1]
„Getting the real stuff there. [...] So, I would say some of the most important impediments are not represented. [...] So, I would say this is one of the biggest challenges. Having or bringing the real topics up there, even if they appeared there already one thousand times." [AM1]
„[...] the problem actually is the same on all levels, but it starts on the highest level of our hierarchy. The higher you get in a company like ours, the less people actually prioritize. […] And the reporting is directly linked to this prioritization." [LM1]
Intention of reporting is not clear to the team
Focusing higher-level reporting on important issues
Missing prioritization of what to report on the higher levels
Challenge C1: Agile teams do not understand the purpose behind reports they are required to do
Challenge C5: Recipients of reports on higher levels do not gain meaningful insights from the reports they receive
Reporting Approaches
Challenges
Potential Solutions

Fig. 1 Example for the used coding system

them to recommend and discuss potential actions to address the identified challenges. The challenges hence served as the interview guide.

For this second round of interviews, we resampled five of the interviewees from the first round at random. We further contacted six additional, new interviewees to also collect unbiased feedback. In total, in the second round we interviewed eleven participants. All the potential interviewees that we contacted agreed to participate. Details on the participants of the second round of interviews are depicted in Table 2.

Again, the interviews were transcribed and coded using the same approach as in the first round of interviews. Based on the coding and analysis of the second round of interviews we refined our challenges and documented the recommendations to reflect practitioners' feedback. As a result, we reduced the number of challenges from nine to seven by merging two challenges into another one. The

Table 2 Participants of the second round of interviews (participants that did not take part in the first round of interviews are outlined in bold)

No.	Alias	Role	Scaling level	Large-scale agile development experience	Softw. dev. experience	Duration (h:m)
1	PO1	Program owner	Program	3–5 years	11–15 years	0:25
2	BE1	Business expert	Portfolio	6 - 10 years	6–10 years	0:38
3	AM2	Agile master	Portfolio	3 - 5 years	6–10 years	0:52
4	**PO4**	Product owner	Team	3–5 years	6–10 years	0:41
5	**PO5**	Product owner	Team	6–10 years	11–15 years	0:37
6	**AM6**	Agile master	Program	6–10 years	11–15 years	0:37
7	**PO6**	Portfolio owner	Portfolio	3–5 years	>20 years	0:38
8	LM1	Line manager	Team	6–10 years	6–10 years	0:38
9	**AM7**	Agile master	Program	6–10 years	>20 years	0:38
10	DEV1	Developer, software architect	Team	1–2 years	6–10 years	0:27
11	PO3	Program owner	Program	3–5 years	11–15 years	0:25

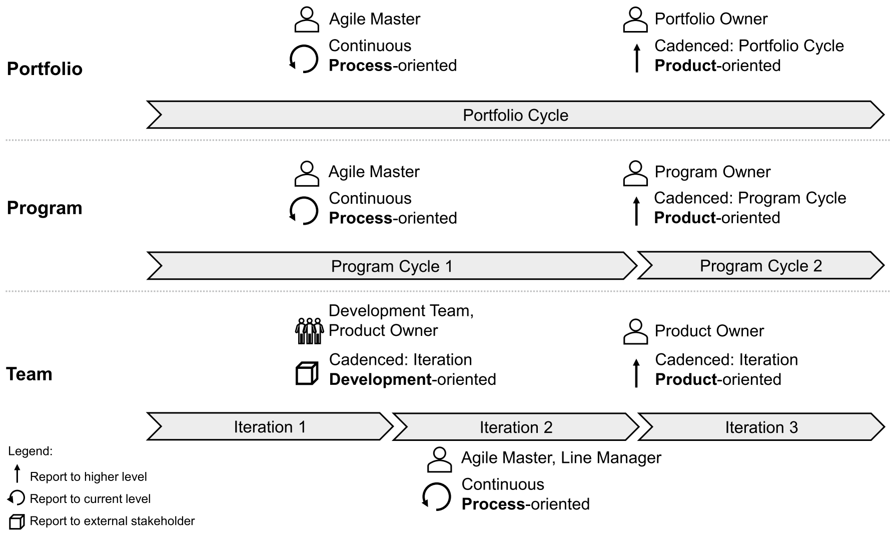

Fig. 2 Visualization of the observed types of reporting on each scaling level, based on the observations from the case study

recommendations were consolidated into a final set of three. These final, refined results are described in Sect. 5.

4 Case description

The industry partner for the case study is a large German car manufacturing company. The organization has well above 100.000 employees, generated a revenue of roughly 100 billion EURO in 2020, and is operating internationally. The case organization has been using agile development methodologies for over 6 years. Large-scale agile programs (in accordance with our definition) have been existing for about 3 years as of 2021.

In general, the organization is structured along major processes that are relevant for auto-making. These processes represent important steps in the value chain of the organization. For each process, the value proposition and especially the served customers are different. The IT department of the organization spans all these processes. This is the case because IT services are necessary in all business functions and process steps of the case organization.

Inside the different departments of the organization, the general organizational structure is standardized and independent of specific agile frameworks. This, in particular, refers to the organizational scaling levels and the roles present at each of these levels. Figure 2 gives an overview of the scaling levels and the different types of reporting at the case organization. The portfolio is the highest organizational scaling level at the case organization. A portfolio contains multiple programs. A program itself usually consists of multiple products. Each product can be developed by one or multiple Agile Teams. The programs that were observed in this study, and from which we sampled interviewees, are all developing software for different purposes in the case organization. These products include software used inside of the produced cars, such as some of the autonomous driving functionalities, as well as software that is used for producing and selling cars, e.g., shopfloor management software or sales software for dealers. Further, the considered programs are using practices from different agile frameworks. While some programs are applying practices from the SAFe framework, such as Program Increment (PI) Plannings, others are using ceremonies from the LeSS framework, for example. Overall, all the observed programs were either using practices from the frameworks SAFe or LeSS, a mixture of practices from these two frameworks, or no specific framework at all. While the case organization does not prescribe which practices can be used, certain roles and responsibilities are standardized across the different programs. In the following sections, we elaborate on these roles at the case organization as well as the types of reporting they are responsible for.

4.1 Agile masters and process-oriented reporting

At each of the organizational scaling levels, an Agile Master is facilitating continuous improvement of the working methodologies. This role is very similar to the Scrum Master role in Scrum but is always present regardless of which agile methodologies or frameworks are applied. In line with the findings by Stettina and Schoemaker (2018), at the case organization, the Agile Masters and sometimes also the Line Managers are responsible for process-oriented reporting. This type of reporting is automated to a high degree, e.g., relying on automated Jira Dashboards and tool-generated reports (*AM2, AM3, AM4, AM5*). It mainly serves the purpose of internal information and facilitating planning and continuous improvement.

At the team level, continuous, process-oriented reporting is the dominant type of reporting. Interviewee *AM5* explained why this is the case:

> *So, in terms of evaluation of the team, it is much more important to focus on the process side of things rather than the output side of things. Because the output is always a result of how well the team operates, and you can't just force the team to produce more. There is always an underlying reason as to why they are potentially producing less. So, that's what we focus on in terms of reporting on the team level [...].*

At the program and portfolio level, the Agile Masters are focusing on trend analysis of several metrics over individual measurements at one point in time in their reporting. For this purpose, Value Stream Dashboards are used in several programs, which are based on data-warehousing approaches for Jira. This allows for sophisticated analysis of trends and variances in Backlogs over time. Further, at the program and portfolio level, Agile Masters are continuously employing automated quality checks on the Backlog Items. They generate statistics on whether

Backlog Items contain proper linkage to parent items, a textual description, estimation in Story Points, definition of acceptance criteria, linkage to addressed defects, and more. Those checks are monitored for Backlog Items in both portfolio and program cycles.

4.2 Product owners and product-oriented reporting

Further, a structure of Product Owners is present across the scaling levels at the case organization. They are in charge of work content and prioritization at different levels. Again, in line with Stettina and Schoemaker (2018), Product Owners at the case organization are responsible for product-oriented reporting. In the case organization, this type of reporting is mostly done via Program and Portfolio Review meetings, as suggested by scaling agile frameworks LeSS (Larman 2016) and SAFe (Leffingwell 2018). The higher in the organizational structure, the more important this type of reporting is. Product-oriented reporting is mainly compiled manually (*PO1, BE1, AM2, LM1*).

Reporting based on previously defined goals plays a major role in product-oriented reporting at the case organization. Goal-based reporting is mandatory at the portfolio level to show progress towards the highest-level goals but is also frequently used at the program level, as explained to us by interviewees *PO1, BE1, AM2,* and *LM1*. This type of reporting is required by top management (*PO1*). Management uses the aggregated insights gained from this reporting to steer the organization and to evaluate the status towards the overall goals of the organization. The rationale behind this mandatory reporting is to have a common baseline of reporting across all programs at the case organization, that allows management to keep an overview. For this purpose, it is mandatory to use standardized document templates at the portfolio level. Approaches to how these templates are filled, however, differ between programs. They are filled in Portfolio Reviews, Program Reviews, Area Retrospectives, PI Plannings, and similar events. This largely depends on the scaling agile framework chosen by a particular program.

4.3 Agile teams and development-oriented reporting

Finally, development-oriented reporting is the responsibility of the Agile Teams at the case organization. Again, this matches the findings by Stettina and Schoemaker (2018). The most common means of development-oriented reporting at the case company is the Sprint Review. It was described to us by interviewees *LM1, BE1, PO3, AM1, AM2, AM3,* and *AM5*.

As outlined in the previous paragraphs, the separation of reporting responsibility at our case organization matches the findings of Stettina and Schoemaker (2018). However, in our case, the separation between development- and product-oriented reporting responsibility is not as clear-cut as it is for process reporting responsibility. We rather find that the separation between development- and

Table 3 List of challenges identified in the case study

ID	Name	Description	Interviewees
Challenges of the creator of a report			
1	Agile teams do not understand the purpose behind reports they are required to do	We observed that teams often are not fully aware of a clear purpose behind a certain report. In these cases, teams perceive reports as reporting for the pure sake of management control	*DEV1*
2	Agile teams lack context information to clearly report how their individual team work contributes to overall product and portfolio goals	We observed that reports of agile teams at the case organization often lack expressive power because the reporting agile team is not able to contextualize the work they are reporting on in the larger program or portfolio environment. Reports assembled by individual agile teams do not clearly highlight how the team's work contributes to the higher-level goals of the program or portfolio at the case organization	*PO1*, *STE1*
3	Reporting demands are limiting the autonomy of agile teams	While teams in agile settings are granted extensive autonomy in planning and executing their work, the large-scale organizational structures at the case organization demand a certain level of reporting to be able to coordinate work and keep track of overall progress. Such reporting demands are "imposed" onto teams at the case organization, which limits teams in their autonomy because they feel pressured to plan and execute their work according to the goals they are required to report towards	*AM4*
4	Increased frequency of inspect and adapt cycles increases the efforts of reporting and automating reports	Caused by the increased frequency of inspect and adapt cycles that is central to agile methodologies, the usual efforts of gathering the data necessary for reporting now have to be undertaken several times more often in a certain period of time than before. At the case organization, a typical reporting cycle on the program level that used to last a quarter of a year now contains four program cycles of three weeks each—each of which requires a program-level report at the end. Further, due to the increased frequency of inspect and adapt cycles and the resulting goal adaptations, existing reports also need to be adapted more often than before the introduction of agile methodologies. However, these frequent changes also inhibit the automation of reports	*BE1*, *PO2*, *LM1*, *DEV1*, *AM4*

Table 3 (continued)

ID	Name	Description	Interviewees
Challenges of the recipient of a report			
5	Recipients of reports on higher levels do not gain meaningful insights from the reports they receive	We observed missing expressiveness of aggregated reports on higher levels in the organization, and interviewees explained to us that important topics from the lower organizational levels often are not represented correctly or not at all by reports on the higher levels. Reports on the higher organizational levels are either too detailed and thus too extensive to grasp, or only represent the issues and prioritization from a subset of all involved teams. Thus, they fail to direct management's focus on the few most central topics	*AM1*, *STE1*, *LM1*
6	Increasingly large scale of agile organization causes delays in the reporting chain	The large size of the case organization often causes reports from lower levels to reach higher levels with a certain delay, because the reported information is reused in another reporting cycle on the next higher level. Such delays are causing the agile inspect and adapt cycles to lag behind the current state of work at the case organization. Ultimately, this might limit the scope of action compared to a more timely report and reaction	*PO2*

Table 4 Determining aspects and impacts of the challenges related to large-scale agility

ID	Name	Aspects of the agility of the organization that determine the challenge	Aspects of the large scale of the organization that determine the challenge	Impacts of the challenge on the organization
1	Agile teams do not understand the purpose behind reports they are required to do	No apparent connection of reports to the continuous improvement process of agile teams	Communication gap between the program / portfolio level and the team level	*Team-level:* Agile teams perceive reporting as a pure control mechanism
2	Agile teams lack context information to clearly report how their individual team work contributes to overall product and portfolio goals	Agile teams autonomously set their goals	Communication gap between the program / portfolio level and the team level	*Team-level:* Agile teams struggle to report their contribution to overall program or portfolio goals
3	Reporting demands are limiting the autonomy of agile teams	Agile teams want to maximize value delivery for their local customer	Program and portfolio need to coordinate their teams and ensure a common direction	*Team-level:* Agile teams are limited in their full autonomy on setting their own goals
4	Increased frequency of inspect and adapt cycles increases the efforts of reporting and automating reports	Frequent inspect and adapt cycles are central to agility and require frequent reporting to gather feedback	–	*Program- and portfolio-level:* The program and portfolio are required to adjust their reports more frequently to changing goals compared to non-agile settings. To allow for some automation, the program and portfolio also have to ensure a certain consistency of the used tools across their agile teams
5	Recipients of reports on higher levels do not gain meaningful insights from the reports they receive	Agile teams autonomously set their goals	Program and portfolio have to keep an aggregated overview across the work of their teams	*Team-level:* Aspects or issues important to the individual agile teams are under- or misrepresented in higher-level reports. *Program- and portfolio-level:* Higher-level reports are either too broad, i.e., too extensive to grasp, or too narrow, i.e., not covering all high-priority aspects
6	Increasingly large scale of agile organization causes delays in the reporting chain	–	*Large-scale:* Information takes time to travel across the organizational hierarchies	*Team-level:* Delayed reports and feedback inhibit inspect and adapt cycles and continuous improvement, and thus the agility, of the agile teams

product-oriented reporting responsibility becomes more evident the higher the organizational scaling level at which the reporting is done. This is because Product Owners on team-level are involved in development-oriented reporting as well as product-oriented reporting, while Program Owners and Portfolio Owners almost exclusively focus on their product-oriented reporting responsibility.

5 Findings

In this section, we describe the reporting challenges that we identified at the case organization. Table 3 summarizes all the identified challenges. Further, Table 4 presents aspects of the agility and the large scale of the case organizations, that each of the identified challenges relates to. We clustered the challenges into the reporting and receiving sides.

5.1 Challenges of the creator of a report

5.1.1 Agile teams do not understand the purpose behind reports they are required to do (C1)

We observed that teams often are not fully aware of a clear purpose behind a certain report, and lack a clear link to their continuous improvement efforts. Interviewee *DEV1*, a member of an agile development team, summarized that "*[...] sometimes reporting feels to be there for the sake of reporting. [...] And not reporting for the sake of trying to solve a problem or help us be more informed about something. That's kind of, like, maybe sometimes a solution in search of a problem*". In these cases, teams perceive reports as reporting for the pure sake of management control, as explicated to us by interviewee *AM4* with reference to an automatic, tool-based report used in their program:

> *I mean, also there [...] the Domain [i.e., portfolio] starts to do some basic checks in all the Jiras of the Products. But this is sometimes really seen as a kind of control effort by management and is not really seen very well by the teams, that management wants to look into their Jiras, if their Jiras have good quality.*

While transparency is a central value to many agile methodologies (e.g., SAFe states transparency as a core value (Leffingwell 2018; Edison et al. 2022)), agile teams at the case organization seem to be uncomfortable with their stakeholders' ability to look at their work and documentation tools at any time.

To summarize, the challenge is determined by a combination of aspects that relate to both the large scale and the agility of the organization. The challenge is determined by the agility of the organization because the agile teams do not see the benefit of the reports to their continuous improvement efforts, which is a central aspect of agile methodologies. Further, the challenge is also determined by the large scale of the organization, because a communication gap is occurring between the agile teams and the higher levels in the organization (program, portfolio). Through the combination of these aspects, the impact of the challenge is seen in agile teams, who perceive the reports as a pure control mechanism.

5.1.2 Agile teams lack context information to clearly report how their individual team work contributes to the overall product and portfolio goals (C2)

We observed that reports of agile teams at the case organization often lack expressive power because the reporting agile team is not able to contextualize the work they are reporting on in the larger program or portfolio environment. Reports assembled by individual agile teams do not clearly highlight how the team's work contributes to the higher-level goals of the program or portfolio at the case organization. Often, also the recipient of a report themselves lacks knowledge of how the progress that is reported to them fits into the larger context. Interviewee *PO1* largely attributes the lack of context to missing top-to-bottom reporting procedures at the case organization:

> *What I am missing, is the connection. You're part of [program name], what's the next step? What's important — I keep asking this question, but I don't always get the link — what corporate goal am I actually contributing to and how? [...] On the one hand, the link to the corporate strategy itself. And the other is that reporting should not always be in one direction, but also in the other. This is something that is often lacking.*

While lower levels are frequently and systematically reporting to their next higher levels at the case organization, higher-level stakeholders often fail to pass down context information of overall strategy and goals in a similarly frequent and systematic manner.

Interviewee *STE1* mentioned, that Backlog Items at the team level frequently lack a linkage to higher-level goals. As a result, the member of the agile team that gets assigned said Backlog Item does not have transparent documentation of how their work fits into the larger organizational context.

To summarize, the challenge is determined by a combination of aspects that relate to both the large scale and the agility of the organization. It is determined by the agility of the organization, because, due to the agile teams' goal-setting autonomy, teams have to make sure themselves to set goals that contribute to the overall progress of the program or portfolio. The challenge further is also determined by the large scale of the organization, because a communication gap is occurring between the agile teams and the higher levels in the organization (program, portfolio). Through the combination of these aspects, the impact of the challenge is seen at the agile teams, who have problems in clearly reporting how their progress towards their own goals contributes to program or portfolio goals.

5.1.3 Reporting demands are limiting the autonomy of agile teams (C3)

At the case organization, several programs are struggling with finding the right balance between required reporting and desired agility. Interviewee *AM4* stated that "*the reporting doesn't always fit to the agile working model, I would say. That's something which comes out of a different kind of organization.*" While teams in agile settings are granted extensive autonomy in planning and executing their work, the large-scale organizational structures at the case organization demand a certain level of reporting to be able to coordinate work and keep track of overall progress. Teams plan the goals for their sprints autonomously, but they are required to also report progress towards higher-level goals of the organization. As such reporting demands are imposed onto teams at the case organization, teams feel limited in their autonomy and pressured to plan and execute their work according to the higher-level goals they are required to report towards. Interviewee *AM4* stated this challenge is especially hard to overcome in organizations that transformed from traditional project management to agile methodologies because people are used to the "old" way of reporting. As a consequence, the teams feel limited in their autonomy:

> *I think one big thing that's going on here is just the kind of mismatch again between old management structures and an interest of management to control, and then autonomy on the team and the product level. And there is quite a wish on the product level and team level to be autonomous and it's a quite common statement by Product Owners to say 'hey, look at what I am delivering every quarter, all of it is fine, why do you want so much reporting? If I wouldn't have to do that reporting, I would be able to implement even more within my product. But there is so much effort on reporting, that hinders me in achieving more goals'.* — Interviewee *AM4*

To summarize, the challenge is determined by a combination of aspects that relate to both the agility and the large scale of the organization. On the one hand, it is determined by the agility of the organization because agile teams want to maximize the value delivery to their customer. On the other hand, the challenge is also determined by the large scale of the organization, because the program and portfolio have to ensure that the work of the individual agile teams is coordinated towards a common direction. This is realized by posing reporting demands to the teams. Through the combination of these aspects, the impact can be seen at the agile teams, who, as a result, are limited in their goal-setting autonomy.

5.1.4 Increased frequency of inspect and adapt cycles increases the efforts of reporting and automating reports (C4)

While gathering the necessary data for reporting is often already demanding in traditional large-scale organizations (*BE1*), we observed that this challenge was further amplified by the introduction of agile methodologies at the case organization. Caused by the increased frequency of inspect and adapt cycles that is central to agile methodologies, the usual efforts of gathering the data necessary for reporting now have to be undertaken several times more often in a certain period of time than before. At the case organization, a typical reporting cycle at the program level that used to last a quarter of a year now contains four program cycles of three weeks each — each of which requires a program-level report at the end. Interviewee *PO2* explained to us:

> *It's really crucial that these artifacts are kept up to date. That's also quite a challenge because it's also some work to do in order to keep the stuff up to date. We have more than 80 to 100 Sagas within each Domain [i.e., portfolio] cycle. Then, within each Saga we derive, let's say, two or three Epics. And within each Epic we have 15 to 20 Stories. So, each of these artifacts needs to be kept up to date. [...] And there's of course the whole review meetings, the preparation of the review meetings, the preparation of the slides, and all this stuff, this takes a lot of time and is not easy sometimes to do. All the preparation of all that stuff.*

The impacts of this challenge, while present for agile teams also in small-scale organizations, can mostly be seen on the program and portfolio level. The introduction of a higher frequency of reporting cycles, tied to the inspect and adapt cycles of the agile methodology, also increases the efforts of reporting on the higher levels in the organizations compared to non-agile times.

To aggravate this challenge, with the large-scale application of agile methodologies also the efforts of automation are increasing on the higher levels in the organizations (*LM1*, *DEV1*, *AM4*). On the one hand, process-oriented reports that collect data and report on the same key performance indicators every iteration (e.g., velocity or throughput) can be automated with relatively less effort using tools such as Jira. As described in Sect. 4, at the case organization several programs are trying to automate process-oriented reporting as much as possible. To that end, observed programs are often introducing new collaboration and work management tools that enable Agile Masters to build automated process-oriented reports (e.g., Jira and Confluence). Interviewee *AM4* explained that only the standardization of tools and Backlog structures across teams and programs would allow them to automate their process-oriented reporting and thus satisfy the increasing information needs of stakeholders (in this case usually the Product Owner) that come with the introduction of agile inspect and adapt cycles.

> *That's... in parts it's very disappointing, there is not as much automation there as possible. [...] Whenever you want to do automated reporting, you need similar structures over the different products. And that's not existing everywhere. As I said, my product works with Sagas and Epics, most of the other products only work with Epics. And so, it's hard to do this kind of automated reporting there.* — Interviewee *AM4*

On the other hand, however, a development-oriented report that focuses on the newly developed features of the product will most likely differ a lot between two iterations, given the use of frequent inspect and adapt cycles is tied to development iterations at the case organization. This makes it hard to automate. Automating the generation of such a report, therefore, is always a trade-off between the stability of the goals reported against and the effort of automation (*LM1, DEV1*). Interviewee *LM1* explained:

> *It's always a question of how much effort you put in the automation and how stable the reportings are. If they are very stable, then it's worth putting a lot of effort into the automation. If it changes from quarter to quarter, then you obviously won't spend that much time, money, and effort in order to automate it. But as a general rule, sure, every reporting should be automated as much as possible.*

To summarize, the challenge is determined by aspects that relate to the agility of the organization, because frequent inspect and adapt cycles also require frequent reporting to gather feedback. The impacts of the challenge, in contrast, are related to the large scale of the organization. While the effort for agile teams remains largely the same in large-scale settings compared to small agile organizations, the effort increases mostly at the program and portfolio level at the case organization. The program and portfolio are required to adjust their reports more frequently to changing goals. To allow for some automation, the program and portfolio further have to ensure a certain consistency of the used tools across their agile teams.

5.2 Challenges of the recipient of a report

5.2.1 Recipients of reports on higher levels do not gain meaningful insights from the reports they receive (5)

We observed missing expressiveness of aggregated reports on higher levels in the organization, and interviewees explained to us that important topics from the lower organizational levels often are not represented correctly or not at all by reports on the higher levels. This challenge is driven by two aspects. On the one hand, the variety of topics coming from the agile teams that need to be aggregated in the higher-level reports. And on the other hand, the lack of a clear prioritization by the stakeholders of what the aggregated reports should focus on. Interviewee *AM1* explained the first aspect of the challenge as follows:

> *Getting the real stuff there. [...] So, I would say some of the most important impediments are not represented. [...] So, I would say this is one of the biggest challenges. Having or bringing the real topics up there, even if they appeared there already one thousand times.*

The autonomy of agile teams to set their team-level goals on their own makes it increasingly hard for reports on the higher levels to capture and aggregate the critical issues from all agile teams. A further complication of this aspect is the large size of the organization (*STE1*) and the resulting high number of teams with autonomously defined team goals.

The second aspect that drives this challenge is the missing prioritization of goals on the higher organizational levels. Interviewee *LM1* explained this aspect:

> *[...] the problem actually is the same on all levels, but it starts on the highest level of our hierarchy. The higher you get in a company like ours, the less people actually prioritize. They decide 'do it' or 'don't do it', but they don't prioritize. [...] I would say that's the biggest challenge. Actually prioritizing on the highest level of our organization. And the reporting is directly linked to this prioritization.*

In several programs, goals on the higher levels are simply treated as overall objectives without any relative prioritization among each other. Because reporting is linked to prioritization (*LM1*), the lack thereof makes it hard for reports to focus on high-priority topics and provide focused insights on these topics.

The combination of these two aspects is causing reports on higher levels in the case organization to lack a clear focus and meaningful insights. Reports on the higher organizational levels are too detailed on all the various issues and thus too extensive to grasp, making it hard for teams to highlight their most pressing topics. Thus, they fail to direct management's focus on the few most central topics.

To summarize, the challenge is determined by a combination of aspects that relate to both the large scale and the agility of the organization. It is determined by the agility of the organization because the agile teams are autonomously setting their goals. It is also determined by the large scale of the organization because the programs and portfolios need to keep an overview of overall progress across all their teams. The misrepresentation of agile teams' important topics and issues as well as the lack of focus on the program and portfolio level represent impacts of the challenge both on the agile teams and on the program and portfolio level.

5.2.2 Increasingly large scale of agile organization causes delays in the reporting chain (C6)

The large size of the case organization often causes reports from lower levels to reach higher levels with a certain delay because the reported information is reused in another reporting cycle on the next higher level. E.g., information reported after a team cycle (e.g., a weekly Scrum Sprint) is used as input for the report at the program level (e.g., monthly product review), and thus will be evaluated and discussed at the portfolio level up to three weeks later. Such delays are causing the agile inspect and adapt cycles to lag behind the current state of work at the case organization. Ultimately, this might limit the scope of action compared to a more timely report and reaction. Interviewee *PO2* explained this challenge to us as follows:

> *I mean, as I already said, we are more than 100 people working on different topics. And sometimes the whole [reporting] chain, when something isn't delivered or when there is some issue, sometimes it happens that this information comes at a quite late stage. [...] So, this whole chain from developer, feature team, product, portfolio... sometimes it's not that easy to have a fluent communication going on.*

To summarize, the challenge is determined by aspects that relate to the large scale of the organization, while it impacts the agility on the team level. The aspect that information takes time to travel across the organizational hierarchy and reporting cycles relates to the large scale of the organization. The agile teams are impacted in their process of continuous improvement, ultimately negatively impacting a central concept of agility.

5.3 Recommendations

With the interviewees, we also discussed potential actions to mitigate or address the challenges that we explained in the previous section. Throughout these discussions, we found that interviewees repeatedly linked challenges with reporting back to issues with the goal-setting procedures used at the case organization. Further, several interviewees brought up the topic of report automation. While they often deemed automation as a valuable tool, it became clear that automation is no silver bullet and should only be used for parts of the reporting spectrum in large-scale agile settings. We expand on these aspects in the following and provide associated recommendations that we learned from the case study.

5.3.1 Consistent usage of goal-setting practices and linkage of goals across organizational levels

On the one hand, agile teams often do not understand the purpose behind certain reports they are required to do (C1) and feel limited in their autonomy (C3), while they even lack the necessary context information they would need to meaningfully do these reports (C2). On the other hand, in many cases, the recipients of said reports do not gain meaningful insights from them (C5). Given these challenges, a disconnect between the creating and receiving side of reports becomes apparent.

Several of the interviewees from the case study explained that they consider improving the goal-setting process to be a suitable remedy to this disconnect. Especially, (1) a consistent usage of goal-setting practices across teams, both horizontally and vertically in the organization, as well as (2) establishing a linked chain of subgoals across organizational levels were recommended in the discussions.

It is crucial to strike a balance between establishing a standardized goal-setting practice across teams, that provides a consistent goal-setting process, and not constraining the autonomy of teams by dictating how work is to be done (Moe et al. 2021). Clearly prioritized, transparent goals can be used for well-structured reporting in the organization while providing a justification why a certain reporting is necessary that is comprehensible to the agile teams. Further, by making the goals and their prioritization transparent for all stakeholders, it enables report creators to focus on the most important goals and issues in their reports. Interviewees argued in favor of goal-setting practices such as Objectives and Key Results (OKRs) (Niven and Lamorte 2016; Stray et al. 2022) or Goal Question Metric (GQM) (van Solingen et al. 2002) over custom, organization-specific practices (*AM2, AM7*). This is to further facilitate transparency and comprehensibility for all stakeholders. These insights from the case study are in line with previous findings from the literature that emphasize a balance between decentralized self-management of the agile teams and centralized, program-level coordination and oversight as a key aspect of large-scale agile development (Dingsøyr et al. 2018; Edison et al. 2022; Moe et al. 2021). While standardizing the practices used for goal-setting across agile teams may constrain the team in some of their autonomy, a balance has to be achieved that also enables programs and portfolios to coordinate the teams and keep an overview (Dingsøyr et al. 2018; Moe et al. 2021).

Interviewees *LM1* and *AM7* explained that the linkage of sub-goals to higher-level goals would be a necessity to ensure the stringent implementation of the organization's overall strategy. By clearly linking each goal (and also Backlog Items at the team level) to the goals on the respective next higher scaling level in the organization, reporting can consequently show how work on lower organizational levels contributes to overall progress. Such a linkage can be achieved by collaborative goal-setting involving stakeholders from both the respective higher and lower organizational scaling levels. When defining goals, stakeholders from both levels should collaboratively position each goal in the context of the overall program, portfolio, or organizational goals. Each goal should be clearly contributing to at least one higher-level goal, and this contribution should be made explicit to all involved stakeholders. However, this also implicates that autonomous agile teams in large-scale settings still have to involve and coordinate with (higher-level) stakeholders to ensure their prioritized goals and work do contribute to overall objectives. Thus, ideally, the agile team does not only work with customers for goal-setting but also actively seeks input from higher levels within the organization.

However, multiple interviewees stressed that, while in general the linkage of goals should be enforced, there may still be goals that cannot be directly linked to any higher-level objective (*AM2, LM1, PO3, PO5, PO6*). A commonly referred example is technical changes to the software product that are identified as necessary by the agile team but do not directly contribute to any higher goal, e.g., a version upgrade to an existing database (*AM2*). So, while a linkage of goals to higher-level goals facilitates goal-oriented reporting of the organization, it should not be forced on every necessary work item in the daily work.

5.3.2 Differentiation of automation approaches for reporting

The increased frequency of reporting and the associated increased effort of creating the reports is a central challenge in our case study (C4). An intuitive remedy to this challenge seems to be the automation of the required tasks. Yet, we also learned that automation itself is non-trivial and brings challenges of its own in agile work settings. From the discussions, we learned that different types of reporting in large-scale agile organizations lend themselves to different extents of automation.

Interviewees *LM1, BE1, DEV1, PO3,* and *PO5* emphasized that automation should be pursued as much as possible to reduce manual efforts of process-oriented reporting. Interviewee *PO5* even went as far as expressing the opinion that any additional, manually created report besides automated analysis of the actual development

Backlogs is an unnecessary overhead for process-oriented reporting. However, *AM2, PO1, PO4,* and *PO5* cautioned to always consider context information when interpreting and discussing automatically generated reports based on metrics. The team or actor in charge of a metric should always be involved in discussions of it, to be able to provide background and context information if needed. Regarding this, interviewee *AM2* stated that "*[...] if I am trying to tell something about a team just by looking at the chart [of a report], yes, I can say something. But I can only interpret it 100% right if someone who is in the team and who was kind of responsible for the numbers gives me feedback or gives me input*".

Besides process-oriented reporting, product- and development-oriented reporting was also affected by the adoption of large-scale agile methodologies at the case organization. Both product- and development-oriented reporting also increased in frequency across all scaling levels, just like process-oriented reporting. However, in contrast to process-oriented reporting, automation of product- and development-oriented reporting is less prevalent and also considered desirable less often at the case organization. Participants *LM1, BE1,* and *PO6* explained that product- and development-oriented reporting should be a mixture of quantitative and qualitative means of reporting, as well as manual and automatic generation. In contrast to process-oriented reporting, product- and development-oriented reporting often require a detailed discussion about recent changes or developments to the product. This is in line with agile values, which emphasize interactions between people and collaboration with stakeholders. Full automation is not possible and also not desired for these types of reporting, even though frequency and efforts increased with the introduction of agile methodologies, because the resulting interactions are deemed as beneficial by agile methodologies.

5.3.3 Tracking of trends over time

When discussing the challenge of delays in the reporting chain due to the organization's size and scaling levels (C6), none of the interview participants had suggestions for targeted actions to address the challenge. However, we learned that a continuous understanding of reporting, which Murphy and Cormican (2015) also argue in favor of, may help to mitigate the delay issue. Individual reports should not be considered isolated, but be compared to and evaluated in the context of previous reports. Especially continuously tracking metrics over time allows programs to identify changes in trends earlier, and thus mitigates the problem that reports reach higher levels delayed in large-scale agile environments. Since metrics and continuous evaluation are particularly relevant for process-oriented reporting, process-oriented reporting should track trends across longer periods of time to improve the ability to recognize changes and make reliable predictions for the next cycle

However, interviewees *AM7* and *PO5* cautioned that such tracking over long periods of time requires certain stability in the organization and team composition, to ensure comparability between different points in time. Given this stability, participants agreed that tracking over long periods of time allows for the identification of extraordinary effects and evolution of teams, which cannot be achieved with individual data points in time (*PO4, LM1, DEV1*).

6 Discussion and conclusion

At the beginning of this paper, we set out with the hypothesis that the combination of reporting and agile methodologies in large-scale organizations comes with challenges and considerable points of friction. To gain a better understanding of these points of friction and generate a more differentiated view of the intricacies of this combination, we chose to conduct a case study. As hypothesized, our case study revealed that the combination of reporting and agile methodologies in large-scale organizations is indeed non-trivial. While agile methodologies emphasize minimal reporting and documentation (Beck et al. 2001; Rubin and Rubin 2011), large-scale organizations face concerns due to their size that require a certain degree of reporting and documentation. We learned that the problems of this combination manifest themselves in multiple, different facets in the studied case company. Our results highlight, that finding a good balance between agile development and reporting procedures is challenging for large-scale organizations. By documenting these different facets in the form of seven challenges we contribute to a detailed and differentiated understanding of the problems. This detailed understanding will help researchers, in the next step, identify suitable remedies in order to achieve effective and efficient reporting in large-scale agile organizations. We documented initial ideas and paths for potential solution designs in the form of three recommendations, which we gathered from our discussions with practitioners. We encourage future research to further look into these recommendations. Similar to what is proposed by the literature on ambidexterity (e.g., Jöhnk et al. (2022), Gregory et al. (2015)), large organizations may need to employ ambidextrous resolution strategies to balance agility with reporting needs. This may allow large-scale organizations to employ some reporting procedures where it is inevitable while still adhering to the agile philosophy by minimizing documentation and reporting wherever possible. Future research should investigate how organizations can find the right balance between agility and reporting for their circumstances and what strategies to realize this balance can look like. The recommendations derived from our study could provide a foundation for research on such strategies.

Not all of the identified challenges are of equal importance for all organizations or programs, however. In our case study, we observed and interviewed practitioners from multiple major departments at the case organization. Not all of the challenges were relevant for all of the programs, which we attribute to the different agile frameworks used, the different types of products being developed, and the different backgrounds of the departments, among other factors. We categorized the identified challenges into the creating and receiving side of reports in large-scale agile organizations. The challenges express several aspects that more precisely describe the previously hypothesized points of friction between reporting and agility.

To discuss the identified points of friction, we formulate a set of questions that should be asked in organizations:

Why are you reporting? We learned that reporting actors, especially agile teams, often lack a clear understanding of the backgrounds and necessities of the reports they are creating.

What are you reporting? Further, reporting actors also often lack an understanding of the larger context of their work in the overall organization. This complicates the creation of concise and expressive reports. Receiving actors, on the other side of reporting, often lack an overview of the individual goals that the many autonomous agile teams are currently pursuing. As a result, both sides struggle with clearly understanding what is actually being reported. Insights gained from reports are thus often minimal. We documented the recommendation to look into consistent goal-setting as a potential remedy to these types of challenges. Future research should investigate how consistent goal-setting across products and agile teams can be implemented and how it can contribute to more transparency and better reporting. Improved transparency on goals could not only contribute to more insightful reports but could also help teams to make autonomous decisions in line with organizational goals (Ramasubbu and Bardhan 2021). We suggest drawing on the goal-setting theory (cf. Locke and Latham (2002, 2006)) to transfer its well-known concepts into the contexts of large-scale agile organizations.

How are you reporting? The mechanics of reporting are also often challenging in large-scale agile settings. Existing tools are often insufficient to contribute significantly to the reduction of reporting efforts. For process-oriented reporting in particular, we learned that automation is desirable. Basic automation can readily be achieved via backlog management tools. The degree of automation and possible types of analysis are quite limited, however. Especially regarding product-oriented reporting, we further found that, while several teams were automatically collecting data on, e.g., static code analysis or test results, such data was not used for product-oriented reporting beyond the team at all. Thus, future research should look into the design of information systems and artifacts that facilitate reporting in large-scale agile organizations. While process-oriented reporting seems to already leverage automation and insights from automatically collected data, future research should also investigate how product-oriented reporting in large-scale agile organizations can benefit from automation. To facilitate automation and build on our recommendation to track process trends over longer periods of time as part of the reporting, research should also look into how to reliably identify potential problems and future risks from such trend reporting. Existing research on modeling systemic risks in information technology portfolio management (e.g., Guggenmos et al. (2019)) could provide interesting starting points for such trend analysis.

Similar to the individual and different types of challenges, we also think that appropriate solutions to address these challenges have to be of different types and be customized to suit the specific organization or program. As outlined above, we motivate research into the design of new systems and artifacts for reporting in large-scale agile organizations. Several interviewees mentioned to us that new reporting approaches based on artificial intelligence and advanced data analytics could be interesting to investigate. Future research should look into how the data generated by already existing backlog management tools can be used for data-based approaches to automating reporting. This could improve large-scale agile organizations' absorptive capacity by achieving better insights from data (cf. Hofmann et al. (2021)), which are of higher relevance and better to understand for decision-makers.

The contribution of our research is two-fold: First, we contribute to the discourse on large-scale agile software development and large-scale agile organizations (e.g., Dikert et al. (2016); Uludağ et al. (2019)) by documenting which challenges arise with reporting in such organizations. Second, we document recommendations on how to address these challenges, guiding the design of large-scale agile organizations. Besides our core research contribution, our documented challenges and recommendations also provide guidance for practitioners in large-scale agile organizations seeking to benefit from reporting. On the one hand, the challenges can help practitioners understand common problems they might encounter in their own organizations. On the other hand, the recommendations provide actionable suggestions to practitioners on how to overcome these challenges, should they face one of the challenges on their own. As outlined in the sections above, based on our results practitioners and scholars can now address the detailed challenges and build on the directions revealed by the documented recommendations.

Our research is not without limitations. The biggest limitation is the singular case that we studied in this paper. Not all the identified challenges may be relevant in every organization, as described in the previous paragraphs. Further, the data we collected could be affected by biases of the interviewees or the selection process, for example, because not all scaling levels could be represented by an equal amount of interviewees in both interview phases. We tried to mitigate these limitations by collecting data from multiple programs and types of sources, to achieve triangulation. Further, using a systematic coding and documentation approach, we tried to realize a transparent chain of evidence. The analysis we conducted could, however, still be affected by researcher bias. In order to mitigate this concern, we conducted a second round of interviews to evaluate our results.

Appendix 1: Guide for first round of interviews

First part: general information

1. What is your large-scale agile development (stakeholder) role?
2. How long have you been working in the field of software development?
3. How long have you been working in agile software development?
4. How long have you been working in large-scale agile software development?
5. How large is the development program/organization?
6. How many teams are co-located and how many teams are distributed?
7. In which countries are the teams located?
8. Which scaling agile frameworks are you using for the software development process?
9. To which level do you scale agile practices?
10. Are we allowed to contact you again for further research?

Second part: reporting team progress

1. How do the teams report their progress towards their goals?
2. How do you categorize this reporting?
3. Why did you choose this type of reporting?
4. Which stakeholder(s) are responsible for this reporting?
5. When and how often is the progress reported?
6. Are you using (planning to use) any tools for the reporting?

Third part: reporting overall progress

1. How is overall program/product progress towards the goals reported?
2. How do you categorize this reporting?
3. Why did you choose this type of reporting?
4. How are individual team reports used for reporting overall program/product progress?
5. Which stakeholder(s) are responsible for reporting overall program/product progress?
6. When and how often is overall program/product progress reported?
7. Which stakeholder(s) are interested in the report of overall program/product progress?
8. Are you using (planning to use) any tools for reporting overall program/product progress?

Fourth part: challenges

1. What are the biggest challenges you are facing in reporting progress?
2. How do you approach these challenges?
3. Which actions can be (ideally) implemented to avoid these challenges?

Fifth part: discussion

1. Do you want to add any further information, comment, or a topic that we missed?

Funding Open Access funding enabled and organized by Projekt DEAL. No funding was received for conducting this study.

Data availability statement The datasets generated during and/or analysed during the current study are not publicly available due to restrictions by the case organization and data protection regulations.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Abrahamsson P, Conboy K, Wang X (2009) lots done, more to do: the current state of agile systems development research. Eur J Inf Syst 18(4):281–284. https://doi.org/10.1057/ejis.2009.27

Baham C, Hirschheim R (2022) Issues, challenges, and a proposed theoretical core of agile software development research. Inf Syst J 32(1):103–129. https://doi.org/10.1111/isj.12336

Beck K (2000) Extreme programming explained: embrace change. Addison-Wesley

Beck K, Beedle M, van Bennekum A, et al (2001) Manifesto for agile software development. https://agilemanifesto.org, accessed: 2022-12-11

Benbasat I, Goldstein DK, Mead M (1987) The case research strategy in studies of information systems. MIS Q 11(3):369–386. https://doi.org/10.2307/248684

Boehm BW, Turner R (2005) Management challenges to implementing agile processes in traditional development organizations. IEEE Softw 22(5):30–39. https://doi.org/10.1109/MS.2005.129

Conboy K (2009) Agility from first principles: reconstructing the concept of agility in information systems development. Inf Syst Res 20(3):329–354. https://doi.org/10.1287/isre.1090.0236

Conboy K, Carroll N (2019) Implementing large-scale agile frameworks: challenges and recommendations. IEEE Softw 36(2):44–50. https://doi.org/10.1109/MS.2018.2884865

Cruzes DS, Dyba T (2011) Recommended steps for thematic synthesis in software engineering. In: 2011 International symposium on empirical software engineering and measurement, pp 275–284, https://doi.org/10.1109/ESEM.2011.36

digital.ai (2021) 15th annual state of agile report. https://digital.ai/resource-center/analyst-reports/state-of-agile-report, accessed: 2022-03-11

Dikert K, Paasivaara M, Lassenius C (2016) Challenges and success factors for large-scale agile transformations: a systematic literature review. J Syst Softw 119:87–108. https://doi.org/10.1016/j.jss.2016.06.013

Dingsøyr T, Moe NB (2014) Towards principles of large-scale agile development. a summary of the workshop at xp2014 and a revised research agenda. In: Agile methods. Large-scale development, refactoring, testing, and estimation. Springer, Cham, DE, pp 1–8, https://doi.org/10.1007/978-3-319-14358-3_1

Dingsøyr T, Fægri TE, Itkonen J (2014) What is large in large-scale? a taxonomy of scale for agile software development. In: International conference on product-focused software process improvement. Springer, Cham, DE, pp 273–276, https://doi.org/10.1007/978-3-319-13835-0_20

Dingsøyr T, Moe NB, Fægri TE et al (2018) Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. Empir Softw Eng 23(1):490–520. https://doi.org/10.1007/s10664-017-9524-2

Dreesen T, Diegmann P, Rosenkranz C (2020) The impact of modes, styles, and congruence of control on agile teams: Insights from a multiple case study. In: Proceedings of the 53rd Hawaii international conference on system sciences, p 6247–6256, https://doi.org/10.24251/hicss.2020.764

Edison H, Wang X, Conboy K (2022) Comparing methods for large-scale agile software development: a systematic literature review. IEEE Trans Softw Eng 48(8):2709–2731. https://doi.org/10.1109/TSE.2021.3069039

Gregory RW, Keil M, Muntermann J et al (2015) Paradoxes and the nature of ambidexterity in IT transformation programs. Inf Syst Res 26(1):57–80. https://doi.org/10.1287/isre.2014.0554

Guggenmos F, Hofmann P, Fridgen G (2019) How ill is your it portfolio?: measuring criticality in it portfolios using epidemiology. In: 40th international conference on information systems

Hackman JR (1986) The psychology of self-management in organizations. Am Psychol Assoc. https://doi.org/10.1037/10055-003

Herrmann W (2022) Beiersdorf-cio annette hamann: Wir müssen weg von der kostendenke. CIO Magazin

Hoffmann D, Ahlemann F, Reining S (2020) Reconciling alignment, efficiency, and agility in it project portfolio management: recommendations based on a revelatory case study. Int J Project Manage 38(2):124–136. https://doi.org/10.1016/j.ijproman.2020.01.004

Hofmann P, Stähle P, Buck C, et al (2021) Data-driven applications to foster absorptive capacity: a literature-based conceptualization. In: Proceedings of the 54th Hawaii international conference on system sciences, Honolulu, HI, pp 4880–4889, https://doi.org/10.24251/HICSS.2021.593

Ingvaldsen JA, Rolfsen M (2012) Autonomous work groups and the challenge of inter-group coordination. Human Relat 65(7):861–881. https://doi.org/10.1177/0018726712448203

Jöhnk J, Ollig P, Rövekamp P et al (2022) Managing the complexity of digital transformation—how multiple concurrent initiatives foster hybrid ambidexterity. Electron Mark 32(2):547–569. https://doi.org/10.1007/s12525-021-00510-2

Kalenda M, Hyna P, Rossi B (2018) Scaling agile in large organizations: practices, challenges, and success factors. J Softw Evol Process 30(10):e1954. https://doi.org/10.1002/smr.1954

Kasauli R, Knauss E, Horkoff J et al (2021) Requirements engineering challenges and practices in large-scale agile system development. J Syst Softw 172(110):851. https://doi.org/10.1016/j.jss.2020.110851

Kischelewski B, Richter J (2020) Implementing large-scale agile - an analysis of challenges and success factors. In: Proceedings of the 28th European conference on information systems (ECIS)

Larman C (2016) Large-scale scrum, 1st edn. Addison-Wesley Professional, Boston, MA

Leffingwell D (2018) SAFe 4.5 reference guide: scaled agile framework for lean enterprises, 2nd edn. Addison-Wesley Professional; Safari, Boston, MA

Limaj E, Bernroider EWN (2022) A taxonomy of scaling agility. J Strateg Inform Syst 31(3):101721. https://doi.org/10.1016/j.jsis.2022.101721

Locke EA, Latham GP (2002) Building a practically useful theory of goal setting and task motivation: a 35-year odyssey. Am Psychol 57(9):705. https://doi.org/10.1037/0003-066X.57.9.705

Locke EA, Latham GP (2006) New directions in goal-setting theory. Curr Dir Psychol Sci 15(5):265–268. https://doi.org/10.1111/j.1467-8721.2006.00449.x

Miles MB, Huberman AM, Saldaña J (2013) Qualitative data analysis. A methods sourcebook, 3rd edn. SAGE Publications, Los Angeles, USA

Moe NB, Šmite D, Paasivaara M et al (2021) Finding the sweet spot for organizational control and team autonomy in large-scale agile software development. Empir Softw Eng 26(5):1–41. https://doi.org/10.1007/s10664-021-09967-3

Müller R, Martinsuo M, Blomquist T (2008) Project portfolio control and portfolio management performance in different contexts. Proj Manag J 39(3):28–42. https://doi.org/10.1002/pmj.20053

Murphy T, Cormican K (2015) Towards holistic goal centered performance management in software development: lessons from a best practice analysis. Int J Inform Syst Project Manag 3(4), 23–36. doi: https://doi.org/10.12821/ijispm030402

Niven PR, Lamorte B (2016) Objectives and key results: driving focus, alignment, and engagement with OKRs. John Wiley & Sons

Nyrud H, Stray V (2017) Inter-team coordination mechanisms in large-scale agile. In: Proceedings of the XP2017 scientific workshops. Association for computing machinery, New York, NY, USA, XP ’17, https://doi.org/10.1145/3120459.3120476

Paasivaara M (2017) Adopting safe to scale agile in a globally distributed organization. In: 2017 IEEE 12th international conference on global software engineering (ICGSE), https://doi.org/10.1109/ICGSE.2017.15

Ramasubbu N, Bardhan IR (2021) Reconfiguring for agility: examining the performance implications of project team autonomy through an organizational policy experiment. MIS Q 45(4):2261-2279. https://doi.org/10.25300/MISQ/2021/14997

Reifer DJ, Maurer F, Erdogmus H (2003) Scaling agile methods. IEEE Softw 20(4):12–14. https://doi.org/10.1109/MS.2003.1207448

Rolland KH, Fitzgerald B, Dingsoyr T, et al (2016) Problematizing agile in the large: Alternative assumptions for large-scale agile development. In: Proceedings of the 37th international conference on information systems (ICIS)

Rubin E, Rubin H (2011) Supporting agile software development through active documentation. Requir Eng 16(2):117–132. https://doi.org/10.1007/s00766-010-0113-9

Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. Empir Softw Eng 14(2):131–164. https://doi.org/10.1007/s10664-008-9102-8

Sambamurthy V, Bharadwaj A, Grover V (2003) Shaping agility through digital options: reconceptualizing the role of information technology in contemporary firms. MIS Q 27(2):237–263

Scheerer A, Hildenbrand T, Kude T (2014) Coordination in large-scale agile software development: a multiteam systems perspective. In: 2014 47th Hawaii international conference on system sciences, pp 4780–4788, https://doi.org/10.1109/HICSS.2014.587

Schwaber K (2007) The enterprise and scrum. Microsoft Press

Schwaber K, Beck K (2002) Agile software development with scrum. Prentice Hall, Upper Saddle River, NJ

van Solingen R, Basili V, Caldiera G et al (2002) Goal Question Metric (GQM) Approach. John Wiley & Sons. https://doi.org/10.1002/0471028959.sof142

Stettina CJ, Schoemaker L (2018) Reporting in agile portfolio management: routines, metrics and artefacts to maintain an effective oversight. In: International conference on agile software development, Springer, Cham, pp 199–215, https://doi.org/10.1007/978-3-319-91602-6_14

Stray V, Moe NB, Vedal H, et al (2022) Adopting safe to scale agile in a globally distributed organization. In: Proceedings of the 55th Hawaii international conference on system sciences, https://doi.org/10.24251/HICSS.2022.883

Uludağ Ö, Putta A, Paasivaara M, et al (2021) Evolution of the agile scaling frameworks. In: Gregory P, Lassenius C, Wang X, et al (eds) Agile processes in software engineering and extreme programming. Springer International Publishing, Cham, pp 123–139, https://doi.org/10.1007/978-3-030-78098-2_8

Uludağ Ö, Kleehaus M, Caprano C, et al (2018) Identifying and structuring challenges in large-scale agile development based on a structured literature review. In: 2018 IEEE 22nd international enterprise distributed object computing conference (EDOC), pp 191–197, https://doi.org/10.1109/EDOC.2018.00032

Uludağ Ö, Harders NM, Matthes F (2019) Documenting recurring concerns and patterns in large-scale agile development. In: Proceedings of the 24th European Conference on Pattern Languages of Programs. Association for computing machinery, New York, NY, USA, EuroPLop '19, https://doi.org/10.1145/3361149.3361176

Uludağ Ö, Philipp P, Putta A et al (2022) Revealing the state of the art of large-scale agile development research: a systematic mapping study. J Syst Softw 194(111):473. https://doi.org/10.1016/j.jss.2022.111473

Williams L, Cockburn A (2003) Agile software development: it's about feedback and change. IEEE Comput 36(6):39–43. https://doi.org/10.1109/MC.2003.1204373

Wińska E, Dąbrowski W (2020) Software development artifacts in large agile organizations: a comparison of scaling agile methods, Springer International Publishing, Cham, pp 101–116. https://doi.org/10.1007/978-3-030-34706-2_6

Yin RK (2014) Case study research: design and methods, 5th edn. SAGE Publications, Los Angeles, USA

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.