User Involvement in Relational Digital Transformation: A Case Study of Agile Software Development Practices in a Large Organization

Morteza Moalagh
Computer Science Department, Norwegian University of Science and Technology (NTNU), Trondheim, Norway *morteza.moalagh@ntnu.no*

Vegard Svesengen
Computer Science Department, Norwegian University of Science and Technology (NTNU), Trondheim, Norway *vg.sves@hotmail.com*

Babak A. Farshchian
Computer Science Department, Norwegian University of Science and Technology (NTNU), Trondheim, Norway *babak.farshchian@ntnu.no*

Marius Mikalsen
Computer Science Department, Norwegian University of Science and Technology (NTNU), Trondheim, Norway *marius.mikalsen@ntnu.no*

Abstract

This study explores how agile software development supports user involvement in the context of relational digital transformation. Relational digital transformation offers an alternative to top-down approaches by viewing technologies as relational constructs embedded within evolving organizational practices. Achieving such transformation requires attention to the relationships that shape technologies, emphasizing sustained collaboration between users and development teams. While software plays a central role, traditional digital transformation often neglects relational dynamics. Agile software development—with its iterative process, adaptability to user needs, and emphasis on user involvement—emerges as a viable approach to implement relational digital transformation. This paper uses a qualitative case study to examine both the user involvement practices employed and users' perspectives on these practices. The findings identify agile software development practices that can effectively support user involvement, while also revealing limitations, particularly the challenges of maintaining long-term participation and embedding user involvement beyond initial deployments.

Keywords: digital transformation, relational digital transformation, agile software development, user involvement.

1. Introduction

Digital transformation (DT) denotes technology-enabled changes to make organizations better equipped to meet present and future challenges[22], [43]. DT is often initiated by conditions external to the organization, for instance, disruptive technologies such as artificial intelligence and societal challenges such as aging populations[26], [43]. At the same time, DT is increasingly perceived as a significant part of an organization's internal practices and should recognize the dynamic interplay between development teams and

users [3, 4], [14]. In other words, a top-down or outside–in approach, where DT is reduced to a product that can be acquired, may ignore the necessary relations among developers and users.

This paper uses the term Relational Digital Transformation (RDT) as an alternative to the above top-down reductionist approaches. RDT is a conceptualization of DT that regards technologies as "made up of relations and entwined in relations that are constantly evolving" [3, p.4]. In order to succeed with RDT, we need to focus on the relations that produce the technologies and embed technology development within organizational practices by emphasizing the involvement of users [6], [24]. Software is a central enabler of RDT. However, the top-down approach to DT often overlooks the importance of embedding software development within social practices and sustaining close collaboration between development teams and users, an essential prerequisite for realizing the principles of RDT [12].

In our view, the approach that stands out as a viable solution to implement RDT is agile software development (ASD). ASD "affords teams the ability to inspect their work and adapt to changes in user requirements and broadens the team's interactions with stakeholders beyond the IT group" [2, p.114]. With its focus on continuous change and user involvement [2], [10], ASD promises to address the above-mentioned relations in RDT –i.e., the relations between development teams, users, and the software.

Our research interest lies in the alignment between RDT and ASD. We are particularly interested in how agile approaches contribute to RDT through emphasis on user involvement, strengthening relations among stakeholders. More specifically, we aim to improve our understanding of ASD as an approach to implementing RDT in large organizations. The research question we aim to address in this paper is: ***How can ASD practices contribute to involving users in the relational digital transformation program?***

To address the research question, we report on a qualitative, interview-based investigation of ASD in the Norwegian Labour and Welfare Administration (NAV in Norwegian). Recently, NAV has undergone a significant transformation of its approach to software development, moving away from its earlier outsourcing strategy and investing heavily in ASD within a modern product organization. This transformation complies with the recommendation by [23] to treat software development as a central part of any RDT program. NAV can be regarded as a rare case where a transformation of software engineering practices preceded the transformation of the business side. NAV's IT department has, therefore, been acting as a strong driver for this transformation.

Our main contribution is an empirical case study from a large-scale RDT program focusing on ASD. Our findings are relevant because they help us unpack the interactions between agile teams and users, demonstrating the complexity of supporting relations in RDT. This case study shows how the organization and its RDT program are impacted by various ASD practices. Earlier studies of ASD in organizations have often employed a software engineering lens, focusing on the practices within and among agile teams. Our study, through its RDT lens, contributes to a better understanding of ASD in an organizational context. Moreover, this case study shows how ASD practices are adapted to specific organizational settings, such as scale. For instance, our findings show that NAV has implemented a phased agile model, employing different user involvement practices in each phase. Our findings also show some of the limitations of ASD for RDT programs, for instance, the challenge of maintaining relations in the long run.

2. Background

2.1. Relational Digital Transformation and User Involvement

Most information systems researchers agree that DT is about fundamental organizational and societal change initiated by new technological possibilities [32], [40]. However, there are different views on how DT can be successfully managed. While some regard DT as a top-down process, others see it as organic and emergent, and still others see it as a combination [22]. On one hand, digital transformation is regarded as an organization-wide "program" initiated by top management to transform the organization into a version of

itself that is deemed to be more fit for future markets [43,44]. On the other hand, DT is an emergent process of evolving relationships between technology, people, organizations, and society [3]. This view, which is called "relational digital transformation (RDT)," emphasizes that "to understand the things we think of as technologies... we need to focus on the relations that constitute them." [3, p.4], aligning with sociotechnical and technologies-in-practice views [36].

Relational digital transformation (RDT) emphasizes that change is driven not merely by technological adoption but by evolving relationships between people, organizations, and technologies. RDT views change as a continuous co-evolution of digital technologies and industry practices shaped through shifting user interactions [20]. The success of RDT increasingly depends on how technology providers, including development teams, involve users during initiatives [42]. Research shows that user involvement not only enhances knowledge sharing but also improves innovation outcomes and firm performance, reinforcing the critical importance of relational engagement strategies [46].

Although RDT stresses strong user involvement, organizations frequently struggle to achieve it. When software development teams operate in disconnected silos, coordinate goals and execute them suffers [15], [17]. A common pitfall is concentrating too heavily on new technology deployments while neglecting collaboration and coordination, which is essential for embedding transformation into everyday practices [38]. Excluding users from the change process leads to resistance, disengagement, and operational setbacks [18]. Without actively involving users and embedding change structurally, sustaining RDT remains an uphill battle [44].

2.2. User Involvement in Agile Software Development

ASD can support RDT by promoting continuous user involvement, iterative delivery, and decentralized decision-making [10], [25]. Approaches like Scrum aim to shift away from isolated development by engaging users and stakeholders throughout the process [10]. Empirical studies suggest that successful software development depends not only on adopting flexible practices but also on embedding user input into daily workflows and routines [19]. However, in practice, user involvement often falls short, with misalignments between developers and users creating communication gaps [7]. This disconnect weakens the iterative feedback loops essential to RDT and risks turning agile approaches into rigid structures that are misaligned with the goal of supporting a user-focused RDT program [8].

This challenge of insufficient and fragmented user involvement in ASD becomes even more pronounced in large and distributed organizations [30]. RDT programs in these contexts typically span multiple departments, technologies, and geographic locations, and the scale and cross-functional nature of such programs introduce substantial complexity, requiring new ways of organizing work, coordinating stakeholders, and delivering value [39]. Although agile approaches promote user involvement, sustaining meaningful involvement across diverse, geographically dispersed teams often proves difficult [1]. Organizational silos, limited user availability, and inconsistent feedback loops further weaken ASD practices in these complex environments [35]. Despite growing emphasis on RDT, there is still limited research on concrete practices for involving non-technical users in large and distributed RDT programs that use ASD [8], [28]. To address this gap, this study investigates practical approaches that support user involvement and explores how users perceive these practices in the RDT programs.

3. Research Method

We have designed the study as a single exploratory and interpretative case study [45]. With a focus on the in-depth investigation of a phenomenon through multiple data sources, the case study method fits well with our goal of increased understanding of a complex sociotechnical phenomenon through cycles of new interpretations [13]. The following section presents the key components of the exploratory case study, including case description, data collection techniques, and data analysis methods.

3.1. Case description

NAV (The Norwegian Labour and Welfare Administration) is a central public agency in Norway, employing over 22,000 people. NAV provides a wide range of services, including employment assistance, financial benefits, and social support. It helps job seekers find employment, offers training programs, and facilitates connections between employers and potential workers. Headquartered in Oslo, NAV's central office oversees 12 county offices ("Fylke" in Norwegian) and 456 local offices across the country [33].

NAV has undergone substantial organizational and technological transformations, particularly within its information technology (IT) department. In 2016, NAV initiated a major shift in its digital transformation program, transitioning away from traditional project structures toward an agile software development (ASD), insourced model [5], [31]. The IT department was reorganized into a matrix structure with different product areas focused on different categories of services [31]. This shift has been widely recognized in recent years, winning a national innovation award [5]. The IT department now has around 800 employees and 90 software development teams who are collaborating with a diverse range of users across the country [34]. In each local office, selected frontline employees work as user representatives and serve as key points of contact between the development teams and the wider organization, assuming roles such as change agents or super users [27]. These individuals are responsible for gathering requirements, conducting system testing, providing user training, and offering ongoing support to facilitate the implementation and usage of the systems [29].

3.2. Data Collection and Analysis

To collect data, this study relied on both interviews and existing documents. 15 NAV employees from both the development teams and the internal users were interviewed. The interviews were semi-structured and centered around two key questions: "How do users take part in agile software development?" and "How do users perceive their influence on the development process?" Because our participants came from different professional backgrounds, we tailored our questions to align with each group's daily tasks and their role in ASD. Each interview lasted between one and two hours. In addition to the interviews, we also reviewed a number of existing documents, focusing on those that shed light on NAV's strategy for DT. Table 1 outlines data sources, assigning each participant a code (P1–P15) for reference in the findings. Identifiable details, including department, age, experience, and office location, were intentionally excluded to protect anonymity.

Table 1- Detail of data sources

Data Collection Technique	Details	#
Semi-Structured Interview	**Internal Users**	15
	Counsellors from different local offices (P1-P6)	
	Case Workers from different local offices (P7-P8)	
	Development Teams	
	Product Leaders (P9-P11) from different development teams	
	Technical Leads (P12-P13)	
	Domain Expert (P14)	
	Designer (P15)	
Documents	NAV public reports, Strategic documents, and the NAV website	5

After data collection, interviews were transcribed and imported into NVivo for analysis using inductive thematic analysis [9]. The unit of analysis was ASD practices and the relationships between development teams and internal users. Initially, transcripts were reviewed to assess relevance to the research topic. In a second round, data were coded using inductive open coding [41] identifying practices such as pilot tests, user requests, and open days. Coding was iterative and involved peer debriefing sessions to mitigate bias [16]. Codes were refined through discussions informed by relevant theory. The finalized results were grouped and are presented in the findings section.

4. Findings

In this section, we first present the Agile Software Development (ASD) practices implemented by the case organization (NAV) during three distinct phases—discovery, initial roll-out, and post-roll-out—to involve internal users in the relational digital transformation (RDT) program. Following this, we report on user perspectives regarding their involvement in ASD. Quotations from participants (details of which are presented in Table 1) are provided as evidence following each corresponding finding.

4.1. Agile Software Development Practices for User Involvement in Digital Transformation

NAV, through its RDT program, prioritizes the continuous involvement of internal employees—particularly frontline staff in local offices—to ensure software aligns with actual user needs. This involvement spans three key phases. The *discovery phase* focuses on collaboration between development teams and users to gather insights. The second, *the initial roll-out phase*, covers the period immediately before and after the release of new software or features, concentrating on early adoption and feedback collection. Finally, the *post-roll-out phase* involves ongoing iterations informed by user input and performance data. Development teams apply Agile Software Development (ASD) practices to ensure software solutions are regularly updated and improved after their release. Subsequent sections detail each phase, user–developer relations, and specific involvement practices summarized in Table 2.

Discovery phase

The discovery phase is the stage where development teams begin to understand and identify their users, gain insights into existing or potential software solutions, and gather the necessary requirements to build and prioritize their backlogs: "*Typically, early in a project, or early in a phase when we are getting a new product or finding a new product, we work very structured with gaining insight [about future users]. That is what we call the discovery phase on the development side"* [P9]. In this phase, the relationship between development teams and users, which is central to RDT, is shaped, and development teams use various practices to identify and involve users and define backlog features.

To recruit users and involve them in the ASD development, teams draw on a variety of approaches, such as tools, user panels, and designated user representatives. For example, some tools can be used to recruit users. These tools help development teams recruit the most relevant users for collaboration by analyzing real user behaviors, feedback, and engagement patterns. Development teams may also practice user panels from different Microsoft Teams channels in the NAV to recruit users to gather data about the new software solutions or features. Moreover, sometimes they may collaborate with user representatives like change agents or super users in different local offices to gather data or ask to introduce relevant users: *"We have a tool called …, which we can recruit, and then there are often interviews digitally or user tests. We also have a user organization that we work quite closely with. A bit like forums, sometimes they are too. We call it a user panel. […] and then we have case managers, that's it. We have two departments that we collaborate with, and I am also the direct contact person"* [P15].

During the discovery phase, backlog feature ideas emerge from both user feedback and regulatory changes driven by new legislation. To construct and prioritize the backlog, development teams apply a range of techniques, including reviewing official government communications, conducting interviews with users or their representatives—such as change agents or super users—and carrying out observational studies to uncover user needs. Additionally, questionnaires are occasionally used to gather input from internal users, such as counselors and caseworkers, on potential new features: "*The law is changing frequently, so the requirements are basically from the changes in the law, which are quite frequent in our case. […] We also go out and visit case workers in different parts of the country and get their feedback. Sometimes, we need to talk to them a bit to say what they are really unhappy about and honestly tell us what they like and don't like about the*

system" [P12].

Initial roll-out phase

The initial roll-out phase refers to the period preceding the release of a new software and the first few weeks following its deployment. During this phase, three distinct practices are identified that shape the relationship between the development team and users: acceptance tests, pilot tests, and open days, all of which are designed to actively involve users in the ASD.

For the acceptance testing, development teams typically engage pre-recruited users, such as counselors or case workers, through designated communication channels. These users are then invited to begin testing the software in a real-world context. During this process, users report any bugs or usability issues they encounter. As a result, development teams can identify which features are essential and which require modification, as well as how existing functionalities may need to be adjusted to better meet user needs and ensure the system's overall effectiveness and usability: *" if we push code out that's been tested and it seems to work fine, we just push it out to our users and we have those users, those case workers, they are fine with that. And then they just start to report the bugs, you know, every time. We have a major release afterwards,"* [P10].

Pilot tests are a common tool used by development teams in the NAV's RDT program when there are significant updates and major releases. Pilot tests are done in select NAV offices with select users. They allow development teams to remove errors in the system, and develop a relationship with users and involve them deeply in the ASD and improve documentation before the main release to all users in the country: *" In a large organization, as NAV is, it is costly to release new code(solutions) to the whole organization. It requires a lot of implementation(in local offices) and a lot of work. So, at least when we are doing larger projects, it(pilot test) helps us answer any questions and solve issues. It helps us know what we have to communicate when we eventually release it to everyone because the pilot gives us an idea of what is easy to understand (for users)"* [P9].

The third practice of user involvement in the initial roll-out phase was the open days. Open days are synchronous online events held in Microsoft Teams rooms –as opposed to channels –where users from different local offices around the country can "meet" developers online. Open days also seem to allow for strategic discussions beyond bugs: "*Right after the roll-out, we held two open days for counselors from the [local offices]. We were available for a scheduled time so everyone who uses the relevant parts of Modia [the IT solution] could meet us. The first time we were available for the whole day from 9 to 15 o'clock. Counselors could show up whenever they wanted, ask questions, give us feedback, and discuss things they were worried about or if they felt any changes were needed"* [P14].

By dedicating one or two days in the open days practice to direct interactions with users, development teams gain timely and actionable insights on the system shortly after its initial release. Users also expressed strong support for the practice, emphasizing that the opportunity to engage soon after deployment provided meaningful input and enhanced their sense of involvement in the ASD: "*We have recently rolled out, in my department, some new and very technical solutions, and we have had an open day with the developers. We could give feedback on what we did not like about the new solution, and also talk about what we liked. They also told us about the upcoming priorities and what was coming soon"* [P6].

Post-roll-out phase

The post-roll-out phase describes the period after the initial roll-out of a software or a feature and the following continuous development. The primary method of user involvement in this phase is user requests. User requests can be sent from internal users like counselors or case workers or from their representatives like change agents, or super users to development teams in several ways, e.g., through an internal channel called Porten on NAV's intranet, or through Yammer (a commercial messaging tool), or through direct contact between developers and users, for example, on Microsoft Teams. All interview participants mentioned using either Porten or Yammer multiple times for user requests:

"*Every request we receive in Porten ends up with me as a Jira-case [Jira is a product management tool]. Some users submit many requests, about things they want to change, and things they want in future deployments. They give feedback on what is missing and what could be better. Then, I will usually create a list with the requests I feel are relevant and realistic. Then we have meetings within the team, which we call product talks, to discuss solutions and future development"* [P14].

In the post-roll-out phase, development teams adopting an agile approach address user requests through iterative updates, ranging from minor enhancements to major system releases: *"We have the channels to tell them about new things coming plans and ask them a questions and they can write into us about problems or ideas or whatever"* [P13]. Depending on the nature of the change, users may either be informed and involved or, in the case of smaller updates, the modifications are usually deployed with minimal communication and without broad announcements: "*There are small notifications in the systems in Modia, which notifies you when there is new functionality. There are often news messages on Navet, our intranet. Then it is all about sorting it; what is relevant for my work?"*[P4]. When new software or major features are released, greater emphasis is typically placed on communication and preparation prior to the launch. A planned release date is usually established, accompanied by training sessions and courses to ensure users are adequately prepared: *"If they release a new program, there will usually be more buzz around it. Usually, we get a release date, specific information about that date, what changes it is, the system's name, and so forth"*[P2].

Table 2. User Involvement Practices in different development phases of agile software development

Development Phase	User Involvement Practice
Discovery phase	• User Recruitment Tool • User Panel • User Representatives • User Interviews • Observational Studies • Questionnaires
Initial roll-out phase	• Acceptance tests • Pilot tests • Open days
Post-roll-out phase	• User Requests

4.2. User perspective on involvement in agile software development

The agile practices used in the RDT program in NAV received mixed feedback from the users who were interviewed. Some see the value in gradually improving a system as it is used. For example, one pointed to the educational aspects of being part of an ASD when comparing the new approach to what was common in the past: "*So if you think about a counselor's role in NAV, [in the past] you would have used Arena's [the old IT system's] features for everything, which takes a long time to learn. In my experience, learning something like that is a more difficult task. Compared to now, with the agile development"* [P2]. Another user also appreciated the opportunity to propose improvements as their familiarity with the software solution increased: *" We have good opportunities to report wishes and requests. Moreover, we can see in the deployed systems that our requests have been seen and heard"* [P1]. They acknowledged that the development teams would receive numerous requests and emphasized the importance of prioritization within the backlog. This involvement was perceived as fruitful by the users, and they reported a sense of satisfaction in seeing their feedback incorporated into the ASD: "*We see the results of the contact [with development teams]. We have been listened to, and what we say has been received on the other end and worked with. Everything we say cannot be done, or maybe even should not be done. That should be up to the developers [to decide], but the fact that we have an opportunity to be heard, I personally think that has been very good"* [P4].

While user involvement in the development process was generally positive, some

participants experienced a degree of frustration as they navigated the demands and challenges of the agile approach. As software solutions are continuously updated, internal users like counselors and case workers can have a challenging job of keeping updated, and can continue using the systems in a less effective way even if effective solutions exist in later releases: "*What is common in NAV now is that we start implementing things before they are finished. It gets built piece by piece. And often that is very useful for us because we can start working with it early. But in the long run, it can become an issue because people learn to misuse it [the solution] before we know how it will work when it is more complete"* [P6]. Moreover, users are expected to be informed and trained regarding all changes. However, due to the high frequency and volume of incremental updates in ASD, development teams often prioritize user training for major releases only. Consequently, users may perceive a lack of involvement or awareness, as numerous minor changes are accumulated over time without training, leading to a knowledge gap: *" In the past, we developed an hour of major upgrades of major changes. But now, [with agile software development], you have lots of small changes along the way, then there are some big ones in between, and there are big peaks. And then it's not like we're going to stand there and kind of have a lot of training. Because it might only take 10 minutes to learn the new part […]. I have very little insight into exactly what hasn't been communicated, and it's something, yes, something that should have been addressed for sure"* [P7].

During the post-rollout phase, users frequently submit numerous enhancement requests based on their evolving needs. However, development teams often face competing priorities, as they must also address strategic directives from organizational leadership or comply with external mandates such as government regulations. As a result, not all user-generated requests can be accommodated, leading to a perception among users that their involvement in the ASD diminishes during this stage. Furthermore, when new features are deployed without prior consultation or adequate communication, users may perceive these changes as misaligned with their actual needs and of limited value: "*Sometimes [the product team] will release some new feature, as they often do with the agile methods. And then, we see this mass hysteria in the office, and many people are unhappy with the change. And then [the product team] changes something quickly without considering who is behind the hysteria [..] In my experience, they sometimes make changes rather quickly [if enough people complain] without considering the goal or reason behind it"* [P3].

5. Discussion

Relational digital transformation (RDT) literature emphasizes the importance of user involvement as part of the co-evolution of technology and organizational practices [3]. Our study demonstrates how such involvement is operationalized concretely through agile software development (ASD) practices at NAV (as the case organization). At NAV, internal users such as counselors and caseworkers were included across structured phases—discovery, initial roll-out, and post-roll-out—to facilitate ongoing feedback and iterative improvements. These practices not only enabled responsiveness to evolving needs but also embodied the continuous relational engagement that RDT theorizes [20].

This study also extends Maruping et al.'s [23] point by illustrating that the way software development is practiced significantly affects how user involvement unfolds within RDT. In the NAV case, ASD practices were not applied generically but adapted through specific techniques, such as user recruitment via tools, user panels, and the use of change agents, to support sustained user interaction. These practices facilitated the relational exchanges emphasized in RDT [3], and fostered a shift from transactional implementation to value-driven collaboration [21]. As users contributed to feature definition, pilot testing, and post-release improvement, they actively shaped the evolution of software solutions, aligning with the sociotechnical view that technologies are constituted through ongoing relations [3]. Moreover, such iterative, feedback-driven practices contribute to the shift from transactional to relational user involvement emphasized in recent literature [42], thereby positioning ASD practices not just as a development method but as a potential relational infrastructure underpinning successful RDT [46].

Our findings also highlight that standard ASD practices may be insufficient for supporting RDT in large-scale organizations. Despite ASD's emphasis on responsiveness and collaboration [10], the NAV case shows that added complexity, due to organizational scale, user diversity, and cross-location coordination, necessitated more structured and phase-based adaptations of agile. These included discovery activities for user recruitment, structured pilot tests, and dedicated open-day events to ensure broad-based user involvement. This supports previous findings that agile methods often require contextual adaptation in large-scale organizations [37]; moreover, it contributes to the RDT literature by showing that tailoring ASD practices to organizational complexity is essential to achieve sustained user involvement [30].

Finally, our study reveals tensions in maintaining user involvement during the later stages of ASD. While initial phases encouraged strong participation, the post-roll-out phase saw declining user involvement due to competing strategic priorities and limited communication around incremental updates. This reflects common challenges in sustaining ASD when user requests are deprioritized in favor of regulatory compliance or top-down directives [18]. This highlights a structural limitation in ASD execution: *despite its iterative ethos, user involvement can wane without mechanisms for consistent involvement*. For RDT, this raises important questions about how organizations can embed and normalize user involvement beyond initial deployments [8]. *As such, sustaining RDT requires not only initial relational practices but long-term strategies for institutionalizing user involvement within ASD.*

6. Implications for Practice

The study presents valuable guidance for practitioners by showcasing real-world user involvement practices from a large-scale organization with over ten years of agile experience. Table 2 and the findings offer practical summaries of how these practices are applied, with insights drawn from both users and development team members. A key takeaway is the importance of fostering collaboration through a bottom-up approach, which enables stronger relationships between users and developers and supports user involvement in relational digital transformation (RDT).

Additionally, this research highlights that agile software development (ASD) is a continuous journey rather than a one-time project. User involvement should not end after delivery but continue through all stages, including support and system evolution. Organizations need to establish consistent ways of involving users to maintain responsiveness to change. This ongoing collaboration ensures that the software remains aligned with user needs and business goals over time, reinforcing the core principles of ASD.

7. Limitations and Threats to Validity

The first limitation of this study is that it is based on a single case involving a large public sector organization in a developed country. As with most case study research, the findings are context-specific and may not be readily generalizable to other organizational types or settings. These contextual factors should be considered when interpreting the study's implications. Secondly, the study's qualitative and interpretive design enables a detailed, context-rich understanding of the research setting. However, this approach also introduces subjectivity. The findings are shaped by the researcher's interpretations; however, as discussed in the method section, some steps were taken to mitigate subjectivity in the data analysis.

The study's construct validity is influenced by the selection of interview participants and data sources. If we were to include more senior management, we would get more insight into the top-down organizational strategies for relational digital transformation. Additionally, although some local offices were included in the study, including even more local offices around the country would capture wider variation in experiences and practices across different internal users.

8. Conclusion and Future Research

This study demonstrates how Agile Software Development (ASD) practices can facilitate sustained user involvement within the context of Relational Digital Transformation (RDT). At NAV, ASD was effectively adapted to the organization's inherent complexity, fostering a dynamic co-evolution between technological solutions and organizational processes—an alignment central to RDT. In this setting, the relationship between development teams and users serves as a key enabler of RDT, achieved through continuous collaboration rather than one-time efforts. The study provides a valuable example of how such collaboration can be operationalized through agile practices. However, challenges in the post-roll-out phase reveal a vulnerability: without institutionalized strategies, user involvement declines over time. This key finding underscores the importance of establishing long-term, organization-wide mechanisms to maintain user involvement and ensure that the benefits of ASD persist beyond the initial development lifecycle.

Future research could explore practices for sustaining user involvement beyond the initial phases of ASD, especially in large-scale, complex, and distributed organizations. A key focus is how institutionalizing agile practices shapes the long-term trajectory of RDT. Comparative case studies across public and private sectors, in both developed and developing countries, are essential to identify contextual differences in user involvement. Such research could reveal diverse challenges organizations face in enabling meaningful, sustained user involvement within ASD across varying institutional and cultural settings.

Acknowledgements

This work was supported by research funds from the Norwegian Labour and Welfare Administration (NAV). We are thankful to the anonymous reviewers for their constructive feedback, and extend our appreciation to all NAV employees who generously contributed to the study through interviews.

References

1. Alyahya, Sultan ; Bin-Hezam, Reem ; Maddeh, Mohamed: Supporting Remote Customer Involvement in Distributed Agile Development: A Coordination Approach. In: *IEEE Transactions on Engineering Management* Bd. 71 (2024), S. 688–701
2. Baham, Corey ; Hirschheim, Rudy: Issues, challenges, and a proposed theoretical core of agile software development research. In: *Information Systems Journal* Bd. 32 (2022), Nr. 1, S. 103–129
3. Bailey, Diane E. ; Faraj, Samer ; Hinds, Pamela J. ; Leonardi, Paul M. ; Von Krogh, Georg: We Are All Theorists Of Technology Now: A Relational Perspective On Emerging Technology And Organizing. In: *Organization Science* Bd. 33 (2022), Nr. 1, S. 1–18
4. Baiyere, Abayomi ; Grover, Varun ; Lyytinen, Kalle J. ; Woerner, Stephanie ; Gupta, Alok: Digital "x"—Charting a Path for Digital-Themed Research. In: *Information Systems Research* Bd. 34 (2023), Nr. 2, S. 463–486
5. Bernhardt, H.B.: Digital Transformation in NAV IT 2016–2020: Key Factors for the Journey of Change. In: Mikalef, P. ; Parmiggiani, E.: *Digital Transformation in Norwegian Enterprises*. 1. Aufl. : Springer, 2022 — ISBN 978-3-031-05275-0, S. 115–134
6. Blanka, Christine ; Krumay, Barbara ; Rueckel, David: The interplay of digital transformation and employee competency: A design science approach. In: *Technological Forecasting and Social Change* Bd. 178 (2022), S. 121575
7. Buchan, Jim ; Bano, Muneera ; Zowghi, Didar ; MacDonell, Stephen ; Shinde, Amrita: Alignment of Stakeholder Expectations about User Involvement in Agile Software Development. In: *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*. Karlskrona Sweden : ACM, (2017)
8. Carroll, Noel ; Conboy, Kieran ; Wang, Xiaofeng: From transformation to normalisation: An exploratory study of a large-scale agile transformation. In: *Journal of Information Technology* Bd. 38 (2023), Nr. 3, S. 267–303
9. Clarke, Victoria ; Braun, Virginia: *Thematic analysis: a practical guide*: Sage, 2022

10. Conboy, Kieran: Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. In: *Information Systems Research* Bd. 20 (2009), Nr. 3, S. 329–354
11. Conboy, Kieran ; Morgan, Lorraine: Beyond the customer: Opening the agile systems development process. In: *Information and Software Technology* Bd. 53 (2011), Nr. 5, S. 535–542
12. Dikert, Kim ; Paasivaara, Maria ; Lassenius, Casper: Challenges and success factors for large-scale agile transformations: A systematic literature review. In: *Journal of Systems and Software* Bd. 119 (2016), S. 87–108
13. Dubois, Anna ; Gadde, Lars-Erik: Systematic combining: an abductive approach to case research. In:*Journal of Business Research* Bd. 55 (2002), Nr. 7, S. 553–560
14. Faraj, Samer ; Leonardi, Paul M: Strategic organization in the digital age: Rethinking the concept of technology. In:*Strategic Organization* Bd. 20 (2022), Nr. 4, S. 771–785
15. Heracleous, Loizos ; Gledhill, David: Why Digital Transformation May Fail – And What Can Be Done About It. In:*The Journal of Applied Behavioral Science* Bd. 60 (2024), Nr. 1, S. 215–219
16. Janesick, V. J.: Peer debriefing. In:*The Blackwell encyclopedia of sociology.* (2007)
17. Jonathan, Gideon Mekonnen ; Rusu, Lazar ; Perjons, Erik: Digital Transformation in Public Organisations: IT Alignment-Related Success Factors. In: *30th International Conference on Information Systems Development, ISD 2022*. Cluj-Napoca, Romania, 2022
18. Klein, Sascha P. ; Spieth, Patrick ; Söllner, Matthias: Employee acceptance of digital transformation strategies: A paradox perspective. In: *Journal of Product Innovation Management* Bd. 41 (2024), Nr. 5, S. 999–1021
19. Kuhrmann, Marco ; Tell, Paolo ; Hebig, Regina ; Klunder, Jil ; Munch, Jurgen ; Linssen, Oliver ; Pfahl, Dietmar ; Felderer, Michael ; u. a.: What Makes Agile Software Development Agile? In: *IEEE Transactions on Software Engineering* Bd. 48 (2022), Nr. 9, S. 3523–3539
20. Lanamäki, Arto ; Väyrynen, Karin ; Laari-Salmela, Sari ; Kinnula, Marianne: Examining relational digital transformation through the unfolding of local practices of the Finnish taxi industry. In:*The Journal of Strategic Information Systems* Bd. 29 (2020), Nr. 3, S. 101622
21. Malik, Mohsin ; Andargoli, Amir ; Chavez Clavijo, Roberto ; Mikalef, Patrick: A relational view of how social capital contributes to effective digital transformation outcomes. In: *The Journal of Strategic Information Systems* Bd. 33 (2024), Nr. 2, S. 101837
22. Markus, M. Lynne ; Rowe, Frantz: The Digital Transformation Conundrum: Labels, Definitions, Phenomena, and Theories. In: *Journal of the Association for Information Systems* Bd. 24 (2023), Nr. 2, S. 328–335
23. Maruping, Likoebe M. ; Matook, Sabine: The evolution of software development orchestration: current state and an agenda for future research. In: *European Journal of Information Systems* Bd. 29 (2020), Nr. 5, S. 443–457
24. Mehmood, Hamid ; Farshchian, Babak A: From Representative To Transformative User Participation – A Case Study Of Public Healthcare Digitalization. In: *In the proceeding of 14th Scandinavian Conference on Information Systems*. Porvoo, Finland, 2023
25. Mergel, Ines ; Ganapati, Sukumar: Agile: A New Way of Governing. In: *Public Administration Review,* Bd. 81 (2020), Nr. 1, S. 161–165
26. Mergel, Ines ; Gong, Yiwei ; Bertot, John: Agile government: Systematic literature review and future research. In: *Government Information Quarterly* Bd. 35 (2018), Nr. 2, S. 291–298
27. Moalagh, Morteza ; Dahl-Jørgensen, Tangni C. ; Svesengen, Vegard ; Farshchian, Babak A.: User Representatives as Boundary Spanners: Case Study of Large-Scale Agile Software Development. In: *In Proceedings of the 23rd EUSSET Conference on Computer Supported Cooperative Work*. Newcastle upon Tyne, United Kingdom. June 30th – July 4th, 2025 : European Society for Socially Embedded Technologies (EUSSET), (2025)
28. Moalagh, Morteza ; Mikalsen, Marius ; Farshchian, Babak A.: Are Team Autonomy and Flexibility Enough for Agile Transformation? A Review of Transformed Practices in a Public Sector Organization. In: *Proceedings of ECSCW 2023*. Trondheim Norway, 2023, publisher: European Society for Socially Embedded Technologies (EUSSET), (2023)

29. Moalagh, Morteza ; Svesengen, Vegard ; Farshchian, Babak A.: Investigating User-Side Representatives in Large-Scale Agile Software Development. In: Peter, S. ; Kropp, M. ; Aguiar, A. ; Anslow, C. ; Lunesu, M. I. ; Pinna, A.: *Agile Processes in Software Engineering and Extreme Programming: 26th International Conference on Agile Software Development, XP 2025, Brugg-Windisch, Switzerland, June 2–5, 2025,* Bd. 545. Cham : Springer Nature Switzerland, (2025)
30. Moe, Nils Brede ; Mikalsen, Marius: Large-Scale Agile Transformation: A Case Study of Transforming Business, Development and Operations. In: Stray, V. ; Hoda, R. ; Paasivaara, M. ; Kruchten, P. (Hrsg.): *Agile Processes in Software Engineering and Extreme Programming*, *Lecture Notes in Business Information Processing*. Bd. 383. Cham : Springer International Publishing, 2020 — ISBN 978-3-030-49391-2, S. 115–131
31. Mohagheghi, Parastoo ; Lassenius, Casper: Organizational implications of agile adoption: a case study from the public sector. In: *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. Athens Greece : ACM, (2021)
32. Nadkarni, Swen ; Prügl, Reinhard: Digital transformation: a review, synthesis and opportunities for future research. In:*Management Review Quarterly* Bd. 71 (2021), Nr. 2, S. 233–341
33. NAV: What is NAV? URL: https://www.nav.no/hva-er-nav/en , (2025)
34. NAV IT: *NAV IT*. URL https://www.detsombetyrnoe.no/ , (2025)
35. Omerkhel, Qudrattullah ; Mohd Yusop, Othman ; Ismail, Saiful Adli ; Azmi, Azri: User Involvement Approach In Agile Software Development: A Systematic Literature Review. In:*Journal of Information System and Technology Management* Bd. 8 (2023), Nr. 32, S. 01–20
36. Orlikowski, Wanda J.: Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. In: *Organization Science* Bd. 11 (2000), Nr. 4, S. 404–428
37. Paasivaara, Maria ; Behm, Benjamin ; Lassenius, Casper ; Hallikainen, Minna: Large-scale agile transformation at Ericsson: a case study. In: *Empirical Software Engineering* Bd. 23 (2018), Nr. 5, S. 2550–2596
38. Philippart, Michel Henri: Success Factors to Deliver Organizational Digital Transformation: A Framework for Transformation Leadership. In: *Journal of Global Information Management* Bd. 30 (2022), Nr. 8, S. 1–17
39. Rolland, Knut H. ; Fitzgerald, Brian ; Dingsøyr, Torgeir ; Stol, Klaas-Jan: Acrobats and Safety Nets: Problematizing Large-Scale Agile Software Development. In:*ACM Transactions on Software Engineering and Methodology* Bd. 33 (2024), Nr. 2, S. 1–45
40. Saarikko, Ted ; Westergren, Ulrika H. ; Blomquist, Tomas: Digital transformation: Five recommendations for the digitally conscious firm. In: *Business Horizons* Bd. 63 (2020), Nr. 6, S. 825–839
41. Strauss, Anselm L. ; Corbin, Juliet: Open coding. Social research methods: A reader. In: *Social research methods: A reader*. New York, NY : Routledge, 2004, S. 303–306
42. Ullrich, André ; Reißig, Malte ; Niehoff, Silke ; Beier, Grischa: Employee involvement and participation in digital transformation: a combined analysis of literature and practitioners' expertise. In: *Journal of Organizational Change Management* Bd. 36 (2023), Nr. 8, S. 29–48
43. Verhoef, Peter C. ; Broekhuizen, Thijs ; Bart, Yakov ; Bhattacharya, Abhi ; Qi Dong, John ; Fabian, Nicolai ; Haenlein, Michael: Digital transformation: A multidisciplinary reflection and research agenda. In: *Journal of Business Research* Bd. 122 (2021), S. 889–901
44. Vial, Gregory: Understanding digital transformation: A review and a research agenda. In: *The Journal of Strategic Information Systems* Bd. 28 (2019), Nr. 2, S. 118–144
45. Walsham, G: Interpretive case studies in IS research: nature and method (1995)
46. Zhao, Feifei ; Meng, Tao ; Wang, Wei ; Alam, Faizan ; Zhang, Bingchao: Digital Transformation and Firm Performance: Benefit From Letting Users Participate. In: *Journal of Global Information Management* Bd. 31 (2023), Nr. 1, S. 1–23