Communications of the Association for Information Systems

Volume 54 Paper 6

2-28-2024

Machine Learning System Development in Information Systems Development Praxis

Samuli Laato
University of Turku, Tampere University

Matti Minkkinen
University of Turku

Matti Mäntymäki
University of Turku

Denis Dennehy
Swansea University

Recommended Citation

Laato, S., Minkkinen, M., Mäntymäki, M., & Dennehy, D. (2024). Machine Learning System Development in Information Systems Development Praxis. Communications of the Association for Information Systems, 54, 133-154. https://doi.org/10.17705/1CAIS.05406

Machine Learning System Development in Information Systems Development Praxis

Cover Page Footnote

This manuscript underwent peer review. It was received 10/06/2022 and was with the authors for twelve months for two revisions. Wasana Bandara served as Associate Editor. This article was submitted and reviewed during the tenure of editor-in-chief Fred Niederman.

Communications of the Association for Information Systems

Research Article DOI: 10.17705/1CAIS.05406 ISSN: 1529-3181

Machine Learning System Development in Information Systems Development Praxis

Samuli Laato
Information Systems Science
University of Turku
Faculty of Information Technology
Communication Sciences
Tampere University
0000-0003-4285-0073

Matti Minkkinen
Information Systems Science
University of Turku
0000-0002-6406-8093

Matti Mäntymäki
Information Systems Science
University of Turku
0000-0002-1981-566X

Denis Dennehy
School of Management
Swansea University
0000-0001-9931-762X

Abstract:

Advancements in hardware and software have propelled machine learning (ML) solutions to become vital components of numerous information systems. This calls for research on the integration and evaluation of ML development practices within software companies. To investigate these issues, we conducted expert interviews with software and ML professionals. We structured the interviews around information systems development (ISD) models, which serve as conceptual frameworks that guide stakeholders throughout software projects. Using practice theory, we analyzed how software professionals perceive ML development within the context of ISD models and identified themes that characterize the transformative impact of ML development on these conceptual models. Our findings show that developer-driven conceptual models, such as DevOps and MLOps, have been embraced as common frameworks for developers and management to understand and guide the ML development processes. We observed ongoing shifts in predefined developer roles, wherein developers are increasingly adopting ML techniques and tools in their professional work. Overall, our findings underscore that ML technologies are becoming increasingly prominent in software projects across industries, and that the incorporation of ML development in ISD models is an ongoing, largely practice-driven, process.

Keywords: Information Systems Development, Software Development Life Cycle, System Development Life Cycle, Machine Learning, AI.

This manuscript underwent peer review. It was received 10/06/2022 and was with the authors for twelve months for two revisions. Wasana Bandara served as Associate Editor. This article was submitted and reviewed during the tenure of editor-in-chief Fred Niederman.

1 Introduction

Information systems development (ISD) has been at the core of information systems (IS) research for over 50 years (Hassan & Mathiassen, 2018; Hirschheim & Klein, 1989; Sidorova et al., 2008). ISD models are cognitive conceptualizations of software development work that direct how software development teams are organized and how they operate (Kühl et al. 2021; Mantei & Teorey, 1989). ISD models are complementary to higher abstraction-level conceptual models, such as project portfolio management models (Dennehy & Conboy, 2019). Since ISD projects are inherently multifaceted, context-laden, and subject to changing user requirements, there is no one-size-fits-all ISD model (Benlian, 2022; Lyytinen & Rose, 2006). Despite attempts to improve the management of ISD projects, these efforts have not always had the desired effect (Baghizadeh et al., 2020; Dennehy et al., 2022; Dwivedi et al., 2015). However, periodically new advances take place, and from contemporary developments, we can note, for example, the agile manifesto originally released in 2001, which led to the embracing of agile software development approaches (Abrahamsson et al., 2017) and the principles of combining development and operations (DevOps) into a single conceptual model (Ebert et al., 2016), which is well-fitted for the increasingly popular live-service platform development.

A relatively recent entry to the contemporary software development business landscape is machine learning (ML) development, which over the recent years has grown to become a prominent part of ISD practices (Laato et al., 2022a; Niederman, 2021). While ML development is a critically important addition to ISD, it also brings new challenges, such as the following. First, ML systems are fundamentally probabilistic (Bawack & Ahmad, 2021; Akkiraju et al., 2020; Ishikawa & Yoshioka, 2019) as opposed to rule-based deterministic software. Second, ML development can require significant data and computation resources (Laato et al., 2022a), and hence, projects may require, for example, plans for storing, handling, and managing data (Jüngling et al., 2020). Third, ML development is often an experimental process and, therefore, unpredictable (Akkiraju et al., 2020). Fourth, ML tools enable developers to automate processes that could not previously be automated (Laato et al., 2022a). This opens new avenues for both development and the types of systems being developed. In addition to these four challenges, the inherent technical complexity and inscrutability of ML systems, risks related to biased outputs and unintended consequences (Asatiani et al., 2021), and raising ethical concerns (e.g., using data that concerns or targets humans) (Berente et al., 2021; Vasist & Krishnan, 2022) all call for particular attention in ML development in ISD practices.

Against this background, we echo previous work (e.g., Jüngling et al., 2020; Laato et al., 2022b; Mucha et al., 2022) that it is critical to understand what new elements these aspects of ML introduce to existing ISD models, practices, and processes. Furthermore, ML development is often handled by expert roles, such as (1) data scientists, who are experts in analyzing and interpreting complex datasets to extract valuable insights and identify patterns or trends, and in experimenting with techniques such as data visualization to design meaningful ways to make use of data, and (2) ML engineers, who are responsible for building the ML systems using available tools, data, and ecosystems. In this work, we refer to both roles with the umbrella term "ML developer". The addition of new roles to software development teams further calls for inquiry into how contemporary ISD models are fit to guide ML development (Jüngling et al., 2020).

Previous research suggests that as new requirements and technologies emerge in ISD, these changes are first incorporated into existing ISD models at the level of practice (Zhang, 2005). However, the ISD literature has only given limited attention to the incorporation of ML development into conceptual ISD models (Sharp & Babb, 2018; Mucha et al., 2022). In addition, the literature has acknowledged that a disconnect exists between IS researchers and practitioners (Rosemann & Vessey, 2008; Gill & Bhattacherjee, 2009), and this calls for studies that aim to bridge this knowledge gap. As a result, in this study, we focus on ISD practices, which we understand as the activities and processes involved in designing and building information systems (Barki & Hartwick, 2001). To this end, we build on practice theory (Dittrich, 2016; Orlikowski, 2000; Schatzki, 1996) to explore *how ML development is being incorporated into ISD practices*. Practice theory is an appropriate theoretical lens to address this research question since it enables studying the interplay between conceptual models and practices in ISD work (Dittrich, 2016; Päivärinta & Smolander, 2015). With this study, we respond to the calls for research on the idiosyncratic challenges of ML development (Jüngling et al., 2020; Laato et al., 2022b). We situate our study 'on the ground' (Trauth, 2017) and analyze empirical data collected from ML and ISD experts.

We used four ISD models (waterfall, spiral, scrum, and DevOps) as sensitizing concepts (Bowen, 2006) and employed the Gioia method (Gioia et al., 2013) to analyze the empirical data. We interpret the findings through the lens of practice theory (Dittrich, 2016; Päivärinta & Smolander, 2015) to shed light on the connections between ISD models as they are appropriated by developers, ISD practices, and ML development.

The remainder of this paper is structured as follows. A review of ML development literature and practice theory is presented. Then, the research methodology and data analysis techniques are outlined. Next, the findings are presented, followed by a discussion, implications of the findings, and limitations. The paper ends with a conclusion.

2 Research Background

2.1 Machine Learning in ISD

ML is a broad term that denotes various computer-based data mining and interpretation techniques used for uncovering complex patterns, particularly in large and complex sets of data (Mohri et al., 2018), to extract insights for classification, prediction, and decision-making purposes (Chinnamgari, 2019; Cui et al., 2006). ML can be divided into subcategories such as supervised, unsupervised, semi-supervised, transfer, and reinforcement learning (Chinnamgari, 2019), and one IS can make use of one or more of these approaches. Both the ML approach and the problem to be tackled with ML have a significant influence on the development process of the ML system and may bring specific challenges to the development process and system operation (Ishikawa & Yoshioka, 2019). For example, in supervised learning, data annotation is a key phase and may contain subjective interpretation, whereas such a phase does not exist in unsupervised learning.

There are several ISD models used in ISD practice, such as the waterfall (Royce, 1970), spiral (Boehm, 1988), scrum (Benlian, 2022), and DevOps (Ebert et al., 2016) models. The models emphasize different approaches to ISD, depending on the context in which they are intended to be used. Prior IS research has emphasized that since ISD projects are heterogeneous, with varying goals, customer needs, and developer preferences, all these models may have their use purpose (Benlian, 2022; Lyytinen & Rose, 2006). This same principle also applies to ISD projects involving ML, and hence, it can be valuable to obtain perspectives on how various popular ISD models can guide ML development work.

Despite recent studies focusing on various aspects of ISD (e.g., Kautz & Bjerknes, 2020; Maruping & Matook, 2020; O'Connor et al., 2022; Öbrand et al. 2019) or specifically ML in the context of ISD (e.g., Jüngling et al., 2020), research on how ML development has been integrated into ISD models has been scarce (Laato et al., 2022b; Mucha et al., 2022; Sharp & Babb, 2018), and has so far mostly taken place through the research regarding ML-specific development approaches such as MLOps (Mucha et al., 2022; Tamburri, 2020). To investigate the integration of ML into ISD models, we employ practice theory as the conceptual backbone of the study.

2.2 Practice Theory

Practice theory offers a theoretical lens for understanding the interplay between ISD models and practices, which, in turn, acts as a basis for investigating the incorporation of ML development into ISD practices. Practice theory is a social scientific theory family that studies social practices as the primary elements of social reality (Cecez-Kecmanovic et al., 2014). Practice theory includes diverse theoretical starting points stemming from foundational concepts such as the habitus (Bourdieu, 1977), structuration (Giddens, 1984), integrated practice (Schatzki, 1996), and epistemic practices (Knorr Cetina, 2005). Within the IS field, practice theory has been used to investigate, for example, technology use in organizations (Orlikowski, 2000), the sociomateriality of IS (Cecez-Kecmanovic et al., 2014), institutional logics in the adoption of healthcare IS (Hansen & Baroody, 2020), and IT-based regulation systems (de Vaujany et al., 2018).

A common thread in these practice-theoretical perspectives is that people construct and continuously recreate their lived social worlds through practices (Oomen et al., 2021). Many practice-theoretical accounts also emphasize the role of materiality and artifacts in influencing practices (de Vaujany et al., 2018; Orlikowski, 2000). Practices are recurrent, organized, and situated activities conducted by human agents (Orlikowski, 2002; Päivärinta & Smolander, 2015). In the software development context, a practice can be understood as *"a commonly agreed upon way of acting that is acknowledged by the team"*

(Dittrich, 2016). Software development practices generally fall under categories such as analysis, design, implementation, and quality management (Päivärinta & Smolander, 2015).

Practice theory is suitable for analyzing the incorporation of ML into ISD models, as it offers a frame through which to observe the interplay of practice and conceptual models and, accordingly, offers a helpful way for problematizing the connection between models and practices in ISD work. This is warranted as studies show that methods and models are creatively appropriated rather than simply adopted. Indeed, according to some studies, as few as six percent of developers follow any particular method rigorously (Dittrich, 2016). In this study, we used practice theory as a sensitizing theoretical lens to direct our analysis and make sense of our inductive findings. Hence, practice theory serves as a basis for supporting and structuring the derived knowledge.

Table 1 summarizes the practice theory perspective on ISD practices and models. From a practice perspective, ISD models are practice patterns (i.e., sets of tool-supported understandings (notations, concepts) and rules (processes, task descriptions)) (Dittrich, 2016). These practice patterns tie knowledge and action together (Päivärinta & Smolander, 2015), and they need to be incorporated into existing practices and adapted to the situations of specific development projects (Dittrich, 2016).

Table 1. ISD Practices and Models from a Practice Theory Perspective

Concept	Explanation	Examples
ISD practices	Recurrent, organized, and situated activities (Orlikowski, 2002); commonly agreed ways of acting acknowledged by the team (Dittrich, 2016)	Acceptance testing, code review, pair programming
ISD models	Practice patterns: sets of tool-supported understandings (notations, concepts) and rules (processes, task descriptions) (Dittrich, 2016)	Spiral model, Scrum, DevOps
ISD models in use	Adopted and adapted practice patterns (Dittrich, 2016): idiosyncratic ways of adapting and mixing ISD models	Combining practices from Scrum and DevOps by automating testing within the Scrum process

In particular, the advantage of a practice theory lens is that it sensitizes us to three key facets at the intersection of ISD models, practices, and ML development. First, new technologies, such as ML, necessitate the continuous evolution of both ISD models and practices (Dittrich, 2016). Second, ISD models acquire meaning and become shared framing devices as they are adapted to software and ML development practices. Third, ISD models in practical use can be seen as adapted and adopted practice patterns (Dittrich, 2016) (i.e., understandings and rule sets that are localized to given contexts).

Figure 1 illustrates the incorporation of ML development into ISD practices from a practice-theoretical perspective. ISD practices are guided by models as practice patterns and also models in use that are adapted and used in particular projects. ML development eventually influences ISD models, but this influence is likely to take place largely through concrete practices that need to be reshaped due to the novel characteristics of ML, such as the reliance on data, the probabilistic nature of the systems, and the three characteristics of ML systems mentioned by Berente et al. (2021): autonomy, learning, and inscrutability.

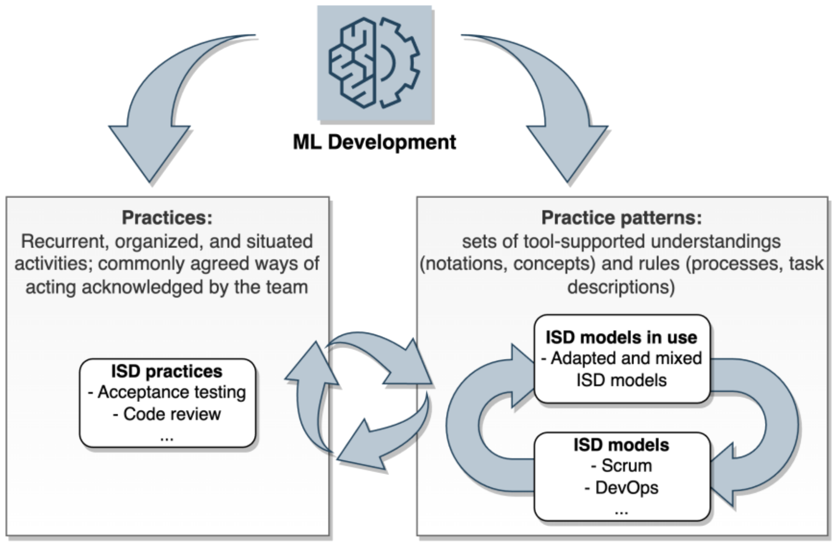

Figure 1. ML Development, ISD Practices, and ISD Models as Viewed through Practice Theory

3 Methodology

3.1 Interview Design and Process

To achieve the aim of this study, we constructed an interview guide following the instructions from Gioia et al. (2013) and Coombes et al. (2009). To sensitize participants to the research topic (Bowen, 2006), we selected four established ISD models as starting points to structure the discussion. These were waterfall, spiral, scrum, and DevOps. We used these models to ground our data collection into shared visual landmarks and, consequently, to prompt the experts to discuss the research topic from multiple perspectives. Next, we elaborate on these models and why we selected them to structure the interviews.

We began the interviews by asking participants about their background and current work in relation to AI and introducing them to the topic. We then proceeded to discuss ML system development from the perspective of the four selected archetypal ISD models: Waterfall, Spiral, Scrum, and DevOps.

The first ISD model presented to the interviewees was the Waterfall. While the original waterfall model presented by Royce (1970) is largely considered obsolete today, its contemporary versions (e.g., Balaji & Murugaiyan, 2012) can still be helpful for designers, developers, and managers today. This model was selected for its popularity and since it is often used as a comparison case for demonstrating the effectiveness of agile approaches.

The second ISD model presented to the informants was the Spiral model (Boehm, 1988), which conceptualizes development in four iterative phases: (1) determining and clarifying objectives; (2) identification of risks and resolving them; (3) developing, coding, and testing the solutions; and (4) planning the next iteration. With each new iteration/cycle in the spiral, the cumulative costs of the project increase. This model was selected for its historical significance.

The third ISD model presented was Scrum, one of the most popular contemporary models used in ISD practice (Benlian, 2022). Similarly, to the spiral model, Scrum is also cyclical. It operates in “sprints,” which are usually two- or four-week cycles, during which the ISD team develops workable features of the intended software (Srivastava et al., 2017). Scrum was selected as it is a hugely popular agile ISD approach and provides a clear ISD model visualizing the development approach.

The fourth model presented to the interviewees was DevOps, which, as the name implies, integrates the previously distinct development and operation stages of software development (Ebert et al., 2016). DevOps aims to deliver software products to customers as quickly as possible, avoid manual labor, and lead to efficient practices (Ebert et al., 2016; Virmani, 2015). DevOps builds on the concept of continuous delivery, where building, testing, quality assurance, verification, and development are largely automated (Virmani, 2015). The rapid deployment process of DevOps makes it unsuitable for critical systems (Ebert et al., 2016). DevOps was selected for its popularity and prominence, and since there are also ML-specific versions of it available, typically discussed under the umbrella of MLOps.

Using the four above-described models as a shared point of departure, we discussed from the vantage point of these models how the participants see ML development in their work, in their company, and in general. More specifically, for each model, we asked participants about (1) their experience with the model; (2) their evaluation of the positive and negative aspects of the model; and (3) the ability of the model to meaningfully describe ML model development work and the characteristics of ML systems. Furthermore, we asked follow-up questions and clarifying questions regarding, for example, whether the participants were already following one of these models in ML system development work in their companies. We discussed the different developer roles involved in ML system creation and how the developers, management, and customers communicate about the project goals, needs, and implementation. Finally, we asked participants to summarize which ISD models they had used in their ML projects, and which model they thought would be optimal for describing ISD projects involving ML.

3.2 Data Collection

The interviewees were recruited using purposeful sampling (Seidman, 2019; Noy, 2008) which enabled us to reach out to respected, knowledgeable, and well-spoken individuals, and to access interviewees through the contact information provided by other interviewees (Noy, 2008). The authors listed potential names and began contacting people for interviews in the spring of 2021. We did not recruit all participants at once but got further names and suggestions from our peers and the first interview participants as we progressed. We looked for interview saturation by observing whether the new participants brought significant new information and details to the topic of this study. After 19 interviews the authors agreed that we had reached sufficient saturation since no new significant information had emerged in the last five interviews.

Due to the Covid-19 pandemic, interviews were conducted remotely via Zoom, and their duration varied from 41 minutes to 74 minutes (see Table 2). Each interview was recorded, transcribed, and annotated. The interviewees had varying roles and expertise, but all were knowledgeable about either ISD models, ML development, or both, as displayed in Table 2. Overall, the participants held a wide variety of positions within the IT industry and academia and could hence be expected to provide a large breadth of expert information regarding the research topic.

Table 2. The Interview Participants, Work Experience, Job Title, Characterization of the Company*

ID	Job title	Work experience (years)	Industry sector	Interview duration
1	Data & security specialist	5	Medium-sized publicly traded company	53 min
2	Chief technology officer	11	Medium-sized unlisted company	59 min
3	Professor of data science	16	Large public university	41 min
4	Competence lead	10	Medium-sized publicly traded company	59 min
5	Senior data scientist	20	Large publicly traded company	63 min
6	Senior software developer	13	Large publicly traded company	59 min
7	Data science research fellow	15	Large public university	73 min
8	Director of a software project	20	Large publicly traded company	57 min
9	Professor of software engineering	20	Large public university	62 min
10	Systems architect	6	Medium-sized unlisted company	32 min
11	Insurance statistician/actuary	7	Large publicly traded company	74 min
12	Professor of software engineering	26	Large public university	74 min
13	Professor of data science	29	Large public university	61 min
14	Director of a software project	22	Medium-sized unlisted company	69 min
15	Senior software developer	26	Large publicly traded company	48 min
16	AI consultant (developer)	11	Medium-sized unlisted company	68 min

17	Data project lead	12	Medium-sized unlisted company	55 min
18	Director of an AI startup	15	Small unlisted company	57 min
19	Professor of software engineering	21	Large public university	51 min
*Note: large is defined as 500+ employees, medium-sized as between 50 and 500, and small as below 50.				

3.3 Data Analysis

Analysis of the data was guided by the Gioia method, which is an inductive approach for rigorously processing qualitative data towards a coherent and representative knowledge structure (Gioia et al., 2013). The Gioia method has gained popularity over recent years in organization studies and IS (Gioia et al., 2022; Wiesche et al., 2017; Mäntymäki et al., 2020). The method builds on top of the concept of open coding (Strauss & Corbin, 1990), where 1st level concepts are identified from the data through a process like open coding from grounded theory, combined to form 2nd order themes and ultimately 3rd order theory-guided aggregate dimensions. While the Gioia method has been criticized for trading interpretive rigor for procedural rigor (Mess-Buss et al., 2022), it brings transparency and structure into qualitative analyses (Gioia et al., 2013). These characteristics are highly desirable since our research area is muddled with ill-defined concepts and overlapping terminology. Hence, being able to transparently trace the derived themes to participant quotes can help readers understand our reasoning and interpretation.

We began the analysis following the Gioia method by coding the data inductively for concepts related to developers' and development teams' practices and models in implementing ML systems (see Table 3). In our interpretation of the data, we focused on the key characteristics, concepts, aspects, and dimensions that the experts discussed. After discovering basic concepts through open coding, similar concepts can be grouped together to form 2nd order themes (Gioia et al., 2013). In the context of this study, this process was iterative, and the authors returned to the data multiple times to affirm that the first-order concepts were interpreted correctly and to combine similar concepts together in a conceptually sound fashion. Continuing with the Gioia method, we finally collated the codes into broader themes to uncover key dimensions of focus in integrating ML system development into ISD practices. The authors refined and iterated the final data structure and themes multiple times.

We applied practice theory (Dittrich, 2016) in the final step of our analysis to guide the formation of the theory-guided dimensions (Gioia et al., 2013). After finalizing the first full iteration of the analysis, we returned to reflect on the resulting data structure from the vantage point of practice theory, aiming to enrich the data-driven findings and elevate the level of abstraction in our utilized concepts. We started by comparing the different accounts of existing practices in software development as they appeared in our interviews, and whether they would still match our existing data structure. For those practices that did not match, we explored whether they were in the process of becoming practices in the future. We refined the data structure at this stage according to our evolving understanding of the topic. After the three researchers participating in the analysis were happy with the resulting interpretation of the data, we continued to examine what practice-theoretical implications the aggregate dimensions would have on software development teams working with ML.

4 Findings

As described in the analysis method, we first extracted first-order concepts from the data. These are displayed in Table 3. Following the Gioia method, we analyzed the data to explore how focus manifests itself in the categories identified in the previous stage (see Figure 2). Through the process of distillation, we moved from the first-order codes to second-order categories. Drawing from practice theory, we then connected the discovered seven second-order themes to theory-guided aggregate dimensions (Gioia et al., 2013). In this section, an in-depth analysis of each of these aggregated themes is presented.

Table 3. The First-Order Codes Identified in this Analysis and Example Quotes FROM THE Informants

Concept	Example quotes
ML becoming an important skill also for non-specialized developer roles	*"I would divide it so that [what differentiates] data scientists [is that they] are able to do custom-ML model development (...) and then an engineer that is able to do basic ML engineering can be called a full stack data scientist *laughs*"* (P8). *"We've had employee trainings regularly from all these techniques and ML included even if they are not straight away used in any of our software"* (P14).
ML engineers working alongside the rest of the development team	*"There is some differentiation, for example in the project I'm now involved with we have a team of two data scientists, and we do a little bit of coding as well and communication with the others too, but the ML parts are our main responsibility"* (P5). *"We work closely with the IT team (...), we have clear systems and pipelines where we push our work - and we have monitoring tools"* (P11).
ML development should be governable and predictable	*"There is a bit of this that [in ML systems]" it's not communicated what's happening under the hood. Using ML can also be just pushing a button without understanding what is being calculated or without being able to fix the system if something goes wrong. You can't also improve system efficiency…"* (P3). *"It has not been many years since someone got the idea that we should apply the practices of software engineering to ML development (...) to understand and follow - and perhaps predict how the development is going"* (P17). *"You need something like this when AI development moves from that kind of artisan-development towards a defined software development process"* (P8). *"The closer we get to a human brain [with AI] the bigger the explainability issue is. (...) so you can put boundaries around the system and then it's easier to control"* (P13).
The demand for management to understand ML technology and its possibility	*"Well, if they [decision makers] don't understand the [AI] systems then it's their loss. The technology has enormous potential and not using it means you will be left behind"* (P13). *"I would say [choosing to apply ML for a business case] is mostly developer-driven these days. The developers select the tools. It would be good if the management understood the IT systems of course"* (P10).
Unpredictability of ML projects	*"Customers are always thinking about their risk and [in ML engineering] the risk mostly comes from these data aspects - and that uncertainty coming from data puts its mark on all the development"* (P5). *"With public side clients they are often based on having some POC [proof of concept] product. (...) We've won the bidding of projects where we have not created the POC and we have lost some where we have. It's supposed to increase the predictability of the outcome of sorts but that's not always the case"* (P16).
DevOps and MLOps are becoming popular conceptual models for development	*"[With DevOps]" you have the advantage that you notice mistakes earlier and fixing them will then be cheaper (...) and quality monitoring is ingrained, which is largely dismissed in the other approaches"* (P1). *"The development part of DevOps can be agile and follow Scrum. When I look at, at least what I've seen, it's [the development work in companies is] DevOps and Scrum mostly"* (P12).
Agile development practices have become the industry standard	*"I know it's mostly all agile these days, but I like the clear checklist kind of visualisation of waterfall"* (P7). *"In any case, from the models so far, agile is the most sensible. (...) The idea that you can react fast is sensible on all levels, and some firms have developed scaling models on top, to be agile on several levels"* (P12). *"Personally, I always prefer that the customer is involved. The whole point of scrum is that the product owner is present. (...) We as a company push towards working with scrum quite heavily"* (P6).
There is no ISD template that fits all projects	*"It is not much use to strictly apply a textbook [ISD] process if the customer does not have the capability to work along with it - we need the ability to pick [from a textbook ISD process] the most useful parts that benefit the customer the most"* (P8). *"I've always talked about [ISD] models and you just say image- and I actually think it's better cause these are just visualisations of the development processes, and no one follows these*

	strictly” (P17). *“As long as you have a human involved, no one is going to repeat [the ISD] process one to one and there are always the variations in there”* (P9).
Product customers may have specific needs regarding the ISD processes	*“In the case of [redacted] we studied how they adopted DevOps, but they could not follow it everywhere and the link from development to production was oftentimes slow because their customers weren’t ready for that”* (P12). *“Easily when you’re doing some bigger project the customer wants to see a waterfall model because it’s rigid, simple, predictable - they understand what’s going to be done (...) then internally we may just follow scrum anyways”* (P16). *“Some customers want to just completely outsource the project and get a working product after, say a year”* (P6). *“The works I’ve been involved have been rather small, there having daily scrum meetings, all other kinds of meetings and all that would just feel like a waste of time”* (P2).
ML system training can have significant data requirements	*“Especially from the viewpoint of GDPR it is important to consider where you store the data”* (P1). *“When our understanding of the data improves, we can like notice that the data does not meet the requirements we originally had and then we need to take steps backwards”* (P3). “We figure out what data we want and what data we need, but oftentimes after that we need to go out and about to gather that data” (P17).
ML system training can have significant computation requirements	*“What has changed is of course deep learning that came about ten years ago and all this GPU-based computation that is needed, and while it is a breakthrough in certain tasks it also increased the need for computation quite a bit”* (P3). *” Google has their own hardware, and this is a good trend as the hyperscalers can develop specialized hardware as they have so much of the mass there and that brings them more of the scaling advantage”* (P16).

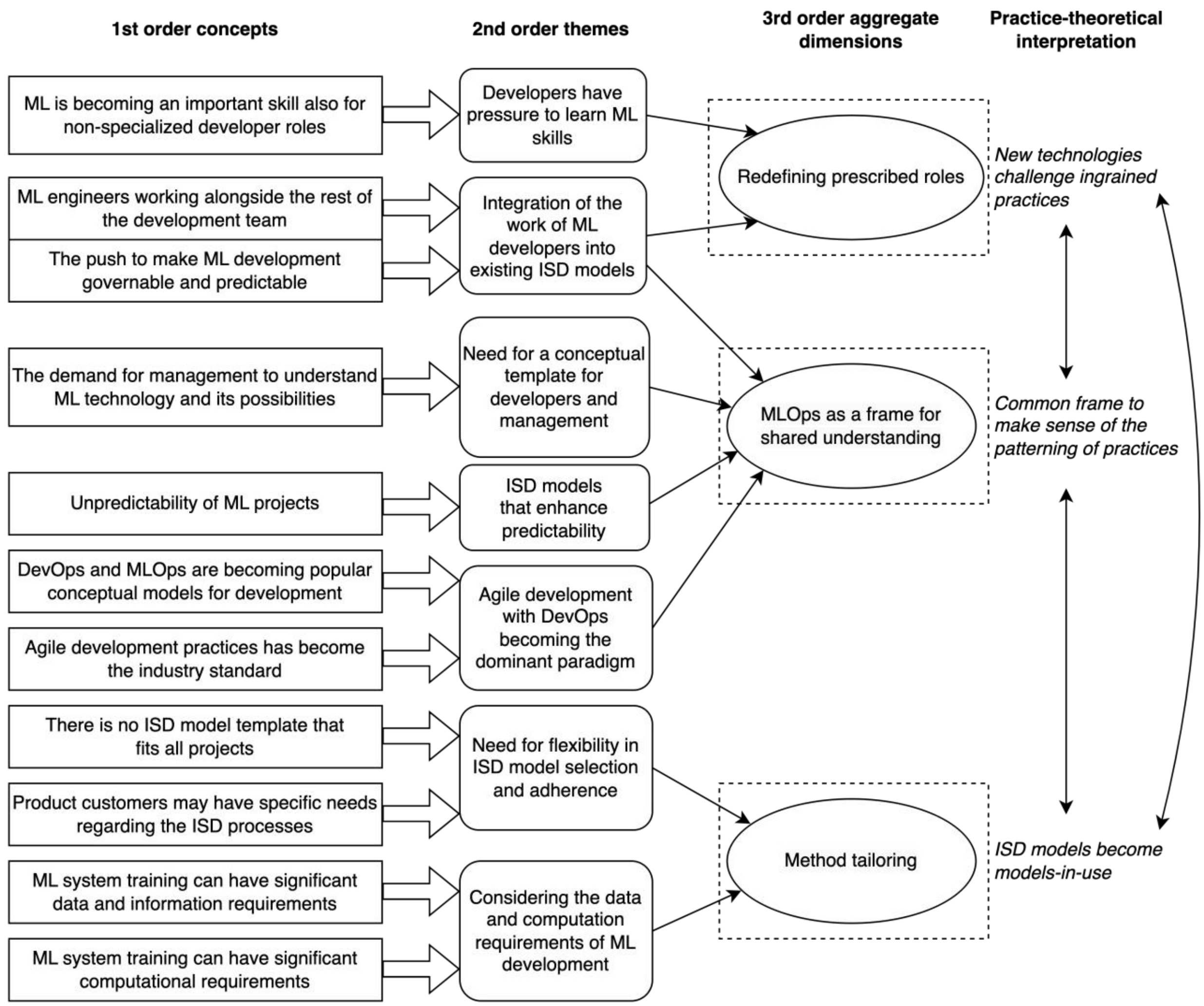

Figure 2. The Stages of the Analysis Process and the Concepts, Themes, and Aggregate Dimensions

4.1 Redefining the Prescribed Roles and Responsibilities in ISD

The interviewees involved in technically developing ML systems (e.g., P5, P6, P11, and P16) reported that they typically work alongside other engineers and follow the ISD model of the entire development team. For example, with Scrum, their sprint tasks could be quite different, but they could still follow Scrum without any issues. A data scientist could also get support for some aspects of their work from the rest of the development team. Implicit in all the interviewees' remarks was that ML system developer was seen as a specialized role, distinct from other software developers. At the same time, however, the results implied that the roles in development teams could be fluid depending on the task at hand and team composition. Even in the primary role of a data scientist, the activities could vary considerably between projects, as the following quotation from the interviewed senior data scientist illustrates:

> *"I've been doing everything [from data extraction to model training and production]. But sometimes we have distinct roles. For example, we had a case where I was together with another data scientist, I was the person who pumped information from the clients and took care of the background stuff while the other guy worked more on data processing and that end."* (P5)

This comment, together with the rest of the interview material, suggested that data scientists and ML engineers also handle some tasks that may be associated with "regular developers". Similarly, however, developers faced pressure to do some tasks otherwise associated with data science. Overall, this speaks of a redefinition of prescribed developer roles, which fits well with the practice-theoretical insight of technologies continuously challenging and recasting ingrained development practices. In the case of ML, it seems that the characteristics of the technological artifacts demand novel arrangements of roles in practices to ensure efficient development. Despite drawing a parallel between specialized development roles and the role of a data scientist, some interviewees brought up AI-specific ISD models and data mining process models such as CRISP-DM and the Essence theory of software engineering. An interviewed professor of software engineering (P12) was particularly critical about the idea of there ever being a model that would fit all or even most software development cases and even challenged the idea of advocating for any specific ISD model: *"I've tried to understand for years why we are still going to this direction where someone presents yet another [ISD] model and again we hear cheering for it from the advocates of the new model. There are always unique structures and organizations, and there will always be the need to fit and apply [ISD] models into that context"* (P12).

Specific developer roles were not well-defined, and developers constantly supported neighboring development work depending on the project needs, resulting to some degree in convergence between the distinct roles of data scientists, ML engineers, and software engineers. For example, the interviewed director of software development (P14) stated that as APIs and practices of creating certain standard ML systems are increasing in maturity, they are simultaneously becoming a part of the toolkit of regular software developers. Similarly, data scientists are using the same tools as software developers. Furthermore, the integration of ML engineering into software development is also piling pressure on software developers to personally develop their skills to learn how to manage large data sets and train basic ML models. The broader theme arising from the interviews regarding the pre-subscribed roles in software development is that they appear to be in flux. The proliferation of data-intensive development is pressuring upper management to acquire an understanding of how to best make use of new opportunities, as well as how the development of ML systems and their ISD model should be managed. Simultaneously data scientists have been forced to increase the level of maturity of their activities, as challenges including AI governance and auditing seem pertinent in most AI systems. In this complex landscape that is in transition, new software roles are likely to emerge, formed, and shaped by new emerging technologies such as ML, digital platforms, and the overall surrounding digital infrastructure. P16, however, had another take, and he believed that instead of moving towards more specialized roles, the roles are in fact converging into one as he states: *"I do honestly believe that in the future we do not have some data developers and software developers, but we only have developers and then we have tools that the developers use"* (P16).

4.2 MLOps as a Frame for Creating a Shared Understanding between Developers, Management, and Clients

The second aggregate dimension is presented under four sub-headings due to its complexity and the richness of informant discussions compared to the other dimensions.

4.2.1 Incorporating ML Engineering into Existing ISD Models

In the interviews, we asked the participants about four archetypal ISD models and how feasible they thought the models were in describing actual ML development work and project work they had been part of. Worth noting is that interviewees did not consider any of the archetypal ISD models to be superior to the others. Individual interviewees also expressed views in support of the spiral model and the waterfall model that were created as early as the 1970s and 1980s (Boehm et al., 1988). The following two quotes illustrate how even these two older models received support from experts:

> *"Of course, I should not probably say this, but the waterfall model is the most readable. You have to remember I've not been involved in very big software projects or the industry, so from that point of view I don't have knowledge of how to get a project of 100 people to actually work. (–) But I would say that instead of slavishly following any of the presented models, I prefer knowing the steps involved in the development process."* (P7)

> *"I like the simple approach and the engineer-like clarity of the spiral model. In a certain way the cyclical work is presented well here, as if peeling an onion one layer at a time, or rather growing an onion one layer at a time. It is a good description of ML model development."* (P5)

In addition to each of the archetypal ISD models receiving support from some experts, there appeared to be a consensus that ISD projects involving ML engineering were unique and highly complex, and for this reason, forcing the projects to operate under a rigid model was not a good strategy. In fact, the project needs often forced the experts to heavily adapt their working practices to fit the project needs. The following quote illustrates how all the presented ISD models could have value depending on the context:

> *"None of these is superior and I'm sure the waterfall is also a good model if you have a very accurate picture about what you are doing, but of course I have never really felt like I would have, unless the problem is really simple."* (P3)

4.2.2 The Need for a Shared Conceptual Frame between Developers, Customers, and Management

One of the goals of ISD projects involving ML was to *"demystify AI and make it an ordinary, boring tool"* as described by P18, who was a director of a successful and growing AI-based startup. According to them, ML had long been seen as an academic curiosity or something in the realm of large corporations that was not practical to utilize in smaller scale consulting processes. However, this was changing rapidly with the increased availability of computing power for training models, existing APIs and libraries that support development work (e.g., TensorFlow and Keras, PyTorch), and cloud services such as AWS, Azure, and Google Cloud offering tutorials and guidelines on how to implement ML systems. While all these developments contributed to an increase in applying ML engineering to solve problems in software consulting projects, it was not clear and self-evident that the customers, or even the project management, fully understood the limitations and capabilities of ML engineering.

From a practice theory standpoint, turning ML into a "boring tool" transforms the presently somewhat idiosyncratic ML development practices into patterned practices supported by established tools and rulesets. The participants expressed the need to communicate issues such as the inscrutability of ML models, risks of bias in the training and testing data, governance issues, and uncertainties to customers and the management. There was an emphasis that data science is *"still much of experimenting, feature engineering, trying things out, what works and what not"* (P7), and these aspects had to be communicated to customers and the management. Here DevOps and particularly the ML-specific version of the ISD called MLOps had become important frames to support and scaffold the building of a shared understanding of the development work between developers, customers, and management. While this approach had obvious advantages, it also required the perhaps non-technical stakeholders (customers and management) to really get into ML engineering on a technical level and understand what was going on.

4.2.3 ISD Models to Improve Predictability of ML Development Processes

Connected to the previous sub-theme and the idea of *"demystifying AI,"* the informants sought to improve the predictability of ML development processes, which would further establish the practice patterns around ML development. When budgeting for ISD projects involving ML it was difficult to put a price tag on the work of data scientists, since there were various issues that could significantly alter the success of the project, including but not limited to (1) the quality and availability of data; (2) the availability of expert workforce to handle the models; (3) external pressure for transparency, explainability, reliability, and understandability; (4) legal compliance; and (5) long-term maintenance of the system. However, according to, for example, P1 and P5, the biggest risk comes from the data. These issues were highlighted particularly when creating something new that had not been done before since software consultancies had no data on previous similar projects and their estimated costs. The lack of predictability was seen as a clear deterrent from adopting ML engineering into practice, but at the same time, the promises of the solutions sometimes outweighed these risks. Furthermore, some developers chose to apply ML to solve some engineering issues on their own, without any strategic decisions to use ML involved.

P16 described their work in the software consulting business as being more volatile when considering bidding for projects that involved ML. There was a structure where companies would create proof-of-concept products for customers, and the customers would then ask software consultancies to make an offer of what it would cost for them to create this product. As sometimes the consultancies had no prior experience of such projects, it was difficult to evaluate the costs. P16 explained as follows:

> *"We've won the bidding of projects where we have not created the proof-of-concept, and we have lost some where we have. It's supposed to increase the predictability of process outcome, (sort of), but that's not always the case."* (P16)

In addition to consultancies doing more ML projects and obtaining more knowledge on the potential costs of them, another important aspect in improving the predictability of ISD involving ML was to use an existing framework for development, such as MLOps. As P11 explained: *"We have very strict processes, and I have my slot where I just put the model running. It's all very clearly defined to minimize the risks."* Thus, by adopting strict clearly defined processes companies can reduce some of the risks involved in ML engineering, improving the predictability of the development work.

4.2.4 Agile Development with MLOps/DevOps as the Dominant Paradigm

According to the participants, the principles of continuous integration and delivery in DevOps were seen to align with ML model engineering, as also reported in extant literature (e.g., Karamitsos et al., 2020). Most of the interviewees, particularly those working in the industry stated that almost all their work these days follows DevOps. With regards to DevOps, participants sometimes discussed the concept of MLOps interchangeably. Interviewees appreciated how the operation side was considered in DevOps/MLOps and considered it essential in ML model development. For example, *"from the viewpoint of my work, in the real world that constantly changes, the model here that shows a DevOps- approach would be best. (–) When the system is in production we can follow it, and from there we can get a trigger that we now need to react and do something"* (P11).

When synthesizing the arguments across all the interviews, MLOps emerged as a promising ISD frame that could work simultaneously as the backbone of development practices and as a conceptualization for the upper management on the state of the IS (i.e., a language to communicate development practices to management). The interviewed data project lead (P17) spoke about the importance of integrating governance and automated testing into MLOps pipelines, hence automating the roles of external auditors and managers to an extent. In this way, MLOps has the potential to bridge the chasm between management and software development teams, as also discussed in section 4.2.1. Overall, these factors have contributed to the rising popularity of MLOps for describing ML system engineering, and according to the interviewed experts, it seems to be becoming the dominant paradigm despite there still being room for all kinds of other ISD conceptualizations depending on the specific project needs.

4.3 Method Tailoring when Fitting ML into ISD

The third theme can be summarized as a shift from adherence-based ISD approaches towards method tailoring, a phenomenon already identified in prior work (Campanelli & Parreiras, 2015; Dingsøyr et al., 2019), but which strongly emerged in our empirical data within the context of the integration of ML into ISD. This theme speaks to the continuous need for adapting and adopting practice patterns in specific projects (i.e., as models-in-use), which creates tension with the shared framing discussed under the previous theme. One of the challenges that software development teams encounter is the strict requirement to adhere to the method approach as prescribed by the method creator without considering the socially embedded contextual nature of software development practices. However, the interview data suggests that development teams are quite liberal in applying the theoretical models in practice, particularly when dealing with ML engineering, meaning they adapt each of the models for the specific use case and context. For example, a senior software developer working for a large software consulting company (P6) reported that they *"always follow some iterative development approach, preferably Scrum",* but that their clients may not have a sufficient level of maturity nor willingness to participate in the Scrum process. Hence the team adjusted their ISD approach to accommodate the customers' needs.

From the perspective of the interviewees, ML engineering is an endeavor consisting of various activities, the most important of which are data collection and data engineering, and model development and testing. Particularly among larger companies, the work of ML engineers also consisted of providing a pipeline where the latest champion model would be provided for use by the rest of the system development team. The interviewed data project lead (P17) illustrates this with an example from their company where two data scientists work relatively independently from the rest of the team during Scrum sprints but would always seek to integrate their work into the rest of the product at the end of the sprint. In doing so, the data scientists were subjected to the same ISD procedures as the rest of the team. The ISD practices are hence being guided by specific development tools and aims, such as the construction of a CI/CD pipeline. The other side of the coin was the customer and their needs, which varied substantially, as P5 explains: *"And companies I've seen who are not working specifically in a data-related field, they can be quite stiff, surprisingly stiff and have quite poorly so far made use of the data and the opportunities they have"* (P5). Thus, particularly in software consultation projects the maturity of both the developer side (e.g., the software consultancy) and the customer must be considered, and the ISD methods tailored accordingly.

Evident from the interviews, and from recent developments with large language models such as OpenAI's GPT-series, training of some ML models requires extensive computational resources. Obtaining computational capacity is possible through hyperscaler cloud services such as AWS, Google Cloud, and Microsoft Azure, but training large models is still highly expensive, putting some limitations on the kind of ML systems that can be developed. However, for many projects, there was no need to develop models from scratch, or even develop models at all, as there were multiple parties offering APIs to access pre-trained models as P16 explains: *"We already see this in that cloud service providers are, for example, offering pre-trained models that can be used through some interface"* (P16).

Overall, the findings suggest that ML engineers and data scientists are already working in development teams following established ISD models. Furthermore, since software engineers are feeling the pressure to learn the basics of ML model development, ML model development has already been integrated into existing ISD models at the level of practice. With the arrival of MLOps, construction of CI/CD pipelines, and use of version control tools such as Delta Lake, it appears that there is a shift in software development that goes beyond the addition of ML engineering into conceptual ISD models, as ML engineering is also shaping the core foundations of software development.

5 Discussion

5.1 Theoretical Contributions

Overall, our findings elucidate the various ways in which ML development has been incorporated into ISD practices. Models such as MLOps, which have been designed to streamline and guide ML development for industrial use (Tamburri, 2020), were popular. From a practice theory viewpoint, these technology-grounded models directing development (e.g., MLOps) provide a shared practice pattern (Dittrich, 2016) that scaffolds a mutual understanding between the development team and non-technical stakeholders. Our findings also highlight the multitude of other ISD models besides MLops used in guiding ML development. This aligns with previous research suggesting that there is no silver bullet ISD conceptual model that all development projects should follow (Baseer et al., 2015; Giardino et al., 2015). This finding also supports the practice-theoretical insight that models, as practice patterns, are adopted and adapted in specific project contexts, and they are useful insofar as they help organize the ISD practices and tie knowledge and action together in that context (Dittrich, 2016; Päivärinta & Smolander, 2015).

Our findings underscore that the incorporation of ML development into ISD practices brings challenges pertaining to system governance and auditing and simultaneously requires various new developer skills, both technical and non-technical. The rise of ML development as part of ISD muddles the established prescribed roles in software development teams, and new roles might be created (cf., Birkstedt et al., 2023). Reflecting on Kim et al. (2016), who showed that even within prescribed roles such as that of a data scientist, there is variance in working style, our findings suggest that the variance in the exact roles that developers acquire in teams can be even more manifold now and in the near future. From a practice theory perspective, we highlight that ML development is merely one of several trends (albeit an important one) impacting the shaping of practice patterns and developer roles and that new AI-based tools such as large language models, new analytics methods, and other advances in hardware, software, and management jointly contribute to shaping and forming ISD practices.

Figure 3 situates the aggregate dimensions described in the findings section into the practice theory framework introduced in section 2.2. Hence, the figure aims to provide a theoretical elaboration of the findings by placing them into a systemic understanding of how a technology-driven change (in this case, ML development) initiates new practices that become established into practice patterns through continuous proactive adaptation (method tailoring) and teams' sense-making.

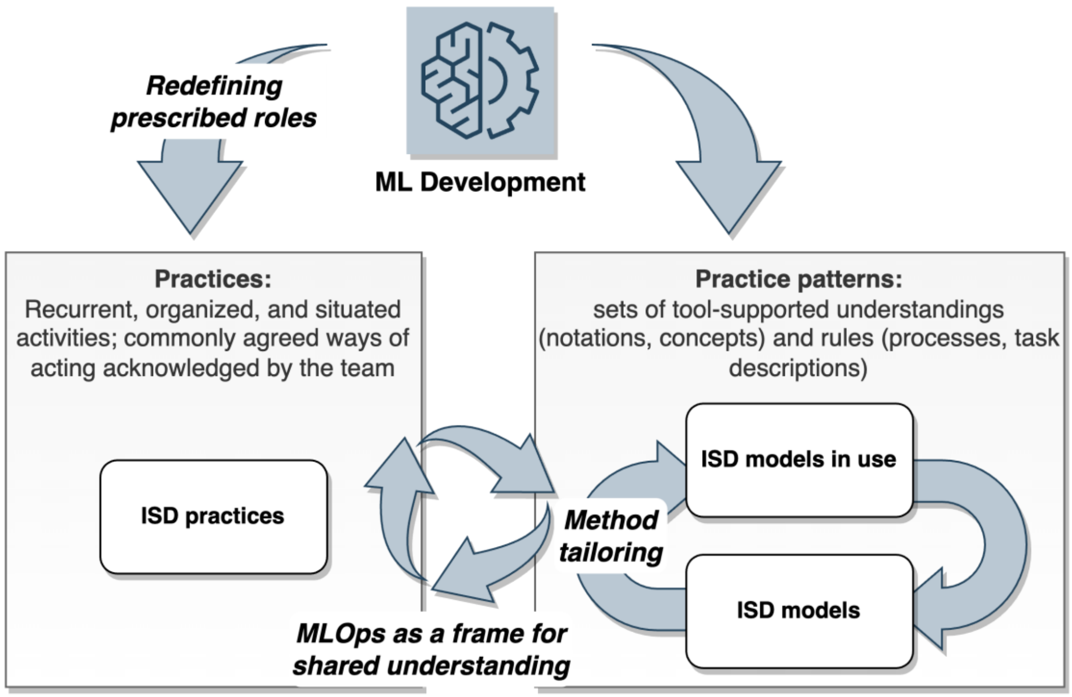

Figure 3. Positioning the Aggregate Dimensions (in italics) in the Practice Theory Framework

As indicated by the top part of Figure 3, we view the arrival of ML development into ISD practices as the initial event that leads to cascading changes in both practices (through redefined developer roles) and practice patterns. Practices are teams' commonly agreed ways of operating. They involve actions, social norms, and values. Under practices, we also have ISD practices, such as agile software development (Abrahamsson et al., 2017). Practice patterns, by contrast, are a combination of practices, organized into a pattern. Here we have conceptual ISD models and the ISD models that are used by the team. Between practices and practice patterns, there is a continuous reciprocal process of the entrenchment of practices into patterns and patterns, in turn, influencing and giving meaning to practices. This process is indicated by the cyclical arrows between practices and practice patterns in Figure 3.

According to our analysis, MLOps offers a frame to make sense of the patterning of practices ("MLOps as a frame for shared understanding"), giving a shared language to the various practices required by ML development (the technological initiator of the changes). In order to be practically applicable, models as practice patterns need to be fitted to specific projects ("Method tailoring"), becoming models-in-use and, thus, hybrid practice patterns supported by diverse tools rather than the clear-cut patterns provided by the archetypal ISD models. The bidirectional arrows in Figure 3 illustrate the dynamic interaction between models providing outlines for models-in-use and models-in-use, in turn, potentially providing feedback to archetypal ISD models.

While we do not claim to theorize how technologically driven changes renew ISD models, we can provide some initial insights to be elaborated in subsequent studies. First, at least in the case of ML development, the technological change seems to be integrated into practices first through new roles, and the more patterned changes to models come afterward through sense-making and entrenchment of practices into patterns. Second, as illustrated in Figure 3, there seem to be dual continuous cycles at play: between practices and practice-patterns (establishing and elaborating patterns as shared frames for understanding) and between ISD models and ISD models in use (tailoring and adapting models in particular contexts). Individually, these insights are familiar from the ISD literature. However, when investigated systemically to understand how technologies drive changes in ISD, they provide clues to the complexities of this process. Rather than technologies, such as ML, merely being integrated into ISD models, there are continuous sense-making and adaptation cycles at play as renewed practices contribute to renewed ISD models.

5.2 Implications for Practice

Our study offers three actionable practical insights. First, it provides a step towards a more systematic understanding of ISD methods and ML development. In our data, the most common ways for ML development to have been integrated into ISD were following the latest industry-led models (MLOps) or simply following the practices that already existed within development teams prior to ML development and attaching data scientists to operate along with the rest of the team. An implication for project managers guiding the integration of ML development into their existing ISD processes is the need to assess the readiness of their project teams and wider organization, and to tailor the methods and approach accordingly, a finding supporting evidence from another recent study (Dennehy et al., 2022).

Second, despite the hype surrounding the potential benefits of ML development, the study findings highlight the continuous and pragmatic patterning of ISD practices into adaptable models that key ISD decision-makers should consider. Since data science is *"still very much experimentation, trial and error"* (P7) decision-makers need to account for the uncertainties surrounding ML engineering. Supporting this view were also the aims of industry stakeholders to move away from the *"mysterious and magical"* toward the mundane and rigorous application of ML techniques. While approaches such as MLOps (Tamburri, 2020) and tools such as Amazon SageMaker (Das et al., 2020) can help development teams reach maturity and predictability in their ML development, it is also important to enhance the maturity of the development team in terms of AI governance and auditing to ensure legal and ethical compliance (Birkstedt et al., 2023).

Third, the findings highlight the importance of a frame for building an understanding between management and developers. As ISD projects are getting more complex, there should be conceptual support that unites all stakeholders to form a shared understanding of the ML development work, what challenges and risks are included, and what support the project needs. In our data MLOps emerged as one potential such frame that could support both the development work as well as project management throughout the project lifecycle. The end-to-end visibility, potential to support compliance and governance, risk management, and resource optimization can become highly valuable, particularly in bigger projects, and perhaps these factors are the reason why the DevOps paradigm has been widely adopted to guide ML development work in practice, leading to the birth of MLOps.

With these contributions, we address calls from previous research to focus on how to integrate the work of ML engineers and data scientists with software development teams (Jüngling et al., 2020; Laato et al., 2022b). This work also addresses the identified gap in the IS literature concerning the lack of research on ML-specific ISD models such as DevOps (Sharp & Babb, 2018). We contribute to these studies by providing an overview of the key characteristics that experts focus on when combining pre-existing ISD practices with ML engineering.

5.3 Limitations and Future Work

As with all research, this study has limitations. The first relates to our empirical investigation that was sensitized by the selected four ISD models that are widely used in contemporary ISD practice. While building the interviews around these models had the advantage of prompting the informants to reflect on the topic from four different viewpoints, they ultimately also guided the resulting discussion, perhaps leading to the omission of some other aspects. The second relates to grounding our research on the ISD research tradition, which inevitably directed the focus of the empirical inquiry. This means that our work might not have captured the more technical aspects of the topic, nor perhaps all the nuances of the social elements involved in the formation of practices and related decision-making. Thus, we encourage future research also in the field of IS to look at complementary empirical approaches to bring further clarity to the topic. In terms of the empirical investigation, we interviewed a diverse set of professionals working within the fields of IS production, ML engineering, and ISD management. However, the sampling does have some blind spots. Mainly, we are missing informants working with large global IT products.

Arising from our findings were also new future research directions. Future research could advance the understanding of the role of external factors (e.g., national culture, regulatory environments), which could help developers to better account for possible external obligations and demands that affect ISD activities. Future research could also examine the ML system development process from one abstraction level higher than the ISD model to better account for the influence of organizational and environmental factors on the ML system development process. The higher abstraction level would connect ISD models to the emerging research stream on organizational AI governance, which is concerned with the processes, practices, and tools for aligning AI systems with organizational strategies and regulatory and ethical requirements (Birkstedt et al., 2023; Mäntymäki et al., 2021; Papagiannidis et al., 2023). Studies investigating ISD models and practices in the context of AI governance could generate a detailed and contextualized understanding of how regulatory, ethical, and other stakeholder requirements are practically tackled in ML development (Minkkinen et al., 2022; Morley et al., 2020; Schiff et al., 2021), complementing the primarily conceptual AI governance literature (Birkstedt et al., 2023). The practice theory lens, possibly coupled with organizational theory and institutional theory (e.g., Minkkinen & Mäntymäki, 2023), could provide fruitful starting points for this. We also encourage future research to carry out case studies of the integration journeys of ML development into the rest of ISD, as our findings suggested that there is much variety within consulting projects and, thus, a need to understand the specific challenges.

Another critical avenue for future research in this domain is analyses that are grounded in technical practices. In our data, there was evidence of the centralization of development practices as well as indications of the role of developer tools in the process. Hence, we encourage future research to perform close reads on tools such as SageMaker or Valohai to understand the roles of these software (and, by extension, the companies developing and offering them) in the formation of ML development practices in ISD. Beyond these tools, we have laws and regulations, as well as ethical guidelines, related to the development of ML systems. These play a critical role in ISD practices because development projects must ensure legal and ethical compliance, and this requires taking extra steps in the development work. Future research could explore these elements also from the perspectives of involving the legal team as well as trust and safety teams in the development work and what changes this brings to the established ISD practices.

Finally, while AI tools have improved automation in various software environments and cyber-physical systems, it has become evident with the advent of large language models such as GPT-4 and solutions such as GitHub co-pilot that they are now also going to raise the level of abstraction of ISD practices in multiple instances. This marks a shift in programming tasks towards prompting code generation as opposed to writing down everything by hand. Soon, this could greatly reduce the amount of time and effort required to develop software, allowing developers to focus on higher-level tasks such as designing user interfaces and experimentation with data. As in the future, these ML tools are likely to become even more sophisticated, we may see a further shift towards a more natural language-based approach to programming. This could have fundamental impacts on ISD practices, processes, and ISD models. Therefore, we envisage this to be a ripe area for future research and encourage IS scholars to closely monitor the developments in this emerging research space.

6 Conclusion

This study highlights the heterogeneity of needs and contextual factors with respect to ML development and ML systems that need to be accommodated in conceptual ISD models. Our results underscore the variability in ML system development and the importance of flexibility in choosing the conceptual frameworks, thinking patterns, and practices to support the development activities. The proliferation of data-intensive development poses challenges for management and project teams. To alleviate issues pertaining to the lack of understanding between developers and management, our findings highlight ISD models that are close to the development paradigm (such as MLOps) as conceptualization that show promise in helping all stakeholders involved in the ISD acquire a shared vision with regards to ML development. Our findings further highlight the diversity of tools and projects under the ML umbrella and the need to understand the implications of specific ML tools on ISD instead of discussing ML broadly. We conclude that while already some adjustments have been made to conceptual ISD models to accommodate ML development, as indicated by the conceptual transformation of the DevOps paradigm into MLOps, the process of incorporating ML development as part of ISD practices remains by large practice driven. However, recent advances in ML systems (e.g., large language model-based systems such as ChatGPT) showcase that we are on a constantly evolving path, and moving forward we may see further and stronger adjustments to conceptual ISD models.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). *Agile software development methods: Review and analysis*. arXiv preprint arXiv:1709.08439. Retrieved from https://arxiv.org/abs/1709.08439

Akkiraju, R., Sinha, V., Xu, A., Mahmud, J., Gundecha, P., Liu, Z, Xiaotong Liu & Schumacher, J. (2020). Characterizing machine learning processes: A maturity framework. In D. Fahland, C. Ghidini, J. Becker, & M. Dumas (Eds.), *International conference on business process management*.

Asatiani, A., Malo, P. Nagbøl, P. R., Penttinen, E., Rinta-Kahila, T., & Salovaara, A. (2021). Sociotechnical envelopment of artificial intelligence: An approach to organizational deployment of inscrutable artificial intelligence systems. *Journal of the Association for Information Systems*, *22*(2), 8.

Baghizadeh, Z., Cecez-Kecmanovic, D., & Schlagwein, D. (2020). Review and critique of the information systems development project failure literature: An argument for exploring information systems development project distress. *Journal of Information Technology*, *35*(2), 123-142.

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. v-model vs. agile: A comparative study on SDLC." *International Journal of Information Technology and Business Management*, *2*(1), 26–30.

Barki, H., & Hartwick, J. (2001). Interpersonal conflict and its management in information system development. *MIS Quarterly*, *24*(2), 195-228.

Baseer, K. K., Reddy, A. R. M., & Bindu, C. S. (2015). A systematic survey on waterfall vs. agile vs. lean process paradigms. *i-Manager's Journal on Software Engineering*, *9*(3), 34.

Bawack, R. E., & Ahmad, M. O. (2021). Understanding business analytics continuance in agile information system development projects: An expectation-confirmation perspective. *Information Technology & People*, *34*(6), 1551-1569.

Benlian, A. (2022). Sprint zeal or sprint fatigue? The benefits and burdens of agile ISD practices use for developer well-being. *Information Systems Research*, *33*(2), 557-578.

Berente, N., Gu, B., Recker, J., & Santhanam, R. (2021). Managing artificial intelligence. *MIS quarterly*, *45*(3), 1433-1450.

Birkstedt, T., Minkkinen, M., Tandon, A., & Mäntymäki, M. (2023). AI governance: Themes, knowledge gaps and future agendas. *Internet Research*, *33*(7), 133-167.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer, 21*(5), 61–72.

Bourdieu, P. (1977). Outline of a theory of practice. Cambridge University Press.

Bowen, G. A. (2006). Grounded theory and sensitizing concepts. *International Journal of Qualitative Methods*, *5*(3), 12-23.

Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring – A systematic literature review. *Journal of Systems and Software*, (110), 85-100.

Cecez-Kecmanovic, D., Galliers, R. D., Henfridsson, O., Newell, S., & Vidgen, R. (2014). The sociomateriality of information systems: Current status, future directions. *MIS Quarterly*, *38*(3), 809-830.

Chinnamgari, S. K. (2019). *R machine learning projects: Implement supervised, unsupervised, and reinforcement learning techniques using R 3.5. 1st Edition*. Packt.

Coombes, L., Allen, D., Humphrey, D., & Neale, J. (2009). *In-depth interviews*. *Research Methods for Health and Social Care* (pp. 197–210). Retrieved from https://www.researchgate.net/publication/313552823_In-depth_interviews

Cui, G., Wong, M. L., & Lui, H.-K. (2006). Machine learning for direct marketing response models: Bayesian networks with evolutionary programming. *Management Science*, *52*(4), 597-612.

Das, P., Ivkin, N., Bansal, T., Rouesnel, L., Gautier, P., Karnin, Z., Dirac, L., Ramakrishnan, L., Perunicic, A., Shcherbatyi, I., Wu, W., Zolic, A., Shen, H., Ahmed, A., Winkelmolen, F., Miladinovic, M., Archembeau, C., Tang, A., Dutt, B., Grao, P., & Venkateswar, K. (2020). Amazon SageMaker autopilot: A white box AutoML solution at scale. In *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning* (pp. 1-7).

Dennehy, D., & Conboy, K. (2019). Breaking the flow: A study of contradictions in information systems development (ISD)*. Information Technology & People*, *33*(2), 477-501.

Dennehy, D., Griva, A., Pouloudi, N., Dwivedi, Y. K., Mäntymäki, M., & Pappas, I. O. (2022). Artificial intelligence (AI) and information systems: Perspectives to responsible AI. *Information Systems Frontiers*, *25*, 1-7.

de Vaujany, F.-X., Fomin, V. V., Haefliger, S., & Lyytinen, K. (2018). Rules, practices, and information technology: A trifecta of organizational regulation. *Information Systems Research*, *29*(3), 755-773.

Dingsøyr, T., Dybå, T., Gjertsen, M., Jacobsen, A. O., Mathisen, T. E., Nordfjord, J. O., Røe, K., & Strand, K. (2019). Key lessons from tailoring agile methods for large-scale software development*. IT Professional*, *21*(1), 34-41.

Dittrich, Y. (2016). What does it mean to use a method? Towards a practice theory for software engineering. *Information and Software Technology*, *70*, 220-231.

Dwivedi, Y. K., Wastell, D., Laumer, S., Henriksen, H. Z., Myers, M. D., Bunker, D., Elbanna, A., Ravishankar, M. N., & Srivastava, S.C., (2015). Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, *17*(1),143-157.

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps*. IEEE Software, 33*(3), 94–100.

Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P. (2015). Software development in startup companies: The greenfield startup model*. IEEE Transactions on Software Engineering*, *42*(6), 585–604.

Giddens, A. (1984). *The constitution of society: Outline of the theory of structuration*. Polity Press.

Gill, G., & Bhattacherjee, A., (2009). Whom are we informing? Issues and recommendations for MIS research from an informing sciences perspective. *MIS Quarterly*, *33*(2), 217-235.

Gioia, D., Corley, K., Eisenhardt, K., Feldman, M., Langley, A., Lê, J., Golden-Biddle, K., Locke, K., Mees-Buss, J., Piekkari, R., Ravasi, D., Rerup, C., Schmid, T., Silverman, D., & Welch, C. (2022). A curated debate: On using “templates” in qualitative research. *Journal of Management Inquiry*, *31*(3), 231-252.

Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2013). Seeking qualitative rigor in inductive research: Notes on the Gioia methodology. *Organizational Research Methods*, *16*(1), 15-31.

Hansen, S., & Baroody, A. J. (2020). Electronic health records and the logics of care: Complementarity and conflict in the U.S. healthcare system. *Information Systems Research*, *31*(1), 57-75.

Hassan, N. R., & Mathiassen, L., (2018), Distilling a body of knowledge for information systems development. *Information Systems Journal*, *28*(1), 75-226.

Hirschheim, R., & Klein, H .K. (1989). Four paradigms of information systems development. *Communications of the ACM*, *32*(10), 1199-1216.

Ishikawa, F., & Yoshioka, N. (2019). How do engineers perceive difficulties in engineering of machine learning systems? Questionnaire survey. In *2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry* (CESI) *and 6th International Workshop on Software Engineering Research and Industrial Practice* (SER&IP).

Jüngling, S., Peraic, M., & Martin, A. (2020). Towards AI-based solutions in the system development - lifecycle. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engi*neering.

Laato, S., Mäntymäki, M., Islam, A. K. M., Hyrynsalmi, S., & Birkstedt, T. (2022a). Trends and trajectories in the Software Industry: Implications for the future of work. *Information Systems Frontiers*, *25*, 1-16.

Laato, S., Birkstedt, T., Mäntymäki, M., Minkkinen, M., & Mikkonen, T. (2022b). AI governance in the system development life cycle: Insights on responsible machine learning engineering. In *Proceedings of the 1st Conference on AI Engineering-Software Engineering for AI*.

Lyytinen, K., & Rose, G. M. (2006). Information system development agility as organizational learning. *European Journal of Information Systems*, *15*(2), 183-199.

Karamitsos, I., Albarhami, S., & Apostolopoulos, C. (2020). Applying DevOps practices of continuous automation for machine learning. *Information*, *11*(7), 363.

Kautz, K. K., & Bjerknes, G. (2020). Information systems development as value co-creation. *Communications of the Association for Information Systems*, *47*(1), 37.

Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2016). The emerging role of data scientists on software development teams. In *Proceedings of the 38th International Conference on Software Engineering* (pp. 96-107).

Knorr Cetina, K. (2005). Objectual practice. In K. Knorr Cetina, T. R. Schatzki, & E. von Savigny (Eds.), *The Practice Turn in Contemporary Theory* (pp. 184-197). Routledge.

Kühl, N., Hirt, R., Baier, L., Schmitz, B., & Satzger, G. (2021). How to conduct rigorous supervised machine learning in information systems research: The supervised machine learning report card. *Communications of the Association for Information Systems*, *48*.

Maruping, L. M., & Matook, S. (2020). The multiplex nature of the customer representative role in agile information systems development. *MIS Quarterly*, *44*(3), 1411-1437.

Mäntymäki, M., Hyrynsalmi, S., & Koskenvoima, A. (2020). How do small and medium-sized game companies use analytics? An attention-based view of game analytics. *Information Systems Frontiers*, *22*(5), 1163-1178.

Mäntymäki, M., Minkkinen, M., Birkstedt, T., & Viljanen, M. (2022). Defining organizational AI governance. *AI and Ethics*, *2*(4), 603-609

Mantei, M. M., & Teorey, T. J. (1989). Incorporating behavioral techniques into the systems development life cycle. *MIS Quarterly*, 13(3), 257–274.

Mees-Buss, J., Welch, C., & Piekkari, R. (2022). From templates to heuristics: How and why to move beyond the Gioia methodology. *Organizational Research Methods*, *25*(2), 405-429.

Minkkinen, M., & Mäntymäki, M. (2023). The institutional logics underpinning organizational AI governance practices. In *14th Scandinavian Conference on Information Systems*.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning. Second edition. Adaptive Computation and Machine Learning*. The MIT Press.

Morley, J., Floridi, L., Kinsey, L., & Elhalal, A. (2020). From what to how: An initial review of publicly available AI ethics tools, methods and research to translate principles into practices. *Science and Engineering Ethics*, *26*(4), 2141-2168.

Mucha, T. M., Ma, S., & Abhari, K. (2022). Beyond MLOps: The lifecycle of machine learning-based solutions. In *Proceedings of the American Conference on Information Systems* (AMCIS).

Niederman, F. (2021). Project management: Openings for disruption from AI and advanced analytics. *Information Technology & People*, 0959-3845.

Noy, C. (2008). Sampling knowledge: The hermeneutics of snowball sampling in qualitative research. *International Journal of Social Research Methodology*, *11*(4), 327-344.

O'Connor, M., Conboy, K., & Dennehy, D. (2022). COVID-19 affected remote workers: A temporal analysis of information system development during the pandemic. *Journal of Decision Systems*, *31*(3), 207-233.

Öbrand, L., Augustsson, N. P., Mathiassen, L., & Holmström, J. (2019). The interstitiality of IT risk: An inquiry into information systems development practices. *Information Systems Journal*, *29*(1), 97-118.

Oomen, J., Hoffman, J., & Hajer, M. A. (2021). Techniques of futuring: On how imagined futures become socially performative. *European Journal of Social Theory*, *25*(2).

Orlikowski, W. J. (2000). Using technology and constituting structures: a practice lens for studying technology in organizations. *Organization Science*, *11*(4), 404-428.

Orlikowski, W. J. (2002). Knowing in practice: Enacting a collective capability in distributed organizing. *Organization Science*, *13*(3), 249-273.

Päivärinta, T., & Smolander, K. (2015). Theorizing about software development practices. *Science of Computer Programming*, *101*, 124-135.

Papagiannidis, E., Enholm, I. M., Dremel, C., Mikalef, P., & Krogstie, J. (2023). Toward AI governance: Identifying best practices and potential barriers and outcomes. *Information Systems Frontiers*, *25*(1), 123-141.

Rosemann, M., & I. Vessey (2008). Toward improving the relevance of information systems research to practice: The role of applicability checks. *MIS Quarterly, 32*(1), 1-22.

Royce, W. (1970). Managing the development of large software systems. In *Proceedings of the 9th International Conference on Software Engineering*.

Schatzki, T. R. (1996). *Social practices: A Wittgensteinian approach to human activity and the social*. Cambridge University Press.

Schiff, D., Rakova, B., Ayesh, A., Fanti, A., & Lennon, M. (2021). Explaining the principles to practices gap in AI. *IEEE Technology and Society Magazine*, *40*(2), 81-94.

Seidman, I. (2019). Interviewing as qualitative research: A guide for researchers in education and the social sciences. Teachers College Press.

Sharp, J., & Babb, J. (2018). Is information systems late to the party? The current state of DevOps research in the Association for Information Systems eLibrary. In *Proceedings of the Americas Conference on Information Systems*.

Sidorova, A., Evangelopoulos, N., Valacich, J. S., & Ramakrishnan, T. (2008). Uncovering the intellectual core of the information systems discipline, *MIS Quarterly*, *32*(3), 467-482.

Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). SCRUM model for agile methodology. In *2017 International Conference on Computing, Communication and Automation* (ICCCA). IEEE.

Strauss, A., & Corbin, J. (1990). *Basics of qualitative research*. Sage publications.

Tamburri, D. A. (2020). Sustainable MLOps: trends and challenges. In *22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (SYNASC) (pp. 17–23). IEEE.

Trauth, E. (2017). A research agenda for social inclusion in information systems. *ACM SIGMIS Database: the Database for Advances in Information Systems*, *48*(2), 9-20.

Vasist, P. N., & Krishnan, S. (2022). Deepfakes: An integrative review of the literature and an agenda for future research. *Communications of the Association for Information Systems*, *51*(1), 14.

Virmani, M. (2015). Understanding devops & bridging the gap from continuous integration to continuous delivery. In *Fifth International Conference on the Innovative Computing Technology*.

Wiesche, M., Jurisch, M. C., Yetton, P. W., & Krcmar, H. (2017). Grounded theory methodology in information systems research. *MIS Quarterly*, *41*(3), 685-701.

Zhang, P., Carey, J., Te'eni, D., & Tremaine, M. (2005). Integrating human-computer interaction development into the systems development life cycle: A methodology. *Communications of the Association for Information Systems*, *15*(1), 29.

About the Authors

Samuli Laato works as a Postdoctoral Researcher at Tampere University, Finland. He earned his Ph.D. from the University of Turku in 2021 and has worked on projects in the fields of information systems, education, and human-computer interaction. Samuli's research has been published in outlets such as the European Journal of Information Systems, Computers in Human Behavior, and Information Systems Frontiers. He has served as a committee member for several conferences such as the European Conference on Information Systems and the CHI conference on Human Factors in Computing Systems. His current research interests include generative AI systems and pervasive technologies.

Matti Minkkinen is a Postdoctoral Researcher in Information Systems at the University of Turku. He conducted his Ph.D. research on privacy protection in the European Union, and his recent research covers socio-technical perspectives on responsible artificial intelligence and systemic foresight processes. Minkkinen has several years of research and teaching experience on the interplay between future visions and socio-technical change as well as foresight methods. His research has been published in journals such as Technological Forecasting & Social Change, Information Systems Frontiers, and New Media & Society.

Matti Mäntymäki is a Professor of Information Systems Science at University of Turku, Finland. His research interests cover psychosocial, organizational, and business implications of digitalization, including governance and social responsibility of artificial intelligence. He has published more than 120 peer-reviewed papers in outlets such as Information Systems Journal, Information & Management, Technological Forecasting & Social Change, International Journal of Information Management, Journal of Business Research, Information Systems Frontiers, Information Technology & People, Journal of Systems & Software, and Communications of the Association for Information Systems, among others.

Denis Dennehy is Associate Professor of Business Analytics and School Research Lead at the School of Management, Swansea University, Wales. His research primarily focuses on the mediating role of technologies and data, and its implications for teams, organisations, and society. He has worked on several industry-oriented research projects funded by UKRI, Enterprise Ireland, Science Foundation Ireland, Erasmus+, and Irish Aid. This research has been published in leading journals including Information and Management, Journal of Business Research, Information Systems Frontiers, International Journal of Operations & Production Management, European Journal of Operational Research, and IEEE. He is co-editor-in-chief for Communications of the Association for Information Systems, senior editor for Information Technology and People, and he has edited many special issues related to his field.