How agile teams make Objectives and Key Results (OKRs) work

Viktoria Stray
University of Oslo, SINTEF
Oslo, Norway
stray@ifi.uio.no

Jan Henrik Gundelsby
Knowit Solutions
Oslo, Norway
jhg@knowit.no

Rasmus Ulfsnes
SINTEF
Trondheim, Norway
rasmus.ulfsnes@sintef.no

Nils Brede Moe
SINTEF
Trondheim, Norway
nils.b.moe@sintef.no

ABSTRACT

Today, many companies allow their employees to work from anywhere, which has changed how employees coordinate their work and align toward the same goals. Objectives and Key Results (OKRs) is a goal-setting framework applied in such distributed settings. This research aimed to investigate how OKRs are used in large-scale agile contexts. We interviewed team members and analyzed documents, including a survey. Our study's results provide both enabling and limiting situations that make team members' utilization of the framework either easier or more difficult. We found that OKRs aided knowledge sharing and improved transparency between teams. We present four strategies used for overcoming challenges and maximizing the benefits of using a goal-setting framework. An important takeaway is that companies that employ OKRs must support their employees, especially in defining key outcomes that align and encourage teams toward a common goal.

CCS CONCEPTS

• **Software and its engineering** → *Software creation and management*.

KEYWORDS

Large-scale distributed agile, OKRs framework, agile software development, goal-setting, longitudinal case study

ACM Reference Format:
Viktoria Stray, Jan Henrik Gundelsby, Rasmus Ulfsnes, and Nils Brede Moe. 2022. How agile teams make Objectives and Key Results (OKRs) work. In *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering (ICSSP'22), May 20–22, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3529320.3529332

1 INTRODUCTION

In large-scale agile, the decision-making focus moves from the project manager to the software development team, and the decision-making process changes from individual and centralized to shared and decentralized [10, 18]. However, balancing the need for alignment and achieving the benefits of a decentralized decision-making structure that involves many autonomous teams in large-scale software projects is not well understood [6, 14, 21]. Furthermore, when many autonomous agile teams work toward the same goal, much additional coordination and management effort is required [20], which in earlier studies on large-scale agile projects was found to be challenging [19]. Although there is a vast literature on large-scale agile, several important questions remain open: If each team were given the authority and responsibility for organizing their work, how can the organization ensure the overall success of a large-scale project? Who is responsible for the overall compliance with the project goals? What are effective structures for cross-team decisions when teams have conflicting goals?

The large-scale organization needs to ensure that all agile teams work toward the same goal, which is traditionally achieved through formal organizational control [16]. When team goals conflict with organizational goals, many teams choose to act in their self-interest [1], and disrespect the larger context [7]. Modern agile organizations seek to involve employees when setting common goals across the organization, and they actively engage employees toward these goals [25]. One way of doing this, which is increasing in popularity, is implementing the goal-setting framework Objectives and Key Results (OKRs) [2, 5, 17]. Introducing a goal-setting framework can be a daunting task for an organization because it involves changing behavior in different levels in the organization, often from the very top, all the way down to the individuals working in teams.

Motivated by the importance of finding strategies for aligning teams toward the same goals in large-scale agile, the following research question drives our research: *What strategies do agile teams apply to make OKRs work for them?* To answer this question, we studied agile teams in a large Norwegian agency that were working distributed because of the COVID-19 pandemic.

2 RELATED WORK

In agile software development, the team has complete authority to do whatever it takes to accomplish the "goal" [22]. However, applying this to a large-scale setting is quite different compared to in smaller projects. In a literature review of the challenges and success factors of large-scale agile transformations, Dikert et al. [4] found that organizations had difficulties in finding a balance between the team's goals and the organization's broader goals. A team could easily prioritize their own objectives over those of the organization as a whole. Thus, being a part of a large organization

ICSSP'22, May 20–22, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9674-5/22/05.
https://doi.org/10.1145/3529320.3529332

may constrain the amount of authority that each team can have regarding goal setting. Guinan et al. [8] also recognized the problem of misalignment between business strategy and the teams own goals, noting that tunnel vision is an issue with large-scale agile. This corroborates [4, 6] where *transparency, end-to-end development, communication overload* and *synchronizing across dynamic fast moving teams* are common challenges. Thus, Guinan et al. [8] proposed four main levers for a digital project team: *diverse and targeted team composition, talent management, continuous learning* and *iterative goal setting*.

Goal setting is a well-established field within organizational psychology, and there is a strong empirical link between goal setting, motivation and task performance [13]. Locke and Latham introduced goal setting theory in 1990 [12], building on studies that date back to the 1960s. The authors explored 400 studies about goals and found that setting specific difficult goals led to higher performance compared to simply asking people to do their best. Interestingly, the direction of the goals, that is, whether the goals were self-set or assigned, showed a limited effect on the performance [13]. These findings may indicate that organizations should set difficult goals.

Goals can be set top-down or bottom-up. Kirsch et al. [11] suggested that organizations could achieve alignment and ensure goal attainment through two major modes: formal and informal control modes. Formal top-down control is exercised through relying on documentation, written rules and procedures that require following particular behaviors to achieve the desired outcomes and could be related to managerial control. Informal or bottom-up control encourages groups and individuals to monitor their own work, which they are able to do if group and personal goals are largely compatible with the goals of the broader organization.

OKR was designed to help organizations achieve their business goals in a more efficient and effective way through employee engagement and decision-making [17]. OKR provides benefits such as focus and frequent priority setting, and implementing OKR instills interteam transparency, allowing teams to cross-functionally align and create better products [2, 5, 17, 26]. Objectives are determined for a specified time, and key results are defined to ensure that these objectives are consistently progressed through tasks. An objective succinctly states what the team wishes to accomplish. A well-defined target should be attainable within three months and reflect the team's collective vision [5]. Key results are quantitative claims and enable the team to monitor their progress and determine whether a specific aim has been met [26].

3 RESEARCH METHODOLOGY

The case organization had 11 permanent and three temporary teams comprised of individuals from several departments. Two years ago, the case company made a strategic change to improve technological- and organizational performance. To achieve these goals, they focused on investing more in digital competence, increasing team autonomy, and reorganizing to make more services and products themselves. The organization has three levels: management, product areas, and team level.

We chose to undertake a case study because it is well suited to explore phenomena in depth within their real-world contexts, particularly when the boundaries between context and phenomena are vague or unclear [27]. We chose five autonomous agile teams (see Table 1). All teams used the OKR framework, which affected how they prioritized their work tasks. We analyzed documents and a division-wide survey about OKRs conducted in December 2020 with 41 respondents. We conducted 12 interviews from the teams with project participants in various roles. Findings from eight of these interviews are reported in [24]. For this study, we conducted a new analysis on all the transcripts. The interviews were transcribed, and analyzed using the qualitative data analysis software NVivo. We began by recording broad impressions, and then reduced, organized, and classified the data. Strauss and Corbin [3] have proposed a coding paradigm that includes context, causal conditions, intervening conditions, strategies, and consequences. We used this paradigm because it helped organize and understand our data and emerging categories when coding.

Table 1: Team characteristics

Team	Size	History	Context
Alpha	10	2 years	Platform team serving other teams
Bravo	7	2 yr.	Product team
Charlie	4	5 yr.	Product team
Delta	7	2 yr.	Platform team serving team Echo
Echo	4	3 yr.	Product team using team Delta's platform

4 RESULTS

The organization grew quickly, and it simultaneously introduced autonomous teams. Some pains resulting from these two factors included difficulties in ensuring organizational alignment, teams lacking a common direction, and no way of knowing if teams were making the right products. Finally, the organization lacked a helpful way of tracking progress. To improve the situation, OKRs was introduced. One of the key findings of our study is that introducing OKR is not easy and straightforward. We found that the local team context affected the introduction of the OKR framework. We next describe four identified strategies that the agile teams applied to make OKRs work for the teams and the organization. Table 2 provides an overview of the four strategies, with intervening conditions (marked IC1–IC10) and consequences (C1–C11).

4.1 S1 - Top-down and bottom-up

OKR was introduced bottom-up, meaning that the teams found their objectives and key results on their own, with sparse direction from management. Many of the interviewees reported that in the beginning, it was confusing if OKR was to be used as a tool for reporting, prioritizing, or measuring team progress (IC1). After a year of using OKR, one developer explained *"The organization has finally done a good job of setting objectives and key results at the top level. Because it has been a bit lacking. They have had something, but it has been so executive level that it has been a bit difficult to understand at the team level."* Another interviewee stated, *"I do not know if we had achieved the overarching company goals. It would have been nice if those goals would have been clearer."*

Many of the interviewees described using OKRs to be both difficult and cumbersome (IC2). Specifically, it was difficult to set clear,

Table 2: Strategies for OKRs

Strategy	Description	Intervening condition	Consequences
S1: Top-down & Bottom-up	OKRs were set both from the bottom-up, through the various developer teams, and top-down, from the management who were setting clear, high-level goals.	IC1 - Perceived purpose of OKRs IC2 - The OKR experience feels difficult and cumbersome IC3 - Different roles, different interests IC4 - No onboarding-process IC5 - Maturing organization	C1: Team focus C2: Performance C3: Alignment C4: Data-driven
S2: Competence building	Building competence in goal setting, and management through: *courses, community of practice, guidelines, mentoring*. The competence building targeted team members and team leaders differently.	IC2 - The OKR experience feels difficult and cumbersome IC6 - Organizational alignment in large scale IC7 - Lack of competence IC8 - Providing feedback	C5: Increased role-specific knowledge C6: Feedback to management C7: Added overhead
S3: Tooling	OKR tracker was used to manage the different goals and key result status. Slack was used for distributed communication and instant messaging. Digital whiteboards were utilized for workshops and retrospectives.	IC9 - Measuring success in the organization IC10 - Distributed teams (COVID-19)	C8: Transparency C3: Alignment C9: Ability to work distributed C4: Data driven
S4: Continuous improvement	The overall use of OKRs was described and guidelines were updated. An annual wheel was used to show how the OKR process aligned with other key dates.	IC5 - Maturing organization IC8 - Providing feedback	C10: Guidelines and annual wheel C11: Surveys C6: Feedback to management

measurable key results. Either the key results became binary (done or not done) or they were at the wrong level of abstraction (not a key result, but an actual activity, initiative or objective). One developer said, *"We use an incredible amount of energy to formulate objectives and key results in the right way. Especially the key results."* Another developer stated: *"The OKR process does not always fit so well with what I, as a developer, feel is important"* - (IC3). Another team member also stated that they struggled with the level of abstraction, *"The OKRs either define high overall goals, or define which work tasks we should have. The middle ground works poorly."*

In the beginning, there was no onboarding for teams on using OKR, and consequently, all teams were conducting their work differently (IC4). The team leaders read various online articles to obtain the knowledge needed to facilitate the workshops.

Many of the interviewees talked about a small shift six months into the work, where they seemed to become familiar with using the framework and adapted to a new everyday working life (IC5). One team lead stated, *"The biggest skeptics are actually the ones who become most satisfied with ORKs, as long as they have worked with it long enough. We have a long time left before this flows well, but every time we set OKRs, we get better and better at setting the right OKRs."*

After twelve months of using OKRs, the management received feedback through a survey that showed the need for clear top-down goals from the organization—both from the top leadership and the product areas responsible for the teams. This resulted in a hierarchy of three levels of OKRs, all aligned top-down. Other feedback from the survey included the need for clearer communication regarding the purpose and importance of OKR. In the survey, when asked what the main potential for improvements were, 37% answered that it was to obtain better goals from the product area leaders (see more survey results in Table 3).

Several interviewees described that utilizing OKR increased team focus (C1) and that they believed it increased the team's efficiency (C2). In addition, it helped the teams with direction and alignment (C3); a team member stated, *"The purpose is really that everyone on the team should at least have a common understanding of where we are going."* Another said, *"I still think it is useful for direction and what we have chosen to focus on for this quarter."*

Although organizational alignment is perceived as an important purpose in utilizing the framework, many of the people we interviewed did not seem interested in the other teams' objectives. Everyone knew that the OKRs were publicly available through an internally developed OKR tracker application, but they seldom peeked at other teams' OKRs. One developer stated, *"Teams are autonomous, teams do not talk to each other."* However, in one case, two teams had an internal competition of reaching their key results first. Understandably, this happened when the team's objectives and products were related. A few team members perceived the OKRs' purpose to be reporting and measuring success (C4).

4.2 S2 - Competence building

The interviewees noted several issues related to the lack of familiarity with the OKR concepts and the process of establishing the specific objectives and key results (IC2), a tech lead explained, *"A lot of people struggle with the OKR process. So now, everyone will get a course on how to define good OKRs and how to apply them to everyday work. In addition, we also get a course as team leads on how to facilitate the OKR process for the teams."* Another said, *"It could have been beneficial, I think, to have had someone who was somehow very experienced with OKR, who could have been an assistant at our meetings when we set our objectives"* (IC7). The survey performed in Q3 2020, one year after starting, showed a varied perception of the usefulness of OKRs (see Table 3). The response from management was to create a yearly training program.

Teams across the large-scale project were then instructed in the use of OKRs once a year. Qualified OKR facilitators with extensive experience delivered the training sessions. These sessions served as the focal point for the teams to grasp better the OKR process and

Table 3: OKR survey results (scores from 1 strongly disagree – 5 strongly agree)

#	Statement	Alpha	Bravo	Charlie	Delta	Echo	Company Avg.
1	The OKR framework is useful for my team and me.	3.4	4.0	4.3	3.6	3.8	**3.8**
2	OKRs helps my team and me align on direction and focus.	3.8	4.3	4.0	3.4	3.8	**3.8**
3	My team uses OKRs actively while working	3.6	4.3	4.0	3.4	3.6	**3.3**
4	Workshops helps my team and me to set good OKRs.	3.8	4.0	3.3	3.5	4.2	**4.0**
5	The OKRs of the product area help my team and me to set good OKRs.	3.1	1.0	1.3	2.2	2.8	**2.3**

assist them in working with OKRs. As such, the OKR training promoted knowledge sharing regarding the proper usage of OKRs. The organization utilized all-hands meetings as a form of community of practice, where the teams presented their OKRs and received feedback from peers.

Several interviewees, especially many team leaders, believed that structured training was highly valuable. Finding a common ground on how to use the framework made a significant difference, together with getting accustomed to the process. The role-specific training provided the developers insight into the purpose and understanding of the concepts, and it provided the team leaders with the expertise to run and facilitate workshops (C5). In addition, the training courses provided an opportunity to get feedback on the OKR process (C6). However, note that this adds overhead to the OKR process (C7), and some saw this as negative for productivity. A developer stated, *"When you have a framework that's there for productivity, then you use several quarters to figure out how it works for your team, then you have to attend a three-day course, and then you have the OKR workshops—is it really that useful?"*

4.3 S3 - Tooling

One of the main purposes for implementing OKR in the organization was to understand and measure the success of the digitalization efforts (IC9). When the pandemic hit, all teams were sent home and needed to work distributed (IC10). This challenged the OKR sessions. According to one back-end developer, *"We had one virtual OKR workshop where some people didn't write a single post-it note. That would not have happened in a physical workshop."* Digital whiteboards quickly improved this situtation. A team lead stated, *"We have started using Miro, and it turns out to be immensi flexible and genius for meetings such as retrospectives and OKR-workshops."*(C9)

The case company had developed their own OKR tracker. A purpose of this tracker was to push the organization towards becoming more data driven (C4). The OKR tracker was a digital dashboard accessible to all members in the project. The tracker facilitated transparency and collaboration across teams (C8). The tool was much used and highly appreciated by the team members. One UX-designer explained, *"I think the OKR tracker works very well. It's nice to see what the other teams are working on. One can easily measure progress. It's fun to update the tracker and see that we are at 60%; it's motivating. Further, I appreciate that it provides transparency across the project."*

In this tool, they could see the percentage that they had reached for each of the key results. This tool also helped with motivation and the alignment between the different teams (C3). A tech lead stated: *"We know that we have to go through the key results in the OKR tracker once a week, so it is important to have fun with it. Therefore, we have made it a bit like an internal competition together with our sister team. So, after we have reviewed our OKR tracker, we always look at theirs and see if we are ahead of them."* However, while the OKR tracker enabled teams to share their progress and goals across teams, many team members were uninterested in or lacked time to stay updated with the OKRs of other teams.

4.4 S4 - Continuous improvement

The teams collaborated to create, analyze, and enhance the OKRs through quarterly OKR workshops. These sessions raised team awareness of what needed to be accomplished. Throughout the quarter, the teams made incremental progress toward key results by completing activities associated with these key results. However, the lack of progress on key results was demotivating, but the situation improved with time. Two of the teams indicated they were constantly developing and had made significant progress in utilizing the framework after a year (IC5). Most teams had regular retrospectives where they reflected on what happened in the iteration and identified actions. A team lead explained, *"OKR has often come up in our monthly retrospectives, often as a kind of pain point"* (IC8). The management recognized the need for more structure and continuously improved how OKRs suited the organization. The overall OKR process was described and the guidelines were updated (C10). The company introduced an annual wheel to show how the OKR process aligned with other critical dates for the team. Furthermore, they held regular surveys (C11), and management became more involved in OKRs (C6).

5 DISCUSSION

Organizations need to ensure that all teams work toward the same goals. When team goals conflict with organizational goals, many teams choose to act in their own self-interest rather than the organization's [1]. One way of setting common goals across the organization is through implementing the goal-setting framework Objectives and Key Results (OKRs) [5, 17]. In our case study, we found that organizational alignment and setting direction were the key reasons for introducing the OKR framework, in addition to supporting team development and measuring success. Furthermore, we found that the framework helped the teams develop goals through a structured process in which they agreed on objectives. Each objective included a set of key results that tracked progress toward the goal.

OKRs helped teams focus on overall outcomes rather than specific tasks because the key results were discussed frequently in team meetings. The OKRs were regularly evaluated to adjust to

the changing environment, and to fit to the teams' agile context. This finding is akin to those of Guinan et al. [8] where *iterative goal setting* was a critical component to digital transformations.

The teams used OKRs to prioritize tasks, acquire a sense of what was most important and have awareness of what other teams were working on. Locke and Latham [13] showed that one of the key mechanisms of goals is to provide a direction through steering toward goal-relevant activities. The objectives in OKR are supposed to be expressed qualitatively, not quantitatively. However, formulating the key results was more challenging as they had to be expressed quantitatively. Our findings are consistent with earlier research suggesting a lack of practical guidance on transforming a qualitative objective into quantifiable key results [25].

There was confusion on the main purpose and importance of using the OKR framework, especially for developers. Using OKRs felt difficult and cumbersome, albeit team leads had an experience that was more positive. According to Locke and Latham [13], goals are important for gaining commitment to them. One way of doing that is having leaders communicate their vision and following up with that vision. Top-management introduced high-level goals to mitigate the uncertainty around the goals, thus combining top-down and bottom-up.

In the beginning, the teams did not have any training in the use of OKRs. Additionally, the team-leads had little experience in facilitating OKR workshops which caused the teams to struggle with setting and creating OKRs. Compared to large-scale agile transformations [4, 6], this is similar to introducing large-scale agile in organizations, where training and coaching are the key success factors for introducing large scale agile frameworks,

Participants were informed about how their own and others' tasks contributed to the objective through sharing knowledge about what was vital to complete, how, when, and by whom. As a result, the findings suggest that OKR enhances awareness across distributed team members. Tools and frameworks to support awareness are vital in distributed settings [23], because a lack of awareness of what distributed members are doing raises the barrier for initiating contact [9]. The tracker can be a way for agile teams to state publicly and commit to their goals, which is found as an important factor according to Locke et al. [13].

Dikert et al. [4] concluded that the most critical factor influencing the success of large-scale agile transformations was adapting the agile approach. The teams in our study modified their ways of working and they experimented with ways to improve the OKR framework for their context. It took over a year before teams could fully utilize the OKR framework, and the key to achieve this was the focus on continuous improvement. Teams learning how to improve their own activities is one of the most important success factors for succeeding with improvement work in agile software companies [15]. Learning occurs through continuous sense-and-response cycles, which identify current weaknesses, initiate new efforts, and implement their results.

There was a need for combining a top-down and bottom-up approach, building competence, applying the right tools, and continuous improving the process to support the implementation of OKR on the team and organizational level.

Our results indicate that team leaders are generally more interested in setting OKRs, while the developers are more interested in following the OKRs already set rather than spending time in formulating them. It was frustrating for the business-orientated roles when they sensed a lack of engagement from the developers regarding formulating the goals. However, both interviews and the survey showed that most project members, including developers, found using the OKR framework useful for focus and direction, as well as a perceived increase in performance and efficiency. Because earlier research on goals has found that whether goals are self-set or assigned does not matter that much as long as the goal's purpose is given [13], perhaps developers do not need to spend too much time on formulating the goals, as long as they agree and feel ownership to the goals set.

6 CONCLUSION AND FUTURE WORK

This study examined how distributed agile teams utilized the goal-setting framework OKRs. We discovered that team members considered setting OKRs as challenging. This demonstrates the importance of OKR training, not only for team leaders, but also for all team members, to acquire shared knowledge.

Although implementing OKR was challenging, our results indicate that the framework benefits teams because it allows them to set agreed goals and revisit them frequently. Teams could view other teams' OKRs via an OKR tracker; however, not everyone took advantage of this feature. Therefore, future research should focus on ways to improve the alignment of shared goals among teams in large-scale distributed projects.

Additionally, because OKR is a long-term method, teams must be supported for a longer duration of time. In our instance, it took more than a year for teams to recognize the frameworks' benefits. We found that different roles have different engagement around formulating the OKRs, but all roles were very engaged in following the framework. Future research should investigate how involved developers need to be in the details of formulating the goals. Many developers experienced the time spent in workshops as frustrating because they wanted to spend time producing code, not sitting in workshops and meetings.

ACKNOWLEDGMENTS

We are thankful to Henrik Vedal for assisting with data collection. The Research Council of Norway through A-team (grant 267704) and Transformit (grant 330104) supported the work.

REFERENCES

[1] Steve Berczuk and Yi Lv. 2010. We're all in this together. *IEEE Software* 27, 6 (2010), 12–15.
[2] Marthe Berntzen, Rashina Hoda, Nils B Moe, and Viktoria Stray. 2022. A Taxonomy of Inter-Team Coordination Mechanisms in Large-Scale Agile. *IEEE Transactions on Software Engineering* 01 (2022), 1–1. https://doi.org/10.1109/TSE.2022.3160873
[3] Juliet Corbin and Anselm Strauss. 2014. *Basics of qualitative research: Techniques and procedures for developing grounded theory.* Sage publications.
[4] Kim Dikert, Maria Paasivaara, and Casper Lassenius. 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software* 119 (Sept. 2016), 87–108. https://doi.org/10.1016/j.jss.2016.06.013
[5] John Doerr. 2018. *Measure what matters: How Google, Bono, and the Gates Foundation rock the world with OKRs.* Penguin.
[6] Henry Edison, Xiaofeng Wang, and Kieran Conboy. 2021. Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review. *IEEE Transactions on Software Engineering* PP, 99 (2021), 1–1. https://doi.org/10.1109/tse.2021.3069039

[7] Amy Farrow and Steve Greene. 2008. Fast & predictable a lightweight release framework promotes agility through rhythm and flow. In *Agile 2008 Conference*. IEEE, 224–228.

[8] Patricia J. Guinan, Salvatore Parise, and Nan Langowitz. 2019. Creating an innovative digital project team: Levers to enable digital transformation. *Business Horizons* 62, 6 (2019), 717–727. https://doi.org/10.1016/j.bushor.2019.07.005

[9] J. D. Herbsleb. 2007. Global Software Engineering: The Future of Socio-technical Coordination. 188–198. https://doi.org/10.1109/fose.2007.11

[10] Rashina Hoda, James Noble, and Stuart Marshall. 2012. Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering* 39, 3 (2012), 422–444.

[11] Laurie J Kirsch, Dong-Gil Ko, and Mark H Haney. 2010. Investigating the antecedents of team-based clan control: Adding social capital as a predictor. *Organization Science* 21, 2 (2010), 469–489.

[12] Edwin A Locke and Gary P Latham. 1990. *A theory of goal setting & task performance*. Prentice-Hall, Inc.

[13] Edwin A Locke and Gary P Latham. 2002. Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American psychologist* 57, 9 (2002), 705.

[14] Marius Mikalsen, Nils Brede Moe, Viktoria Stray, and Helga Nyrud. 2018. Agile digital transformation: a case study of interdependencies. *International Conference on Information Systems 2018, ICIS 2018* (2018).

[15] Nils Brede Moe. 2013. Key challenges of improving agile teamwork. In *International conference on agile software development*. Springer, 76–90.

[16] Nils Brede Moe, Darja Šmite, Maria Paasivaara, and Casper Lassenius. 2021. Finding the sweet spot for organizational control and team autonomy in large-scale agile software development. *Empirical Software Engineering* 26, 5 (2021), 1–42.

[17] Paul R. Niven and Ben Lamorte. 2016. *Objectives and key results: Driving focus, alignment, and engagement with OKRs*. John Wiley & Sons.

[18] Maria Paasivaara and Casper Lassenius. 2014. Communities of practice in a large distributed agile software development organization–Case Ericsson. *Information and Software Technology* 56, 12 (2014), 1556–1577.

[19] Maria Paasivaara, Casper Lassenius, and Ville T Heikkilä. 2012. Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work?. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*. 235–238.

[20] Kai Petersen and Claes Wohlin. 2010. The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering* 15, 6 (2010), 654–693.

[21] Johan E Ravn, Nils Brede Moe, Viktoria Stray, and Eva Amdahl Seim. 2022. Team autonomy and digital transformation. *AI & SOCIETY* (2022), 1–10.

[22] K. Schwaber and Beedle. 2001. *Agile Software Development with Scrum*. Upper Saddle River: Prentice Hall.

[23] Igor Steinmacher, Ana Paula Chaves, and Marco Aurélio Gerosa. 2013. Awareness support in distributed software development: A systematic review and mapping of the literature. *Computer Supported Cooperative Work (CSCW)* 22, 2-3 (2013), 113–158.

[24] Viktoria Stray, Nils Brede Moe, Henrik Vedal, and Marthe Berntzen. 2022. Using Objectives and Key Results (OKRs) and Slack: A Case Study of Coordination in Large-Scale Distributed Agile. In *Proceedings of the 55th Hawaii International Conference on System Sciences*.

[25] Bianca Trinkenreich, Gleison Santos, Monalessa Perini Barcellos, and Tayana Conte. 2019. Combining GQM+ Strategies and OKR-Preliminary Results from a Participative Case Study in Industry. Springer, 103–111.

[26] Christina R Wodtke. 2017. *Radical focus: Achieving your most important goals with objectives and key results*. Boxes and Arrows.

[27] Robert Yin. 2017. *Case Study Research and Applications: Design and Methods* (6 ed.). SAGE Publications.