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.

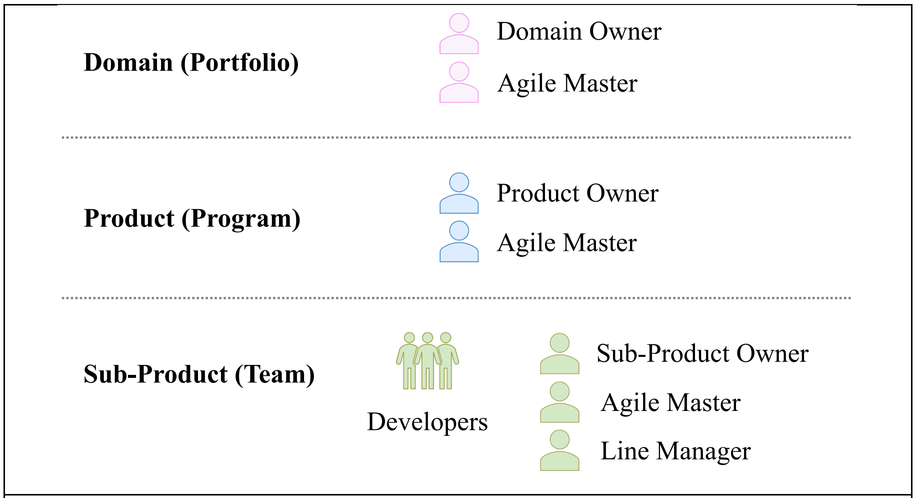

Figure 1. Overview of the Scaling Levels at the Case Organization; the Notation is Based on the One used by Stettina and Schoemaker (2018)

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/3z5knE0

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.

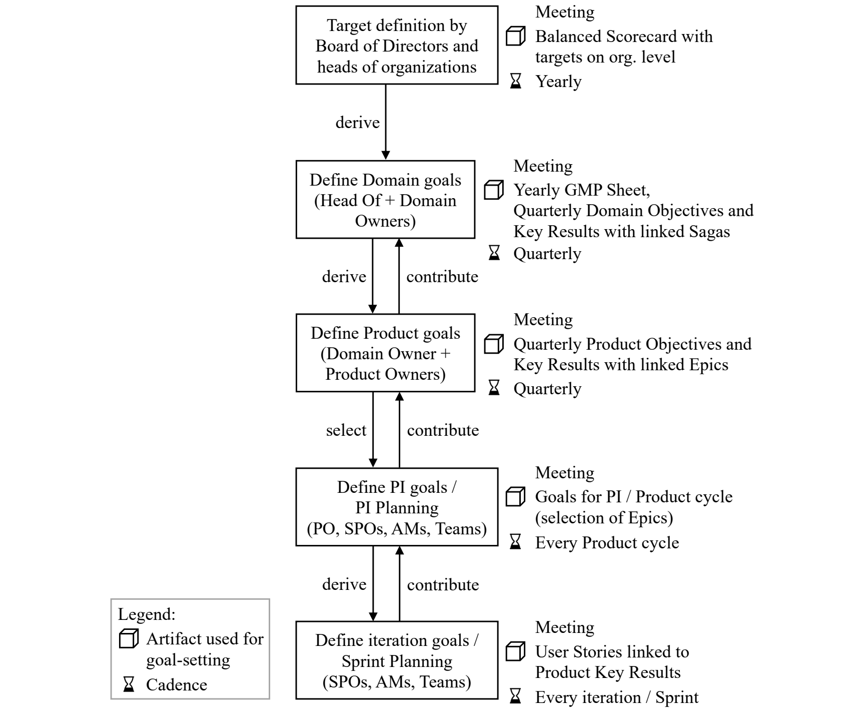
Target definition by Board of Directors and heads of organizations
Meeting
Balanced Scorecard with targets on org. level
Yearly
derive
Define Domain goals (Head Of + Domain Owners)
Meeting
Yearly GMP Sheet, Quarterly Domain Objectives and Key Results with linked Sagas
Quarterly
derive
contribute
Define Product goals (Domain Owner + Product Owners)
Meeting
Quarterly Product Objectives and Key Results with linked Epics
Quarterly
select
contribute
Define PI goals / PI Planning (PO, SPOs, AMs, Teams)
Meeting
Goals for PI / Product cycle (selection of Epics)
Every Product cycle
derive
contribute
Legend:
Artifact used for goal-setting
Cadence
Define iteration goals / Sprint Planning (SPOs, AMs, Teams)
Meeting
User Stories linked to Product Key Results
Every iteration / Sprint

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

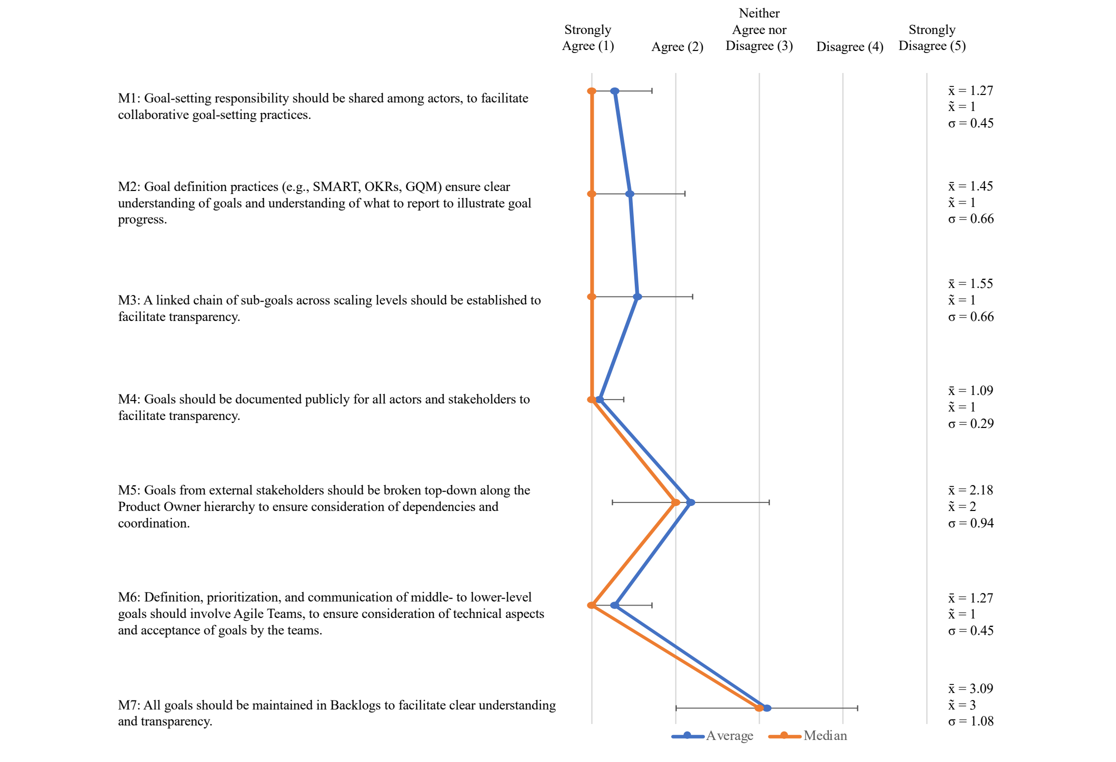

Figure 3. Evaluation Results of the Mitigation Propositions

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.