Why and How do Large-scale Organizations Operationalize DevOps

Inauguraldissertation

zur

Erlangung des Doktorgrades

der

Wirtschafts- und Sozialwissenschaftlichen Fakultät

der

Universität zu Köln

2023

presented

by

Dipl.-Wirt.-Inf. Michael Hüttermann

from

Kirchhellen, jetzt Bottrop

First reviewer: Prof. Dr. Christoph Rosenkranz
Second reviewer: Prof. Dr. Lea Ellwardt
Date of oral defense: September 20, 2023

Abstract

An essential part of organizational efforts is to provide products to customers. To sustain competitive positions on existing markets, and to expand into new markets, firms utilize and continuously optimize approaches to efficiently provide effective products. Meanwhile, applying agile practices is a commoditized way for organizations to better adapt to changes during the development of their products. For bringing products to customers, more than their development is required. Typically, multiple organizational functions, all with individual goals and practices, are included in the development and delivery of products. This is often associated with friction points between those functions, and hinders the optimization of effectiveness and efficiency in providing products to customers. In retrospective, not all firms were able to recalibrate themselves and find back to former success after they had once missed to (again) innovate by timely addressing changes on their existing markets, discovering unmet or changed customer needs, and providing new products that bring together emerging technology with evolving customer demands (Nokia Corporation, 2023). This potential threat now appears to be omnipresent with the ongoing proliferation of digitalization through the practical world of all of us.

The emerging phenomenon of DevOps, a portmanteau word of "development" and "operations", describes approaches to streamline development and delivery of products across organizational functions, to efficiently provide effective products, and to enable organizational digitalization efforts. This dissertation sheds light on reasoning, configurational factors, and dynamics behind DevOps implementations in large-scale. The composition of four independent yet interrelated scientific papers, the cornerstones of this dissertation, answers why and how large-scale organizations operationalize DevOps. In sum, this dissertation adds systematic and foundational knowledge, presents new applications and nuanced concretizations of scientific empiric approaches, connects allied but distinct research communities, and provides guidance for practitioners acting in this timely, relevant and interesting domain.

Table of Content

List of Figures

List of Tables

Abbreviations

DevOps	Development Operations
ISSM	Information Systems Success Model
IS	Information Systems
ISD	Information Systems Development
AISD	Agile Information Systems Development
IT	Information Technology
ISSM	Information System Success Model
GTM	Grounded Theory Method

1 Dissertation Overview

1.1 Introductory Summary

In industry, to develop and deliver information systems (IS) products, the approach of *DevOps*, a portmanteau word of “development” and “operations”, is widely spread to integrate organizational functions and to maximize the probability of success (digital.ai, 2022; Google, 2022). However, in contrast to its pervasive industrial proliferation (digital.ai, 2022; Google, 2022), research on the emerging DevOps phenomenon is still scant. DevOps means different things to different people (Qumer Gill *et al.*, 2018). Prior studies have provided very valuable preliminary insights on DevOps and some of its probable characteristics, mechanisms, and effects (Gall and Pigni, 2021; Hemon-Hildgen, Rowe, *et al.*, 2020; Wiedemann *et al.*, 2020), while recognizing DevOps as a set of principles for collaborative work between organizational functions and a next step in evolving agile information systems development (ISD) (Gall and Pigni, 2021). However, a foundational grounding of DevOps is largely missing. There is little systematic knowledge about why and how large-scale organizations implement DevOps.

My dissertation aims to further unpuzzle the emerging DevOps phenomenon. It is written cumulatively, and is based on a compilation of three conference publications and one journal manuscript (which is currently under review), see Table 1-1. For the sake of simplicity and better readability, across the dissertation, I refer to those items as “papers” and use Roman numerals to reference them. The list of papers is ordered chronologically, with paper III being my last paper that was published already at this moment in time.

My dissertation is the product of my own work; however, the entire project was not run in a social vacuum. Table 1-2 documents the research team per paper and the specific contributions of the named (co-)authors. All papers articulate from the perspective of “we” to involve unknown reviewers of the papers and the community the papers engage with and talk to.

The results of this dissertation contribute to a further understanding of the DevOps phenomenon. They shed light on reasoning behind DevOps implementations, and how DevOps is operationalized in large-scale organizations. Grounded on data, this dissertation uncovers that:

- DevOps is an approach to reconcile different ways to measure IS success.

- DevOps is a vehicle to implement organizational goals, for example digitalization.
- Success, practices and actors are the most dominant factors to initially introduce DevOps in large-scale organizations.
- The DevOps continuum is a set of environmental characteristics, with gradual transitions between their instantiations ("flavors"), influencing how DevOps is implemented in a specific context.
- Two mechanisms, "effectiveness adaptation" and "efficiency adaptation", are causal forces that may lead to new DevOps implementations in large-scale organizations.
- DevOps is an approach to reconcile "business-IT alignment". DevOps may lead to a fusion of business and IT (with its subunits development and operations) toward autonomous teams, and a way to implement "digital business strategies".

Paper	**Title**	**Outlet and status**	**Type**	**CON/JNL***
I	DevOps: Walking the Shadowy Bridge from Development Success to Information System Success	International Conference on Information Systems (ICIS), Munich 2019 (accepted).	Short paper	CON (VHB: A)
II	The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations	European Conference on Information Systems (ECIS), Marrakech 2021 (accepted).	Research paper	CON (VHB: B)
III	The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations	Hawaii International Conference on System Sciences (HICSS), Hawaii 2023 (accepted).	Research paper	CON (VHB: C)
IV	From Loose Coupling to Fusion: How Does DevOps Operationalize Digital Business Strategies and Contribute to Digital Innovation?	Information Systems Research (ISR): the revised and resubmitted manuscript after first review round. Manuscript now again in revision after receiving 2nd round of review with the offer/request to revise and resubmit.	Research article	JNL (VHB: A+)
*CON: Conference; JNL: Journal. Ranking of VHB: German Academic Association for Business Research.				

Table 1-1. The dissertation is based on four papers.

Besides adding systematic knowledge by suggesting novel theory and integrating with existing concepts and models, this dissertation also adds nuanced concretizations of research approaches, links discourses on related but largely disconnected research streams, and offers valuable guidance for practitioners. Organizations planning either to initially implement DevOps or to change their current DevOps implementation find valuable hints about DevOps reasoning, configurational factors and dynamics behind respective DevOps implementations. Assisted by this work, they might find the DevOps implementation that is the "best-fit" for their specific context. They can profit from the empirical evidence this dissertation provides about statics and dynamics of DevOps implementations, and how DevOps contributes to successful implementations of team goals and organizational goals, for example digitalization, to balance effectiveness and efficiency, to gain or maintain differential value for customers on competitive markets, to optimize overall IS success.

Paper	I. DevOps: Walking the Shadowy Bridge from Development Success to Information Systems Success (ICIS)	II. The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations (ECIS)	III. The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations (HICSS)	IV. From Loose Coupling to Fusion: How Does DevOps Operationalize Digital Business Strategies and Contribute to Digital Innovation? (ISR)
Research Team	Hüttermann Rosenkranz	Hüttermann Rosenkranz	Hüttermann Rosenkranz	Hüttermann Rosenkranz
Named (co-) authors	Hüttermann Rosenkranz	Hüttermann	Hüttermann Rosenkranz	Hüttermann Rosenkranz
Research Design	Hüttermann	Hüttermann	Hüttermann	Hüttermann Rosenkranz
Literature research	Hüttermann	Hüttermann	Hüttermann	Hüttermann
Data Collection	Hüttermann	Hüttermann	Hüttermann	Hüttermann
Data Analysis	Hüttermann	Hüttermann	Hüttermann	Hüttermann
Theorizing	Hüttermann	Hüttermann	Hüttermann	Hüttermann Rosenkranz
Write Up	Hüttermann Rosenkranz	Hüttermann	Hüttermann	Hüttermann
Supervisor / Review	Rosenkranz	Rosenkranz	Rosenkranz	Rosenkranz

Table 1-2. Research team per paper.

1.2 Structure of this Dissertation

This cumulative dissertation is organized as follows. The first chapter provides an introductory summary for this scientific work, and how this dissertation is structured. Chapter two presents the research project. Influenced by a common layout of publications in the underlying domain of systems development and aligned with this specific purpose, it offers an introduction with motivation and research question, enrolls the theoretical background and related work, summarizes the four papers my dissertation is based on, provides details of the research design, discusses the main findings and integrates them with prevalent literature, summarizes contributions and avenues for future research, and ends with a conclusion. The respective content of that chapter stems from the lens of the overarching research project. It explains how the four papers are contextually linked as the pillars of the research project to answer the overall research question. The chapter balances the required level of detail of information needed to understand the overall research project, its main results, and the role of the contextually linked papers (aligned with the structure of the dissertation), while this introduction enables the reader to zoom in to the individual papers for more details. For this reason, the following elements from papers I-IV were reused and embedded into chapter two of the dissertation: Figure 2-5, Figure 2-6 (summary), Figure 2-7 (updated), Figure 2-10, and Table 2-6. Chapter three to six contain the four papers. Each paper is included as submitted/published and was reformatted to fit into the overall format of this dissertation. As detailed below, as part of its publication, the online appendices A-E of paper III were made available at osf.io. In addition, for the sake of completeness (i.e., to include all parts to this document), they are added to this document as appendices at the end of chapter 5. Online appendices of paper IV, the paper which is under review at ISR, were uploaded to the outlet's author online system. I have added them as-is at the end of chapter 6, the chapter that contains the paper IV.

2 Unpuzzling the DevOps Phenomenon: Why and How do Large-scale Organizations Operationalize DevOps?

2.1 Introduction

Information systems development (ISD) comprises a set of development activities (e.g., planning, analysis, design, building, testing), undertaken by organizational actors, to create working information systems (IS products) (Greenwood *et al.*, 2020; Matook *et al.*, 2021). To gain or maintain competitive advantages, organizations seek for approaches to develop and deliver their IS products faster and better aligned with customer needs (digital.ai, 2022; Google, 2022). To accomplish this, since the 1990s, organizations often utilize flexible and adaptable *agile* approaches (Dingsøyr *et al.*, 2012; Dybå and Dingsøyr, 2009; Rajlich, 2006; Siau *et al.*, 2022). Agile approaches such as Scrum (Beck, 1999; Dybå and Dingsøyr, 2008) emphasize the continuous delivery of valuable outcome to optimize customer satisfaction (Conboy, 2009; Fowler and Highsmith, 2001).

Meantime, the application of agile approaches is widely spread in industry (digital.ai, 2022; Google, 2022). This proliferation "across almost every corner of contemporary software development" (Carroll *et al.*, 2023) evolves ISD toward agile information systems development (AISD) (Conboy, 2009; Dingsøyr *et al.*, 2012; Matook *et al.*, 2021). With AISD, the functions of business and *IS development* (typically a part of an organizational IT function) integrate closer with each other to work together on new versions of the respective IS product (Dybå and Dingsøyr, 2008, 2009; Fitzgerald *et al.*, 2006). However, AISD does only minimally emphasize the "go-live" transition and the operations of the IS product (Siau *et al.*, 2022). Due to a division of labor into distinct organizational units (Greenwood *et al.*, 2020; Lwakatare, Karvonen, *et al.*, 2016), the *IS operations* function (those who make available the IS product on production systems; typically a part of an organizational IT function) is often not included in the efforts of AISD (Lwakatare, Karvonen, *et al.*, 2016), see Figure 2-1, leading to barriers and friction points between IS development and IS operations (Siau *et al.*, 2022). On a similar footing, as a result of the division of labor (Greenwood *et al.*, 2020), organizational functions often have micro-optimized goals, leading to diverged incentives of functions (Hüttermann, 2012). For example, the IS development function aims to bring new changes to production frequently (Recker *et al.*, 2017), while the IS operations function typically has the goal to maximize the stability of the IS product in production and to minimize maintenance time (Dekleva, 1992) – two conflicting goals since changing the production system may risk its stability and cause maintenance time (Hüttermann, 2012). In sum, these challenges led to the rise of the DevOps phenomenon.

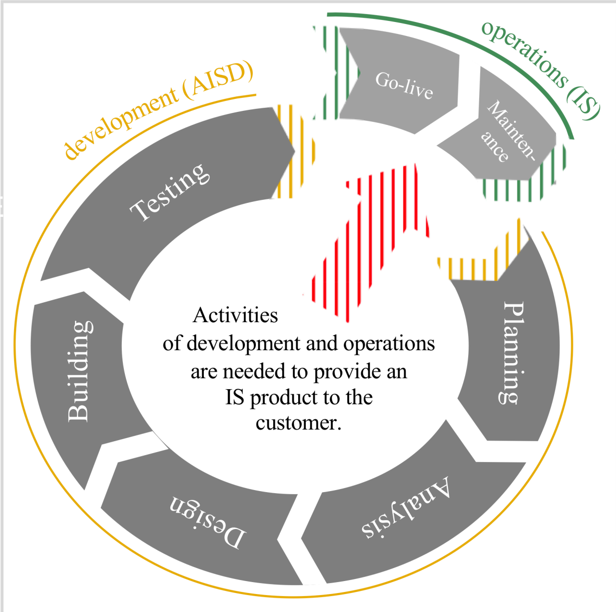

Figure 2-1. Activities of IS development cover a subset of all activities needed to provide an IS product to the customer. Operational activities are often located in a dedicated function. Activities adopted from Matook et al. (2021) and Siau et al. (2022).

DevOps describes principles for collaborative work across organizational functions (Fitzgerald and Stol, 2017; Gall and Pigni, 2021), specifically IS development and IS operations[1], to continuously deliver valuable outcome to customers (Fowler and Highsmith, 2001; Wiedemann, Forsgren, *et al.*, 2019). Popular parlance coined the term DevOps in 2009 (Fitzgerald and Stol, 2017) to describe shared practices and forms of collaboration between the two often siloed (and micro-optimized) organizational functions of development and operations (Hüttermann, 2012; Kaiser, 2023).

The emerging phenomenon of DevOps has gained growing attention from IS research (Sharp and Babb, 2018). First studies suggest that the orchestration of automation, sharing and risk management is moderated by work conditions and positively impacts job satisfaction if AISD teams evolve toward DevOps (Hemon-Hildgen, Rowe, *et al.*, 2020), propose a model of alignment inside organizational IT

[1] Different roles are subsumed into development (e.g., architect, developer, tester) and operations (e.g., database, network, security). The industry-driven name of the DevOps phenomenon might be too exclusive regarding specific roles being involved in development and operations of the IS.

functions (Wiedemann *et al.*, 2020), and suggest a conceptual model relating concepts of monitoring, culture and automation (Gall and Pigni, 2021).

However, given the emergence of DevOps in industrial practice (digital.ai, 2022; Google, 2022) and its importance and relevance (Sarker *et al.*, 2013), the phenomenon is apparently under-researched (Locke and Golden-Biddle, 1997; Sandberg and Alvesson, 2011): for example, the literature can be problematized as *incomplete* in (a) not providing empirical support and theoretical grounding to explain why and how DevOps is operationalized in large-scale organizations and why/how these implementations change and (b) not providing systematic knowledge of the concrete role of DevOps in digital business strategy (Bharadwaj *et al.*, 2013) and digital innovation (Nambisan *et al.*, 2017, 2020), given that DevOps is seen as part of the root metaphor for (digital business) strategy in digital innovation (Berente, 2020), and as *inadequate* in suggesting that reaching a maturity level is sufficient (Gupta *et al.*, 2017; Rafi *et al.*, 2021) or a set of skills is needed (Hemon-Hildgen, Lyonnet, *et al.*, 2020; Wiedemann and Wiesche, 2018) to implement DevOps successfully. Along similar lines, the underlying research project of this dissertation aims to "think [about DevOps] differently, instead of what is already known" (Sandberg and Alvesson, 2011). For example, this dissertation integrates the scientific conversation about a novel phenomenon (Colquitt and George, 2011) across disciplinary boundaries by integrating the domains of systems development and digitalization[2] – complementary but distinct communities in the interdisciplinary field of IS research (Nambisan, 2003) with its highly diverse disciplines across disciplinary boundaries (vom Brocke *et al.*, 2015). In sum, systematic knowledge is missing why and how large-scale organizations implement DevOps. Consequently, this research asks:

"Why and how do large-scale organizations operationalize DevOps?"

To further unpack the DevOps phenomenon, to answer this research question and to shed light on different aspects of this overarching research question, it is split into sub-questions, which are answered in the scientific papers I-IV, as detailed in Table 2-1.

---

[2] In fact, these domains can be split into sub-domains. For example, from the perspective of this research, systems development in the lines of development of the IS and aspects of its operations, e.g. the ISSM, and digitalization in the lines of digital innovation and digital business strategy.

Paper	Title	Research Question (RQ)
I	DevOps: Walking the Shadowy Bridge from Development Success to Information Systems Success	RQ1: "What are the origins of success criteria enterprises use for their DevOps initiatives?" RQ2: "What is a suitable conceptualization and definition of DevOps from an ISSM perspective, and how does current practice-in-use match to this?"
II	The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations	RQ: "Which characteristics influence how DevOps is implemented in organizations?"
III	The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations	RQ: "Which factors are important to initially introduce DevOps in organizations, and how does DevOps relate to digitalization?"
IV	From Loose Coupling to Fusion: How Does DevOps Operationalize Digital Business Strategies and Contribute to Digital Innovation?	RQ1: "What characteristics influence how DevOps is implemented in organizations?" RQ2: "How and why do teams adapt their DevOps implementations?" RQ3: "How does DevOps relate to the fusion of business and IT?"

Table 2-1. Research questions per paper.

2.2 Theoretical Foundation and Related Work

2.2.1 DevOps

As a term coined by industry in 2009 (DevOpsDays, 2009), the DevOps phenomenon describes shared practices and forms of collaboration between organizational functions to maximize success (Fitzgerald and Stol, 2017). Its rise stems from friction points related to the division of labor across functions while firms develop and deliver IS products (Greenwood *et al.*, 2020; Hüttermann, 2012). This division of labor is largely rooted in the "Plan-Build-Run" paradigm (Urbach and Ahlemann, 2019) with its main focus on optimizing efficiency (Bosch, 2019; Greenwood *et al.*, 2020). Practitioners advocated to expand the use of agile approaches from IS development to IS operations and closer integrate those organizational functions (e.g., Debois (2008)).

2.2.1.1 DevOps, and Success

In industry, AISD is a commoditized way for organizations to maximize the probability of IS development success (Carroll *et al.*, 2023; Dybå and Dingsøyr, 2009). Success from the development side (Siau *et al.*, 2010) typically focuses on staying within scope, time and cost requirements (Chow and Cao, 2008; Lee and Xia, 2010), software functionality or process performance (Recker *et al.*, 2017). Based on the division of labor in organizations (Greenwood *et al.*, 2020; Lwakatare, Karvonen, *et al.*, 2016), AISD often does not include operational factors (Dybå and Dingsøyr, 2008, 2009). Instead, an IS operations function is responsible for running and maintaining the IS product in production, with primarily measures such as “maintenance time” (the time required to keep up the IS in operations), “mean time between failures” (MTBF) (the time between two failures of the IS in production), and “mean time to repair” (MTTR) as the time needed to restore parts of or the entire IS product after an incident occurs (Dekleva, 1992), see Figure 2-2.

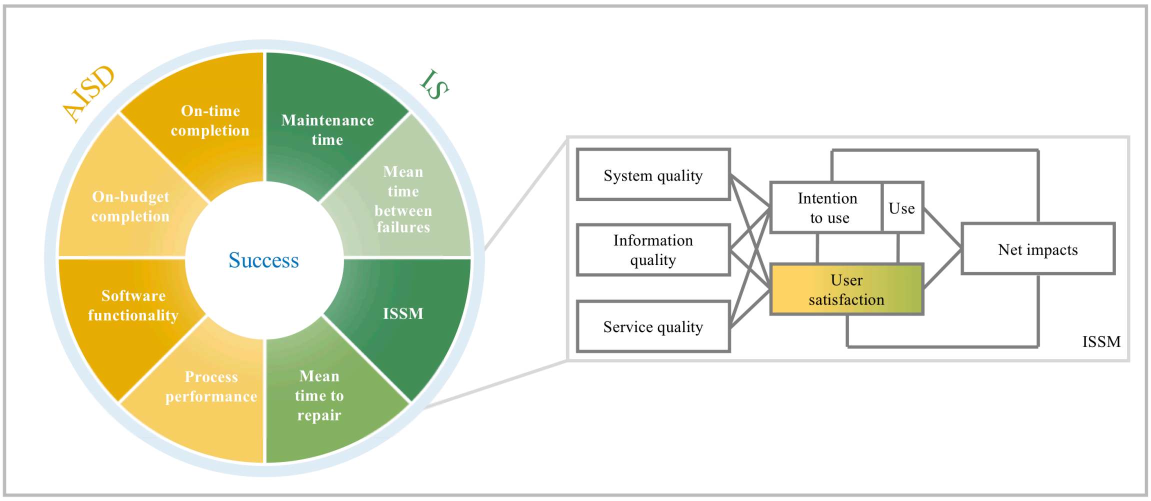

Figure 2-2. Success from perspectives of development (AISD) and operations (IS), with ISSM included as a simplified adoption from DeLone and McLean (2016).

One of the most important contributions (Urbach *et al.*, 2009) to comprehensively measure IS success is the IS Success Model (ISSM) that conceptualizes success as a blend of multiple dimensions including user satisfaction[3] (DeLone and McLean, 1992, 2003; Petter *et al.*, 2013). While the ISSM emphasizes success from the operational side, their authors call for its further refinement and to explore its relationships with ISD (DeLone and McLean 2016, p. 93).

To streamline the development, “go-live”, and operations of the IS product (Siau *et al.*, 2022), and to integrate these functions with each other and with their respective goals, DevOps has gained increasing attention in industry (digital.ai, 2022; Google, 2022; Hüttermann, 2012). As witnessed by Carroll *et al.* (2023), research often follows new and emerging phenomena in software development. Similarly,

[3] The other dimensions are system quality, information quality, service quality, intention to use/use, net impact.

parallel to its rise in industrial practice (digital.ai, 2022; Google, 2022), the emerging phenomenon of DevOps has gained attention from IS researchers (Sharp and Babb, 2018).

2.2.1.2 Toward Studying Implementations of DevOps

DevOps has been recognized as an attempt to scale agile practices (AISD) toward operations (Gall and Pigni, 2021) to maximize the overall IS success (DeLone and McLean, 1992). Existing research has proposed culture, automation, lean, monitoring and measurement, and sharing as substantive dimensions for DevOps (Fitzgerald and Stol, 2017; Gall and Pigni, 2021; Hemon-Hildgen, Rowe, *et al.*, 2020; Wiedemann, Forsgren, *et al.*, 2019), has suggested a DevOps model relating concepts of monitoring, culture and automation (Gall and Pigni, 2021), has uncovered that the orchestration of automation, sharing and risk management is moderated by work conditions and positively impacts job satisfaction if teams move to DevOps (Hemon-Hildgen, Rowe, *et al.*, 2020), and has unfold individual componentization, integrated responsibility and multidisciplinary knowledge as alignment mechanisms in an intra-IT alignment DevOps model (Wiedemann *et al.*, 2020).

Stemming from existing research, this dissertation conceptualizes DevOps as a set of continuously improved practices for collaborative work implemented between organizational functions to develop and deliver an IS product toward continuous delivery of valuable outcome for customers (see paper IV). This definition specifically highlights (a) the necessity of involvement and integration of multiple organizational functions to provide an IS product to the customer (Greenwood *et al.*, 2020), (b) the provision of the IS product has to be done timely and frequently ("efficiently") (Bosch, 2019; Fitzgerald and Stol, 2017; Fowler and Highsmith, 2001), (c) and has to incorporate (customer) feedback to provide "the right IS product" that satisfies the customer ("effectiveness") (DeLone and McLean, 2003; Fowler and Highsmith, 2001), underpinned by (d) the ability of the team to adapt to change (Conboy, 2009) to remain open for or even create change in business environments and competitive markets. Further, this conceptualization (e) reflects that the time span an IS product is in production for customer use is typically much longer than the time required to develop its functionality (Siau *et al.*, 2022), and (f) captures that DevOps aims to find the right balance between effectiveness and efficiency in a given context (Lee and Xia, 2005), supposedly by providing ambidexterity on the team level (Birkinshaw and Gibson, 2004).

2.2.2 Digitalization: Digital Innovation, Digital Infrastructure, and Digital Business Strategy

Similar to DevOps, the concept of digitalization can be tailored to sub-concepts where actors across organizational teams and levels play an important part in (Furr *et al.*, 2022; Nambisan *et al.*, 2020; Osmundsen and Bygstad, 2021) to achieve success (DeLone and McLean, 1992; Furr *et al.*, 2022). Digitalization is recognized as an emerging term in IS research, meaning more than a shift in labels of existing IS research streams ("digital x") (Baiyere *et al.*, 2018, 2023). While it is a "broad and complex phenomenon that does not easily fit into a given theory" (Furr *et al.*, 2022), it is attributed with the ongoing reorganizing and inventing of novel social and technical structures and their relationships (Baiyere *et al.*, 2023). Digitalization comprises different lines of research and challenges a "separation of disciplines" (Tilson *et al.*, 2010).

2.2.2.1 Digital Innovation

Innovation includes the process of developing and implementing new ideas (Van de Ven *et al.*, 2008). "Digital first" (Baskerville *et al.*, 2020) ventures, for example those summarized with the acronym GAFAM (Google, Amazon, Facebook, Apple, Microsoft), have created new perspectives on innovation (Alt *et al.*, 2021). They harness digital technology during innovation leading to *digital innovation* (Nambisan *et al.*, 2017, 2020). Digital innovation is the creation of and change in business processes and models, and market offerings (Nambisan *et al.*, 2017). Stemming from a product-centric perspective (Kohli and Melville, 2019; Yoo *et al.*, 2010), this research conceptualizes digital (product) innovation as organizational efforts leveraging digital technology in between an internal organizational environment and an external competitive environment (Kohli and Melville, 2019) leading to outcomes of (a) new IS products (Yoo *et al.*, 2010), either functional or through new meanings (Wang *et al.*, 2022), and (b) improved customer satisfaction (DeLone and McLean, 1992; Kohli and Melville, 2019).

2.2.2.2 Digital Infrastructure

Digital product innovation results from the use of *digital infrastructure* (Osmundsen and Bygstad, 2021), which this research conceptualizes as a network of digital technologies and capabilities (Osmundsen and Bygstad, 2021; Sandberg, 2014). Digital infrastructure is utilized to "keep in touch with evolving customer needs" (Sia *et al.*, 2016), to create novel value creation and differential value for customers (Bharadwaj *et al.*, 2013; Nambisan *et al.*, 2017; Urbach and Ahlemann, 2019), to maximize IS success (DeLone and McLean, 1992). Digital infrastructure is becoming the central driver of value creation (Osmundsen and Bygstad, 2021; Peppard and Ward, 2004). Digitally innovating ventures recognize IT not only as an industrial fabric function, rather its pervasiveness in other organizational functions such as business and operations (Berente, 2020), to enable digital innovation in increased pace (Berente, 2020; Yoo *et al.*, 2010), either gradual or radical (Alt *et al.*, 2021; Van de Ven *et al.*, 2008; Weick and Quinn, 1999). Digital product innovation with its impact on user-centric factors prompts calls for a new logic of competitive strategies that recognizes the central role of digital infrastructure in delivery of IS products (Woodard *et al.*, 2013). According to the "Innovation-Design-Transform" paradigm (Urbach and Ahlemann, 2019), with its main focus on optimizing effectiveness, organizations need the digital infrastructure to innovate to respond to market events and opportunities to design effective IS products (Sia *et al.*, 2016; Siau *et al.*, 2022), and to transform by changing the (design of the) installed base of infrastructure (Osmundsen and Bygstad, 2021; Woodard *et al.*, 2013) and the structures and processes accordingly (Urbach and Ahlemann, 2019).

2.2.2.3 Digital Business Strategy

According to Bharadwaj et al. (2013), *digital business strategy* is an organizational strategy across and integrating functions executed by leveraging digital infrastructure (Bharadwaj *et al.*, 2013, p. 472). This conceptualization implies an organizational understanding of the importance of the role of digital infrastructure (Kahre *et al.*, 2017), with a merge of IT and business strategies (Kahre *et al.*, 2017) to one digital (business) strategy (Bharadwaj *et al.*, 2013; Peppard *et al.*, 2014; El Sawy, 2003) to enable digital innovations (Bharadwaj *et al.*, 2013; Kahre *et al.*, 2017). Ventures pursuing a digital business strategy are enabled to respond to changing business environments or to identify market opportunities, for example by digitally innovating (Bosch and Olsson, 2021; Olsson and Bosch, 2020; Sia *et al.*, 2016, 2021), and successfully leveraging digital infrastructure (with its technologies and practices) to gain and maintain competitive advantage (Huang *et al.*, 2017) in environmental turbulences (El Sawy *et al.*, 2010). Digital business strategies enact a different form of integration and collaboration between organizational functions (Siau *et al.*, 2022), reconciling "business-IT alignment" (Gerow *et al.*, 2014; Luftman *et al.*, 1999; Reynolds and Yetton, 2015; Tallon and Pinsonneault, 2011), which is an alignment to leverage digital infrastructure for transforming organizations (Henderson and Venkatraman, 1999). As a result, instead of aligned functions of business and IT (with IS development and IS operations), the functions are fused within one distinct autonomous team (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013; Overby *et al.*, 2006; El Sawy, 2003).

2.2.2.4 Summary

Foundational knowledge on the why (including its role in IS success) and the how of DevOps operationalizations is scant. Although DevOps is seen as part of the root metaphor for (digital business) strategy in digital innovation (Berente, 2020), their relation is not yet sufficiently addressed in the neighboring streams of literature of systems development and digitalization (with its lines of research), specifically how the sought-for fusion between business and IT can be achieved to pursue a digital business strategy (Bharadwaj *et al.*, 2013) to enable digital innovation (Kohli and Melville, 2019; Nambisan *et al.*, 2020). Looking through the lens of digitalization, this work is located in the conceptual overlay of digital business strategy (a strategy across and integrating functions) (Bharadwaj *et al.*, 2013) that leverages digital infrastructure (the pair of digital technologies and digital capabilities) (Osmundsen and Bygstad, 2021) to digitally innovate (resulting in the twofold outcome of changed/new IS products and maximized IS success) (Nambisan *et al.*, 2017, 2020), see Figure 2-3.

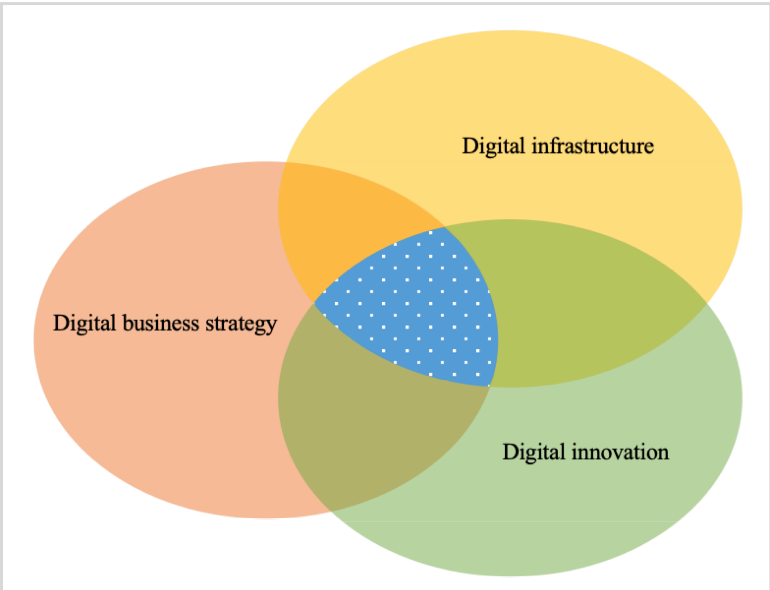

Figure 2-3. This work is located in the conceptual overlay of digital business strategy that leverages digital infrastructure to digitally innovate.

Table 2-2 lists the core concepts as part of the theoretical foundation.

Core concepts	Short descriptions (in relation to this research)	Key references
DevOps	A set of continuously improved practices for collaborative work implemented between organizational functions to develop and deliver an IS product toward continuously delivery of valuable outcome for customers.	(Bosch, 2019; Conboy, 2009; Fowler and Highsmith, 2001; Hemon-Hildgen, Rowe, *et al.*, 2020; Wiedemann *et al.*, 2020)
IS success	A goal and its measurement to efficiently provide effective IS products to gain/maintain differential value for customers, with the ISSM being a model that conceptualizes success as an array of dimensions including user satisfaction. Effectiveness and efficiency support providing differential value for customers while being responsive to constantly changing environments.	(Bharadwaj *et al.*, 2013; Bosch, 2019; DeLone and McLean, 1992, 2003; Urbach and Ahlemann, 2019)
Digital innovation	The creation of and change in market offerings through provision of new/changed IS products resulting from the use of digital infrastructure to maximize IS success. Occurs in between two environments: internal organizational and external competitive.	(Alt *et al.*, 2021; Furr *et al.*, 2022; Huang *et al.*, 2017; Kohli and Melville, 2019; Nambisan *et al.*, 2017, 2020; Yoo *et al.*, 2010)
Digital infrastructure	The installed base of digital technologies and practices (capabilities, as a collection of routines) used by the team to develop and deliver an IS product. Being from strategic importance, both technologies and practices are a subset of the space of opportunities conditioned by digital evolution.	(Henfridsson and Bygstad, 2013; Osmundsen and Bygstad, 2021; Sandberg, 2014; Tilson *et al.*, 2010)
Digital business strategy	An organizational strategy across, integrating or fusing functions leveraging digital infrastructure. Contributes to digital innovation to maximize IS success.	(Bharadwaj *et al.*, 2013; Furr *et al.*, 2022; Henry Lucas *et al.*, 2013; El Sawy, 2003)

Table 2-2. Pivotal concepts as part of the theoretical foundation for this research.

2.3 Overview of Papers

This chapter provides an overview of each of the four papers to explain their respective focus as part of the overarching research project, how they are contextually linked, with their main results, to answer the research question why and how do large-scale organizations operationalize DevOps. The four individual (self-contained) papers integrate with existing literature (and theory). This dissertation builds theory to explain and predict DevOps implementations (Gregor, 2017). It develops new concepts and novel insights to develop a plausible understanding of a poorly understood phenomenon (Sarker *et al.*, 2018), along the themes *sensitize*, *initialize*, *implement*, *adopt*. These themes are inspired by key components of an innovation journey (Van de Ven *et al.*, 2008, p. 25) and span questions of why and how of organizational DevOps initiatives, see Table 2-3.

Paper I contains a review of background literature and builds preliminary theory. The paper presented the state of this research to uncover the dominant incentives of organizations for their DevOps efforts.

A case research was designed to qualitatively explore the emerging DevOps phenomenon, and a preliminary DevOps model was suggested that identified diverse success criteria of development and operations.

Paper	I. DevOps: Walking the Shadowy Bridge from Development Success to Information Systems Success	II. The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations	III. The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations	IV. From Loose Coupling to Fusion: How Does DevOps Operationalize Digital Business Strategies and Contribute to Digital Innovation?
Aspect	Qualitative exploration of the problem domain. Focus on the relation of DevOps and success. Theoretical grounding.	Qualitative exploration of the problem domain. Focus on factors influencing DevOps implementations in large-scale organizations, and its relation to IS success.	Qualitative exploration of the problem domain. Focus on initially introducing DevOps in large-scale organizations, and its relation to digitalization.	Qualitative exploration of the problem domain. Focus on how DevOps implementations change, and on DevOps as a means to implement digital business strategies and to reconcile business/IT alignment through fusion of functions.
Theme	"Sensitize"	"Implement"	"Initialize"	"Adopt"
Role of theory	Inductively developing theory (that is generalizable across settings) from case study. Identify relation to and integrate with ISSM.	Inductively developing theory (that is generalizable across settings) from case study. Integrate with ISSM: DevOps a means to reconcile different approaches to measure IS success. Suggest the "DevOps continuum" with characteristics and flavors influencing how DevOps is implemented in large-scale organizations.	Inductively developing theory (that is generalizable across settings) from case study. Suggest the "DevOps funnel" with the factors most dominant to initially introduce DevOps. Unfold relation of DevOps with digitalization: DevOps an antecedent for digitalization in large-scale organizations.	Inductively developing theory (that is generalizable across settings) from case study. Integrate with ISSM: DevOps a means to reconcile different approaches to measure IS success. Suggest the "DevOps continuum" with characteristics and flavors influencing how DevOps is implemented in large-scale organizations, and mechanics of their causal structure leading to changing DevOps implementations. Integrate with business/IT-alignment and digital business strategy: DevOps a vehicle to operationalize digital business strategies through fusion of functions and leveraging digital infrastructure to digitally innovate to maximize IS success.

Table 2-3. Focus of each paper.

The paper highlights the relation of DevOps and success, specifically the IS success model (ISSM) (DeLone and McLean, 1992). As part of this work, I also started the conversation with the research community of systems development (and DevOps).

Guided by the overall research question, paper II continues the exploration of DevOps. As the study explored an emerging phenomenon and aimed to build novel theory, I followed a qualitative case research approach. I focused on shedding light on factors influencing existing DevOps implementations in large-scale organizations and its relation to success. Based on data, I explored a variety of characteristics, and how these characteristics influence a DevOps implementation in a different context. I uncovered that DevOps implementations are gradual transitions on a "DevOps continuum".

Paper III aimed to explore factors important for large-scale organizations to initially introduce DevOps. I designed a two-staged research of systematic literature review and multiple-case study. To focus on publications that combine high level of scientific rigor in IS research with strong relevance in the underlying domain of IS, I built upon an existing study that provided an inventory of DevOps literature (Sharp and Babb, 2018). As a result of the systematic literature review, I created a concept matrix with its concepts serving as a starting point for stage two of that study. Starting with these provisional codes, as part of the second stage, I applied process coding to analyze data from cases, and to theorize the "DevOps funnel".

Paper IV adds more data and expands the lens of paper II to the *dynamics* of DevOps implementations in large-scale organizations, and studies in-depth the relation of DevOps with digitalization – evidence for its existence delivered paper III. With business and IT roles being an explicit part of an autonomous DevOps team, the integration of organizational functions may even go beyond "business-IT alignment" (Gerow *et al.*, 2014; Luftman *et al.*, 1999; Reynolds and Yetton, 2015), and may achieve the sought-for fusion of business and IT (Bharadwaj et al., 2013; Henry Lucas et al., 2013) to digitally innovate and to maximize IS success. This journal submission leverages two views on the origins of knowledge: empirical evidence grounded on representative, generalizable facts of reality, based on positivist case research, and mechanisms acting in context leading to causal, observable (empirical) outcome, based on critical realist case research.

2.4 Research Design

My research draws attention to an emerging phenomenon that is still not sufficiently addressed in literature. An appropriate research design was chosen to answer the research question(s), see Table 2-4 for an overview of the research design per paper.

2.4.1 Multiple-case Study Research

I followed an inductive, qualitative case-study research to explore the emerging DevOps phenomenon and to generate new theory from case study (Eisenhardt, 1989; Eisenhardt and Graebner, 2007; Yin, 2018). A multiple-case study is considered an appropriate method to explore a contemporary phenomenon within its real-life context (Benbasat *et al.*, 1987; Dubé and Paré, 2003; Yin, 2018) to build and extend concepts and theory (Dubé and Paré, 2003; Yin, 2018).

I relied on theoretical sampling, the process of data collection for generating theory, for deciding what data to collect next and where to find them, in order to develop theory as it emerges (Eisenhardt, 1989; Glaser and Strauss, 1967), to choose cases for theoretical reasons to replicate previous cases and to assert and extend emergent theory (Eisenhardt, 1989), until closure and theoretical saturation is reached (Dubé and Paré, 2003; Eisenhardt, 1989; Yin, 2018), with cases expecting to strongly implement DevOps (case group A) or implementing DevOps moderately (case group B), thus examining contrasting (rival) cases (Dubé and Paré, 2003; Eisenhardt, 1989; Yin, 2018).

Thoroughly assessed case sites candidates were large-scale, international organizations, see Figure 4-1, Table 5-1 and Table 6-1. This research expected them to have established structures and processes, with many persons across organizational functions involved in development and delivery of the IS, and that they go through a thorough decision-making process in complex organizational setups, before they introduced DevOps (Dikert *et al.*, 2016; Dingsøyr *et al.*, 2023; Lindvall *et al.*, 2004).

Two case sites were already part of the professional network, I contacted all case sites in purpose of this research. Interviews were the main data source, with initial entries recommending other potential informants ("snowball sampling", (Eisenhardt, 1989)). Referrals were pre-examined for applicability for the study and were asked for participation (Dubé and Paré, 2003; Eisenhardt and Graebner, 2007). I gave full anonymity and transparency to the research participants, for example, by providing thorough information about the study (Yin, 2018). During the course of incrementally creating comprehensive writeups for the cases, and recognizing that organizations may have multiple DevOps implementations, the unit of analysis moved from the organization to teams.

Paper	I. DevOps: Walking the Shadowy Bridge from Development Success to Information Systems Success	II. The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations	III. The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations	IV. From Loose Coupling to Fusion: How Does DevOps Operationalize Digital Business Strategies and Contribute to Digital Innovation?
Research design	Background literature review. Multiple-case study (preflight with pilot cases). (Yin, 2018)	Exploratory, inductive, embedded multiple-case study. (Yin, 2018)	Systematic literature review. Exploratory, inductive, embedded multiple-case study. (Levy and Ellis, 2006; Webster and Watson, 2002; Yin, 2018)	Exploratory, inductive, embedded multiple-case study. (Yin, 2018)
Case topology	Cases in two rival, revelatory groups, group A and group B. Cases from NASDAQ-100, DAX40, or similar.	Four cases, two in group A, two in group B. Cases across segments (software, streaming, industry, bank), "digital first" and incumbents, from Europe and USA. Large-scale organizations.	Seven cases. Cases across segments (online products, manufacturing, insurance, software, banking), "digital first" and incumbents, from Europe and USA. Large-scale organizations.	Eight cases. Cases across segments (streaming, manufacturing, software, banking, insurance), "digital first" and incumbents, from Europe and USA. Large-scale organizations.
Data collection	Assessing case site candidates, and piloting semi-structured, open ended interviews in 12 different firms over a period of ten months. Preflight with pilot cases. (Brinkmann and Kvale, 2014; Yin, 2018)	Semi-structured, open ended interviews, with 17 key informants. Participant observation, project documentation, public tech blogs, presentations. (Brinkmann and Kvale, 2014; Yin, 2018)	Semi-structured, open ended interviews, with 26 key informants. Project documentation, public tech blogs, presentations. Partly reused data from Paper II. (Brinkmann and Kvale, 2014; Yin, 2018)	Semi-structured, open ended interviews, with 30 key informants. Project documentation, public tech blogs, presentations. Partly reused data from Paper II. and III. More/other data. (Brinkmann and Kvale, 2014; Yin, 2018)
Data analysis	Preflight, with eclectic coding, as a "first draft" coding. Explore applicable coding methods. (Saldaña, 2016)	Influenced by GTM coding techniques/families, applied initial coding, pattern coding, axial coding, theoretical coding. (Glaser, 1978; Glaser and Strauss, 1967; Miles *et al.*, 2020; Saldaña, 2016)	Open coding, theoretical coding, provisional coding, process coding. Provisional codes from literature review served as input for process coding, grounded in data. (Levy and Ellis, 2006; Miles *et al.*, 2020; Saldaña, 2016; Webster and Watson, 2002)	Influenced by GTM coding techniques /families, applied initial/pattern/ axial/theoretical/process coding. (Glaser, 1978; Glaser and Strauss, 1967; Miles *et al.*, 2020; Saldaña, 2016) Applied principles for conducting critical case study research, influenced by the CMO framework. Identified key events resulting in timelines; applied retroduction to identify mechanisms. Empirical and theoretical elaboration. (Bhaskar, 2008; Danermark *et al.*, 2019; Mingers *et al.*, 2013; Pawson and Tilley, 1997; Wynn and Williams, 2012)

Table 2-4. Overview of the research design per paper (excerpt).

2.4.1.1 Positivist Case Study Research

Positivist case study researchers aim to discover causal relationships to establish causes and effects based on observations with theory being empirically tested and thus verified or falsified (Benbasat *et al.*, 1987; Sarker *et al.*, 2018; Urquhart, 2013; Wynn and Williams, 2020). I followed suggestions for scientific rigor for explorative positivist case study research (Dubé and Paré, 2003), for example designing a multiple-case study, stating clear research questions and the unit of analysis, and executing pilot cases with pilot interviews with 12 different firms over a period of ten months, while shaping relevant lines of questions and detailing data collection plans, and probing applicable coding methods, for example via eclectic coding (Saldaña, 2016)) ("design area"), executing data triangulation and multiple data collection methods ("data collection area"), and utilizing quotes for evidence and comparison with extant literature while demonstrating the chain of evidence ("data analysis area"). I developed concepts and their relationships, and theory emerged in sequential within-case analysis, before I looked for similar concepts and relationships across multiple cases, to build, refine, and assert the emerging theory and to use theory to generalize from case study (Dubé and Paré, 2003; Eisenhardt, 1989). Applying first cycle (initial coding) and second cycle coding methods (pattern coding, axial coding and theoretical coding) (Saldaña, 2016), I synthesized "flavors" and "characteristics" and theorized the "DevOps continuum" (Eisenhardt and Graebner, 2007), see Figure 4-2.

I applied open coding (Miles *et al.*, 2020; Saldaña, 2016) in conjunction with an executed systematic literature review (Webster and Watson, 2002) to identify the dominant concepts of each publication relevant for the study. I streamed in the dominant concepts into a concept matrix, see Figure 5-3. I used theoretical coding (Glaser, 1978; Saldaña, 2016) as a second cycle coding method to model the integration of codes while integrating concepts from the background literature. I applied provisional coding (Saldaña, 2016) as an exploratory coding method to build theory from case studies. The data analysis was inspired by the coding family "process" (Glaser, 1978; Saldaña, 2016) to inductively evidence factors important to initially introduce DevOps in large organizations (Van de Ven and Poole, 1990), see Figure 5-4. As a result, I theorized the DevOps funnel with the three dominant factors "success", "actors" and "practices" to initially introduce DevOps in organizations (cf. Table 5-3), and suggested the visual representation of an enterprise sandglass hinting to the top-down and bottom-up interaction paths between management and employees, see Figure 5-1.

2.4.1.2 Critical Case Study Research

Based on empirical data and qualitative analysis, I unveiled the DevOps continuum with gradual transitions of DevOps implementations of environmental characteristics with their respective flavors (paper II). In conjunction to and in extension of this *static* lens, I theorized the *dynamics* behind changing DevOps implementations (paper IV). Henfridsson and Bygstad (2013) argue that applying positivist approaches while studying dynamics of strategic evolutions, for example of digital infrastructure, and its relation to business success, may lead to too overly simplistic explanations, and that critical realist approaches provide a better philosophical underpinning for research on change of (strategic) use of IS, aligned with business needs, and using IT for competitive advantage (Morton, 2006). In line with their suggestion, my analysis was influenced by critical realist approaches (Sayer, 2000). This helped to explore the complex causal structure behind (changing) DevOps implementations. From a critical realist perspective, a phenomenon is explained by a causal structure (Bhaskar, 2008). A mechanism as part of the structure explains an empirical outcome (Bygstad *et al.*, 2016), where the reality has three stratified domains (Mukumbang, 2023; Zachariadis *et al.*, 2013): the *real* consists of structures of entities that exist independent of our perception of them, with capacities for behavior called *mechanism*. These mechanisms generate events in the level of the *actual*. These events may or may not be observed. The *empirical* contains the subset of the actual that is observed (Volkoff and Strong, 2013). My analysis makes use of Pawson and Tilley's (1997) proposal for realist evaluation of a given phenomenon to find ways of identifying, articulating, testing, and refining conjectured CMO configurations (p. 77). According to their generic context-mechanism-outcome (CMO) structure (Pawson and Tilley, 1997), generative contingent mechanisms may generate observable events that depend on context (Bhaskar, 2008; Henfridsson and Bygstad, 2013; Pawson and Tilley, 1997).

I followed suggestions for scientific rigor for critical realist case study research (Wynn and Williams, 2012), for example the (a) explication of events and structure/context, (b) applying retroduction, and (c) empirical corroboration, see Table 6-5: (a) I studied change over time (Poole *et al.*, 2000; Van de Ven *et al.*, 2000), by applying a process coding procedure (Miles *et al.*, 2020; Saldaña, 2016), see Figure

6-8, streaming identified key events (observed in the empirical) into chronological timelines for each case (Figure 6-9 to Figure 6-16). I expatiated on the context of the CMO structure (Pawson and Tilley, 1997). I used my analytical finding of the DevOps continuum for the *context* with the settings of teams served as the structure, with the team being the unit of analysis and the point of departure (Delanda, 2006), and theorized "success" as the "*outcome* of interest" (Iannacci *et al.*, 2021), see Figure 6-7. (b) I uncovered candidate *mechanisms* through *retroduction*, a creative process (Reichertz, 2007), a "thought trial" (Weick, 1989), working backward from observed events (Bygstad *et al.*, 2016; Wynn and Williams, 2020) (in the "empirical real") to underlying mechanisms, that could logically have produced those events (Danermark *et al.*, 2019; Sayer, 2000; Volkoff and Strong, 2013). Thus, I moved "from experiences in the empirical domain to possible structures or mechanisms in the real domain." (Mingers *et al.*, 2013), taking an empirical observation and hypothesizing a mechanism that might explain that particular outcome (Henfridsson and Bygstad, 2013; Mingers *et al.*, 2013; Sayer, 2000). Figure 6-8 shows a visual representation of key events at MovieStream together with illustrations of the coding process. (c) I challenged the emergent understanding in view of other mechanisms (Sayer, 2000). Accompanied by integration with literature, out of an array of candidate mechanisms, I compared their explanatory power and identified "effectiveness adaptation" and "efficiency adaptation" as the most powerful to explain the causal structure behind DevOps implementations.

2.4.2 Systematic Literature Review

In addition to respective reviews of extant background literature and aligned with the research design, I conducted a systematic literature review (paper III) (vom Brocke *et al.*, 2015; Levy and Ellis, 2006; Webster and Watson, 2002) to reconcile the existing body of knowledge on DevOps to build upon an existing study that offered an inventory of DevOps publications of the IS community (Sharp and Babb, 2018), see Figure 5-2. After screening, filtering and testing applicability for the specific study (with its design and research question), combined with a forward/backward search, I streamed an array of 35 publications into the resulting concept matrix. The assembled concept matrix, see Figure 5-3, with its (provisional) codes served as a starting point for the subsequent stage of the study, a multiple-case study, to apply process coding (Miles *et al.*, 2020; Saldaña, 2016), resulting in the suggestion of the "DevOps funnel", see Figure 5-1.

2.4.3 Philosophical Perspectives

As rendered above, the fourth paper adds a critical realist view to examine process and conditions of causality in the empirical (Zachariadis *et al.*, 2013) to leverage a configurational perspective (Henfridsson and Bygstad, 2013) to gain powerful explanations of the dynamics behind changing DevOps implementations, see Table 2-5 for an overview of epistemological stances of each paper. According to Sarker *et al.* (2018), boundaries of (epistemological) stances are not set in stone, unless their application is "thoughtful" and it provides guidance concerning standards by which the scientific inquiry can be judged. As documented above, this research carefully aligned with existing guidance including those with regard to epistemological questions, and documents how the theory was constructed, how the scientific knowledge was acquired, and what criteria were applied to judge the soundness and rigor (Gregor, 2017; Sarker *et al.*, 2013, 2018).

Paper	I. DevOps: Walking the Shadowy Bridge from Development Success to Information Systems Success	II. The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations	III. The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations	IV. From Loose Coupling to Fusion: How Does DevOps Operationalize Digital Business Strategies and Contribute to Digital Innovation?
Epistemological stance	Positivist case research with inductive theory building to explore a contemporary phenomenon (Benbasat *et al.*, 1987; Dubé and Paré, 2003; Eisenhardt, 1989; Sarker *et al.*, 2018; Yin, 2018).	Positivist case research with inductive theory building to explore a contemporary phenomenon (Benbasat *et al.*, 1987; Dubé and Paré, 2003; Eisenhardt, 1989; Sarker *et al.*, 2018; Yin, 2018).	Positivist case research with inductive theory building to explore a contemporary phenomenon (Benbasat *et al.*, 1987; Dubé and Paré, 2003; Eisenhardt, 1989; Sarker *et al.*, 2018; Yin, 2018).	Positivist case research with inductive theory building to explore a contemporary phenomenon (Benbasat *et al.*, 1987; Dubé and Paré, 2003; Eisenhardt, 1989; Sarker *et al.*, 2018; Yin, 2018). Critical realist case research (Mingers *et al.*, 2013; Sarker *et al.*, 2018; Wynn and Williams, 2012, 2020) to identify mechanisms in a context-mechanism-outcome structure (Danermark *et al.*, 2019; Pawson and Tilley, 1997; Sayer, 2000).

Table 2-5. Overview of the epistemological stances, for each paper.

2.5 Discussion

For scaffolding, and convenience for the reader, the following discussion is aligned with papers' themes (see Table 2-3), from sensitive over initialize and implement to adopt, to offer an alternative structure for reading and zooming in to the narrower sliced papers along the research (sub)questions of why and how do large-scale organizations operationalize DevOps, see Figure 2-4.

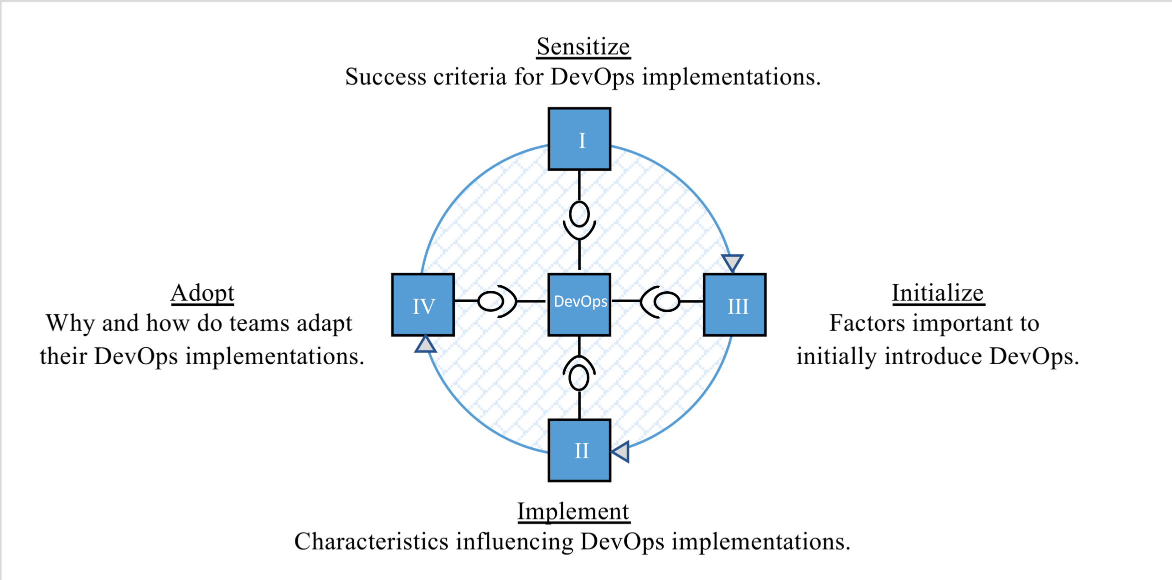

Figure 2-4. A composition of four papers answers the overarching research question.

This dissertation discusses the presented results of research on DevOps to uncover the why and how behind DevOps implementations in large-scale organizations. As detailed above, the papers with their respective research (sub)questions shed light on specific parts while breaking down the overall research question. While the first publication is the grounding paper that draws attention on the *why* of DevOps implementations (it identifies the relation to the ISSM), papers II-IV cover and detail both, the *why* and the *how*, with paper II shedding light on DevOps as a means to reconcile success, a vehicle to bridge effectiveness and efficiency, and a position on the DevOps continuum, with paper III shedding light on factors important to initially introduce DevOps, and its relation to digitalization, and paper IV shedding in-depth light on the static and dynamic view, with DevOps implementations being a result of transitions on the DevOps continuum, and how DevOps contributes to the implementation of organizational goals. One example goal is digitalization by integrating teams across functions and leveraging digital technology to digitally innovate to maximize IS success.

2.5.1 Sensitize DevOps

Prior studies have proposed a diverse and varied set of ISD success conceptualizations and measures, for example staying within scope, time and cost requirements (Chow and Cao, 2008; Lee and Xia, 2010), software functionality and process performance (Recker *et al.*, 2017). IS success similarly can be defined in multiple ways, for example maintenance time, mean time between failures (MTBF) and mean time to repair (MTTR) (Dekleva, 1992). The ISSM with its user-centric point of departure is a logical starting point to conceptualize the operational aspects of the IS, and to integrate with DevOps. A holistic approach to AISD and thus a closer integration of IS development and IS operations (namely DevOps) is needed, to maximize the probability of overall success. Figure 2-5 represents a (preliminary) model with both functions, IS development and IS operations, having their respective success criteria. These independent variables impact success, either ISD success or IS success, contributing to overall success. Evidence exists that DevOps highlights customer satisfaction (Bhattacherjee, 2001; Fowler and Highsmith, 2001; Recker et al., 2017) that can be integrated with "user satisfaction" of the ISSM (DeLone and McLean, 1992). DevOps replaces more siloed measures for success, either from the developmental side or from the operational side.

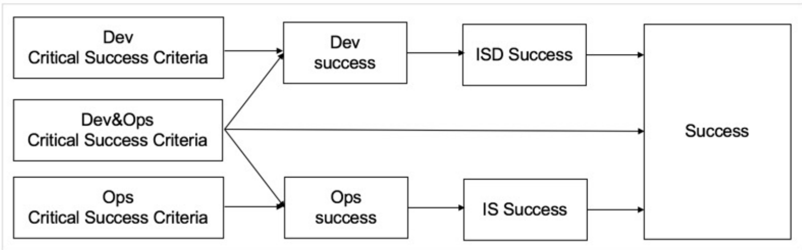

Figure 2-5. A conceptualization of DevOps from a success perspective.

As evidence suggests, DevOps emphasizes the sociotechnical nature of IS (Sarker et al., 2019), and aims to reconcile the plethora of different approaches trying to measure IS success and ISD success. Success is the outcome of interest of DevOps implementations (Iannacci *et al.*, 2021).

2.5.2 Initialize DevOps

Evidenced by data, the factors *success*, *organizational actors*, and *practices*, are the most dominant for organizations to initially operationalize DevOps, confirming the importance of actors, practices, and outcomes in the thematic advances in knowledge in IS research (Matook *et al.*, 2021). Through shared practices (Hemon-Hildgen, Rowe, *et al.*, 2020; Wiedemann *et al.*, 2020) across actors (Greenwood *et al.*, 2020; Hemon-Hildgen, Lyonnet, *et al.*, 2020; Osmundsen and Bygstad, 2021) from different organizational teams and levels (Osmundsen and Bygstad, 2021), DevOps positively impacts success (DeLone and McLean, 1992; Petter *et al.*, 2013). All factors have a top-down (with the management as the point of departure) and bottom-up (with the team as the point of departure) view. Following the "top-down" path, success is articulated to the team as a goal, for example "digitalization" (Burgelman, 1983; Greenwood *et al.*, 2020; Nambisan *et al.*, 2020). DevOps is introduced by the team with its actors to achieve its own goals ("bottom-up"), and organizational goals ("top-down"). DevOps links an organization's digitalization effort (the "what"), defined by management (Furr *et al.*, 2022), to the "how", how the team develops and delivers the IS product (Gall and Pigni, 2021). Success informed by digital innovation (Nambisan *et al.*, 2020) is based on and evolves the installed base of digital infrastructure (Henfridsson and Bygstad, 2013).

The "top-down" and "bottom-up" directions form a funnel, see Figure 2-6. Its graphical representation looks like an enterprise sandglass rotated counterclockwise by 90 degrees, and draws on work of Osmundsen and Bygstad (2021) who identified interaction cycles between management and employees[4]. According to interview partners, digitalization with existing long-term inventory is recognized to be a challenge ("brownfield digitalization", in an informant's voice). Incumbent firms with a "different set of liabilities of aging and bigness" (Van de Ven *et al.*, 2000) may address this challenge by technically decoupling IS products based on layered modular architectures (Yoo *et al.*, 2010), to achieve digitalization goals specific for their organization with its actors (Furr *et al.*, 2022). As informants reported, DevOps is more difficult to introduce when older technology (i.e., the installed base of digital infrastructure) is used (Bygstad, 2017; Yoo, 2013), see Table 5-4 for an overview of digital technology per case.

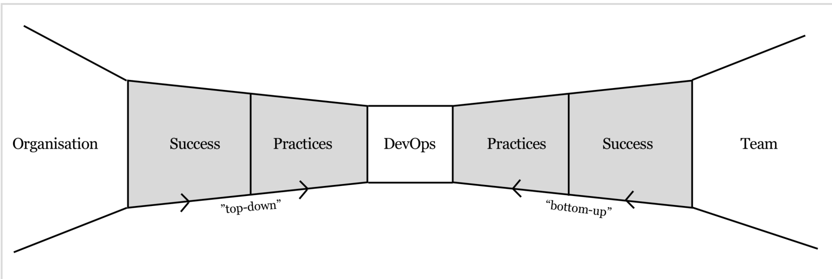

Figure 2-6. DevOps funnel: Success and practices across actors, with interaction paths top-down and bottom-up (summary).

2.5.3 Implement DevOps

Data indicate that *success* is a characteristic of a team's *inner environment*. It relates to the contextual reasons why DevOps is implemented at all. Other characteristics are *organizational standards*, *dealing*

[4] The scope of their work was also not the change/adaption of methods, rather unfolding the interaction cycles.

with change, *applied practices*, and, as part of the *outer environment*, *market type*, *product type*, and *digital technology*, see Table 2-6.

<table>
<tr><th colspan="2">Characteristics</th><th>Short descriptions</th></tr>
<tr><td rowspan="4">Inner environment</td><td>Organizational standards</td><td>Rules and norms the team must align with.</td></tr>
<tr><td>Dealing with change</td><td>How and when the team responds to changes.</td></tr>
<tr><td>Applied practices</td><td>Practices applied by the team.</td></tr>
<tr><td>Success</td><td>Success, and its measurement, the team aims to achieve to fulfill its task.</td></tr>
<tr><td rowspan="3">Outer environment</td><td>Market type</td><td>Attributes of the demand of the IS product the team provides, in its market.</td></tr>
<tr><td>Product type</td><td>Attributes and identifying features or core affordances of the IS product the team provides.</td></tr>
<tr><td>Digital technology</td><td>Digital technical means the team utilizes to develop and deliver the IS product.</td></tr>
</table>

Table 2-6. Characteristics of teams' DevOps implementations, with short descriptions.

The sum of characteristics, with typical *flavors* (instantiations), relates to how a team executes DevOps in a specific context. Because of different flavors of environmental characteristics, even in one organization, different DevOps implementations may exist (Table 6-6 lists examples of characteristics of the inner environment and Table 6-7 for examples of the outer environment). Variations of DevOps implementations are gradual transitions on the DevOps continuum between two edges, siloed (detached) functions on the left, full coalescence of functions on the right with autonomous teams (one organizational function) fully responsible for the IS product and all activities needed to develop and deliver it to the customer (see Figure 2-1), and gradual configurations in the middle, with functions assisting each other while being (partly) immersed in the other function (El Sawy, 2003) and the IT function strategically developing into an innovation partner within the company (Urbach and Ahlemann, 2019). Figure 2-7 presents the DevOps continuum with typical flavors and exemplary forms of collaboration as part of the organizational standards.

Effectiveness (doing the right thing) (DeLone and McLean, 1992; Fowler and Highsmith, 2001) and efficiency (doing the thing right) (Bosch, 2019; Fowler and Highsmith, 2001) define success (Lee and Xia, 2005). Informed by their openness for and willingness for creation of change (Conboy, 2009), teams constantly balance the intertwined concepts of effectiveness and efficiency through DevOps. For example, leveraging modern technology to gain insights from the IS product running on production systems helps the team to optimize its functionality. This interplays with DevOps emphasizing holistic, humanistic goals (leading success flavors on the right side of the continuum, e.g., *customer satisfaction*)

(DeLone and McLean, 1992; Sarker *et al.*, 2019) with the help of holistic instrumental goals: the *delivery performance* defines how fast and often an IS product can be made available to the user, with its subconcepts "lead time" (a measure of time between inception of a change in the IS product and making the change available to the user), "deployment frequency" (a measure of how often new versions of the IS product are made available to the user), and *mean time to repair (MTTR)* (Forsgren, 2018; Sarker *et al.*, 2019). Findings support the existence of the social-technical continuum and its axis of cohesion with a positive synergy between instrumental and humanistic goals in IS (Sarker *et al.*, 2019).

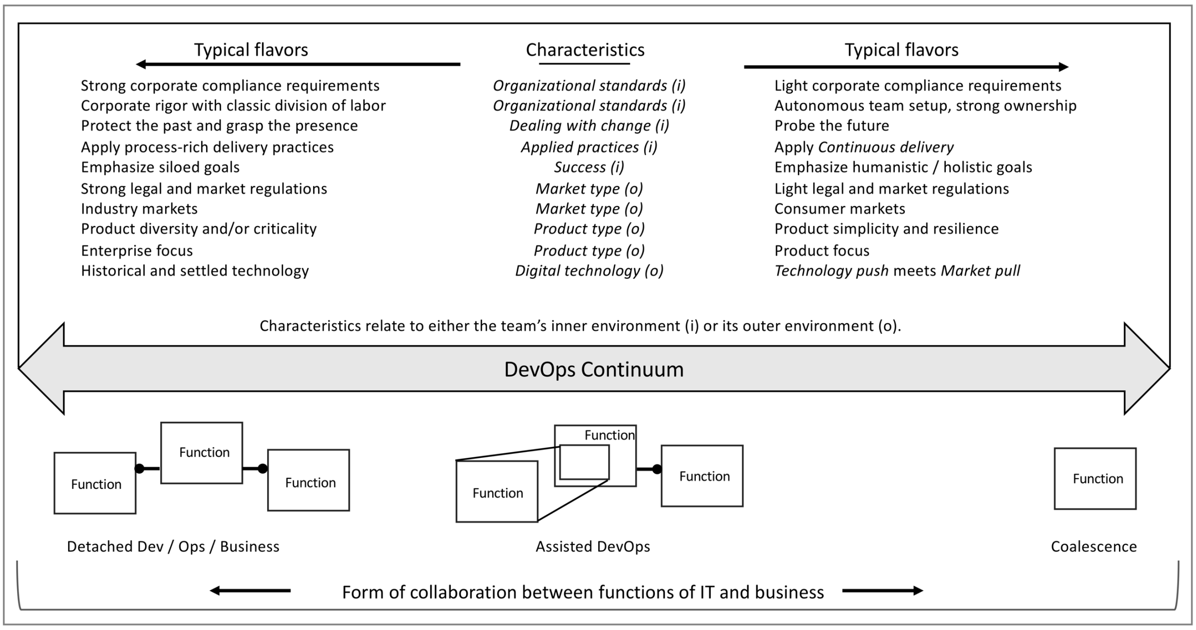

Figure 2-7. Characteristics of teams' DevOps implementations, with typical flavors, can be arranged along a continuum, including different forms of collaboration.

This research and the finding of the configurational factors of the DevOps continuum unwrap the role of context in DevOps (Davison and Martinsons, 2016). For example, a team's DevOps operationalization is different if it provides an enterprise-focused industry product (outer environment) under strong corporate rigor (inner environment) (e.g., from ManuFact, with trains and powerplants), compared with a team that provides a technically decoupled and globally distributed consumer product with a light set of corporate compliance rules (e.g., from MovieStream, with a product for online streaming), or a team, active in a regulated market, that is required to authorize changes in an IS product conformable with the "four eyes" principle (Basle Committee on Banking Supervision, 1997) following a segregation of functions by dual control of assets (e.g., AutoBank, with its financial products).

All these environmental settings lead to different positions of the respective team on the DevOps continuum, and impact how the team defines and measures success. Ambidexterity relies on this configurational context to help an organizational entity to achieve two seemingly conflicting goals (Birkinshaw and Gibson, 2004; Werder and Heckmann, 2019), in this case effectiveness and efficiency. The ambidextrous nature of DevOps has explorative (links to effectiveness) and exploitative (relates to efficiency) elements (Gregory *et al.*, 2015; March, 1991; Sia *et al.*, 2021; Werder and Heckmann, 2019; Xue *et al.*, 2012). Findings confirm the need for ambidexterity in organizations to follow digital business strategies by being able to pursue two separate things (effectiveness and efficiency) at the same time as part of a continuum (Cao *et al.*, 2016; March, 1991; Werder and Heckmann, 2019).

2.5.4 Adopt DevOps

The mechanisms of *effectiveness adaptation* and *efficiency adaptation* are the two most powerful to explain the causal structure of DevOps implementations, see Figure 2-8. Stemming from the concept of *morphing*, they comprise significant changes and profound transformations to the IS product, along with reconfigurations of resources, capabilities, and structures (Rindova and Kotha, 2001).

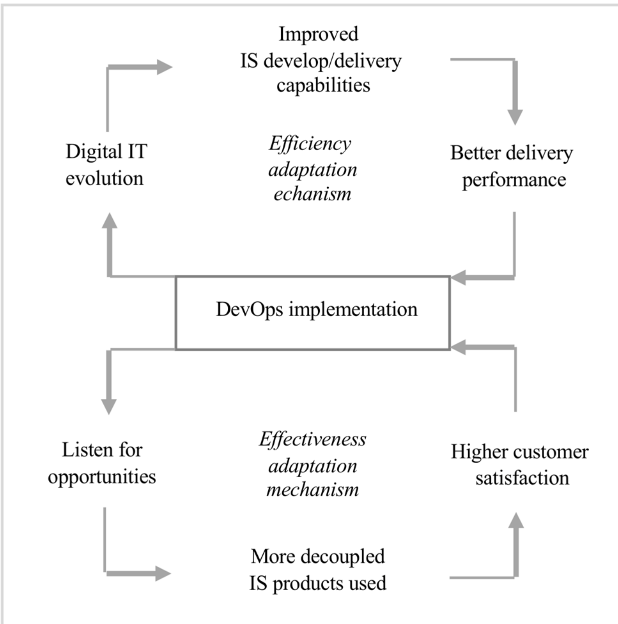

Figure 2-8. Effectiveness adaptation mechanism and efficiency adaptation mechanism.

Once actualized they may change a team's position on the DevOps continuum and thus how the team implements DevOps. Effectiveness adaptation comprises the process by which technically decoupled IS products are created as teams increasingly collaborate across functions of business and IT, listen for opportunities to innovate, and to achieve higher customer satisfaction. Efficiency adaptation describes the process by which teams increasingly collaborate across organizational functions and enhance their IS delivery capabilities as they innovate and utilize new digital infrastructure to achieve better delivery performance, see Table 6-3 for a tabular summary. Figure 6-3 and Figure 6-4 depict examples of the causal structure and how the two mechanisms were actualized (effectiveness adaptation at SoftwareDev and efficiency adaptation at MovieStream).

Both mechanisms are self-reinforced in that they generatively feed on themselves (Bygstad *et al.*, 2016; Huang *et al.*, 2017): they may lead to new IS products and improved develop/deliver capabilities, and a changed context (a new position on the DevOps continuum, thus a new DevOps implementation, including what is seen as success as the outcome of interest (Iannacci *et al.*, 2021), see Figure 2-9) – a self-referential loop.

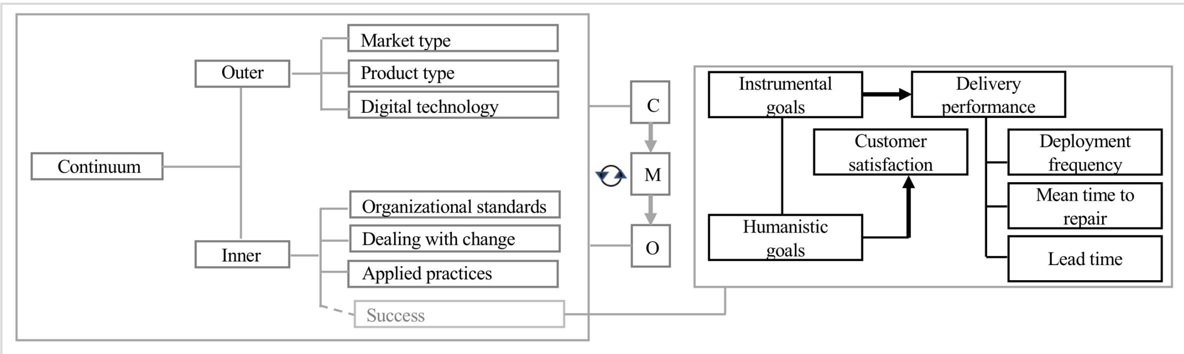

Figure 2-9. The DevOps continuum defines how the team implements DevOps. From a team's perspective, it is its context (C). Actualized mechanisms (M) result in outcome (O): a change in the context (different flavors) and, as part of it, a maximized success (the outcome of interest).

Based on another, more abstract level of theorizing (Berlin, 1950), the composition of both mechanisms balances digital innovation (Nambisan *et al.*, 2020; Yoo *et al.*, 2010) and develop/delivery capabilities (Osmundsen and Bygstad, 2021) to sense and keep in touch with evolving customer needs (Sia *et al.*, 2016) to maximize IS success (DeLone and McLean, 1992). Value creation and digital innovation (effectivity) as well as delivery capability (efficiency) enact the DevOps implementation. To innovate, teams implement digital business strategies (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013) through a composition of configurational characteristics including collaboration/integration of functions and (practices of) leveraging modern technology (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013). A specific (and extreme) form of collaboration is a fusion[5] of business and IT (including development and operations) (Sia *et al.*, 2016, 2021) to an autonomous team, with coalescent functions, that is fully responsible for developmental and operational concerns during the entire lifecycle of the IS product with its activities (see Figure 2-1). Grounded on data, fusion is not the best solution in every context. A different level of coupling of functions can be a better fit in a given situation (Greenwood *et al.*, 2020; Orton *et al.*, 1990), see examples in section 2.5.3 and paper IV, with loose coupling, distinct but connected functions of business and IT, or a tight coupling, setups with distinct and responsive functions of business and IT, for example business or operations immersed in the development function. In its representation, Figure 2-10 synthesizes the characteristics of the DevOps continuum, and the two mechanisms; actualized mechanisms may lead to new flavors (and new DevOps implementations), specifically a different organizational standard for the coupling of functions.

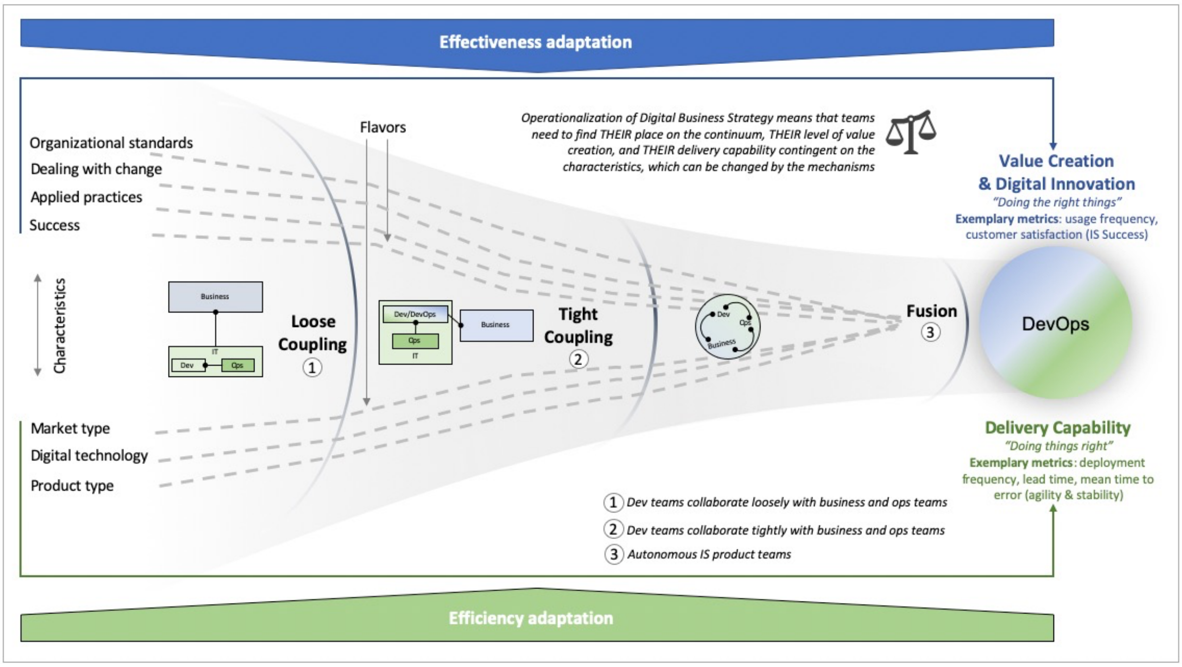

Figure 2-10. A DevOps implementation as a sum of characteristics with their specific flavors at a given time, with actualized mechanisms leading to a change of a team's position on the continuum.

All nuanced forms of collaboration along with different levels of coupling are executed DevOps implementations. Concurring what Bygstad (2017) calls "heavyweight IT" and "lightweight IT", this

---

[5] If we think of business, development and operations as three distinct functions, this could actually be labelled as "double-fusion".

may lead to different DevOps implementations inside one organization, and the underlying ambidextrous structures can be linked with each other, e.g. through organizational standards (Berente and Yoo, 2012; Bygstad, 2017).

2.5.5 Summary

DevOps is a set of continuously improved practices for collaborative work implemented between organizational functions to develop and deliver an IS product toward continuously delivery of valuable outcome for customers (Beck *et al.*, 2001; DeLone and McLean, 1992; Greenwood *et al.*, 2020). Organizations operationalize DevOps to reconcile siloed goals and to maximize overall IS success (DeLone and McLean, 1992), specifically customer satisfaction.

Success, actors and practices are the most dominant factors to initially introduce DevOps. Serving as a funnel, DevOps supports to streamline "top-down" and "bottom-up" interactions between management and teams (Osmundsen and Bygstad, 2021). DevOps is an antecedent for organizations to operationalize organizational goals, for example digitalization, specifically a digital business strategy (through integration of functions and leveraging digital technology) (Bharadwaj *et al.*, 2013), to digitally innovate (Nambisan *et al.*, 2020), to gain and maintain competitive advantages (Urbach and Ahlemann, 2019), to maximize IS success (DeLone and McLean, 1992). Teams find their best spot on the DevOps continuum, a set of environmental factors with respective flavors (instantiations) at a given time. Contextual flavors, from both the team's inner and outer environment, enact a specific DevOps implementation.

Reconciling "business-IT alignment" (Gerow *et al.*, 2014; Luftman *et al.*, 1999; Reynolds and Yetton, 2015), in specific settings DevOps supersedes "alignment" and may result in fused (coalescent), autonomous teams. Researchers agree that digital infrastructure can provide competitive advantage that influence corporate success (Kahre *et al.*, 2017; Peppard and Ward, 2004). With fused teams, the digital infrastructure is defining the business strategy rather than supporting it. Depending on the context of the team-in-focus and its (inner and outer) environment, loosely coupled (e.g., seeded in corporate norms of strict division of labor, work regulations and necessity of 24x7 monitoring) and tightly coupled (e.g., a centralized technical platform must be used) can also be part of "valid" DevOps implementations.

DevOps balances effectiveness (to innovate and to deliver the right IS product to optimize customer satisfaction) and efficiency (to deliver the IS product right while leveraging delivery capabilities), two intertwined things pursued at the same time with the opposing elements form part of the same continuum (Cao *et al.*, 2016; Lee and Xia, 2005; March, 1991; Werder and Heckmann, 2019). Two mechanisms, effectiveness adaptation and efficiency adaptation, are causal forces that, once actualized, may lead to new forms of collaborations and moves on the DevOps continuum and thus to new DevOps implementations.

Influenced by Woodard *et al.* (2013) and their report of design moves in digital business strategy, effectiveness and efficiency can be related with each other as two dimensions on a map. The "DevOps Contingent Map" in Figure 2-11 represents a conceptual matrix offering an illustration of the coupling between functions as the team's form of collaboration at a point in time, with possible moves (i.e., mechanisms of effectiveness adaptation and efficiency adaptation actualized) from one form to another. Since many nuanced forms of collaboration exist as reported above, this map provides a simplified "intuitive" illustration (cf. Woodard *et al.* (2013)). Effectiveness and efficiency are intertwined, and the associated mechanisms compose the contingent forces which may lead to a changed DevOps implementation, leading to the "best-fit" for the team with its configurational environment and respective goals at a given time. Better develop/delivery capabilities improve the team's delivery performance and their ability to use and combine the installed digital technology to decouple IS products to maximize success (Osmundsen and Bygstad, 2021; Rindova and Kotha, 2001; Tarafdar and Gordon, 2007). The ambidextrous and interweaving operationalizations of effectiveness and efficiency constitute the space of rationality (Dybå and Dingsøyr, 2009), i.e. they guide the reasoning of the team to provide decoupled IS products and their capabilities to deliver them to the customer. Ambidexterity and integration of functions affect innovation (Tarafdar and Gordon, 2007).

The visualized quadrants contain example structures for coupling: AISD setups (Dingsøyr *et al.*, 2012; Siau *et al.*, 2022) for tightly coupled forms of collaborations in quadrants B and C, technically/functionally structured (loosely coupled) layout of functions in quadrant A, and a fused team (function) in quadrant D[6]. While all examples illustrate "valid" and executed DevOps implementations, this alternative perspective opposes existing research on necessity of skill sets (Hemon-Hildgen, Lyonnet, *et al.*, 2020; Wiedemann and Wiesche, 2018) and importance of maturity (Gupta *et al.*, 2017; Rafi *et al.*, 2021) moving toward DevOps.

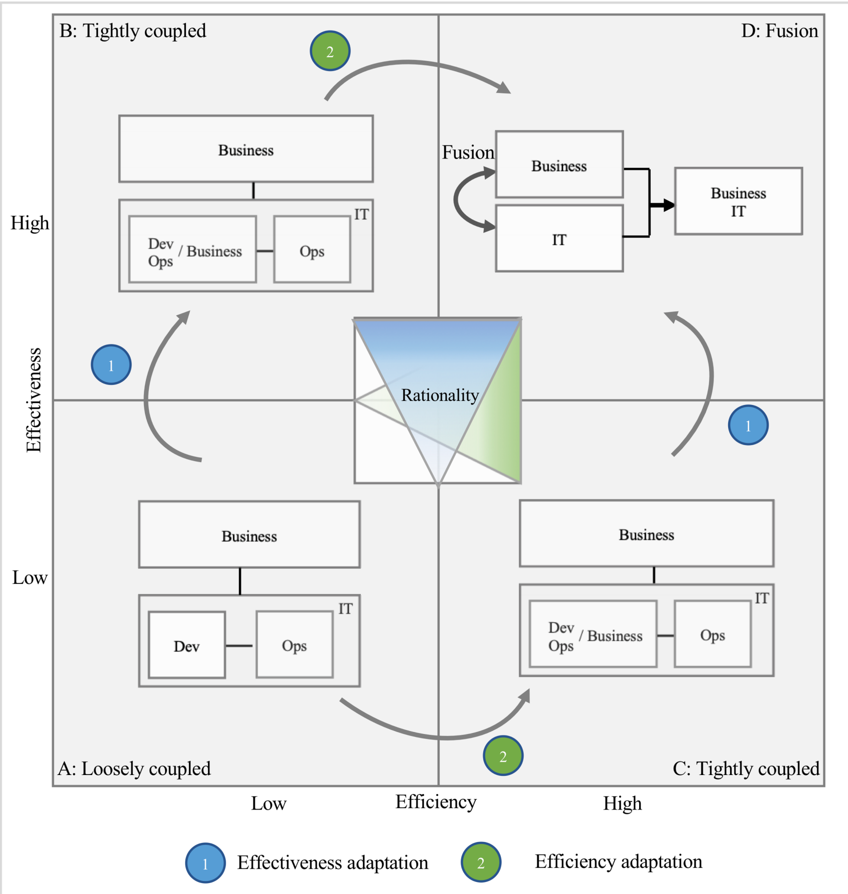

Figure 2-11. The "DevOps Contingent Map": Coupling in relation to effectiveness and efficiency, with examples. Mechanisms lead to new forms of collaboration between functions.

2.6 Contribution

Based on the key contributions of each paper, see Table 2-7 for a summary, this scientific inquiry presents three core contributions. (1) While suggesting the "DevOps funnel" and the "DevOps continuum", it contributes novel and nuanced theoretical grounding of the emerging DevOps phenomenon. It adds systematic and foundational knowledge about DevOps and the how and why of its implementations in large-scale organizations. (2) This research draws upon prior studies and

[6] The integration of business and IT to one fused function was visually accentuated in quadrant D of the figure.

complements and expands existing concepts and models. For example, this research reconciles DevOps with the ISSM (DeLone and McLean, 1992) to align developmental and operational factors for providing IS products, while responding to the authors' call to further refine the ISSM and to explore its relationships with ISD (DeLone and McLean 2016, p. 93), and it reconciles DevOps with "business-IT alignment" toward the sought-for fusion of business and IT to operationalize digital business strategies (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013; El Sawy, 2003), while leveraging digital infrastructure (Henfridsson and Bygstad, 2013; Osmundsen and Bygstad, 2021; Tilson *et al.*, 2010), to foster digital innovation (Nambisan *et al.*, 2020; Yoo *et al.*, 2010). Thus, it addresses the call of Kahre *et al.* (2017) to provide evidence how organizations implement digital business strategies. (3) Stemming from the scientific approaches used and the domains covered, this research significantly contributes to advancing IS research in three ways. First, according to Henfridsson and Bygstad (2013), studies of evolution of complex organizational phenomena as part of established streams of IS research have a voiced paucity of empirical research underpinned by critical realist approaches, and only few early examples exist (e.g., Volkoff and Strong (2013)). To understand the complex dynamics of DevOps implementations (including integration of organizational functions), I applied a critical realist case research (Mingers *et al.*, 2013; Wynn and Williams, 2012, 2020). This way I addressed the voiced paucity identified by Henfridsson and Bygstad (2013) and added an interesting study to the IS field utilizing this epistemological stance while complementing literature on DevOps using other epistemological stances (e.g., Wiedemann *et al.* (2020)). Second, in a similar vein, this research contributes to a broader and more nuanced application of the CMO framework of Pawson and Tilley (1997) as part of their guidance for realist evaluation. While this inquiry resonates with the generic CMO framework and identifies mechanisms acting in context leading to causal outcome (Bygstad *et al.*, 2016; Henfridsson and Bygstad, 2013; Meyer *et al.*, 2007; Pawson and Tilley, 1997), and is aligned with critical realist case research guidance (Mingers *et al.*, 2013; Williams and Karahanna, 2013; Wynn and Williams, 2012, 2020), this work is the first that concretizes the general CMO framework by the special case of a context change during actualization. It theorizes the changed context (in the moment a mechanism is actualized) as the context (in the CMO sense), not the outcome of causation (that is the success), since success is the outcome of interest from a team's perspective, however both (the context and as part of it the success; success is an environmental characteristic on the DevOps continuum) change. Third, this foundational research with its empirical and theoretical elaboration connects the distinct research domains of systems development and digitalization, links their respective sub-communities (e.g., DevOps, success, digital innovation, digital business strategy), and provides a mutually beneficial perspective on the substantive role of DevOps (in digitalization) along with fruitful discussions and research in and across lines of research.

Paper	I. DevOps: Walking the Shadowy Bridge from Development Success to Information Success	II. The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations	III. The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations	IV. From Loose Coupling to Fusion: How Does DevOps Operationalize Digital Business Strategies and Contribute to Digital Innovation? (ISR)
Contribution	Preliminary DevOps conceptualization, with a suggested model of DevOps from a success perspective (ISSM). Theoretical grounding, and opening paths for subsequent research.	Empirically theorized the DevOps continuum, with inner/outer environment and typical flavors (instantiations), that influences how DevOps is implemented in large-scale organizations. Unpacked DevOps as a link between effectiveness and efficiency. Integrated DevOps with the ISSM and identified a spin from instrumental/ siloed goals to humanistic/ holistic goals.	Reconciled existing knowledge on DevOps, and empirically theorized the "DevOps funnel". Identified relationship between DevOps and digitalization, and linked the discourses of the respective allied but largely disjoint research streams. Revealed DevOps as a form of collaboration that can lead to fused functions of business and IT (development, operations), and thus reconciles "business-IT alignment".	Empirically theorized the DevOps continuum: A DevOps implementation is a sum of characteristics with their specific flavors at a given time. It is a position on the DevOps continuum. Uncovered success as both a characteristic on the continuum and the "outcome of interest" for a changed DevOps implementation. Actualized mechanisms of effectiveness (driving innovation, "doing the right things") and efficiency (driving delivery capability, "doing things right") lead to a change of the team's position on the continuum. DevOps as a form of collaboration can lead to tightly coupled (fused) functions of business, development, operations, with leveraged digital resources, the sought-for implementation of "digital business strategies". Applying critical realist case research approaches to explore dynamic evolutions in complex organizational settings resulting in business success. Adding a nuanced application of the generic CMO framework, with the context changed during actualization of mechanisms.

Table 2-7. Overview of contribution of each paper.

This dissertation provides several relevant and interesting practical implications. Based on the presented findings, practitioners can better understand reasoning, configurational factors, and dynamics behind DevOps initiatives. While investigating international, large firms across segments, including both “digital first” and incumbent (traditional, industrial-native) firms, this work offers nuanced understanding on the why and how of DevOps implementations, and spots attention to DevOps as a mash of configurational settings including the form of organizational collaboration, with gradual transitions on the DevOps continuum. Challenging prior studies which did not sufficiently incorporate different perspectives and views of the DevOps phenomenon (Locke and Golden-Biddle, 1997), specifically research on required DevOps skill sets (Hemon-Hildgen, Lyonnet, *et al.*, 2020; Wiedemann and Wiesche, 2018) and importance of DevOps maturity (Gupta *et al.*, 2017; Rafi *et al.*, 2021), this dissertation argues that DevOps does not require the need for actors to build up a specific skill set or work toward a DevOps maturity model. Since more and more ventures seek for practical, systematic guidance about how DevOps and digitalization can be organizationally implemented, the dissertation with its empirical and theoretical elaboration can serve as the missing piece to glue together organizational efforts of DevOps and digitalization.

2.7 Limitations and Future Research

I note several limitations of this research prompting avenues for future research. First, stemming from research design choices and inherent to case research, the suggested theory is grounded in the situation of the large-scale, international cases across segments, including “digital first” and traditional, industrial-native firms. The emerged theory of the DevOps funnel and the (statics of the) DevOps continuum was built and asserted from case study, while enfolding literature, to be generalizable across settings. I call for other studies, with other cases, from other segments, relate DevOps implementations to the funnel and to the continuum, to provide supporting and contrasting theory on DevOps in large-scale organizations. Although this cross-sectional research deliberately focused on established, large-scale organizations, the sampled basket of case sites also contained teams that act similar to start-ups. Even if similar findings can be expected from studying start-ups, further research of DevOps implementations in start-ups and early ventures might add further insides into the why and how of DevOps implementations. Complementary, future research might execute an in-depth, longitudinal, single case study research, empirically exploring the how and why behind DevOps implementations, to provide further insights on the suggested theory, to further examine the mechanisms leading to a rebound with a reverse change in coupling (e.g., the move from quadrant D to B, not from B to D; Figure 2-11), to further assert the best "fit" for a DevOps implementation through the lens of organizational change (including adaption of methods) and contingency theory (Ven and Drazin, 1984; Volberda *et al.*, 2012). Further, aligned with the research design and scientific rigor, the systematic literature review applied defined inclusion/exclusion filters on found publications from defined data sources informed by the research design. A concept matrix and provisional codes were the starting point for theorizing the DevOps funnel. Future research might choose other publications, and assert the suggested model. In addition, Bygstad and Munkvold (2011) argue identifying mechanisms with positivist causation is possible, but much more difficult compared to applying a critical realist perspective. They state that the critical realist approach spurs creative thinking and provides more precise explanations. I call for further studies of mechanisms of DevOps dynamics, with same or other epistemological views, to create supporting or contrasting theory.

2.8 Conclusion

This research investigated the why and how of DevOps implementations in large-scale organizations and adds systematic knowledge to literature. It reports interesting findings along with empirical and theoretical elaborations, integrates with and expands existing concepts and theory, and suggests novel theory. With DevOps being a natural evolution step of AISD, and given its pervasive proliferation in industrial practice along with the scant (although growing) literature on the emerging DevOps phenomenon, this foundational research is important and relevant, and opens avenues for future research. On another layer, this inquiry presents interesting new applications and nuanced concretizations of scientific empiric approaches in IS research, connects the distinct but allied communities of systems development and digitalization, and, grounded on the sound theoretical foundation, lays out fruitful discussion points for conversations in and across domains. I encourage other researchers to continue to study DevOps and add further systematic knowledge.

3 Paper I. “DevOps: Walking the Shadowy Bridge from Development Success to Information Systems Success”.

Hüttermann, Michael and Rosenkranz, Christoph, "DevOps: Walking the Shadowy Bridge from Development Success to Information Systems Success" (2019). ICIS 2019 Proceedings. 10. https://aisel.aisnet.org/icis2019/is_development/is_development/10

DEVOPS: WALKING THE SHADOWY BRIDGE FROM DEVELOPMENT SUCCESS TO INFORMATION SYSTEMS SUCCESS".

In recent years enterprises have observed that a holistic approach to agile information systems development and a closer integration of information systems development with information systems operations is essential to maximize the probability of success, leading to the emerging DevOps phenomenon. While past research delivers insights about success criteria in information systems development as well as information systems operations, conceptual inclusion of DevOps is missing. We propose a multi-staged qualitative research design including literature review and multiple-case study to explore and identify origins of critical success criteria used to measure success by the two major IT-related enterprises functions IT development and IT operations. Based on that, we aim to contribute a "DevOps model", from a success criteria perspective, and reconcile information systems development with the Information Systems Success Model. In addition, our research significantly fosters understanding of the DevOps phenomenon and identifies paths for future research.

3.1 Introduction

In today's industrial practice, applying agile methods such as Scrum (Beck, 1999) or Extreme Programming (Schwaber and Beedle, 2002) to *information systems development (ISD)* is increasingly one promising way for enterprises to maximize the probability of *ISD success* (Dybå and Dingsøyr, 2009). These approaches, collectively labelled under the umbrella of *agile information systems development (AISD)* (Conboy, 2009), currently are the dominant way to ISD in industry (VersionOne, 2018). AISD approaches-in-use within companies are often characterized as a combination of specific *agile practices* (e.g., daily stand-up meeting, pair programming, etc.) taken from different methods, which team members can choose to enact as a methodology-in-use (Recker *et al.*, 2017).

Although a lot of ISD research has investigated specific facets of AISD such as the use and effects of selected agile practices (e.g., Balijepally 2009; Holmqvist and Pessi 2006; van Oorschot et al. 2018) or the implementation and adoption of AISD methods to teams (e.g., Cao et al. 2009; Mangalaraj et al. 2009), there is still a dearth of research that tries to link AISD in general to ISD success. On the one hand, we lack insights about how to measure and evaluate AISD and ISD success (Conboy, 2009). On the other hand, most extant AISD research has focused solely on developmental factors, the ISD process itself, and the developers' perspective, with little attention being paid to operational factors, the resulting IS product, or its long-term operation and use (Diegmann *et al.*, 2018).

In industry, the simultaneous consideration of both developmental and operational factors is central to the emerging phenomenon of *DevOps* (Wiedemann, Forsgren, *et al.*, 2019). DevOps is a portmanteau word of "development" and "operations". Proponents of the concept argue that the two major IT-related enterprise functions *IT development* and *IT operations* increasingly apply shared goals and use shared practices across both functions, bringing together team members from both development and operations, in order to implement AISD in a comprehensive way (Lwakatare et al. 2016; Qumer Gill et al. 2018; Wiedemann 2017). In essence, DevOps appears as a logical extension of AISD, aiming to bridge development to its resulting IS product and its systems' operation (Lwakatare et al. 2016; Qumer Gill et al. 2018). Thus, companies applying DevOps streamline their IT development and IT operations to overcome classic barriers and friction points between those two often siloed IT functions.

Since DevOps is a very recent phenomenon, only few studies up to now exist. Extant work so far has investigated various disparate aspects of DevOps, for example, the continuous software-engineering pipeline of approaches such as DevOps (Fitzgerald and Stol, 2017), the different roles for knowledge sharing within systems management, characteristics of support tools for systems maintenance (Forsgren *et al.*, 2016), adoption challenges for DevOps (Lwakatare, Karvonen, *et al.*, 2016), or necessary skill sets of team members for DevOps (Wiedemann and Wiesche, 2018). The lack of formal definition for DevOps and various different theoretical conceptualizations among these studies, however, prevent a thorough understanding of what DevOps entails and signifies.

As a starting point for building such an understanding and conceptualization, the *Theory of Information Systems Success* (DeLone and McLean, 1992, 2003) and its resulting *IS Success Model (ISSM)* is one of the most important contributions to comprehensively measure information systems (IS) success on the operational side. By responding to the call to further refine the ISSM as well as to explore its relationships with ISD (DeLone and McLean 2016, p. 93), we aim to reconcile the emerging DevOps phenomenon and the issue of evaluating ISD success with an updated version of the model. In doing so, we aim to deliver a theoretically and empirically grounded conceptualization and definition of

DevOps, stemming from an IS success perspective, and thus foster our understanding of the emerging DevOps phenomenon.

In essence, to gain understanding about what DevOps is, we want to explore and find out whether the use of DevOps is (a) a way to optimize a dimension of the ISSM (e.g., service quality), (b) whether it is a success factor and driver of the ISSM itself, (c) whether it is a means that influences IS success (i.e., a cause and effect factor), and (d) what role the application of agile practices plays. Therefore, we aim to answer the following research questions:

RQ1. What are the origins of success criteria enterprises use for their DevOps initiatives?

RQ2. What is a suitable conceptualization and definition of DevOps from an ISSM perspective, and how does current practice-in-use match to this?

For answering the first question, we aim to empirically explore which success criteria used in industrial practice have their origins mainly in IT development, which ones have their origins mainly in IT operations, and which ones have their origins in both entities. Subsequently and closely related to this, we aim to develop and conceptualize a "DevOps model" based on an updated ISSM (integrating developmental and operational success measures) and empirical studies for answering the second question.

Section 2 of this paper summarizes related work and discusses the theoretical background. Section 3 describes the preliminary research model and research design. In Section 4, the expected contribution is discussed and an outlook is provided.

3.2 Related Work and Theoretical Background

3.2.1 DevOps and its Measurement: Bridging Development and Operations

DevOps as one way of applying agile approaches across IT functions (Hüttermann, 2012) has gained increasing attention from IS research (Sharp and Babb, 2018). Due to lack of specific research, models, and definitions, DevOps can describe different things, including team structure, success criteria/factors as well as concepts and tooling (Qumer Gill *et al.*, 2018). In this study, we focus on the success aspect, since increasing the probability of success is the underlying motivation to implement DevOps, shared incentives between development and operations are crucial for success (Hüttermann, 2012), and metrics and measures are often an essential, grounding part of existing DevOps initiatives (Google 2019; Wiedemann et al. 2019).

Accordingly, *measurement* is a major characteristic of DevOps and *useful metrics* is a pattern of DevOps practice (Lwakatare et al. 2016), suggesting that both development teams and operations teams should be incentivized and rewarded by the same metrics, and that one function should use feedback from the other (e.g., progress in development is measured in terms of having a working system in the production environment). DevOps and its focus on measuring addresses major management concerns, including business productivity and cost reduction, IT and business alignment, and business agility and speed to market (Luftman et al. 2013). DevOps accepts that functions do often have different or even conflicting success criteria; nonetheless, it tries to align the success criteria of the two major IT functions found in enterprises, IT development and IT operations, with each other, in order to reduce friction points, and widening the scope of agility from traditional ISD teams in order to prevent micro-optimized silos (Hüttermann, 2012).

What remains unclear, however, is how DevOps is applied and utilized from a success criteria perspective, which measures typically *are* used in emerging DevOps initiatives, and which measures *should* be used. First, we *lack insights about and measurement of success criteria.* For example, "software functionality" is typically used as an ISD success respectively IT development measure (Recker *et al.*, 2017), and "mean time between failures" is typically used as an IS success respectively IT operations measure (Dekleva, 1992). Applying DevOps, it is unclear how both relate to each other, if both should be used, if one should be prioritized above the other, or if both are useful at all. This makes it difficult to understand the motivation of, and, in practice, to improve on those criteria.

Second, it also remains unclear to which extend previously identified AISD success criteria, for instance, *simplicity*, *adding value*, and *learn through change* (Conboy, 2009), are commonly utilized and incorporated in measurement in industry, and how these ISD measures do relate to the resulting IS (e.g., in terms of product quality or customer satisfaction). In practice, such success criteria identified by research could be a good starting point for DevOps initiatives, however, often they appear as neither measurable nor actionable, and information about metrics are still fuzzy. There is a need for developing a set of metrics to evaluate actual performance outcomes under each component of AISD (Conboy, 2009). Similarly, we need more detailed knowledge about specific success criteria and their link to the related DevOps phenomenon.

Third, most measures lack a convincing rationale and theoretical grounding. Behind any good concept or theory should be a strong underlying logic and rationale (Whetten, 1989). Although existing research shaped our understanding of AISD, a parsimonious theory is still missing (Abrahamsson *et al.*, 2009; Conboy, 2009; Whetten, 1989), particularly about the role of DevOps. Although AISD may not be the best solution for all given setups, and in practice often a blend of AISD practices is used (e.g., Fitzgerald et al. 2006), it is widely recognized that AISD is strongly accepted and adopted in industry (VersionOne, 2018). However, AISD approaches are not adopted nor enacted "by-the-book" in most enterprises, and often a blend of practices is used. Meanwhile, extant research on AISD mainly has focused on the developers' perspective (Diegmann *et al.*, 2018), with little research focusing on holistic approaches such as DevOps or end users' responses to AISD, or on factors related to the enterprise function of IT operations.

3.2.2 Information Systems Development Success and Information Systems Success

What many of the approaches both on ISD success and on IS success have in common is that they conceptualize *success* as a dependent variable, and identify multiple success sub-concepts and concepts closely related to success.

ISD success is often related to individual aspects such as staying within scope, time, and cost requirements (Chow and Cao, 2008), rapid change (Lee and Xia, 2010), or delivering high quality (Siau *et al.*, 2010). It has been discussed from different social-technical perspectives, either from an organizational and team perspective such as diversity (Ramasubbu *et al.*, 2015), or from a behavioral and methodological perspective such as communication (Hummel *et al.*, 2013) or learning (Conboy, 2009). Additionally, enterprises often tailor their respective AISD initiatives to also adapt *lean* approaches (Poppendieck and Poppendieck, 2003) to manage and improve measures of flow time and cycle time (Anderson and Reinertsen, 2010; Wang *et al.*, 2012; Wetherbe and Frolick, 2000), which in turn also relate to ISD process-based success criteria. In sum, a diverse and varied set of ISD success conceptualizations and measures have been proposed.

IS success similarly can be defined in multiple ways. There are nearly as many success measures as there are studies (DeLone and McLean 1992). Starting with the data processing era in early 1950s, IS have continuously evolved, reaching the customer-focused era in the new millennium (DeLone and McLean, 2016). For research, measurement of IS success was always crucial. For example, early studies discussed the User Information Satisfaction instrument (Ives *et al.*, 1983). The Technology Acceptance Model (TAM) focused on comprehensive measure of IS success as well (Davis, 1989). Complementary contributions include, amongst others, SERVQUAL, ITIL, or the Balanced Scorecard (DeLone and McLean, 2016).

One of the most adopted and influential contributions (Urbach et al. 2009) in this domain is the *IS Success Model (ISSM)*, tackling success as a blend of six dimensions. According to the ISSM, the inter-related variables of IS success are: system quality (the desirable characteristics of an IS), information quality (the desirable characteristics of the IS outputs), service quality (the quality of the support that system users receive), use (the degree and manner in which employees and customers utilize the capabilities of the IS), user satisfaction (users' level of satisfaction with reports, web sites, support services etc.), and net impact (the extent to which the system is contributing to the success of individuals, groups, or organizations) (DeLone and McLean 2016). The model serves as a base for other work, for example, summarizing the measures of the ISSM applied to the evaluation of IS success and by examining the relationships that comprise the ISSM (Petter *et al.*, 2008). Later work focuses on independent variables as success factors and identifying their relationships to the ISSM, for example, project characteristics such as project management and developer skills (Petter *et al.*, 2013).

In essence, the ISSM attempts to reconcile the plethora of approaches trying to measure IS success, and results in a multi-dimensional framework for understanding the multi-dimensionality of IS success, giving a combined view on existing process and variance models. Its six interdependent variables must all be measured and/or controlled to measure IS success. As such, it is a logical starting point to conceptualize the operational aspects of IS.

3.2.3 Combining Information Systems Success Model and DevOps: A Preliminary Model

Based on this brief discussion and the variables introduced above, Figure 3-1 provides a preliminary conceptualization of the DevOps phenomenon from an ISSM perspective. Within our preliminary model, both IT development (Dev) and IT operations (Ops) have their respective *critical success criteria*. These independent variables impact success, either *ISD success* or *IS success*, and are moderated by *Dev success* and *Ops success*.

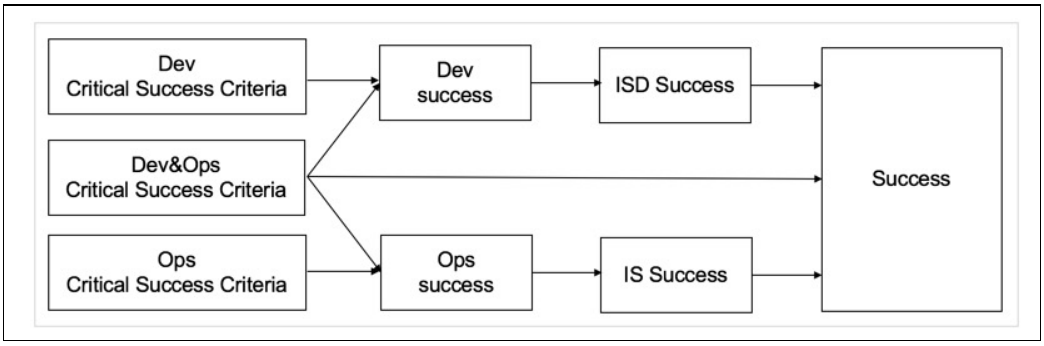

Figure 3-1. A conceptualization of DevOps from a success perspective.

With DevOps, *shared goals* do affect success as well. If the goals of both functions are unrelated or even conflicting, desired behavior of stakeholders in the Dev function as well as the Ops function positively moderates success of the respective enterprise entity, and negatively moderates the success of the respective other entity. Both interfacing functions, Dev and Ops, are aligned with each other through DevOps, which in turn relates to the ISSM. Similar to the ISSM, success in DevOps is also multi-dimensional. Particularly, DevOps can describe both *success criteria* (defining the success of an initiative, such as cost, time, performance, quality, team satisfaction) as well as *success factors* (that are made up by important influences that contribute to IS or ISD success) (Baccarini 1999).

Table 3-1 provides exemplary conceptualizations of the two concepts ISD success and IS success with respective exemplary sub-concepts (e.g., on-time completion, cycle time). The table also contains two examples for shared goals between development and operations, hinting to a strong implementation of DevOps: both IT functions utilize the same success sub-concept (e.g., cycle time), which in turn is part of both ISD success as well as IS success.

Authors	Concepts / Major Focus	Success Sub-concepts
Lee and Xia (2010)	ISD Success / Development	On-time completion, on-budget completion, software functionality
Recker et al. (2017)	ISD Success / Development	Customer satisfaction, process performance, software functionality
DeLone and McLean (2003)	IS Success / Operations	System quality, Information quality, service quality, Intention to use/use, User satisfaction, Net impact
Dekleva (1992)	IS Success / Operations	Maintenance time, Mean time between failures MTBF, Mean time to repair MTTR

Wetherbe and Frolick (2010)	ISD Success / Development IS Success / Operations	Cycle time
Poppendieck and Poppendieck (2003)	ISD Success / Development IS Success / Operations	Flow and cycle time

Table 3-1. Exemplary DevOps Conceptualizations.

3.3 Preliminary Research Design

DevOps is a new and emerging phenomenon, where its role, its context, and the boundaries between phenomenon and context are not clearly evident (Yin, 2018). Consequently, we propose to apply a *case study design* for developing explanatory theory (Eisenhardt, 1989). Case studies allow researchers to develop a deep understanding of the phenomena within its natural setting and real-life context, especially when the boundaries between phenomena and context are not clearly evident (Yin, 2018). Our preliminary research design consists of four stages (see Figure 3-2).

First, a *structured literature review* (Webster and Watson 2002) examines the current body of knowledge on both DevOps and on success criteria, which are used in enterprises to measure IS success and ISD success, and their respective origins in either IT development or IT operations. To synthesize relevant work in the body of knowledge, and to accumulate and identify concepts, we implement a concept-centric approach (Webster and Watson 2002). To identify relevant concepts, major contributions are likely to be published in leading journals. Therefore, our focus is on high quality, peer-reviewed outlets published in journals such as the "Seniors Scholars' Basket of Journals" and selected outlets from Software Engineering and Project Management (e.g., "Empirical Software Engineering", "Journal of Systems and Software"). Due to the broad context, we will also follow their advice to scan the journals' table of contents to pinpoint articles that would not be caught by a strict keyword approach. Second, we then go backward by reviewing the citations of articles identified in stage 1, and go forward by identifying articles citing the articles of the previous stage to identify more articles. Afterwards, we will create a *systematic literature map* with streams and lines of research (Creswell and Creswell, 2018). We expect the review of current literature about the emerging topic of DevOps to be shorter compared to a detailed review for a mature topic where an accumulated body of research exists. Where possible, this study primarily focuses on already performed literature reviews on success criteria, since this is a more mature line of literature.

Third, we will perform a *multiple-case study* to explore the contemporary phenomenon in depth, to understand its how and why, and its real-life context. We are convinced that our design is the best fit to achieve high external validity, including gathering complementary as well as possibly contradictional insights from multiple, theoretically useful cases, and checking whether the findings can be generalized, and bringing together the academic with the practitioners' views. Prior to starting the theory-building case study, important variables are identified, with reference to extant literature (e.g., based on the ISSM), to nail down knowledge about the domain, whereas the theory, including the relationships between variables and theory, is built as part of the case study (Eisenhardt, 1989).

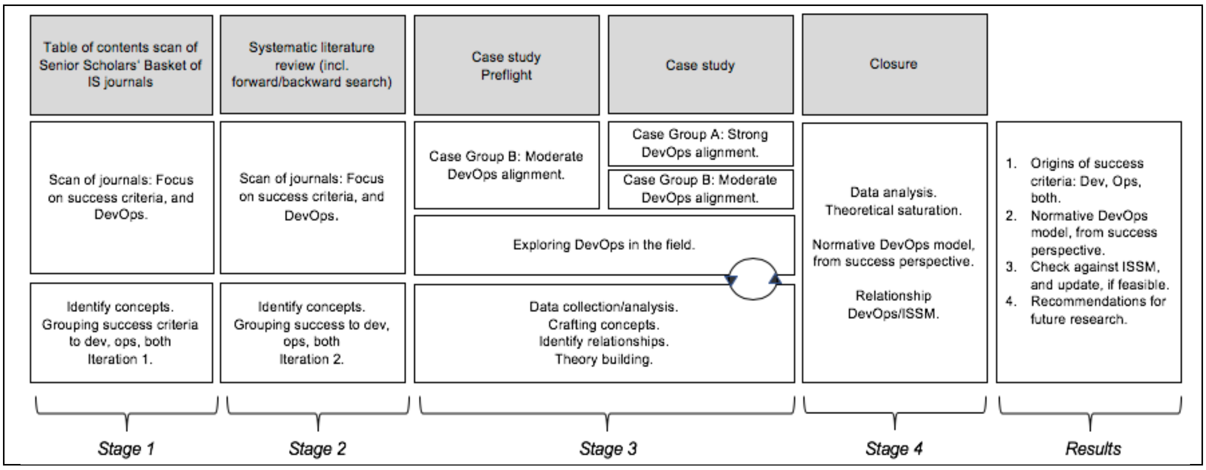

Figure 3-2. Preliminary research design.

Our phased case study (Eisenhardt 1989; Yin 2018) will start with a preflight including a pilot case study (shaping relevant lines of questions and conceptual clarifications for the research design including detailing data collection plans) and a pre-test case study (including finalizing and pre-testing data collection plans). Afterwards, we will then study additional cases treating the series of cases similar to a series of experiments, with each case being a discrete experiment that serves as a replication, contrast, and extension to the evolving DevOps model. We predict to gain meaningful insights with sufficient theoretical saturation (i.e., minimal incremental learning combined with minimal incremental improvement to the theory) after performing four cases studies.

Cases are assigned to two groups. For group A, case sites are both revelatory and extreme while being successful in their domain that understands IT as a core asset and an inherent driver for success.

Companies of this section comprise the largest value of all enterprises worldwide, based on market capitalization. As a result, these companies are important and do run a holistic approach to IS/D because business is driven by IT as a core asset for maximizing business value (Overby *et al.*, 2006). These companies presumably are better integrated internally and arguably do successfully soften conceptional barriers between development and operations, and are listed in NASDAQ-100 (or similar). They are expected to strongly apply DevOps. Members of group B are comparable with members of group A except that they run a more conservative, classic approach to IT. We expect them to apply DevOps moderately, thus providing a contrast to group A. Via theoretical replication, due to contrasting results for anticipatable reasons, the rival group is supposed to help to gain better understanding of the studied DevOps phenomenon. The main unit of interest for both groups is the respective enterprise, the smallest units of interest are embedded teams of enterprises' IT development function and IT operations function. Preflight cases are from group B to better prepare the theory building (e.g., grounding with a more classic approach to IT). For the additional cases, two are of group A and two are of group B.

Data will be collected via interviews (including learning possible new questions from the interviews, and *snowball sampling* to find out more knowledgeable informants), observations, and document analysis, in order to triangulate data collection techniques (Eisenhardt, 1989; Yin, 2018). During a two weeks on-site period for each case, semi-structured interviews are planned with different key informants across organization functions, represented by specific individuals with assigned specific roles including developers, operation engineers, and managers. Observations also will be conducted by one of the authors being a participant-as-observer (Cassell and Symon 2012). Data collection and data pre-analysis do overlap in the field and are part of the overall process of building theory from case study research (Eisenhardt 1989): Aligned with the inductive, exploratory nature of this research, concepts and relationships are developed in sequential within-case analysis, before looking for similar concepts and relationships across multiple cases, to build and continuously further shape theory and to use theory to generalize from case study.

Access to appropriate case sites is possible based on the personal network of the authors. One member of the research team is also an independent freelance consultant on DevOps. Cassell and Symon (2012) argue that some degree of researcher bias is not only inevitable to the study, rather it is beneficial since such studies cannot be carried out in a social vacuum. In addition to highest standards of the scientific method, including replicability and independence of the research as well as the comprehensive chain of evidence, tactics to minimize participant bias include taking frequent breaks from the field to continue on theory building, and to reflect and analyze collected data (Cassell and Symon, 2012).

Fourth, based on insights of these stages, this research will primarily inductively develop a conceptualization and explanatory model of DevOps, from a success perspective, and update the ISSM accordingly.

3.4 Expected Contribution and Outlook

This research adds important value to the existing body of knowledge in three ways. First, the findings will help understanding AISD and IS through the lens of the emerging DevOps phenomenon, and reconcile ISD with the ISSM – a part which hitherto has been largely neglected. We expect our results to considerably foster our understanding about initiatives in AISD, often driven by success criteria/factors, bridging different organizational entities, leading to the DevOps movement.

Second, exploring DevOps from this angle is timely, because in recent years enterprises have observed that a holistic approach to AISD and a closer integration of ISD with IS is needed - namely DevOps - to maximize the probability of success. The DevOps approach is often a holistic mash of different success criteria originated in different enterprise functions, and understanding of their origins does significantly contribute to further understanding.

Third, insights how DevOps and the ISSM can be reconciled will add significant understanding about DevOps to the academic field. The theoretical reasoning of the conceptualized DevOps model can be embedded into the broader debate of IS success and ISD success, and helps to bridge both domains.

We are convinced that this study is a significant contribution. The new DevOps movement lacks a theory, and once this is available, also directions for future research can be identified, e.g. extending the provided DevOps model beyond the foundational success perspective. This research aims to also provide guidance for practitioners since DevOps has strong momentum in the field of enterprise AISD and IS. After we have preliminary designed the research, with an understanding of the underlying domains including their concepts, and identified the research problem, as described above, we plan to finalize the research design, to perform the structured literature review, and to run the pilot case study, by end of 2019.

4 Paper II. “The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations”.

Hüttermann, Michael, "The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations" (2021). *ECIS 2021 Research Papers*. 78. https://aisel.aisnet.org/ecis2021_rp/78

THE DEVOPS CONTINUUM: WALKING THE SHADOWY BRIDGE FROM INFORMATION SYSTEMS DEVELOPMENT TO OPERATIONS

In recent years, enterprises have observed that a holistic approach to agile information systems development and a close integration of information systems operations is essential to maximize the probability of success, leading to the emerging DevOps phenomenon. While first studies have delivered first preliminary insights about DevOps, a foundational understanding of DevOps implementations and theoretical grounding of DevOps is still missing. To close this gap, we conducted a multiple-case study to explore a variety of characteristics of DevOps, and how these different characteristics influence its implementation in different contexts. We find that variations of DevOps implementations are gradual transitions on what we call a DevOps continuum. Based on this insight, we propose a conceptual model which fosters our understanding of the DevOps phenomenon, relates DevOps to existing theories, and identifies new paths for future research.

4.1 Introduction

In industry, the simultaneous consideration of both the development and the operations of information systems (IS) is central to the emerging phenomenon of *DevOps* (Hüttermann, 2012; Wiedemann, Forsgren, et al., 2019). DevOps is a portmanteau word of "development" and "operations". Proponents of the concept argue that the two major information technology (IT)-related enterprise functions "IT development" (those that build IS) and "IT operations" (those that run and maintain IS) increasingly apply shared goals and use shared practices across both functions, bringing together team members from both development and operations, in order to implement information system development (ISD) in a comprehensive way (Lwakatare, Kuvaja, et al., 2016; Qumer Gill et al., 2018; Wiedemann, 2017). In essence, DevOps appears as a logical extension of agile information systems development (AISD) (Hemon-Hildgen, Rowe, et al., 2020), aiming to bridge development to its resulting IS product and its systems operations. Thus, companies applying DevOps streamline their IT development and IT operations to overcome well-known barriers and friction points between those two often siloed IT functions. DevOps aims to address major management concerns, including business productivity and cost reduction, IT and business alignment, and business agility and speed to market (Luftman and Zadeh, 2011).

Until now, scant studies have investigated DevOps because it is a very recent phenomenon. For example, existing research has proposed a tripartite model of intra-IT alignment, made up of individual componentization, integrated responsibility, and multidisciplinary knowledge, to shed light on how DevOps aligns development with operations (Wiedemann et al., 2020). Others have examined the orchestration of automation, sharing, and risk management in DevOps teams and their relationship to work conditions and job satisfaction (Hemon-Hildgen, Rowe, et al., 2020). While these first studies offer very valuable preliminary insights about DevOps implementations in specific settings, they focus on particular aspects such as how alignment in DevOps teams can be achieved (Wiedemann et al., 2020) or how DevOps affects job satisfaction (Hemon-Hildgen, Rowe, et al., 2020). However, similar to AISD, the development of DevOps is primarily driven by industry. No formal definition for DevOps exists, and many practitioners argue that this is intentional because it allows teams and organizations to adopt a definition that works for them (Wiedemann et al., 2019). This lack of understanding, conceptualization, and theorizing is challenging for academic research and studies on DevOps.

In sum, we are missing a theoretical grounding of DevOps, and we are also lacking a foundational understanding of what leads to a specific DevOps implementation (i.e., how DevOps is executed), depending on a given context in industry, or what the effects of different DevOps implementations, configurations, or approaches are. Consequently, this leads us to the following research question for our study: "*Which characteristics influence how DevOps is implemented in organizations?*"

To answer the research question, we conducted an exploratory, multiple-case study in order to derive and suggest a model explaining DevOps. For IS research, we contribute to the nascent understanding of DevOps by providing an empirically grounded theoretical model. Since DevOps has a strong momentum in industrial practice, our results may offer valuable guidance to practitioners who execute (diverse) DevOps initiatives in their respective contexts. Next, we discuss the theoretical background and related work of our study. Afterwards, we describe our research method, present our main findings and discuss the resulting DevOps model. We conclude with highlighting avenues for further research.

4.2 Related Work and Theoretical Background

4.2.1 Origins of DevOps

In today's industrial practice, applying AISD methods such as Scrum (Beck, 1999) or Extreme Programming (Schwaber and Beedle, 2002) is increasingly one promising way for enterprises to maximize the probability of ISD success (Dybå and Dingsøyr, 2009). Research revealed, however, that focusing solely on the development factors does by-and-large neglect the importance of operational factors, the resulting IS product, and its long-term operation and use (DeLone and McLean, 1992; Fitzgerald and Stol, 2017; Petter *et al.*, 2013). This suggests that IS success (DeLone and McLean, 1992) can only be optimized if the organizational subunits of IT development and IT operations are integrated or aligned with each other (Wiedemann et al., 2020). Spanning development and operational factors, *efficiency* ("delivering the product right") and *effectiveness* ("delivering the right product") (Chandler Jr., 2018) contribute to IS success. Efficiency relates to how much IS functionality is shipped in a given time, whereas effectiveness focuses on the amount of customer value created by the shipped functionality (Bosch, 2019). Building on this understanding, industry often sees DevOps as an extension of AISD (Wiedemann, Forsgren, et al., 2019), which aims to optimize overall IS success and strives for better results of efficiency and effectiveness (Hüttermann, 2012).

Similar to its rising application in industry, DevOps has gained increasing attention from IS research (Sharp and Babb, 2018). With its roots in AISD, DevOps stems from the idea to extend the use of agile practices from ISD to IS operations, and thus bridges the often siloed IT functions of IT development and IT operations (Hüttermann, 2012). Due to the lack of specific research, models, and definitions, DevOps can describe different things, including team structure, success criteria (i.e., goals), concepts, or tooling (Qumer Gill et al., 2018).

4.2.2 First Steps Towards Conceptualizing DevOps

As a first step towards a conceptualization of DevOps, researchers have suggested dimensions such as culture, automation, lean, measurement, and sharing (CALMS) (Fitzgerald and Stol, 2017; Humble and Molesky, 2011; Wiedemann, Forsgren, *et al.*, 2019). According to this understanding, DevOps aims to address *cultural* barriers between the development and operations functions, it strives for *automation* of development and delivery processes across the functions, it emphasizes *measurement* and joint use of metrics across the functions, it applies *lean* principles such as removing waste, and it highlights the practice of *sharing* culture, goals, measures, automation and tooling between the functions. Extant research also argues that culture and measurement can be partially subsumed in the concept of sharing (Hemon-Hildgen, Rowe, et al., 2020), that sharing information and practices across functions leads towards finding a common ground and understanding of measures, and once sharing is practiced, automation of processes can be implemented more efficiently towards continuous delivery of software. In industrial practice, automation and the related concept of *continuous delivery*, the production of software in short temporal cycles, resulting in building, testing, and releasing software with greater speed and frequency, often have the highest priority in order to satisfy customers through continuously providing valuable software (Fowler and Highsmith, 2001). A prerequisite for continuous software delivery is *lean thinking* to eliminate bottlenecks along the value chain across all contributing parties (Khan and Sarker, 2002; Poppendieck and Poppendieck, 2003; Wetherbe and Frolick, 2000), above all IT development and IT operations (Fitzgerald and Stol, 2017; Hüttermann, 2012)[7].

Based on this, we adopt previous attempts to define DevOps (Hemon-Hildgen, Rowe, et al., 2020) and combine them with AISD's "ability to adapt to change" (Conboy, 2009) and the concept of "lean thinking" to holistically improve processes to continuously deliver valuable outcome to the customer (Fitzgerald and Stol, 2017; Poppendieck and Poppendieck, 2003). Our resulting conceptualization defines DevOps as "*a set of continuously improved principles for collaborative work implemented between the IS development function and the IS operations function, along with potentially other stakeholders, which is founded on the sharing of culture, goals, measures, automation tools and automated processes towards continuous delivery of valuable outcome*." The latter aspect – continuously deliver value to the customer – provides a reason why DevOps in industry often bridges efficiency (e.g., through management and improvement of cycle time, Anderson and Reinertsen 2010; Wetherbe and Frolick 2000) and effectiveness (e.g., to deliver valuable outcome to the customer).

7 We subsume different developmental roles such as architect and developer into development, being aware that the industry-driven name of the DevOps phenomenon might be too exclusive regarding specific roles being involved in development and operations of the IS.

4.2.3 First Steps Towards Theorizing DevOps

Existing empirical research on DevOps so far has investigated various disparate aspects. Among those are, for example, the continuous integration of software development and the operation of the resulting IS product (Fitzgerald and Stol, 2017), the adoption challenges for DevOps (Lwakatare, Karvonen, et al., 2016), the necessary skill sets of team members for DevOps (Hemon-Hildgen, Lyonnet, et al., 2020; Wiedemann and Wiesche, 2018), or the control-alignment view for product orientation in DevOps teams (Wiedemann et al., 2019). These studies have also resulted in the proposition of two specific theoretical models.

First, a theoretical model of job satisfaction suggests that the orchestration of automation, sharing and risk management is moderated by work conditions and positively impacts job satisfaction if AISD teams move towards DevOps (Hemon-Hildgen, Rowe, et al., 2020). Second, a model on intra-IT alignment extends operational alignment's focus on IT infrastructure and processes to alignment of development and operations functions (Wiedemann et al., 2020). Given mismatching interoperability, conflicting goals, different procedures, and various competencies between development and operations functions, the model proposes interrelated mechanics of integrated responsibility, individual componentization, and multidisciplinary knowledge in order to produce intra-IT alignment.

Although these two models provide first very valuable preliminary insights on DevOps and some of its probable characteristics, mechanics, and effects, they offer only a narrow focus on very specific effects (job satisfaction) and mechanisms (intra-IT alignment), and are based on a single case study (a large service firm) and a very specific sample (smaller software products such as online shops) respectively. Important aspects that may lead to different DevOps implementations, characteristics, mechanisms, and effects are missing, for example, more complex IS products or larger organizational settings have been ignored so far. Existing studies have emphasized the importance of contextual factors while developing IS products (e.g., Baham and Hirschheim, 2021), and have highlighted the important role of contextual factors for transitions towards DevOps, without explicitly identifying and further examining them (for DevOps executions) (cf. Luz et al., 2019). Gaining understanding which contextual characteristics influence DevOps implementations and mechanisms considerably helps to unlock the DevOps phenomenon. Considering the lack of research about DevOps in general, and the missing understanding which contextual characteristics exist that influence a specific DevOps implementation in particular, we explore the characteristics influencing current DevOps implementations. Our goal is to offer an empirically grounded understanding of factors that result in variations of DevOps implementations, and based on that, to abstract theory and to derive predictions for further DevOps implementations.

4.3 Research Method

4.3.1 Design Overview

We used an exploratory, inductive case study design (Eisenhardt, 1989). First, we identified appropriate case site candidates implementing DevOps. Previous research on DevOps also influenced the selection because we wanted to include similar but also contrasting case types to understand which characteristics led to these specific DevOps implementations.

Sampling criteria focused on revelatory cases (Dubé and Paré, 2003) in two groups of cases. The first group (group A) included case sites that are both revelatory and extreme while being successful in their domain, understanding IT as a core asset and an inherent driver for overall company success. These companies do run a holistic approach to ISD because business is driven by IT as a core asset for maximizing business value (Overby et al., 2006). These companies presumably are better integrated internally, and arguably do successfully soften conceptional barriers between development and operations. Case sites of the second group (group B) are comparable with members of the first group except that they run a more conservative, classic approach to IT. We expected them to apply DevOps moderately, thus providing a contrast.

We thoroughly examined pre-selected case site candidates that matched the sampling criteria and agreed to participate, and ran pilot interviews with 12 different firms over a period of ten months. The 12 firms were either part of our professional network, or were actively contacted for the research purpose at leading industry conferences. We started with one company from this set for each group, and iteratively added case sites to each case group. Since we gained meaningful insights with sufficient theoretical saturation (i.e., minimal incremental learning combined with minimal incremental improvement to the theory) after investigating four cases in total, we decided to end our data collection for this phase of the study.

The respective level of DevOps intensity (aligned with our DevOps definition) was the leading driver to understand the characteristics (leading to specific DevOps implementation) and to split the case groups, thus we did not highly prioritize the segments the firms are active in. However, we carefully paid attention to diverse sites across segments, leading to firms being in different businesses (i.e., online streaming, enterprise software, manufacturing, and financial industry in conjunction with automotive industry since the bank is a subsidiary of a leading car manufacturer), see Figure 4-1.

Case Site* / Attributes	**Group A**		**Group B**	
	SoftwareDev	**MovieStream**	**ManuFact**	**AutoBank**
Segment	Software	Streaming	Industry	Bank
Headquarters	Europe	USA	Europe	Europe
Employees	100.000+	<10.000	400.000+	<10.000
Product offering	Medium to large business products	Large customer entertainment product	Large, complex industrial products	Medium to large business products
No. of interviews	6 (2 DevOps engineers, 2 Product owners tools, 1 Product owner, 1 Site reliability engineer)	3 (3 Platform engineers)	4 (3 DevOps lead architects, 1 Dev manager)	4 (1 Dev manager, 1 Ops manager, 1 IT manager, 1 Service manager)
***Names anonymized and number of employees in ranges for confidentiality reasons**				

Figure 4-1. Case description.

As part of this step we also determined the unit of analysis to understand the context of organizations tailoring their respective DevOps implementation. The unit of analysis is the respective enterprise. During the execution step, and cycling between the different steps of the execution stage, we recognized that even inside enterprises multiple different DevOps implementations may be in place, thus the unit of observation transitioned to teams of 10-30 persons who participate in development and delivery of IS products, including platform teams serving numerous other product teams.

4.3.2 Data Collection

Data was collected via interviews, including learning possible new questions from the interviews and snowball sampling to find out more knowledgeable informants (Eisenhardt, 1989). Complementary data sources included project documentations, public tech blogs and presentations, and, in parts[8],

[8] Participant observation was stopped due to the outbreak of the COVID-19 pandemic.

observation as a participant-as-observer (Myers, 2009; Yin, 2018). Aligned with the inductive, exploratory nature of this research, concepts and relationships were developed in sequential within-case analyses, before looking for similar concepts and relationships across multiple cases, to build and continuously further shape theory and to use theory to generalize from the case study.

We conducted 17 semi-structured interviews with highly knowledgeable informants from the case organizations spanning roles of engineers, architects, and decision makers. This allowed us for data triangulation within and across cases so that we could examine the phenomenon from different angles and gather as much alternative explanations as possible (Dubé and Paré, 2003). The open-ended interviews emphasized context and narratives (Brinkmann and Kvale, 2014; Mishler, 1986), were recorded, and lasted between 36 minutes and 107 minutes. We started our interviews with entry questions about the background of the informant, then discussed the transformation towards DevOps and DevOps dimensions asking for goals, practices, tools and any effect DevOps has on complexity, communication, or knowledge management. We closed interviews asking what next steps are planned in the team's DevOps journey, asked for anything the informant wanted to add and for a recommendation for a colleague to talk to. Additionally, we triangulated data and addressed potential biases in several ways. First, one member of the research team is also an independent freelance consultant on DevOps. While this raises a potential bias, Cassell and Symon (2012) argue that some degree of researcher bias is not only inevitable to the study, rather it is beneficial since such studies cannot be carried out in a social vacuum. Other argue the interviewer needs to be knowledgeable about the interview topic (Brinkmann and Kvale, 2014; Yin, 2018). Second, tactics to minimize participant bias included taking frequent breaks to continue on theory building, and to continuously reflect and analyze collected data. We also emphasized using multiple data sources and writing case writeups (Eisenhardt, 1989; Yin, 2018). In addition, we gave anonymity and transparency to our research participants, for example, by providing thorough information about the study.

4.3.3 Data Analysis

Following guidelines for building theory from case studies (Eisenhardt, 1989; Eisenhardt and Graebner, 2007), we analyzed within-case and cross-case without any a priori hypotheses. The analysis was further influenced by the Grounded Theory Method (Strauss and Corbin, 2015). Particularly, we did not begin our analysis with any provisional codes, rather, we were guided by theoretical sampling following the flow the informants offered while asking exploratory questions. We always took special attention to remain open to alternative explanations, which helped us to understand the context-sensitivity of DevOps implementations. Our model emerged through continuously cycling between analyzing the data and unfolding literature. We used the software MAXQDA for coding the transcribed interviews and additional documentation from the case study database. Data analysis followed established guidelines for coding practices for qualitative research (Saldaña, 2016; Strauss and Corbin, 2015).

Specifically, as part of our first analysis cycle of initial coding, we broke down the rich data to discrete parts, and examined and compared them with other codes. Combined with pattern coding, we summarized segments of data and results and identified what we call "flavors of the continuum characteristics", for example, *continuous delivery*. We then applied axial coding to identify the dominant codes and to explore theoretical relations. This led to categories that form our *characteristics*. *Applied practices* and *success* are example categories produced in this iterative step. Theoretical coding synthesized the categories derived from coding and analysis to create the emerging model. Example concepts that resulted from this step are *inner environment* and *outer environment* making up the core category *environment*, that in turn suggested that a specific DevOps implementation is highly context-sensitive. Figure 4-2 describes the coding process with illustrations.

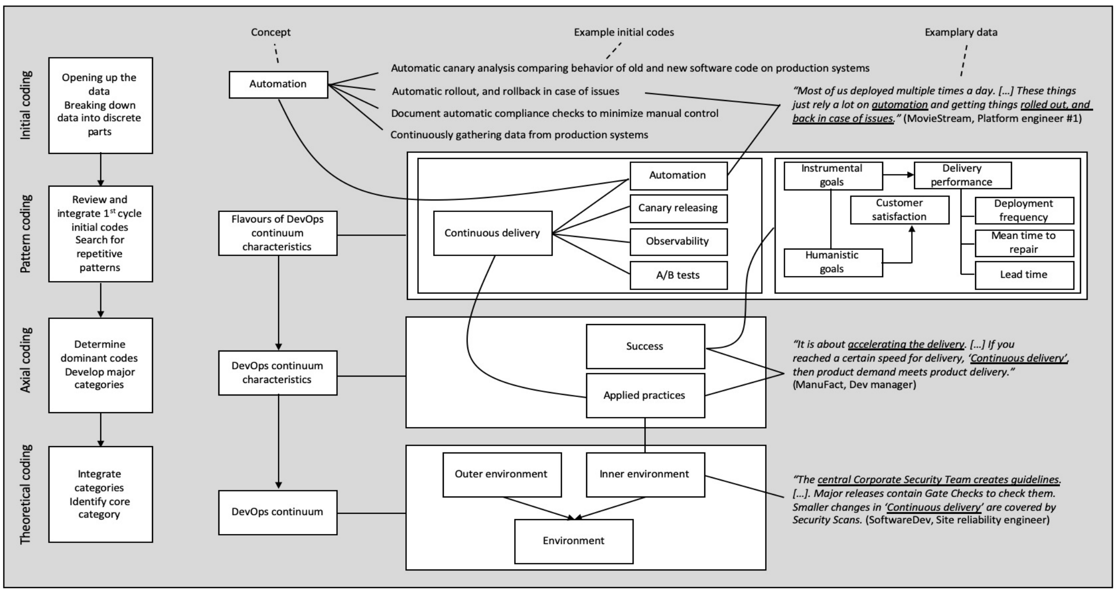

Figure 4-2. Coding process with illustrations.

In addition, the analysis relied on memos, whiteboard sketching, tables, and graphs (Miles et al., 2020). While closing the cross-case analysis, as a blend of case evidence, prior research and stand-alone logic, we finalized the main findings as well as the model of the DevOps continuum (Eisenhardt and Graebner, 2007).

4.4 Findings

We assessed our cases to explore which characteristics impact how they implement DevOps. In the following we analytically summarize our findings and integrate the unveiled characteristics into a model. Our study reveals that the individual team aligns its DevOps initiative with characteristics of its *inner environment* and its *outer environment*, see Figure 4-3.

Characteristics		**Short descriptions**
Inner environment	Organizational standards	Rules and norms the team must align with.
	Dealing with change	How and when the team responds to changes.
	Applied practices	Practices applied by the team.
	Success	Success, and its measurement, the team aims to achieve to fulfill its task.
Outer environment	Market type	Attributes of the demand of the IS product the team provides, in its market.
	Product type	Attributes and identifying features or core affordance of the IS product the team provides.
	Digital technology	Digital technical means the team utilizes to develop and deliver the IS product.

Figure 4-3. Characteristics of teams' DevOps implementations, with short descriptions.

The inner environment expresses characteristics having its root in the team or inherited from the organization it is embedded in. Although, from a team perspective, all other organizational entities in the same organization could be considered to be part of the outer environment, here, with outer environment, we express the context of the team to the field, its users and market. The inner environment and the outer environment make up the environment and summarize characteristics that influence how the team implements DevOps and thus determines the respective DevOps implementation. In sum, this leads not to a pre-defined set of configurations, but rather to a *DevOps continuum*, with gradual transitions of DevOps implementations in between two extremes, see Figure 4-4.

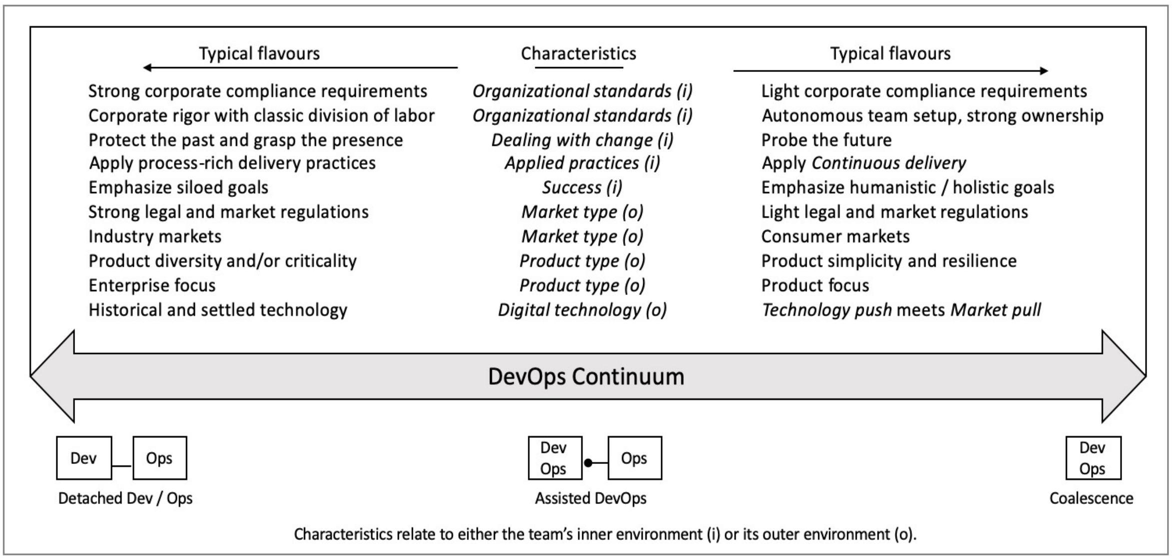

Figure 4-4. Characteristics of teams' DevOps implementations, with typical flavors, can be arranged along a continuum.

According to our DevOps definition, we define 'no integration' between two distinct functions development and operations as *Detached Dev / Ops* on the far left edge of the continuum, 'full integration' as *Coalescence* on the far right edge of the continuum, and *Assisted DevOps* as a 'middle ground', with dedicated IT functions of development and operations with strong collaboration, with many nuances in between. Implementations of DevOps at SoftwareDev span the entire continuum, including classic enterprise resource planning (ERP) products on the left side and other teams with their respective context on the right side. MovieStream's teams mostly can be positioned on the right side, not on the far right edge, since teams, although working strongly autonomously, utilize centralized platform teams. Platform teams are teams of highly specialized experts (e.g., security) providing an internal product or services (e.g., tools, infrastructure) for product teams (teams delivering IS products for a market as part of their outer environments). A context-sensitive variant of division of labor is formed where the whole team, with exception of highly specialized experts, is made up in a cross-functional way spanning different roles to develop and deliver the IS product. DevOps implementations of ManuFact, with its complex and heterogenous industry portfolio, can be positioned along the entire continuum. The DevOps implementation of the examined team at AutoBank, with its specific context of two distinct functions of development and operations with only gentle collaboration, can be placed on the left side of the continuum.

4.4.1 The DevOps Continuum's "Inner Environment"

We define characteristics of the *inner environment* as organizational standards, dealing with change, applied practices, and success. The first characteristic is the *organizational standards.* We define these standards to be corporate, normative rules the team must confirm to, the team setup, and the responsibilities of the team related to other organizational entities. Informants report that these standards contribute considerably to their working culture. Our data indicate that specific instances (i.e., flavors) of this characteristic vary. For example, employees of MovieStream only have to align with a few organizational requirements, leading to large degree of freedom of product teams to freely decide on how they accomplish their tasks. In contrast, teams at ManuFact have to align with more and stronger rules. For example, they must typically use centralized tools, provided and supported by other teams, for version management and binary management, to align with organizational compliance requirements. Different flavors also exist for division of labor across organizational entities. For example, while at MovieStream product teams also operate their product when it is running on production systems ("you build it, you run it"), they develop and freely share and switch operational activities across the product team. At AutoBank it is quite the opposite. There, due to a strict division of labor, colleagues from development do not have access to production systems and are not involved in operational concerns at all.

As regards the second characteristic, we define *dealing with change* as how and when the team responds to changes. Here as well different flavors exist, spanning a range from proactively seeking change and reactively avoiding any change at all. SoftwareDev pointed to situations where members of specific teams question changes and claim that they have solved a specific task the same way for 30 years. On the other side, at MovieStream, change is seen as an opportunity to learn and to understand customer needs. Continuously optimizing processes is mandatory for them if the team aims to deliver their product to production frequently.

The third characteristic as part of the inner environment, *applied practices*, concerns the methods the team uses to develop as well as deliver their IS product. Specific instances of applied practices span a broad range of flavors, from process-rich and manual, with long release cycles (e.g., at AutoBank, SoftwareDev), to continuous delivery (e.g., at MovieStream, SoftwareDev). Automation and sharing contribute to the concept of continuous delivery, where, in its extreme, software is delivered in a pace where the customer does not recognize that a new version of the IS product is in place at all. An important practice across all cases is to utilize automation particularly to foster collaboration and to streamline communication between development and operations. As a Product Owner from SoftwareDev succinctly puts it:

> *"Automating everything, and improving, is important to insure that it is running successfully, even if we know that we have gaps, but we need enough buffer to bring forward our product, in order to continue to make our customer happy".* (SoftwareDev, Product Owner)

Automation of handovers from development to operations reduces communication between involved parties. The entire IS delivery can be automated including compliance checks. The team reportedly feels to be much closer to the customer and being able to better react on their preferences, and thus improve customer satisfaction. In order to gather feedback faster, the team must be able to deliver faster, that is accomplished by automation.

Customer feedback is gathered continuously by implementing "observability": Insights from user behavior and data in production are continuously gathered and metrics are shared across involved parties. As part of continuous delivery, "machine learning" supports to analyze issues on production systems, and various test-related activities are also executed on production systems including "experiment-driven development" (the behavior of the IS product on production systems is checked against defined expectations in form of hypotheses), "A/B tests" (the impact of the difference of two distinct production setups are compared with each other), and "Canary releases" (a new version of the IS product is made available on production systems to only a part of the user base). Across the cases, these practices are often combined and can lead to automatic rollbacks of production changes. A platform engineer of MovieStream gives an example.

> *"We have such high numbers of requests basically, you can learn a lot from metrics. Metrics is something extremely important. And any change being rolled out [to production], its metrics are monitored, basically to see if there's any performance improvement or issues with it. The average change being rolled out is done with canary releases. There is a lot of automation about these things, and it's very metrics-driven".* (MovieStream, Platform engineer #1)

The fourth and last characteristic is *success* (and its measurement), which we define as the measure the team aims to achieve to fulfill its task. For example, AutoBank has siloed success criteria where a development function focuses on completing many product features and the operations function focuses on stable production systems. In contrast, at MovieStream the approach to success is holistic and aligned with the experience of the user. They strongly measure how the customer uses the product and which business opportunities are possibly missed. Informants across all cases report efficiency as a business goal and DevOps as the enabler to get the speed to drive digital transformation.

4.4.2 The DevOps Continuum's "Outer Environment"

We uncovered from our data that the outer environment of DevOps implementations includes the characteristics market type, product type, and digital technology. *Market type* subsumes the characteristic that defines attributes of the demand, that is the market pull. This may lead to a growing, fast changing business. The market type determines market conditions the team must face during their daily work. For example, in regulated markets and industries, it may be required by law or regulation to authorize changes in the company's application systems, that staff must be trained before a new release or version of the IS can be put to production, that the team has to store all data of the application systems in data centers in a specific region, or that the IS product is to be hosted on production systems located in the customers' technical network and must thus align with their compliance rules. A DevOps lead architect provides an example:

> *"Big companies ask for changes, but they might be stacked in order to check there on-site against local compliance tests, and then they are deployed to facility. Processes are different to ours, since we do not run our software".* (ManuFact, DevOps lead architect #3)

Unsurprisingly, we find that flavors of this characteristic differ widely between "consumer markets" and (online) business-to-customer products compared to "industry markets" and business-to-business products. For instance, to contrast to ManuFact, MovieStream is operating in a less-regulated consumer market; hence deploys to production multiple times a day, and can quickly rollback in case of any issues. The deployments strongly rely on automation.

We define the *product type* as the identifying feature or the core affordance of the provided IS product. For example, for MovieStream, the provided product allows users to consume streamed movies. In this case, the resulting IS product needs to be resilient and simple. Resilient means it acts robust against failure situations and recovers from failures quickly. Simple means that MovieStream focuses on the feature that is directly available for the user. This very intense ”product focus” stands in contrast to an enterprise focus for products, for example ERP systems used in enterprises. The product focus at MovieStream leads to few and simple use cases such as the ability to sign up for the service and the ability to press the play button.

The last characteristic of the outer environment is *digital technology*. We define this as the digital technical means to develop and deliver the IS product. Customers ask for new products and new features regularly, and nowadays the team can typically fulfill this demand without facing many technical limitations. This is the balance between “market pull” and “technology push”, as a DevOps lead architect from ManuFact explains:

> *“Software is used in many more industrial areas, that is the market pull, and on the other side we have the technology push, many data, and no restrictions such as memory, network bandwidth, or solutions in the cloud. The technology pushes. Digitization leads to many innovations and changes, I want to react on quickly. Agile software development is the development and DevOps includes the Ops part, and this helps the entire business to stay agile, to react on changes. If you don't have this in the digitization age, ... it is like a life policy.”* (ManuFact, DevOps lead architect #2)

The used digital technology needs to be managed. That is, although teams may have the autonomy and freedom to choose the technology they use to accomplish their work, they may use platforms and services, centralized in distinct organizational entities, to work efficiently and leverage their own resources on working on their IS product.

4.4.3 DevOps as a Funnel to Link Efficiency and Effectiveness

The data-based analysis of the teams' environments and what is perceived as “IS success” offers an interesting lens on efficiency and effectiveness. On the one hand, DevOps is a means to develop and deliver a product in an efficient way, e.g. by continuously delivering new versions of the IS. On the other hand, DevOps bridges to effectiveness, because continuous delivery and automatically gathered insights derived from usage of the IS on production systems, support the teams to adjust, extend, or remove current IS functionality, or even find entirely new business opportunities:

> *"Actually, you do not want to ship features, you want to generate outcome for the users. If it is easy to deliver output, then you start to think about the value of this output."* (ManuFact, Dev manager)

Continuous delivery leads to a "shift left", with development- and test-related activities running on the technical system the customer accesses to use the IS. The practice of continuous delivery spans involved IT functions of IS development and operations. It includes continuously probing the team's environment to check whether the team should re-align based on given changes in its environment. Part of this is "sharing" across teams. Sharing emphasizes the context, as platform engineer of MovieStream points out while talking about the relationship to other autonomous teams, and his roles as a leader:

> *"I'm going to go over and talk to them, because I understand what they're trying to do, but understand how they're not going to get there. So, that context sharing is part of the responsibility of leaders within the organization to make sure that there's no information that stays so localized that other teams can't either leverage it or make use of it or understand what's going on. So, it actually required good communication and curation of information in leadership, so the teams can continue to operate in their locally specialized area, but not be disadvantaged. That's the socio part of our socio-technical system."* (MovieStream, Platform engineer #3)

Meanwhile, the team also must be able to deliver faster in order to gather feedback faster. Continuous delivery of this kind appears to get more difficult if the product type does not support that (e.g., if the IS product is not resilient or not enough meaningful metrics are available due to a low volume of user requests) or the software is produced for and runs in industry markets (e.g., if the IS product is hosted at the customer and must confirm to their respective compliance rules, is shipped in an embedded way together with hardware, or many stakeholders are involved). A DevOps lead architect provides an example:

"We have this with our trains. If you deliver the IS product together with hardware, then it is getting difficult. Because then also the customers are specific, then you cannot just simply automate, because too many stakeholders are involved" (ManuFact, DevOps lead architect #1).

According to the perception of the studied cases, as evidenced by data indicating organizational standards, the more autonomously a team is, the faster and more flexible it can react based on gathered information and changing context. Our data analysis uncovered that with DevOps and high-performing teams, the fine-grained, autonomous team is an agent that can, based on simple rules, continuously adapt itself to fit to the changing inner and outer environment, and organically realign with its collaborative connections to other teams, while executives dynamically refit the architecture of teams, through adding, eliminating, combining and splitting teams. This happens under the permanent weighing the organizational balance between efficiency and effectiveness and probing new organizational standards. As a service manager from AutoBank states:

"Don't just give the autonomy to this small unit, rather also enable the team. Give them the methodology technically and socially. […] Create small units and empower them to decide themselves and support their collaboration with other capsules, to the other teams." (AutoBank, Service manager)

Digital technologies, including those provided by other teams, can support the respective team to fulfil its tasks, but they can also thwart the team in case of restrictive organizational standards. If the team is empowered to decide on its best balance between efficiency and effectiveness, it steadily realigns with its inner environment. This brings the team closer to the edge of the right side of the continuum. For example, at MovieStream and SoftwareDev, teams may even completely bypass centralized platform products and services if they have good reasons to do so and can ensure that they still deliver aligned with operational requirements as part of organizational standards. From the team perspective, products and services provided by another team maybe considered to be "waste" in lean thinking and in this specific context, although, from an enterprise viewpoint, synergies may be missed. A product owner for tools from SoftwareDev emphasizes the continuous trade-off:

"If a team comes with new ideas, new languages, until the whole organization catches up with them, you need to find a way to support them. So, we invest into making easy what most teams

are using. But on the other hand, we want to go in the direction of supporting what's coming in. And that's always a tradeoff, how much do you invest in that." (SoftwareDev, Product owner tools #2)

Focusing too much on supportive activities that are already provided by other (platform) teams, may bind resources that could otherwise contribute to the creation of valuable outcome for the customer. In summary, empirical evidence suggests that DevOps is a vehicle to optimize the balance between efficiency and effectiveness.

4.5 Discussion

Our research aimed to shed light on which characteristics influence DevOps implementations. Similar to enterprises who customize AISD (Fitzgerald et al., 2006), enterprises tailor their respective DevOps implementations depending on their context. Surprisingly, we uncovered a broad set of characteristics, with typical flavors, each of them related to how DevOps is executed in a concrete context. What we observed is that even in one organization many different DevOps implementations may exist. The reason for this is, due to our analysis, that in one team, its environment may be very different compared to the environment of another team, both in one enterprise.

Our results show that variations of DevOps implementations are gradual transitions on a DevOps continuum between two edges, siloed IT functions of development and operations on the left, and full coalescence on the right edge. Coalescence practically means that an autonomous team (Lee and Xia, 2010) is fully responsible for developmental and operational concerns during IS development and owns the IS product (Cao *et al.*, 2009).

Informed by the DevOps continuum, we unfold a streamlined role of success in DevOps initiatives. While implementing DevOps, goals and the definition of success evolve. Success can be conceptualized either from the developmental side (Siau et al., 2010), e.g. staying within scope, time, and cost requirements (Chow and Cao, 2008), functionality (Lee and Xia, 2010), rapid change (Lee and Xia, 2005), or delivering high quality (Siau et al., 2010), or from the operational side, e.g. the industry standard "mean time to restore" (Dekleva, 1992), MTTR in short, as the time required to restore the IS product after an incident occurs that makes the system unavailable. One of the most influential contributions (Urbach et al., 2009) focusing on the operational side of success is the IS Success Model (ISSM) (DeLone and McLean, 2016; Petter *et al.*, 2013).

The characteristic success relates to the reasons why DevOps is implemented at all. All cases worked with defined goals and strive for success. Cases articulated the strong importance of both instrumental as well as humanistic goals, indicating that *instrumental goals* are typically related to *delivery performance* (through continuous delivery) that is how fast and often an IS product can be made available to the user (Forsgren, 2018; Poppendieck and Poppendieck, 2003; Wetherbe and Frolick, 2000), a leading success criteria of DevOps implementations on the right side of the continuum. *Customer satisfaction* (DeLone and McLean, 2016; Forsgren *et al.*, 2016) can be considered to be a *humanistic goal* (Sarker et al., 2019). Thus, evidence exists, that customer satisfaction (Bhattacherjee, 2001; Fowler and Highsmith, 2001; Recker et al., 2017) becomes a leading success criteria, replacing more siloed measures for success, either from the developmental side or from the operational side, and emphasizing the concept of “user satisfaction” of the ISSM. Instrumental goals and humanistic goals are both holistic (spanning the IT functions of development and operations) and in its conjunction emphasize the sociotechnical nature of IS (Sarker et al., 2019). DevOps aims to reconcile the plethora of different approaches trying to measure IS success and ISD success.

Both, efficiency as well as effectiveness, can be a goal and thus define success. Traditionally, enterprises aim to optimize efficiency through cost synergies, e.g. to share given resources to serve a specific market, and single organizational entities, typically business units (BUs), aim to optimize their respective effectiveness. Those (with DevOps often fine-grained) units often decide which market to serve and find efficient ways to do so as long as the decision generally fits into the corporate strategy (Eisenhardt and Piezunka, 2011; Greenwood et al., 2020).

While DevOps at the first glance optimizes the efficiency of how teams develop and deliver the IS (the focus on delivery performance), it also optimizes effectiveness (the focus on customer satisfaction). Both concepts are intertwined. A team can perform efficiently while working not effectively (e.g., continuously delivering the wrong product) and vice versa (e.g., delivering the right product only once a year). High-performing teams strive for an optimal balance between efficiency and effectiveness. Continuously finding an optimal balance can only succeed if change is inherent ingredient of the process, “time-paced” evolution is in place and change is “learning-based” (Brown and Eisenhardt, 1997; Conboy, 2009), that in turn is a prerequisite to continuously deliver valuable IS products (Abrahamsson et al., 2009; Brown and Eisenhardt, 1997; Markus and Robey, 1988; Todnem By, 2005). With DevOps instances on the right side of the continuum, e.g. teams of SoftwareDev and MovieStream, platform teams aim to optimize efficiency while serving as an enabler for product teams with their respective foci on effectiveness.

In summary, DevOps is a funnel to link efficiency and effectiveness, for example, through insights gained from monitoring, fast feedback through continuous delivery, or continuously adapt to changes in its environment. Following this logic, DevOps is an enabler for digitization, it supports teams to concretize and continuously update the functional scope of the developed and delivered IS, and it includes teams permanently asserting their environments to deliver their “right outcome right”. The study reveals that DevOps optimizes a balance between efficiency and effectiveness, since a team on the right side of the continuum continuously (“efficiently”) delivers valuable outcome to the customer (“effectiveness”) (Bosch, 2019).

4.6 Conclusion

This research adds significant and timely understanding of the emerging DevOps phenomenon to the academic field. First, it identifies characteristics that influence DevOps implementations. Characteristics, with their flavors, lead to DevOps implementations that can be positioned on a broad range that makes up the DevOps continuum. This model can support to explain existing and to predict future DevOps implementations. Second, this research contributes to existing concepts and theories. We uncovered the importance of “success” as a characteristic of the team’s inner environment, and the role of DevOps as a funnel to link efficiency and effectiveness. We are convinced that the results of this study help to further conceptualize DevOps and to unpuzzle the emerging DevOps phenomenon, and through the theoretical reasoning also contributes to understanding and theoretical grounding of the studied phenomenon. For practitioners, the suggested model can provide valuable guidance for initiatives towards DevOps. After we contributed to the basic understanding of DevOps and characteristics influencing implementations, future research might replicate this research to improve our confidence of generalizability of our findings. We call for studying other DevOps implementations, relate them to the continuum, and assess and expand the suggested model with its characteristics. We expect more possible lines of research about the DevOps phenomenon, once the theoretical grounding exists. We are convinced that our research is a valuable contribution into this direction, to gain understanding about DevOps as a logical evolution of AISD (that spans agile approaches across IT functions) and which characteristics influence their respective implementations.

5 Paper III. "The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations".

Hüttermann, M., & Rosenkranz, C. (2023). *The DevOps Funnel: Introducing DevOps as an Antecedent for Digitalization in Large-Scale Organizations*. Proceedings of the 56th Hawaii International Conference on System Sciences, pp. 5603–5612. https://hdl.handle.net/10125/103317

Online appendices:
https://osf.io/bkrhz/?view_only=84493d8e2b8743be97754abd2717ee93

THE DEVOPS FUNNEL: INTRODUCING DEVOPS AS AN ANTECEDENT FOR DIGITALIZATION IN LARGE-SCALE ORGANIZATIONS

Business productivity and speed to market are among the top priorities of IS managers to stay successful. To achieve these goals, a change in business processes or models is often required, which is often linked to the phenomenon of digitalization. Enterprises have observed that a holistic approach to agile IS development is essential to enable this change, leading to the concept of "DevOps". While past studies have delivered first insights about DevOps, an understanding of which factors are important to introduce DevOps in organizations, and how DevOps relates to digitalization is still missing. To close this gap, we conducted a two-staged study of literature review and a multiple-case study. Our findings suggest that DevOps links success and practices for development and operations across actors of different organizational levels. We find that DevOps supports digitalization efforts, contribute to the understanding of the DevOps phenomenon, and identify worthwhile paths for further research.

5.1 Introduction

Business productivity, cost reduction, and speed to market (Luftman and Zadeh, 2011) are among the top priorities of information systems (IS) managers to stay successful. However, the traditional division of labor in IS and its split into distinct organizational subunits often hinders organizations to successfully pursue these goals (Hemon-Hildgen, Rowe, *et al.*, 2020). In industry, proponents of the recent DevOps phenomenon argue that it brings together team members from both, IT development (those that build IS) and IT operations (those that run and maintain IS) in order to implement IS development (ISD) in a comprehensive way (Qumer Gill *et al.*, 2018; Wiedemann *et al.*, 2020).

With DevOps, those two major IT-related enterprise functions increasingly apply shared goals and use shared practices across both functions (Conboy, 2009; Hüttermann, 2012; Sharp and Babb, 2018). Thus, DevOps appears as a logical extension of agile ISD (Hemon-Hildgen, Lyonnet, *et al.*, 2020; Hemon-Hildgen, Rowe, *et al.*, 2020; Wiedemann *et al.*, 2020) and as such a means to scale application of agile ISD (AISD) across the organization (Hüttermann, 2021). Consequently, companies applying DevOps streamline their ISD and IS operations to overcome well-known barriers and friction points between those two often siloed IT functions (Wiedemann *et al.*, 2020).

As part of the root metaphor for strategy in digital innovation (Nambisan *et al.*, 2020), DevOps is a new and promising path to support business-IT alignment in order to enable organizational agility (Horlach *et al.*, 2020). Enterprise-wide organizational agility (through sensing and responding) supports achieving management goals in turbulent environments, including strategic flexibility and market orientation (Overby *et al.*, 2006) Business-IT alignment is an influencing factor for enabling organizational agility (Tallon, 2007), which in turn is an important antecedent for digital innovation (Osmundsen and Bygstad, 2021). Stemming from agile ISD, DevOps expands agility across IT functions; moreover, with business and IT roles being an explicit part of an autonomous DevOps team, it supposedly goes even beyond business-IT alignment, and may achieve the sought-for tight fusion of business and IT (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013).

First studies in IS have examined the emerging DevOps phenomenon. For example, existing research has proposed a tripartite model of intra-IT alignment (Wiedemann *et al.*, 2020), examined the orchestration of automation, sharing, and risk management in DevOps teams and their relationship to work conditions and job satisfaction (Hemon-Hildgen, Rowe, *et al.*, 2020), or has studied which characteristics influence how DevOps is tailored in organizations (Hüttermann, 2021). While these studies offer very valuable preliminary insights about DevOps, their research questions are grounded on already implemented and existing DevOps implementations (Hemon-Hildgen, Rowe, *et al.*, 2020; Hüttermann, 2021; Wiedemann *et al.*, 2020) and focus on particular aspects such as how alignment in DevOps teams can be achieved (Hemon-Hildgen, Rowe, *et al.*, 2020), or how DevOps affects job satisfaction (Hemon-Hildgen, Rowe, *et al.*, 2020). However, we are still missing a foundational understanding of DevOps in general, how DevOps supports digitalization, and which factors are important to initially implement DevOps in organizations. Consequently, we ask the following research question: *"Which factors are important to initially introduce DevOps in organizations, and how does DevOps relate to digitalization?"*.

To answer our research question, we followed a two-stage approach. First, to identify dominant concepts of the DevOps phenomenon relevant for our study, influenced by (Webster and Watson, 2002), we conducted a literature review. The results served as a baseline from past studies of the IS community on DevOps (or covering or mentioning it in a relevant way) and to abstract provisional codes as input for the second stage of our study. Second, we conducted a multiple-case study of seven organizations implementing DevOps. Based on provisional codes of the first stage and analyses of the case data, we theorize what we call the “DevOps funnel”, which orchestrates the three core concepts success, actors, and practices, and bridges success and practices across different organizational levels to implement DevOps and foster digitalization.

We contribute to the understanding of DevOps by shedding light on DevOps’ implementation in organizations and the role of DevOps in digitalization. By drawing on existing concepts and suggesting the DevOps funnel as a new concept, we unpack the role of the emerging DevOps phenomenon in digitalization initiatives. Our results are of value to practitioners who want to start a DevOps initiative and might suffer from an unclear systematic knowledge of the role of DevOps and missing guidance of which factors are important to initially implement DevOps in organizations spanning multiple organizational levels.

Next, we discuss the related work of our study. Afterwards, we describe our research method and present our main findings. We summarize our contributions by suggesting two propositions. We conclude with avenues for future research.

5.2 Theoretical Background

5.2.1 Towards Studying Implementations of DevOps

DevOps can describe different things, including team structure, success criteria, concepts, or tooling (Qumer Gill *et al.*, 2018). Since it suggests a relation to collaboration across functions, innovation and agility, we draw on the DevOps definition suggested by (Hüttermann, 2021) that emphasizes DevOps as way of collaborative work between functions to continuously deliver valuable outcome. This also poses a bridge to optimize overall IS success: DevOps aims to reconcile the plethora of different approaches trying to measure IS success and ISD success (Hüttermann, 2021). Adding to AISD, DevOps aims to include operational factors, the resulting IS product or its long-term operation and use (DeLone and McLean, 1992; Fitzgerald and Stol, 2017; Petter *et al.*, 2013). With DevOps, siloed goals, for example, amount of functionality (Lee and Xia, 2010) as a developmental goal, or “mean time to restore” (Dekleva, 1992) as an operational goal, are becoming less important. Instead of these instrumental goals, holistic and humanistic goals, for example, rapid change (Lee and Xia, 2005), or customer satisfaction (DeLone and McLean, 2016; Petter *et al.*, 2013), are often preferred.

While DevOps is a recent phenomenon, past studies so far have resulted in the proposition of three distinct theoretical models to explain various aspects related to DevOps. First, a theoretical model of job satisfaction suggests that the orchestration of automation, sharing and risk management is moderated by work conditions and positively impacts job satisfaction if AISD teams move towards DevOps (Hemon-Hildgen, Rowe, *et al.*, 2020). Second, a model on intra-IT alignment extends operational alignment's focus on IT infrastructure and processes to alignment of development and operations functions (Wiedemann *et al.*, 2020). Third, a model on environmental context of DevOps implementations suggests implementations to be gradual transitions in between the two extremes "detached" and "full coalescence" of integration of development and operations functions (Hüttermann, 2021).

Although these three models provide very valuable preliminary insights on DevOps and some of its probable characteristics, mechanics, and effects, they offer only a narrow focus on very specific effects (job satisfaction) and mechanisms (intra-IT alignment), and are based on a single case study (smaller software products such as online shops), or empirically center on environmental characteristics of DevOps implementations (Hüttermann, 2021), respectively. Moreover, these studies all have investigated effects of existing DevOps implementations (Hemon-Hildgen, Rowe, *et al.*, 2020; Hüttermann, 2021; Wiedemann *et al.*, 2020). While one study we know of has investigated barriers while adopting DevOps (in small to medium-sized enterprises) (Krey, 2022), the reasoning behind the initial introduction of DevOps in large-scale situations or an evaluation of its effects using empirical data is missing.

5.2.2 DevOps in the Digitalization Age

DevOps is part of the root metaphor for strategy in digitalization (Nambisan *et al.*, 2020). While digital transformation is a "broad and complex phenomenon that does not fit easily into a given theory" (Furr *et al.*, 2022), challenging a "separation of disciplines" (Tilson *et al.*, 2010), it can be disaggregated into concepts where actors across organizational teams and levels play an important part in (Furr *et al.*, 2022;

Nambisan *et al.*, 2020; Osmundsen and Bygstad, 2021) to achieve success (Furr *et al.*, 2022; Hüttermann, 2021) – similar to DevOps.

Similarly, digital innovation (and, more broadly, digitalization) can be defined as the creation of and change in market offerings, business processes, or models that result from the use of digital technology (Nambisan *et al.*, 2020). Innovation is encompassing and includes the process of developing and implementing new ideas (Van de Ven *et al.*, 2008). In the digitalization age, innovation can be seen as a self-enforced mechanism that, based on a space of possibilities and assembly of components, may lead to new products and services in IS infrastructure, the installed base of organizations, systems and users (Bygstad *et al.*, 2016). Generative innovation is based on a composition of technology and users and a lightweight IT that is well suited for a specific task and provides immediate value for the user (Bygstad, 2017).

The evolution of the composition of technology is the process by which managers initiate and implement changes in an organization for increasing the alignment between its IT resources and strategic imperatives (Henfridsson and Bygstad, 2013; Nambisan *et al.*, 2020). It is a means to stay competitive in its decoupled structure of actions (Greenwood *et al.*, 2020). AISD emphasizes agility on the team level (Horlach *et al.*, 2020). However, to digitally innovate as an organization to continuously deliver valuable outcome to customers (Fowler and Highsmith, 2001) and to improve the speed of IS delivery (Bharadwaj *et al.*, 2013), agility has to be perceived as an enterprise-wide concern (Horlach *et al.*, 2020). The ability of enterprises to sense environmental change and respond readily is called enterprise agility (Overby *et al.*, 2006). This ability across functions connects multiple actors: The effectiveness of an interconnected, historically grown IS is conditioned on an installed base of extant socio-technical arrangements (Henfridsson and Bygstad, 2013; Yoo, 2013), with innovation emerging from interactions of actors – either interaction of different IT specialists across roles or interactions of powerful users with IT product specialists (Bygstad, 2017).

Business-IT alignment (Gerow *et al.*, 2014; Reynolds and Yetton, 2015) can be understood as an influencing factor for enabling this organizational agility (Horlach *et al.*, 2020; Tallon, 2007). Business-IT alignment describes the orchestration of the separate entities of business and IT to have them work together towards a common goal (Luftman *et al.*, 1999), as a convergence of strategies of business and IT (Bharadwaj *et al.*, 2013; Overby *et al.*, 2006).

Linking all of this to DevOps, the orchestration of ISD in contemporary digital systems is deeply impacted by alignment of its decoupled structure (Wiedemann *et al.*, 2020), above all development and operations (Hüttermann, 2012), and the evolution of its environment (Hemon-Hildgen, Rowe, *et al.*, 2020). The concept of DevOps is a promising path to contribute to business-IT alignment (Gerow *et al.*, 2014) and to reconcile different alignments models (Reynolds and Yetton, 2015; Tallon and Pinsonneault, 2011). While expanding AISD toward DevOps, autonomy of teams is increased by having parts of the business (rooted in AISD) and operations within the team (Fitzgerald and Stol, 2017; Hüttermann, 2021). As digitalization (Sandberg *et al.*, 2020) continues, companies therefore increasingly implement DevOps in order to become more agile (Hemon-Hildgen, Rowe, *et al.*, 2020), to constantly innovate (Hüttermann, 2021), and to achieve better business-IT alignment, where alignment is ranged on a continuum including shared goals and practices between functions (Hüttermann, 2012), assisted DevOps, to full coalescence (Hüttermann, 2021), that, in its extreme, may lead to a fused relationship of business and IT (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013). Due to the emergent role of DevOps and its relationship to business-IT alignment, there are increasing calls for studies on successes and failures of implementing DevOps, specifically in large-scale enterprises (Maruping and Matook, 2020). Considering the lack of research about DevOps in general, and the missing understanding of dominant factors while introducing DevOps in large-scale organizations, and how this relates to digitalization, our goal is to theorize which factors are important to initially introduce DevOps in organizations to shed light on the role of DevOps in digitalization efforts.

5.3 Research Method

5.3.1 Literature Review

First, we conducted a review on IS literature to identify dominant concepts derived from past studies with relevance to our research question. Influenced by (Webster and Watson, 2002), literature was selected to identify concepts, and provide provisional codes as input for data analysis of the second stage of our study in form of a multiple-case study. The literature review is split into four phases: screening, filtering, testing, and packaging the final basket of literature. Screening included determining useful databases, identifying the search term, and developing the search strategy (vom Brocke *et al.*, 2015). A recent literature review in IS research (Sharp and Babb, 2018; Xu and Ramesh, 2007) ends with presenting an inventory of current publications. To focus on publications that combine high level of scientific rigor in IS research with strong practical relevance in the IS domain, we build upon the existing study (Sharp and Babb, 2018). This way, we accumulated a "relatively complete census of relevant literature" (Webster and Watson, 2002) of the IS domain aligned with our research design. The initial, complete background literature review also included other publications, for example, those of the software engineering domain and industry. Since the IS literature references these types of publications as well and to focus on the IS research stream (cf. Sharp and Babb (2018b)), we focused our literature review of stage one on high-quality IS publications to extract relevant literature and the provisional codes for stage two (vom Brocke *et al.*, 2015).

The initial background literature review unveiled that, although growing, the number of articles about the emerging DevOps phenomenon is relatively scant. For this reason, the search term was "DevOps" while we targeted peer reviewed papers, that were written in English (vom Brocke *et al.*, 2015). Via AIS eLibrary (cf. Sharp and Babb (2018b)), we searched in proceedings of AIS conferences plus the highly ranked, affiliated HICSS conference, to identify recent work of IS research covering the emerging DevOps phenomenon. Via EBSCOhost, we added relevant journal articles of the IS community. As part of the filtering phase, we fully read the resulting set of publications, checked how the subject domain was covered, and analyzed for relevance (Bandara *et al.*, 2015). Analysis entailed identifying importance of the information being presented in respect of our research question (Bandara *et al.*, 2015).. This testing for applicability led to 23 articles in proceedings and 8 articles in journal outlets.

In the last phase, a forward/backward search uncovered four more yet undiscovered papers (according to our quality criteria), leading to a final list of 35 publications. Once all articles went through all of the above phases, the publications were streamed into our concept matrix (Webster and Watson, 2002) which emerged successively. Closing the last phase, we reordered lines and columns. The initial review of the background literature (the body of knowledge on DevOps, and digitalization) as well as the literature review (identifying the provisional codes for the second stage) were executed by one researcher and multiple student assistants. A graphical summary of the literature review and the concept matrix with all selected publications are available as Online Appendix A (Figure 5-2) and B (Figure 5-3)[9].

5.3.2 Multiple-case Study and Data Collection

To achieve our goal to further unpuzzle the emerging DevOps phenomenon and its relationship to digitalization, the concept matrix from stage one identified dominant, relevant concepts as provisional starting codes for stage two. For stage two and as part of an overarching multiple-case study, we sampled a set of seven cases.

A multiple-case study is considered an appropriate method to build and extend concepts and theory (Dubé and Paré, 2003; Yin, 2018). Our cases are spread across industry segments including online entertainment, manufacturing, software, and finance. Cases are large-sized, international organizations. We expected them to go through a thorough decision-making process in complex organizational setups before they introduced DevOps. We started our study with the organization being the unit of analysis. As inside organizations several different DevOps implementations may be in place, the unit of analysis transitioned to teams developing and delivering IS products.

The main data source was interviews. Participants of the semi-structured interviews were highly knowledgeable informants spanning roles of developers/engineers, architects, and decision makers, see Table 5-1 for an overview of the demographics (names of organizations are anonymized for confidentially reasons). We followed a predetermined interview guideline emphasizing the narrative character of the open-ended interviews, and remained open for emerging themes (Brinkmann and Kvale, 2014). Case writeups were a complementary vehicle for theorizing and to triangulate evidence across participants and cases (Dubé and Paré, 2003; Yin, 2018).

Organization	Segment	Interviews
EntertainCorp	Online products	3 Platform engineers
StampCorp	Manufacturing	1 Manager, 1 Platform engineer
CashCorp	Insurance	1 Chief digital transformation officer, 1 Application domain officer
SoftwareCorp	Software	2 DevOps engineers, 3 Product owners, 1 manager
ManuCorp	Manufacturing	3 DevOps leads, 1 manager

9 The location of all online appendices is: https://osf.io/bkrhz/?view_only=84493d8e2b8743be97754abd2717ee93

LiquiCorp	Banking	3 Manager, 1 Dev engineer, 1 Product owner
InsureCorp	Insurance	1 Digital transformation manager, 1 IT manager, 1 Project manager, 1 Coach

Table 5-1. Multiple-case study: demographics.

5.3.3 Data Analysis

Following (Saldaña, 2016), we applied different coding strategies, see the coding process with illustrations (excerpt) in Online Appendix C (Figure 5-4). First, as part of stage one, we executed our literature review influenced by (Webster and Watson, 2002). It was assisted by applying open coding (Miles *et al.*, 2020) to identify the dominant concepts of each publication relevant for our study. Second, we applied theoretical coding (Glaser, 1978; Saldaña, 2016) as a second cycle coding method to model the integration of codes. We integrated the codes with concepts from the background literature. The resulting concept matrix in turn served as provisional starting codes for the second stage.

In the second stage, we applied provisional coding (Saldaña, 2016) as an exploratory coding method to build theory from case studies. Our predetermined list of provisional codes was extended, augmented, and integrated with new, emergent codes if needed. The analysis of the qualitative data was further inspired by the coding family "process" (Glaser, 1978; Saldaña, 2016). We found that family helpful to theorize about dominant factors to introduce DevOps in large organizations (Van de Ven and Poole, 1990). A "process" is a way to group together sequencing steps to a phenomenon. If one factor is more dominant than other factors, or is part of a sequence, this offers hints for introducing DevOps.

The concepts were the basis for our theory building (Sarker *et al.*, 2013) as a "thought trial" (Weick, 1989). Informed by our DevOps definition and the integration with background literature (see Table 5-2), this led to our primary concept of the "DevOps funnel". Its representation draws on work of Osmundsen and Bygstad (2021) who identified interaction cycles between management and employees. We asserted our emerging new concept and triangulated evidence across cases (Yin, 2018).

5.4 Findings

Our analysis reveals "success" and "practices" as most dominant to introduce DevOps in organizations. In its orchestration, these concepts act as a funnel to link success and practices across organizational levels, see Table 5-2. Online Appendix D (Table 5-3) provides further details on how the studied teams implement the funnel. We find that DevOps is an antecedent to achieve organizational goals such as "digitalization".

The first dominant factor is "Success", which we define as the goal or the measure an actor aims to achieve. Our study reveals that a team may have success criteria originally rooted in the team itself ("bottom-up"), for example, that value has to be delivered to customers in high frequency to increase customer satisfaction ("continuous delivery"). If defined "top-down", success is articulated to the team as a criterion, requirement, or as part of rules and norms given by the organization and its management, for example "digitalization", cultural change transformation, or shared economics to support teams to find the next most valuable thing to work on.

The second dominant factor is "Actors". Actors can be internal producers or consumers of an IS product, or producers of products or services needed by the team, or other parties inside the organization defining rules and norms that influence how the team produces their IS product. Actors span different organizational levels (management, employees) and functions (business, IS development and IS operations). At the organizational level, the upper management consists of actors who initialize practices and define success (e.g., digitalization). On a team level, with its roots in AISD (including development and the business), DevOps adds the operations function.

The third dominant concept is "Practices". Practices concerns the methods defined or used by actors to develop as well as deliver IS products. Practices are either rooted in the team ("bottom-up"), for example "automation" and "continuous delivery", or methods or initiatives originated by management ("top-down"), for example "sharing" across functions. As part of practices, teams utilize digital technology, see Online Appendix E (Table 5-4) for details per case. The three concepts "Success", "Actors", "Practices" are intertwined. The concepts Success and Practices are funneled top-down and bottom-up across organizational levels. We define this orchestration as the "DevOps funnel", see Figure 5-1. The "top-down" and "bottom-up" directions of the funnel visualize that functions not only align with each other, but rather "fuse" to autonomous DevOps teams. As part of DevOps, besides developmental and operational roles, the team also contains business roles. In its extreme, this leads to a fusion of business and IT, horizontally (fusion on the team level) and vertically (fusion across organizational levels).

In these cases, for providing value to customers in short cycles, IT and business are deeply integrated.

Concepts	References from background literature	Short descriptions	Short descriptions Funnel perspective	Exemplary quotes from case study
Success	DeLone and McLean, 1992; Hüttermann, 2021; Petter *et al.*, 2013	A goal and its measurement	Top-down: DevOps is a means to achieve organizational goals, e.g. sharing or digitalization.	*"Once you have established DevOps, it is definitely an accelerator for digital transformation."* CashCorp, Chief Digital transformation officer
			Bottom-up: Goal, and its measurement, the teams aims to achieve.	*"Cause it's the intention of the way of working that you deploy, that you add, that you constantly provide business value. […] Having short cycles, small portions, and adding business value every time, it really helps in customer satisfaction."* LiquiCorp, Product owner
Organi-zational actors	Greenwood *et al.*, 2020; Hemon-Hildgen, Rowe, *et al.*, 2020; Osmundsen and Bygstad, 2021	Human individuals (or a group of) as a source of action, and as such an entity that interacts with other actors, and the IS. Either on the team level (employee) or part of top management.	Top-down: Upper management, who initialize practices, define success, or their "buy-in" to those.	*"Digital transformation is innovation, automation and education. […] It is pushed by upper management. Our COO is an IT person. He wants to rebuild our IS. The teams always wanted it, but without having someone in the top management, it will not work."* CashCorp, Chief Transformation Officer
			Bottom-up: Actors at team level, who use and/or initialize practices, and act with others, while striving for goals.	
Practices	Hemon-Hildgen, Rowe, *et al.*, 2020; Hüttermann, 2021; Wiedemann *et al.*, 2020	Rules and practices that define the relationship and organization among actors and the IS.	Top-down: Practices defined or initialized by upper management.	*"We have programs initialized by C-Level executives. One past program run a cultural change transformation, and DevOps was part of it."* SoftwareCorp, DevOps engineer
			Bottom-up: Practices initialized and applied by the team.	*"Within minutes, that new code can be taking customer traffic, it's typically hours before that new code is servicing all of customer traffic."* EntertainCorp, Platform engineer

Table 5-2. The Concepts of the DevOps Funnel.

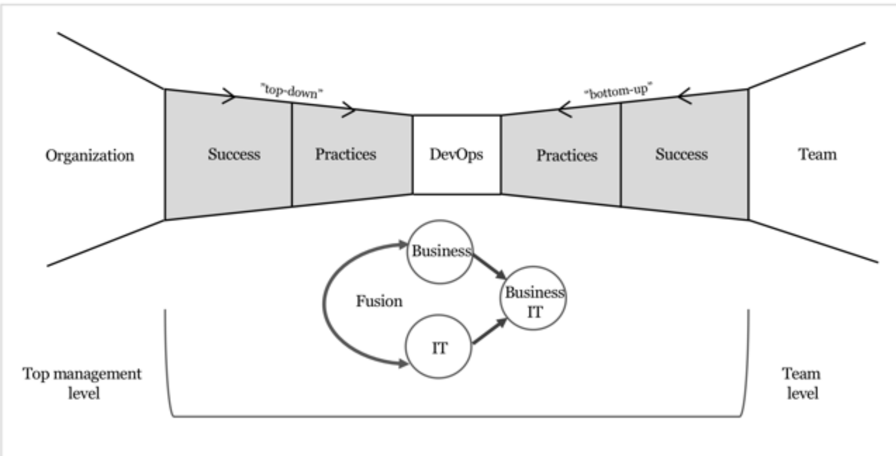

Figure 5-1. Representation of the DevOps funnel.

A manager explains the necessity of short iterations to stay innovative, with many actors included across functions and organizational levels:

"You have to keep innovating. Before, projects took years. Why can't I do that in small steps? So, give the customer something a little bit every time. And the good thing about that is that you get the feedback from users immediately, […] sometimes you only need to deliver 20% of the features and the twenty percent delivers 80% of the value. […] before we had the business department, with their own CEO. And they were delivering the business and they were telling us what to, but we skipped that. Now the business is part of the team." (LiquiCorp, Manager)

This leads to teams of IT and business who are fully responsible for their product.

"Now the teams are fully responsible for everything. We do not have a segregated responsibility for production or another team is responsible for the product. They are together fully responsible. So, it is a collaboration not a segregation model." (LiquiCorp, Manager)

If the goal of an actor is to deliver valuable outcome to customers, related actors must be identified to reach that success criterion (the customer and other internal teams), and practices have to be employed that enable the team to reach its goal (e.g., continuous delivery IS products). For example, if the team's success criterion is to support organization's digitalization efforts (given by upper management decision), actors and practices are identified and employed accordingly. A DevOps lead explains the role of DevOps with regards to supporting digitalization efforts to stay competitive:

"We have to hit market windows of opportunity. Competition makes these windows increasingly shorter. It is an important program for our company to get closer to the customer, to react more agile to their demands. […] Digitalization is driving the entire company. We continuously collect user feedback. We work in short iterations with learning and improving and removing things if they don't work. […] Digitalization is pushed by the top management. They don't define the How. You need a good How in order to deliver the right products. With DevOps, we want to gather feedback what is the right product. And for that, it is of benefit to span the driver digitalization around everything." (ManuCorp, DevOps lead)

With digitalization as on organizational goal given by management (the "what"), DevOps is considered to be the implementation (the "how"). DevOps reportedly serves as an enabler for digitalization efforts of the organization.

5.5 Discussion

Our research aimed to shed light on which factors are important for organizations to introduce DevOps, and how this relates to digitalization. Similar to enterprises who customize AISD (Fitzgerald and Stol, 2017), organizations tailor their respective DevOps implementations. Our findings show that the DevOps implementation of a specific team depends on success criteria, actors, and practices. Those three concepts dominantly influence DevOps' initial implementation in organizations. Depending on the defined success, actors are involved, and practices are employed, in order to fulfill the success criteria in the initial iteration of DevOps implementation. Since the concepts are interweaved (for example, new actors may result in new goals or practices), the team may later iteratively cycle between the steps, once the first DevOps implementation is in place as a foundation for further adaption (Dingsøyr *et al.*, 2012; Lee and Xia, 2010).

With DevOps, success and practices are orchestrated across functions (business, IT development, IT operations) and organizational levels (management, employees). This finding supports previous studies which connect DevOps with the concept of measurement and sharing (Fitzgerald and Stol, 2017; Humble and Molesky, 2011; Hüttermann, 2012) across stakeholders (Hüttermann, 2021), and the importance of practices and actors for innovation and digitalization (Bygstad, 2017). We capture this orchestration of concepts through our first proposition:

> *Proposition 1: Through shared practices across actors from different organizational levels, DevOps positively impacts success.*

We unfold an important role of success in DevOps initiatives. Success can be conceptualized "bottom-up" and "top-down", fusioning teams across different actors and linking organizational levels (Burgelman, 1983; Greenwood *et al.*, 2020; Osmundsen and Bygstad, 2021). Conceptualized bottom-up, holistic and humanistic success criteria are often preferred (see Online Appendix D; Table 5-3), for example to increase customer satisfaction (DeLone and McLean, 2016; Petter *et al.*, 2013) in competitive environments where power shifts to the customer can be observed (Horlach *et al.*, 2020). As a holistic approach to development and operations of IS products, DevOps also intends to link to the "top-down" view: Success in DevOps initiatives conceptualized "top-down", as part of a digital innovation (Nambisan *et al.*, 2020), evolves the installed base of extant socio-technical arrangements

(Henfridsson and Bygstad, 2013), with a generative innovation as essential part of the development culture (Bygstad, 2017). Although intertwined, digitalization is typically an effort on the organizational level (Burgelman, 1983; Greenwood *et al.*, 2020; Nambisan *et al.*, 2020). The team (as the unit of analysis) with its actors introduce DevOps to achieve their goals, and organizational goals (for example digitalization). Supporting the findings of Osmundsen and Bygstad (2021), teams interact with the management by continuous back coupling through sensemaking and supporting ("top-down") and adopting and sense-giving ("bottom-up").

As evidence suggests, DevOps often leads to autonomous teams (Hüttermann, 2021). They are fully responsible for their respective IS product (Wiedemann *et al.*, 2020), with continuous synchronization ("top-down", "bottom-up") between management and employees (Osmundsen and Bygstad, 2021). In DevOps implementations where teams also include developmental, operational and business roles, DevOps positively impacts business-IT alignment (Reynolds and Yetton, 2015). Because business and IT inherently share the same strategy and work together toward shared (business) goals around the customer value flow (Horlach *et al.*, 2020), this goes even beyond a business-IT alignment, and may arguably achieve the sought-for tight fusion of business and IT (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013). This fusion might suggest future paths to reconcile different alignment models (Reynolds and Yetton, 2015; Tallon and Pinsonneault, 2011).

Across all studied cases, the DevOps funnel enables the organizations with its teams to continuously innovate (Alt *et al.*, 2021; Osmundsen and Bygstad, 2021) to champion the continuous innovation (digitalization) of the organization (Osmundsen and Bygstad, 2021), by being able to quickly react in two ways: adaptively to disruptive changes in the environment and entrepreneurially to opportunities in market offerings (Chakravarty *et al.*, 2013), by adapting business models and processes (Nambisan *et al.*, 2020), to maintain or gain a competitive advantage (Bharadwaj *et al.*, 2013). The suggested funnel contributes to scale agility to the entire organization. It is an antecedent for digitalization. Consequently, we posit:

Proposition 2: The DevOps funnel with its autonomous teams does positively contribute to digitalization efforts.

While the funnel streamlines the organization with its teams, with their diverse goals, actors and practices (Davison and Martinsons, 2016; Hüttermann, 2021), our research indicates that its operationalization might look different across organizations in their diverse contexts(Davison and Martinsons, 2016; Hüttermann, 2021). It might be intuitive that the studied case of a market-leading provider of an online product (EntertainCorp) has a fusion of business and IT (actors in autonomous teams sharing success criteria and practices) (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013), and thus inherently maximizes the level of alignment of functions (all functions are fused in one team).

On the other hand, it is surprising that, according to informants of LiquiCorp, a classic bank successfully transformed itself to a "software house with a banking license", thus a fully digital business. In both cases, IT has become both fabric and fusion (Henry Lucas *et al.*, 2013), with DevOps playing an essential part to share success and practices across actors, horizontally and vertically (Reynolds and Yetton, 2015). From another example, in the case of ManuCorp, the industrial sector is again different. As a global manufacturing organization with a diverse product portfolio including power plants and trains, DevOps is reportedly also an antecedent on their digitalization way, however, digitalization with existing long-term inventory is considered to be a challenge ("brownfield digitalization", in the voice of a DevOps lead of ManuCorp). As suggested by Yoo *et al.* (2010), incumbent firms may address this challenge while combining physical and digital elements when digitalizing by applying the practice of using a layered modular architecture, to achieve transformation goals specific for their firm with its actors (Furr *et al.*, 2022).

To implement a modular architecture (and to implement digitalization in general), the installed base of digital infrastructure (Bygstad, 2017; Yoo, 2013) can be a barrier as well. As informants reported, DevOps is more difficult to introduce when older technology is used (see Online Appendix E; Table 5-4). Data further indicates challenges for organizations to find the right balance between IT exploration and IT exploitation while implementing fused teams (Gregory *et al.*, 2015), specific to teams' respective environments, to bridge effectiveness and efficiency (Hüttermann, 2021) .

In summary, DevOps links an organization's digitalization effort (the "what"), defined by management (Furr *et al.*, 2022), to "the how", how the team is developing and delivering the IS product (Gall and Pigni, 2021). These fused teams are fully responsible for their IS product (Bharadwaj *et al.*, 2013; Henry

Lucas *et al.*, 2013; Wiedemann *et al.*, 2020), scale agility across functions (Hüttermann, 2021), and innovate by reacting quickly to changes and business opportunities (Luftman and Zadeh, 2011; Nambisan *et al.*, 2020; Osmundsen and Bygstad, 2021). In its visual appearance, the representation of the suggested concept of the DevOps funnel looks like an enterprise sandglass, a funnel, that is rotated counterclockwise by 90 degrees.

Our study is not without limitations. First, provisional codes are based on selected contributions of past studies of the IS community. Other studies may employ other filter and applicability criteria, broaden the basket of literature and include other publications as well (e.g., local AIS conferences, or publications from other conferences, industry papers, and outlets).

Second, we identified the three dominant concepts success, actors, and practices as the main factors for introducing DevOps. After initially introduced, adaptions of DevOps might lead to other, more, or different concepts. DevOps implementations are dynamic and may evolve over time.

Third, according to our research question and design, we aimed to reconcile existing knowledge of the body of the IS research for theory building, and to empirically assert our findings. Future research might detail the suggested concept "DevOps funnel", for example, by executing an in-depth, longitudinal, single case study research, empirically exploring how the case site does initially introduce DevOps, to detail on the "how" of introducing DevOps (in the sense of change and adaption of methods), and to provide further details on how DevOps funnels the organization and its teams, in a specific context, including small-scale organizations, with barriers and challenges, to implement digitalization.

5.6 Conclusion

Existing literature on DevOps is diverse, and the role of DevOps in digitalization lacks systematic knowledge. To reconcile the growing yet scant set of studies of DevOps, we aimed to review valuable insights of past studies of DevOps from the IS domain, to empirically theorize and assert the suggested concept of "DevOps funnel", and to relate DevOps to digitalization. We contributed to and linked the discourses of research streams on DevOps and digitalization. In spite of our what we think timely and relevant findings to explain the role of DevOps that was imperfectly understood beforehand, and hopefully the valuable guidance it might provide for practitioners, this is not the end. We pointed to possible avenues for future studies on DevOps in order to further investigate the emerging DevOps phenomenon and how it relates to digitalization.

5.7 Appendices of Paper III

This chapter contains the appendices of this paper III as copies of the original appendices located at osf.io.

Online Appendix A

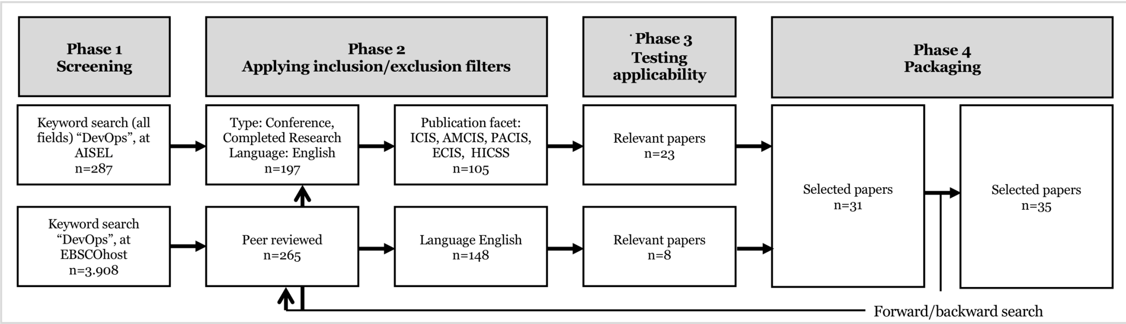

Figure 5-2. Graphical summary of the literature review.

Online Appendix B

						Success							Actors								Practices						
ID	Title	Author	Year	Journal/ Conference	Type	Job satisfaction	Risk management	Reslilience	Time reductions	Better quality	More customer feedback	Supporting digitalization	Develop-ment	Operations	Quality Assurance/ Testers	Compliance/ security	Business (analyst)	Customer	Project managers	Autonomy	Sharing/ culture	Learning	Monitoring/ measurement	Cloud	Continuous integration/ delivery	Automation	Intra-IT alignment / Orchestration
J1	Taking DevOps mainstream: a critical review and conceptual framework	Gall et al.	2021	European Journal of Information Systems	Journal								x	x						x	x		x			x	
J2	Understanding how DevOps aligns development and operations: a tripartite model of intra-IT alignment	Wiedemann et al.	2020	European Journal of Information Systems	Journal								x	x			x			x	x					x	x
J3	Orchestrating automation and sharing in DevOps teams: a revelatory case of job satisfaction factors, risk and work conditions	Hemon-Hildgen et al.	2020	European Journal of Information Systems	Journal	x	x					x	x	x							x						x
J4	An exploratory study on the DevOps IT alignment model	Hart et al.	2020	Interdisciplinary Journal of Information, Knowledge, and Management	Journal								x	x			x	x			x			x	x	x	
J5	Continuous Software engineering: A roadmap and agenda	Fitzgerald et al.	2018	Journal of Systems and Software	Journal				x	x	x	x	x	x	x	x	x		x	x	x		x		x	x	
J6	DevOps for information management systems	Qumer Gill et al.	2018	VINE: The Journal of Information & Knowledge Management Systems	Journal								x	x	x	x					x		x		x		
J7	DevOps in practice: A multiple case study of five companies	Lwakatere et a.	2019	Information and Software Technology	Journal				x	x	x	x	x	x						x	x		x		x	x	
J8	Adopting DevOps in the real world: A theory, a model, and a case study.	Luz et al.	2019	Journal of Systems and Software	Journal			x		x			x	x	x	x	x	x	x		x		x		x	x	
J9	Making Sense of Continuous Development of Digital Infrastructure	Osmundsen et al.	2021	Journal of Information Technology	Journal							x	x	x							x				x		
P1	DevOps Adoption: Challenges & Barriers	Krey et al.	2022	HICSS	Proceeding				x	x		x	x	x				x			x		x		x	x	
P2	A Survey of DevOps in the South African Software Context Morgan	Rowse et al.	2021	HICSS	Proceeding				x	x			x	x	x					x	x		x		x		
P3	SKI: A New Agile Framework that supports DevOps, Continuous Delivery, and Lean Hypothesis Testing	Saltz et al.	2020	HICSS	Proceeding					x			x	x						x	x				x		
P4	Implementing the Planning Process within DevOps Teams to Achieve Continuous Innovation	Wiedemann et al.	2019	HICSS	Proceeding				x	x		x				x	x										
P5	Dependency Management in Large-Scale Agile: A Case Study of DevOps Teams	Stray et al.	2019	HICSS	Proceeding				x	x	x						x		x		x						
P6	DevOps Continuous Integration: Moving Germany's Federal Employment Agency Test System into Embedded In-Memory Technology	Sultanow et al.	2019	HICSS	Proceeding								x	x	x										x	x	
P7	How Enterprises Adopt Structures: A Multiple-Case Study	Gerster et al.	2019	HICSS	Proceeding							x	x	x			x		x	x	x						
P8	Process Efficiency - Adapating Flow to the Agile Improvement Effort	Verbruggen et al.	2019	HICSS	Proceeding				x			x	x	x													
P9	IT Governance Mechanisms for DevOps Teams - How Incumbent Companies Achieve Competitive Advantages	Wiedemann et al.	2018	HICSS	Proceeding				x			x				x				x							
P10	Increasing the Agility of IT Delivery: Five Types of Bimodal IT Organization	Horlach et al.	2017	HICSS	Proceeding							x	x	x			x			x	x			x			
P11	Towards DevOps in the Embedded Systems Domain: Why is It So Hard?	Lwakatare et al.	2016	HICSS	Proceeding				x				x	x							x			x	x		
P12	DevOpsBL: DevOps-based Learning on Information Systems Higher Education	Grotta et al.	2021	AMCIS	Proceeding																x	x					
P13	How to Implement Clan Control in DevOps Teams	Wiedemann et al.	2018	AMCIS	Proceeding								x	x	x	x	x	x	x	x	x						
P14	Is Information Systems Late to the Party? The Current State of DevOps Research in the Association for Information Systems Systems eLibrary	Sharp et al.	2018	AMCIS	Proceeding				x			x									x						
P15	Uncertainty, Personality, and Attitudes toward DevOps	Shropshire et al.	2018	AMCIS	Proceeding		x					x	x	x					x	x	x		x				
P16	Key Capabilities of DevOps Teams and their Influence on Software Process Innovation: A Resouce-Based View	Wiedemann et al.	2017	AMCIS	Proceeding							x	x	x	x	x	x	x	x		x						
P17	IS Career Anchors, Professional Growth and Mobility Intentions: A DevOps Job Learning Effect?	Hemon et al.	2019	ICIS	Proceeding								x	x	x	x	x	x	x			x					
P18	A Control-Alignment Model for Product Orientation in DevOps Teams-A Multinational Case Study	Wiedemann et al.	2019	ICIS	Proceeding								x	x	x	x	x	x	x		x		x				
P19	Coexisting Plan-driven and Agile Methods: How Tensions Emerge and Are Resolved	Laux et al.	2019	ICIS	Proceeding														x	x	x						
P20	Job Satisfaction Factors and Risks Perception: An embedded case study of DevOps and Agile teams	Hemon et al.	2018	ICIS	Proceeding	x	x						x	x													
P21	Designing and Executing Digital Strategies	Ross et al.	2016	ICIS	Proceeding				x			x	x	x							x				x		
P22	DevOps: Concepts, Practices, Tools, Benefits and Challenges	Ghantous et al.	2017	PACIS	Proceeding					x											x				x		
P23	A New Form of Collaboration in IT Teams - Exploring the DevOps Phenomenon	Wiedemann et al.	2017	PACIS	Proceeding								x	x						x	x		x		x	x	
P24	The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations	Hüttermann et al.	2021	ECIS	Proceeding				x			x	x	x						x	x		x		x	x	
P25	Reconceptualising Business-IT Alignment For Enabling Organisational Agility	Horlach et al.	2020	ECIS	Proceeding				x		x	x	x	x			x			x	x				x		
P26	Are You Ready for DevOps? Required Skill Set for DevOps Teams	Wiedemann et al.	2018	ECIS	Proceeding								x	x							x						

Figure 5-3. Concept matrix.

Online Appendix C

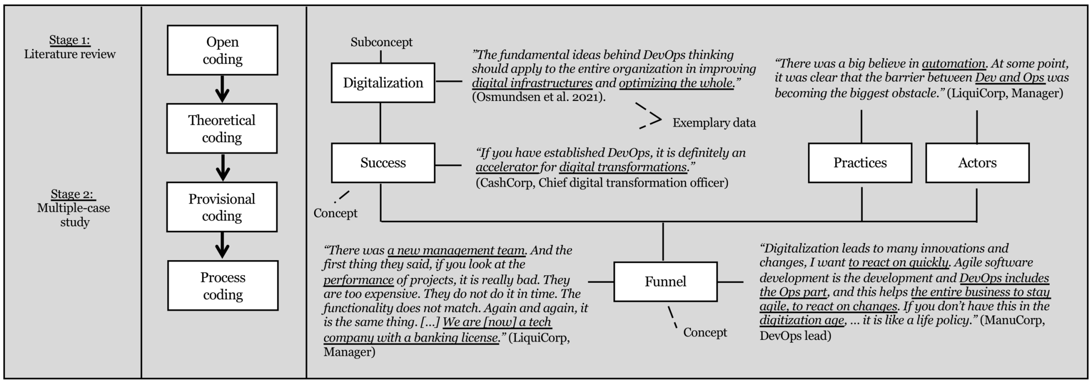

Figure 5-4. Coding process with illustrations (excerpt).

Online Appendix D

Case / Concept	Actors	Applied Practices	Success
SoftwareCorp	20.000+ software engineers, with 1.000+ products (external, and internal, e.g. platform), team size: 10-30 persons	Both, continuous delivery and process rich, depending on the team-in-focus: • Continuous delivery: some teams have highly automated tasks, and deliver releases very quickly. For example, based on an "Inner Source" approach, a shared Jenkins code library is continuously updated and can always be used in its most recent versions. Product owner for CI/CD tools exist. Compliance is automated: *"We must confirm to over 200 formal criteria (defined by company, customer, country regulations) to bring a change to production. […] What has changed is the perception, that those things can be automated"*. (DevOps engineer) • Process rich: other teams have rich processes. For example, for a product, there are defined maintenance windows: *"That is determined by contract. Tuesday and Thursday, early in the morning, maintenance operations on production can be done."* (Product owner)	Both, humanistic and instrumental goals: Blend of both customer satisfaction and delivery performance. Delivery performance is the main goal to positively affect customer satisfaction. However, it may also negatively affect customer satisfaction if product demand is not aligned with product delivery: *"We aim to deliver every day. But the customer is not happy to detect new features every day, […] we are now delivering weekly, and it turns out, that this is an acceptable frequency also for the customer."* (Product owner) Top management programs: *"We have Top-X programs, for example for customer success, productivity, and cultural change transformation."* (DevOps engineer)
EntertainCorp	Platform teams >200 software engineers, Product teams >2000 software engineers, engineers per team: ~10	Continuous delivery: Product teams can use platform and central libraries to keep up the flow. They deploy multiple times a day to production. As part of continuous delivery, modern approaches are used: *"We have a lot of resilience tooling that can be used during canary processes."* (Platform engineer)	Primarily humanistic goals: Customer satisfaction has top priority (e.g., usage of "login" and "play" functionality, and hiding of technical issues). Instrumental goals contribute to humanistic goals. Digitalization inherent due to sector the organization is active in. *"This is so much in our DNA. It is not a topic."* (Platform engineer)
LiquiCorp	Tribe has 2+ Chapters, and 2+ Squads, each about 5-10 dev and ops engineers	Continuous delivery: It is considered to be important to constantly deliver business value, as a product owner states *"Short cycles, small portions, but adding business value every time,*	Both, humanistic and instrumental goals: blend of both, customer satisfaction and delivery performance. *"There is a DevOps pipeline and we build it for all engineers so that*

Case / Concept	Actors	Applied Practices	Success
	plus product manager, constantly changing; 1000+ Squads in total; Tribes for platform	*this really helps in customer satisfaction*" (Product owner)	*they can deliver software, that's the main goal for our end customer".* (Chapter lead) Digitalization is an organizational goal. *"Our former CIO said that we are indeed not a bank with an IT department but it's the other way around."* (Product owner)
StampCorp	Platform team, plus 5 product teams, 5-10 employees each	Continuous delivery: Codified and automated delivery steps. *"I was part of a spearhead in front of a lot of awesome, amazing people to help move that into continuous delivery. And now we release that piece of software five times a day".* (Platform engineer)	Humanistic goals: Customer satisfaction has top priority, and instrumental goals contribute. *"Everything we do, we have to look at the customer. And whether that's our employees considering our external customers and how we want to make them successful. Or whether it's an internal team looking at how we make other teams successful."* (Engineering director) Top initiatives implemented by all teams. *"We have an initiative called 'faster, safer', which is around technology moving faster and safer."* (Platform engineer). *"A lot comes down to economics and going what is the next most valuable thing we get the team to work on."* (Engineering director)
ManuCorp	Corporate function of DevOps research group; 30 persons BUs, with product teams, e.g. medical tech, with 20 dev teams and two ops teams, up to 10 persons	Continuous delivery: Lot of automation across the entire development and delivery in order to enable continuous delivery. Continuously gather user feedback to improve the products. *"We have a closed feedback loop. That is actually the clou. [...] not only the path to the customer, but also the way back, then I can really learn, and the gathered data, [...], I can learn from them, and again stream in improvements at the start, in order to continuously improve".* (DevOps lead architect)	Both, humanistic and instrumental goals, depending on team-in-focus. *"We use software delivery performance metrics, such as lead time, cycle time and failure recovery. [...] And we use classic metrics, e.g. ISO-25010, because availability is important for us. Not 99.99, that's simple. Instead we often have five nines or six nines." (*DevOps lead architect) *"In the plannings, we aim to focus on the (valuable) outcome, instead of just producing features, and to produce those features as output".* (Dev manager) *"Product demand is equal to product delivery. This is the continuous delivery effect."* (Dev manager) Digitalization is a top initiative. *"We work on 'master digital transformation at scale'. Through DevOps, we get the speed we need for that."* DevOps lead architect
CashCorp	Group IT and business line; transformation team under CEO; Squads with 10	Continuous delivery: Frequent delivery of small, decoupled IS products, relying on APIs, to test business models. *"We try to test new business models, by bringing small*	Both, humanistic and instrumental goals, depending on team-in-focus. The top management defines goals to bring out products quickly. *"Our COO wants that".* (Chief digital transformation officer)

Case / Concept	Actors	Applied Practices	Success
	employees each	*products to the market quickly."* (Chief digital transformation officer)	
InsureCorp	Global IT (e.g. Cloud, ERP) and Regional IT, Business line; Product teams with employees of Global IT, Regional IT, Business; Guilds e.g. for transformation and security	Both, continuous delivery and process rich, depending on the team-in-focus: • Continuous delivery: Some teams have highly automated tasks, and deliver new versions of IS products very quickly • Process rich: Tend to organize and schedule up front and all steps in advance. *"It happens that it is preferred to do planning weeks, months more, just to avoid to make wrong decisions".* (Coach)	Both, humanistic and instrumental goals, depending on team-in-focus. Goals are distributed to teams: *"We work on distribute metrics to teams, that teams can define their own metrics, their own goals. […] We don't predefine goals, because teams are too different."* (IT manager) Digitalization is an organizational goal. *"We are on a big transformation journey. […] This is digital transformation. In a through years, we will have a complete different organizational structure. I don't think we will still have this distinction between business and IT."* (Digital transformation manager)

Table 5-3. The funnel and its concepts per case, with examples.

Online Appendix E

Case	Digital Technology
SoftwareCorp	Both, old and modern, depending on the team-in-focus: • Old: classic, settled technology such as ERP. *"It is a big challenge to run an ERP in the cloud, and that it works correctly".* (Site reliability engineer) • Modern: Cloud-based, with an own cloud system (SoftwareCorp provides a cloud solution) and usage of other cloud providers. *"Our own cloud product is not the leading one any more".* (Product owner) Cloud technology is associated with faster delivery. *"And then we said, we want to more engage with Cloud. […] we want to develop and deliver faster."* (Product owner)
EntertainCorp	Modern: Cloud, distributed. *"That is a global distributed Content Delivery Network. […] Customer traffic doesn't even have to broach their ISP's border to come get their video bits. [..] On the backend side, we operate primarily out of three AWS global regions".* (Platform engineer)
LiquiCorp	Modern: Cloud as a strategic platform. *"We decided to go to Azure DevOps because they have data centers all over the world. And also Microsoft, because we are a bank."* (Chapter lead) Entire firm is considered to be a tech company with banking license. *"Our chairman has also said that we are a tech company with a banking license, then the most important enabler for the banking business is tech."* (Tribe lead)
StampCorp	Diverse modern technologies are used: *"We have some stuff on virtual machines in the cloud. We have some stuff on an old Kubernetes cluster and we have some stuff on a new Kubernetes cluster".* (Platform engineer) Technology is also distributed: *"Because it is a physical manufacturing company, […] we also have physical points of presence. We have the warehouses where the products get manufactured. And so we have infrastructure running in those locations as well".* (Engineering director)
ManuCorp	Both, old and modern, depending on the team-in-focus: • Old: classic, settled technology, existing facilities, "brownfield". • Modern and distributed, heterogenous: *"We have everything from cloud, to on-prem, to edge, then different execution platforms, and of course: how do I come (with my software) into the hospital or into the nuclear power plant or into the train."* (DevOps architect) *"In our case, we take data from the hospital, and send them into the cloud".* (Dev manager)
CashCorp	Both, old and modern, depending on the team-in-focus: • Old: classic, settled technology. Some teams have highly automated tasks, and deliver new versions of IS products very quickly • Modern: Cloud. *"Everybody has to develop in an AWS-enabled way. It must run in the cloud, that is the common denominator".* (Application domain officer)
InsureCorp	Both, vintage and modern, depending on the team-in-focus:

	• Old: classic, settled technology. *"Central operations is done for our mainframe platforms. […] There, our old monoliths are running." (*Digital transformation manager) • Modern: Cloud: *"They can order everything by one-click and then use it themselves, either on AWS, or Microsoft Azure, or our own Open-PaaS environment"* (Digital transformation expert). Cloud is strategically important to delivery efficiently, although it is not considered to be unique selling point for itself: *"Typically, an insurance company does not have any innovation pressure. […] You normally don't gain any competitive advantage through technology."* (Coach)

Table 5-4. Digital technology per case, with examples.

6 Paper IV. "From Loose Coupling to Fusion: How Does DevOps Operationalize Digital Business Strategies and Contribute to Digital Innovation?"

Michael Hüttermann
Prof. Dr. Christoph Rosenkranz

Under review, at Information Systems Research

FROM LOOSE COUPLING TO FUSION: HOW DOES DEVOPS OPERATIONALIZE DIGITAL BUSINESS STRATEGIES AND CONTRIBUTE TO DIGITAL INNOVATION?

Organizations operationalize digital business strategies to entrepreneurially explore opportunities on competitive markets and to better react on environmental changes. To support this effort, they increasingly employ a holistic approach to information systems development and a close integration of information systems operations to align organizational functions. This leads to the emerging DevOps phenomenon. While studies have delivered first insights about DevOps, a theoretical grounding of DevOps implementations including how DevOps relates to digital business strategy and digital innovation is still missing. We conducted a multiple-case study to explore the different characteristics of DevOps and how these influence its implementation in different contexts. We find that DevOps implementations are gradual transitions on a DevOps continuum, with two mechanisms, effectiveness adaptation and efficiency adaptation, leading to a new or the change of an existing implementation. Furthermore, we find that, by building on fused teams of people from business, information systems development, and information systems operations, DevOps operationalizes digital business strategies and contributes to digital innovation. These insights foster our understanding of the DevOps phenomenon, relate DevOps to digital business strategy and digital innovation, and identify new paths for future research.

6.1 Introduction

To effectively bring forth business and value creation innovations (Van de Ven and Poole, 1990), to build them on information technology (IT) and to maximize information systems (IS) success (DeLone and McLean, 1992), to develop corresponding IS solutions efficiently, and to subsequently realign their own company to remain competitive, supposedly demands a repositioning of the organizational functions that can meet the companies' requirements for capabilities in innovation, design, and transformation (Henry Lucas *et al.*, 2013; Urbach and Ahlemann, 2019). This has led to the call to *fuse* business and IT (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013) – fusion meaning that IT is fused with the business environment in such a way that business and IT are indistinguishable (El Sawy, 2003). This is argued to be needed for successful *digital business strategies*, that is, organizational strategies that see IT not as a commodity, but as an integral part of business models and processes to create differential value for customers (Bharadwaj *et al.*, 2013; Urbach and Ahlemann, 2019). Organizations must ask themselves what a collaboration between business and IT departments should look like in their respective context to operationalize a digital business strategy and to foster digital innovation and differential value creation (Urbach and Ahlemann, 2019). However, no systematic knowledge exists on how this sought-for fusion between business and IT can be achieved to pursue a digital business strategy (Bharadwaj *et al.*, 2013) so that the company can successfully respond to new threats and opportunities (Sia *et al.*, 2016).

Stemming from agile IS development (AISD) (Hemon-Hildgen, Rowe, et al., 2020), the emerging phenomenon of *DevOps* has been suggested as a potential way to achieve this integration across organizational functions in both literatures on systems development (Greenwood *et al.*, 2020; Hemon-Hildgen, Rowe, *et al.*, 2020; Hüttermann, 2012; Wiedemann *et al.*, 2020) and on digital business strategy and digital innovation (Bosch and Olsson, 2021; Olsson and Bosch, 2020; Sia *et al.*, 2016, 2021). Following the concept's origin in the software industry, DevOps is a portmanteau word of "development" and "operations". Proponents of DevOps argue that the two major IT-related enterprise functions for "IS development" (those that build IS) and "IS operations" (those that run and maintain IS) increasingly apply shared goals and use shared practices across both functions, bringing together team members from both development and operations, in order to implement and run IS in a comprehensive way (Hemon-Hildgen, Rowe, *et al.*, 2020; Qumer Gill *et al.*, 2018). DevOps aligns the two IT functions for development and for operations more closely (Wiedemann *et al.*, 2020). Moreover, with business roles such as product owners (Hemon-Hildgen, Rowe, *et al.*, 2020; Wiedemann *et al.*, 2020) being an explicit part of an autonomous, cross-functional DevOps team, DevOps even goes beyond aligning the two IT functions by clearly involving the business side in forming a product team. As part of the root metaphor for strategy in digital innovation (Berente, 2020), DevOps thus appears to reconcile alignment models (Tallon and Pinsonneault, 2011) and to support the sought-for fusion of business and IT departments (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013; El Sawy, 2003). This bridging of business and development to its resulting IS product and its systems operations (Greenwood *et al.*, 2020) supposedly empowers many of the elements of digital value creation for the overall organization (Alt *et al.*, 2020).

Up until now, however, few studies have investigated DevOps or its effects (Maruping and Matook, 2020). For example, existing research has proposed a tripartite model of intra-IT alignment to shed light on how DevOps aligns development with operations (Wiedemann et al., 2020). However, this focuses on intra-IT collaboration and disregards the connection to business. Others have examined DevOps on an individual level, for instance, its relationship to work conditions and job satisfaction (Hemon-Hildgen, Rowe, et al., 2020). While these studies offer valuable preliminary insights about DevOps implementations in specific settings, they focus on particular aspects of its working such as how intra-alignment of IT functions in DevOps teams can be achieved (Wiedemann et al., 2020), or on particular effects such as how DevOps affects job satisfaction (Hemon-Hildgen, Rowe, et al., 2020). Moreover and similar to AISD (Conboy, 2009), the development and discussion of DevOps is primarily driven by industry.

Similarly, studies on digital business strategies and digital innovation at least partially appear to consider the fusion of business and IT departments for granted. For example, existing research explains how the effects of fusion can be leveraged for implementing digital business strategies and to digitally innovate (Bosch and Olsson, 2021; Olsson and Bosch, 2020; Sia *et al.*, 2016, 2021), conceptualizes the logic of digital business strategy in terms of design capital and design moves (Woodard *et al.*, 2013), or suggests the importance of designing IS products and the required delivery capability to incubate and accelerate emerging digital innovation (Bharadwaj *et al.*, 2013; Sia *et al.*, 2016). However, these studies do neither consider how the needed fusion is achieved in the first place, nor what the contextual dynamics are behind adaption of the integration of organizational functions, or how a design of IS products and the corresponding delivery capability are achieved to enable digital innovation through digital business strategy (Bharadwaj *et al.*, 2013; Sia *et al.*, 2016).

As a result, the literature in both streams creates an unrealistic expectation that a company's DevOps teams can – and should – only need to reach a specific skill set (Hemon-Hildgen, Lyonnet, *et al.*, 2020), a high maturity level (Gupta *et al.*, 2017; Rafi *et al.*, 2021) or a fusion of business and IT (Sia *et al.*, 2016, 2021) for it to work successfully. This lack of understanding, conceptualization, and theorizing of DevOps, how DevOps is implemented, and how DevOps teams work is problematic for academic studies and for companies looking for guidance on how to achieve IT-business fusion, digital innovation, and differential customer value with the help of DevOps. Consequently, recent calls for research (Maruping and Matook, 2020) ask for studies that engage theoretically grounded with the DevOps phenomenon to reach deeper levels of understanding, and for empirical studies that explore the drivers and effects of DevOps (Gall and Pigni, 2021). This includes questions such as how actors in DevOps collaborate, why unique forms of DevOps emerge, and how they change over time (Maruping and Matook, 2020). In sum, we are missing a theoretical grounding of DevOps, we are lacking an understanding of what leads to specific DevOps implementations, we do not know the effects of different DevOps implementations, and we do not know if and how specifically DevOps relates to the propagated fusion of business and IT departments, digital business strategy, and digital innovation. Consequently, we ask the following research questions: *RQ1: "What characteristics influence how DevOps is implemented in organizations?", RQ2: "How and why do teams adapt their DevOps implementations?", and RQ3: "How does DevOps relate to the fusion of business and IT?".*

To answer these questions, we conducted an exploratory, multiple-case study to derive a model that explains the underlying rationale for and implementation form of DevOps. Based on our analysis, we develop new theory on what characterizes DevOps, how and why organizations implement and adapt their DevOps implementations, and how DevOps relates to business-IT fusion, digital business strategy, and digital innovation.

Our study makes several contributions. First, we offer a theoretical model describing and explaining contextual characteristics of DevOps implementations, which we call the 'DevOps continuum'. It describes the characteristics that influence how teams implement DevOps. Second, we propose two causal mechanisms – effectiveness adaptation and efficiency adaptation – through which teams adapt their DevOps implementations, accompanied by changes of the instantiations of characteristics, which we call 'DevOps flavors'. Through these mechanisms, teams manage to optimize IS success in terms of creating differential value for customers by balancing effectiveness and efficiency, contingent on their specific situation and success definition. Third, we suggest that the actualized mechanisms lead teams to find their "best fit" of IT-business collaboration (ranging from loose coupling to fusion) with regards to their specific context and success definition. This also entails that IT-business fusion is not the penultimate goal for all situations. Contrasting the popular narrative that a DevOps team only needs to reach a specific skill set or a high maturity level to solve friction points between IT functions, we find that teams find the best spot for them on the DevOps continuum, or the specific level of coupling of functions needed, in their specific situation and context. Thus, we also add to research on digital business strategy and digital innovation by showing that fusion is not necessarily a binary 'yes/no' decision, and that beneficial loose as well as tight couplings between functions exist. Together with leveraging digital resources, we suggest that this operationalizes digital business strategy and leads to digital innovation. We thereby contribute to a more foundational view of DevOps, a more nuanced view of business-IT fusion, and DevOps' role for digital business strategy and value creation, which is grounded in empirical data.

6.2 Related Work and Theoretical Background

6.2.1 Digital Innovation and Digital Business Strategy

The last two decades have seen the rise of technology-driven organizations or digital ventures such as Amazon, Apple, or Google leveraging digital technology to gain competitive advantage (Huang *et al.*, 2017). As IT has become pervasive and ubiquitous, the possibilities of digital evolution have been perceived as being able to exceed previous waves of innovation for all industries (Yoo *et al.*, 2010). *Digital innovation* can be understood as the creation of and change in market offerings, business processes, or models that result from the use of digital technology (Nambisan *et al.*, 2020) to achieve novel value creation and differential value for customers (Nambisan *et al.*, 2017) to maximize IS success (DeLone and McLean, 1992). Thus, innovation can be seen as a self-reinforced mechanism that, based on a space of possibilities and assembly of components, may lead to new products in IS infrastructure, the installed base of organizations, systems, and users (Bygstad *et al.*, 2016). Successful and innovative companies thus have recognized the importance of digital technology for competitive advantages and to create differential value for customers (Bharadwaj *et al.*, 2013). This means going beyond the traditional view of IT, thinking of IT as an industrial fabric function within firms, instead recognizing the pervasiveness of digital resources in other functional areas such as business and operations (Berente, 2020). This emphasizes elevating the performance implications of IT strategy beyond efficiency and reliability metrics to user-centric factors that drive competitive advantage and strategic differentiation (Bharadwaj *et al.*, 2013; DeLone and McLean, 1992).

We follow Bharadwaj et al. (2013) and define *digital business strategy* as an organizational strategy (across functions) formulated and executed by leveraging digital resources to create differential value (Bharadwaj *et al.*, 2013, p. 472). Existing studies have shown that companies pursuing a digital business strategy are able to respond to digital threats and opportunities, for example, by digitally innovating (Bosch and Olsson, 2021; Olsson and Bosch, 2020; Sia *et al.*, 2016, 2021) or by engaging into specific design moves leveraging its digital resources under certain constraints (Woodard *et al.*, 2013). According to this emerging "Innovation-Design-Transform" paradigm (Urbach and Ahlemann, 2019), IT is becoming the central driver of value creation, which means needing an IT organization with the capabilities to innovate (respond to market events and opportunities), to design (an effective IS product), and to transform (by changing structures and processes accordingly) (Urbach and Ahlemann, 2019). In sum, digital business strategies lead companies to see IT not as a commodity but as an integral part of business models and processes, to integrate organizational functions (Bharadwaj *et al.*, 2013), to leverage digital resources and to incubate and accelerate emerging digital innovation (Sia *et al.*, 2016), and to design IS products and the required delivery capability to create and capture business value (Bharadwaj *et al.*, 2013; Woodard *et al.*, 2013). In industry as well as in first research studies (Bosch and Olsson, 2021; Olsson and Bosch, 2020; Sia *et al.*, 2021), this is often linked to the emerging DevOps phenomenon.

6.2.2 DevOps

Organizations are increasingly using cross-functional team-based structures to innovate (Larson *et al.*, 2022). In today's industrial practice, applying AISD methods is a dominant paradigm and way for enterprises to attempt maximizing the probability of *IS development success* (Dybå and Dingsøyr, 2009). Success from the developmental side (Siau et al., 2010) can be operationalized, for example, by staying within scope, time, and cost requirements (Chow and Cao, 2008), functionality (Lee and Xia, 2010), rapid change (Lee and Xia, 2005), or delivering high quality (Siau et al., 2010). However, focusing solely on the development factors does by-and-large neglect the importance of operational factors, the resulting IS product, and its long-term operation and use (DeLone and McLean, 1992; Fitzgerald and Stol, 2017). This success from the operational side can be operationalized, for example, by using the industry standard "mean time to restore" (Dekleva, 1992) as the time required to restore the IS product after an incident occurs that makes the system unavailable. This suggests that overall *IS success* (DeLone and McLean, 1992) and thus a differential value for customers can only be optimized if development and operations are integrated or aligned with each other (Wiedemann et al., 2020).

Spanning development and operational factors, *effectiveness* ("delivering the right product") and *efficiency* ("delivering the product right") both contribute to IS success (Chandler Jr., 2018). Effectiveness focuses on the quality of and amount of customer value created by the shipped functionality, whereas efficiency relates to how much IS functionality is shipped in a given time (Bosch, 2019). Building on this understanding, industry often sees DevOps as an extension of AISD (Wiedemann, Forsgren, et al., 2019), which aims to optimize overall IS success and achieve better results for both effectiveness and efficiency (Hüttermann, 2012), supposedly by providing ambidexterity on the team level (Birkinshaw and Gibson, 2004).

In parallel to its rising application in industry, DevOps has gained increasing attention from empirical software engineering and IS researchers (Sharp and Babb, 2018). Due to the lack of specific research, models, and definitions, DevOps can describe different things, including team structure, success criteria (i.e., goals), concepts, or tooling (Qumer Gill et al., 2018). As a first step toward a unified conceptualization of DevOps, researchers have suggested dimensions such as culture, automation, lean, monitoring and measurement, and sharing as constitutional for DevOps (Fitzgerald and Stol, 2017; Gall and Pigni, 2021; Hemon-Hildgen, Rowe, *et al.*, 2020; Wiedemann, Forsgren, *et al.*, 2019). In industrial practice, automation and the related concept of *continuous delivery*, the production of software in short temporal cycles resulting in building, testing, and releasing software with greater speed and frequency, often have the highest priority in order to satisfy customers through continuously providing valuable software (Fowler and Highsmith, 2001).

Based on these views in research and industry, we adopt a previous definition of DevOps (Hemon-Hildgen, Rowe, et al., 2020) and combine it with AISD's "ability to adapt to change" (Conboy, 2009) and to continuously deliver valuable outcome to the customer (Fitzgerald and Stol, 2017; Fowler and Highsmith, 2001). Our resulting conceptualization defines DevOps as "*a set of continuously improved practices for collaborative work implemented between organizational functions to develop and deliver an IS product toward continuous delivery of valuable outcome for customers*." The latter aspect – continuously deliver an outcome of *value* to the customer – provides the fundamental reason why DevOps in industry often is presumed to bridge effectiveness (e.g., to deliver valuable outcome to the customer) and efficiency (e.g., through management and improvement of cycle time, Wetherbe and Frolick 2000). This definition emphasizes that DevOps is not only something situated in IT departments, but is seen as a way of collaborative work between IT functions *and* business to continuously deliver valuable outcome (Matook *et al.*, 2021).

Existing research on DevOps has investigated, for example, the adoption challenges for DevOps (Lwakatare, Karvonen, *et al.*, 2016), the continuous integration of software development and the operation of the resulting IS product (Fitzgerald and Stol, 2017), the attributes characterizing DevOps (Gupta *et al.*, 2017), the necessary skill sets of team members for DevOps (Hemon-Hildgen, Lyonnet, *et al.*, 2020), the control-alignment view for product orientation in DevOps teams (Wiedemann et al., 2019), or projectable maturity of DevOps implementations (Alt *et al.*, 2021; Mishra and Otaiwi, 2020; Rafi *et al.*, 2021). While the AISD research stream already gained an understanding about the importance of integration of functions of business and IS development (Conboy, 2009; Fitzgerald *et al.*, 2006; Hemon-Hildgen, Rowe, *et al.*, 2020), no studies exist that shed light on DevOps, how and why it aims to include the two functions of IS development and IS operations, or its suggested relation to digital business strategies and digital innovation.

Although DevOps is seen as part of the root metaphor for strategy in digital innovation (Berente, 2020), the relation is not yet sufficiently addressed in the allied fields and largely disjoint streams of literature of digitalization (including digital business strategy and digital innovation) and systems development. We argue that gaining an understanding of what contextual characteristics influence DevOps implementations and what mechanisms are at work may help to considerably unpuzzle the concept of DevOps and its effects, specifically for the sought-for fusion of business and IT against the backdrop of digital business strategy and digital innovation. Considering the lack of research about DevOps in general, the absence of conceptualization and theorizing, and the missing understanding which contextual characteristics exist that influence a specific DevOps implementation (Maruping and Matook, 2020), we examine how and why organizations continually adapt their DevOps implementations, and which mode of collaboration between functions of business and IT they choose, to efficiently leverage digital resources to achieve differential value for customers.

6.3 Research Method

6.3.1 Design Overview

We used an exploratory, inductive case study design (Eisenhardt, 1989). First, we identified appropriate case site candidates that had self-reported to have implemented DevOps. Previous empirical studies on DevOps (e.g., Hemon-Hildgen, Rowe, et al., 2020; Wiedemann et al., 2020) also influenced the selection because we wanted to include similar but also contrasting case types to understand which characteristics led to these specific DevOps implementations (Eisenhardt, 1989; Yin, 2018). Sampling criteria focused on revelatory cases (Dubé and Paré, 2003) in two groups of cases. The first group (group A) included case sites that are successful in their domain, understanding IT as a core asset and an inherent driver for overall company success. These companies do follow a holistic approach to ISD because business is presumed to be driven by IT as a core asset for maximizing business value (Gall and Pigni, 2021; Maruping and Matook, 2020). Case sites of the second group (group B) are comparable with members of the first group except that they run a more conservative, classic approach to IT. We expected them to apply DevOps less and only moderately, thus providing a contrast. We thoroughly examined pre-selected case site candidates that matched the sampling criteria and agreed to participate, and ran pilot interviews with 12 different firms over a period of ten months (Dubé and Paré, 2003). The 12 firms were either part of our professional network or were actively contacted for the research purpose at leading industry conferences. We started with one company from this set for each group, and iteratively added case sites to each case group to assert and extend emergent theory (Eisenhardt, 1989). Since we gained meaningful insights with sufficient theoretical saturation (i.e., minimal incremental learning combined with minimal incremental improvement to the theory) after investigating eight cases in total, we decided to end our data collection. We did not highly prioritize the industries or segments the firms are active in. However, we carefully paid attention to diverse sites across segments, leading to firms being in different businesses (i.e., online streaming, enterprise software, manufacturing, insurance, and financial industry). Table 6-1 gives an overview of our case sites.

As part of this step, we also determined the unit of analysis. Since we were interested in how and why specific organizations implement DevOps, at the beginning of our study, the unit of observation was the respective company. During our data analysis, we recognized that multiple different DevOps implementations may be in place inside one company, thus the unit of analysis transitioned to teams.

Group A	**Group B**			
FinePrint	**ManuFact**	**AutoBank**	**CashFlow**	**InsureMe**
Manufacturing	Manufacturing	Banking	Insurance	Insurance
UK	Germany	France	France	France
<1.000	> 400.000	<10.000	> 5.000	> 150.000
Medium complex consumer products	Large, complex industrial products	Medium to large business products	Medium to large business products	Medium to large business products
Platform team, plus 5 product teams, 5-10 employees each	Corporate function of DevOps research group: 30 persons; BUs, with product teams, e.g. medical tech, with 20 dev teams and two ops teams, up to 10 persons each	Regional subsidiary: IT function: 50 software engineers, including Ops: 15 Dev: 25 Support: 5	Group IT and business line; transformation team under CEO; Squads with 10 employees each	Global IT (e.g. Cloud, ERP) and Regional IT, Business line; Product teams with employees of Global IT, Regional IT, Business; Guilds e.g. for transformation and security

** Names anonymized and number of employees in ranges for confidentiality reasons*

Case Site* / Attributes	SoftwareDev	MovieStream	TransAct
Industry segment	Software	Streaming	Bank
Headquarters	Germany	USA	Netherlands
Number of employees	> 100.000	<10.000	> 50.000
Product offering	Medium to large business products	Large customer entertainment product	Medium to large business products
Team topology	20.000+ software engineers, with 1.000+ products (external, and interal, e.g. platform), team size: 10-30 persons	Platform teams >200 software engineers, Product teams >2000 software engineers, and engineers per team: ~10	Tribe has 2+ Chapters, and 2+ Squads, each about 5-10 dev and ops engineers plus product manager, constantly changing; 1000+ Squads in total ; Tribes for platform

Table 6-1. Cases Overview.

6.3.2 Data Collection

We collected data between October 2019 and April 2021. The primary mode of data collection was interviews. Complementary data sources included project documentations, public tech blogs and presentations, and, in parts[10], observation as a participant-as-observer (Myers, 2009; Yin, 2018). Appendix A (Table A1; Table 6-4) summarizes our data sources, including details on the interviews.

In total, we conducted 30 semi-structured, open-ended interviews with highly knowledgeable informants from the case organizations, spanning different business, developmental, and operational roles. Based on initial entry contacts, influenced by Graebner and Eisenhardt (2004), we used snowball sampling to identify further informants (Eisenhardt, 1989). Informed by the recommendations of Eisenhardt (1989), we reached closure and stopped adding informants and cases when the incremental learning was minimal and we observed similarities in previous descriptions of the phenomenon. We triangulate views within and across cases, and extend and assert emerged theory (Eisenhardt, 1989). We used an interview protocol for primary data collection (see Appendix A2). The interviews allowed

[10] Participant observation was stopped in April 2020 due to the outbreak of the COVID-19 pandemic.

for probing and emphasized context and narratives (Brinkmann and Kvale, 2014). Depending on background and preferences of the informants, interviews were either conducted in English or German. Questions from the interview guideline and transcribed answers of interviewees were translated by the authors.

We triangulated data and addressed potential biases in several ways. First, one member of the research team is also an independent freelance consultant on DevOps. While this raises a potential bias, Cassell and Symon (2012) argue that some degree of researcher bias is not only inevitable to the study, rather it is beneficial since such studies cannot be carried out in a social vacuum and the interviewer needs to be knowledgeable about the interview topic (Brinkmann and Kvale, 2014; Yin, 2018). Second, tactics to minimize participant bias included taking frequent breaks to continue theory building, and to continuously reflect and analyze collected data. Third, we emphasized using multiple data sources and writing case writeups (Eisenhardt, 1989; Yin, 2018). In addition, we gave anonymity and transparency to our informants, for example, by providing thorough information about the study.

6.3.3 Data Analysis

Our data analysis and theorizing are split into three parts. To answer the first research question ("Which characteristics influence how DevOps is implemented in organizations?") and aligned with the exploratory, inductive nature of our research, we developed concepts and relationships in sequential within-case analyses, before looking for similar concepts and relationships across multiple cases. To analyze the data, we followed established guidelines for qualitative, inductive research (Charmaz, 2006; Eisenhardt, 1989; Eisenhardt and Graebner, 2007; Miles *et al.*, 2020). Our analysis was influenced by Grounded Theory Method (Strauss and Corbin, 2015). Online Appendix B provides further details on the analysis and gives examples for the coding process.

To answer the second research question ("How do teams adopt their DevOps implementations?"), our analysis applied critical realist approaches (Mingers *et al.*, 2013; Pawson and Tilley, 1997; Sayer, 2000), and was influenced by the methodological principles for conducting critical realist case study research suggested by Wynn and Williams (2012). Due to our generative role in the analytical process (Weick, 1989), we did not use their suggested principles as a strict template, but rather as a guidance to align our qualitative analysis with theory development (Grodal *et al.*, 2021; Sarker *et al.*, 2013). Online Appendix C provides further details on the analysis and gives examples for the coding process.

To answer our third research question ("How does DevOps relate to the fusion of business and IT?"), influenced by Danermark *et al.* (2019), we distinguish between an empirical and an theoretical elaboration. In our theoretical analysis (see 6.5.3 "DevOps in Relation to Fusion of Business and IT"), we suggest a generalization based on another, more abstract level of theorizing (Berlin, 1950).

6.4 Findings

6.4.1 The DevOps Continuum

Overall, our study reveals that the individual team aligns its DevOps initiative with characteristics of its *inner environment* and its *outer environment* (see Table 6-2). The inner environment expresses characteristics having its root in the team or inherited from the organization it is embedded in. Although, from a team perspective, all other organizational entities in the same organization could be considered to be part of the outer environment, here, with outer environment, we express the context of the team to the field, its users, and its market.

Characteristics		**Short Description**
Inner env-ironment	Organizational standards	Comprises rules and norms the team must align with (e.g., corporate, normative rules the team must confirm to as well as the general team setup and the responsibilities of the team related to other organizational entities).
	Dealing with change	Describes how and when the team responds to changes of contextual factors relative to the team (i.e., changes of the team's inner or outer environment).
	Applied practices	The work practices for IS development and IS operations (e.g., continuous delivery) or process rich and manual, applied by the team.
	Success	The type(s) of success, and its measurement, the team aims to achieve to fulfill its task.
Outer env-ironment	Market type	Attributes of the demand side of the IS product that the team provides, in its market. The market type determines market conditions the team must face during their daily work.
	Product type	Attributes of the offer side and identifying features or core affordances of the IS product that the team provides.
	Digital technology	Digital components and technical means which the team utilize to develop and deliver the IS product.

Table 6-2. Characteristics of Teams' DevOps Implementations.

Taken together, the inner environment and the outer environment summarize characteristics that influence how the team implements DevOps. In sum, these characteristics lead not to a pre-defined set of configurations or a penultimate maturity level for a DevOps implementation to strife for, but rather to a *DevOps continuum*, with gradual transitions of DevOps implementations (i.e., the sum of characteristics with their flavors) in between two extremes, see Figure 6-1. Specifically, we highlight that this does not present a ranking or ordering in the sense of a maturity level, rather the continuum entails gradual transitions between two edges, with specific positions being the "best fit" for a team to implement DevOps in its context to achieve respective IS success. Appendix D provides detailed examples of characteristics of the inner environment (Table D1; Table 6-6) and of the outer environment (Table D2; Table 6-7).

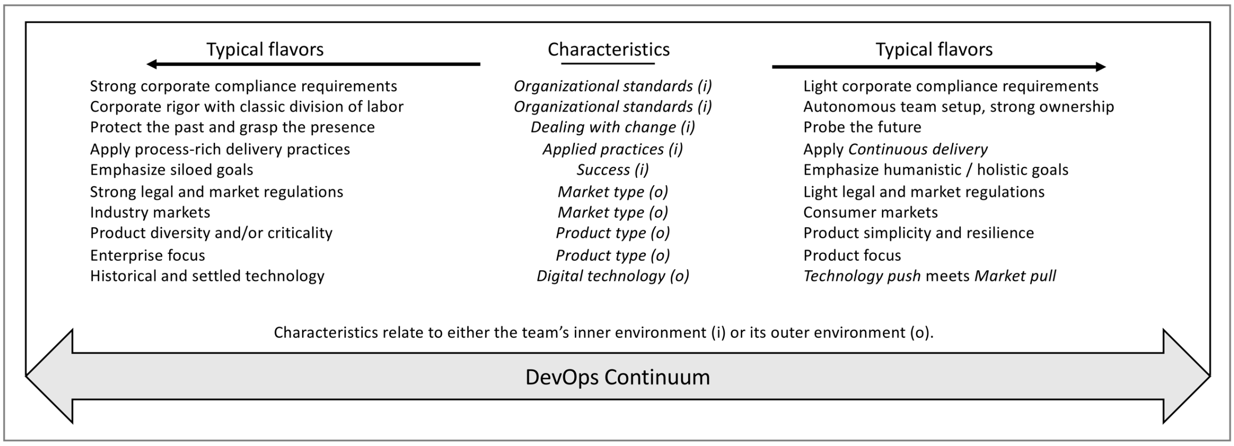

Figure 6-1. Characteristics of team's DevOps implementations, with typical flavors arranged along a continuum.

6.4.2 The DevOps Continuum's "Inner Environment"

We define characteristics of the *inner environment* as organizational standards, dealing with change, applied practices, and success. These characteristics are also intertwined and influence each other. The first characteristic is the *organizational standards.* We define these standards to be corporate, normative rules the team must confirm to, the team setup, and the responsibilities of the team related to other organizational entities including division of labor. Informants report that these standards contribute considerably to their working culture. Our data indicate that specific instances (i.e., flavors) of this characteristic vary. For example, employees of MovieStream only must align with a few organizational rules and requirements, leading to large degree of freedom and autonomy for product teams to freely decide on what counts as success, what practices to apply, and how to accomplish their tasks, whereas ManuFact must align with more and stricter organizational rules and requirements. For example, they must use centralized tools, provided and supported by other teams, for version management and binary management, to align with organizational compliance requirements (e.g., linked to regulatory requirements for safety of provided trains and power plants). Different flavors also exist for division of labor and form of collaboration across organizational entities. For example, at MovieStream product teams are integrated across functions, and develop, deliver, and then operate their IS product when it is running on production systems ("you build it, you run it"). They develop and freely share and switch operational activities across the product team. We could observe similarities to this at TransAct. These integrated teams are fully responsible for their respective IS product. For example, a "Tribe lead[11]" at TransAct emphasizes the important role of collaboration and the very tight integration of functions with business and IT as parts of one team:

> *"The teams are fully responsible for everything. We do not have a segregated responsibility for production or another team is responsible for the product. They are together fully responsible. So, it is a collaboration not a segregation model."* (TransAct, Tribe lead Paul)

One striking instance of a more nuanced positioning as regards of the working together of functions are specialized (platform) teams taking over specific tasks to support development and operations of IS products. For example, at SoftwareDev, (platform) teams serve as an integrational assistance between respective development and operations teams with focus on product delivery and problem management. In contrast, at AutoBank, due to a strict division of labor, employees from IS development do not have access to production systems and are not involved in operational concerns at all. In contrast, they operationalize DevOps differently by following an approach of distinct functions of development and operations as subunits of IT, and between the business as a separate function.

As regards the second characteristic, we define *dealing with change* as how and when the team responds to changes occurring in its inner and outer environment. Here as well different flavors exist, spanning a range from proactively seeking change to reactively avoiding any change at all. SoftwareDev pointed to situations where members of specific teams question changes and claim that they have solved a specific task the same way for 30 years. On the other side, for other teams at SoftwareDev as well as at ManuFact and MovieStream, change is seen as an opportunity to learn and to understand customer needs. Continuously optimizing the delivered IS product is mandatory for them and is only possible if the team delivers their product to production frequently. A DevOps lead architect explains:

---

[11] A Tribe lead is responsible for and supports coordination across a collection of autonomous teams ("Squads").

> *"You just need to get well enough applications into the field quickly, and check gathered technical as well as user- and business-based feedback, whether it works in production. You need to have this iteration, the DevOps cycle, to learn, to improve, and to also remove those things from production again, which are not well accepted or just do not work."* (ManuFact, DevOps lead architect Bob)

The third characteristic as part of the inner environment, *applied practices*, concerns the routines, ways of working, and methods the team uses to develop as well as deliver their IS product. Specific instances of applied practices span a broad range of flavors, from process-rich and manual, with long release cycles (e.g., at AutoBank, SoftwareDev, CashFlow), to continuous delivery using automated deployment of releases (e.g., at MovieStream, SoftwareDev, FinePrint). Automation and sharing contribute to the concept of continuous delivery, where, in its extreme, software is delivered in a pace where the customer does not recognize that a new version of the IS product is in place at all. An important practice across all cases is to utilize automation particularly to foster collaboration and to streamline communication across functions. Teams from cases utilize technical 'delivery pipelines' to automatically bring new versions of IS products to production. Our data indicate that customer feedback is gathered continuously by explicitly implementing observability as part of the IS product. This means that data of user behavior and data in production are continuously gathered to get insights on user behavior.

The fourth characteristic as part of the inner environment is *success* with its subconcepts *customer satisfaction* and *delivery performance*, the latter with its subconcepts *lead time* (a measure of time between inception of a change in the IS product and making the change available to the user), *deployment frequency* (a measure of how often new versions of the IS product are made available to the user) and *mean time to repair* (a measure of how long it takes to repair a defect on a production system). Customer satisfaction and delivery performance are intertwined and interconnected: customer satisfaction contributes to delivery performance, and vice versa. For example, the customer may be satisfied by short mean time to repair (or dissatisfied by long mean time to repair), or the customer satisfaction is high since deployment frequency is high and thus new user feedback is gathered constantly, and streamed back into improved quality and functionality of IS products. A dev manager gives an example of how those two concepts are interconnected and specifically, how they are related to digital innovation and to delivering differential value to the customer:

> *"Actually, you do not want to ship features, you want to generate outcome for the users. If it is easy to deliver output, then you start to think about the value of this output."* (ManuFact, Dev manager Lea)

6.4.3 The DevOps Continuum's "Outer Environment"

The outer environment of DevOps implementations includes the characteristics market type, product type, and digital technology. *Market type* subsumes the characteristic that defines attributes of the demand, that is, the market pull. The market pull may lead to a growing, fast changing business. The market type determines market conditions the team must face during their daily work. For example, in regulated markets and industries, it may be required by law or regulation to authorize changes in the company's application systems, that the "four eyes" principle (Basle Committee on Banking Supervision, 1997) must be followed as a segregation of various functions (by cross-checking, double signature, and dual control of assets), that staff must be trained before a new release or version of the IS can be put to production, that the team has to store all data of the application systems in data centers in a specific region, or that the IS product is to be hosted on production systems located in the customers' technical network and must thus align with their compliance rules. Unsurprisingly, we find that flavors of this characteristic differ widely between "consumer markets" and (online) business-to-customer products compared to "industry markets" and business-to-business products. For instance, to contrast to ManuFact, FinePrint and MovieStream are operating in a less-regulated consumer market; hence they deploy to production multiple times a day, and can quickly rollback in case of any issues. The deployments strongly rely on automation.

We define the *product type* as the identifying feature or the core affordance of the provided IS product. For example, for MovieStream, the provided product allows users to consume streamed movies. In this case, the resulting IS product needs to be resilient and simple. Resilient means it acts robust against failure situations and recovers from failures quickly. Simple means that MovieStream focuses on the feature that is directly available and easy to use for the user. The product focus leads to comparatively few and simple use cases such as the ability to sign up for the service and the ability to press the play button. This very intense "product focus" stands in contrast to an enterprise focus for products, for example, ERP systems (an IS product provided by SoftwareDev) used in enterprises.

The last characteristic of the outer environment is *digital technology*. We define this as the digital components and technical means (Hanseth and Modol, 2021) to develop and deliver the IS product. Customers ask for new products and new features regularly, and teams can typically fulfill these demands without facing many technical limitations. A trade-off exists between "market pull" and "technology push", as a DevOps lead architect from ManuFact explains:

> *"Software is used in many more industrial areas, that is the market pull, and on the other side we have the technology push, many data, and no restrictions such as memory, network bandwidth, or solutions in the cloud. The technology pushes. Digitization leads to many innovations and changes, I want to react on quickly. Agile software development is the development and DevOps includes the Ops part, and this helps the entire business to stay agile, to react on changes. If you don't have this in the digitization age, ... it is like a life policy."* (ManuFact, DevOps lead architect Finn)

The leveraged digital technology needs to be managed. That is, although teams may have the autonomy and freedom to choose the technology they use to accomplish their work, they may use platforms and services, centralized in distinct organizational entities, to work efficiently and leverage their own resources on working on their IS product. Newly available digital means are readily and promptly used if they provide specific benefits and technical innovations offer an advantage to provide either more customer value or being more efficient in the served market. In its extreme case, a company does not distinguish between business and IT, rather it understands itself as an "IT company" or "digital company". As a chapter lead[12] at TransAct makes clear:

> *"We call ourselves an IT bank. When you enter our company, you just meet people with IT background. It is a big IT company."* (TransAct, Chapter lead Emil)

[12] A Chapter is a group of specialists. Its members are distributed across autonomous teams ("Squads").

6.4.4 Causal Structure of DevOps Implementations

Our analysis identified contingent causal patterns explaining how teams' DevOps implementations changed as a result of configurational changes. Specifically, the teams across cases engaged in a set of similar mechanisms responding to a set of triggering conditions to achieve a similar set of outcomes. As ingredients of our causal structure, we conceptualized two interconnected mechanisms that serve as contingent causal forces and non-deterministic motors of empirically observable events (Henfridsson and Bygstad, 2013; Sayer, 2000) under given intertwined contextual conditions to work toward a respective team's "success". Whether a mechanism is actualized or not is contingent (Pawson and Tilley 1997). These intertwined mechanisms lead to innovating teams who maximize their IS success, as a Chapter lead emphasizes:

> *"You have to keep innovating. Before, projects took years. Why can't I do that in small steps? Give the customer something a little bit every time. And the good thing about that is you get feedback from users immediately."* (TransAct, Chapter lead Ahmed)

Out of several candidates, we identified two fundamental mechanisms: *effectiveness adaptation* and *efficiency adaptation*, see Table 6-3 and Figure 6-2. Both mechanisms stem from the concept of *morphing*. "Morphing" comprises significant changes and profound transformations to the IS product, including changes in the range of products and services offered, along with reconfigurations of resources, capabilities, and structures (Rindova and Kotha, 2001).

Mechanism	Description
Effectiveness adaptation	Self-reinforcing process by which decoupled IS products are created as teams increasingly collaborate across business and IT functions (IT-business collaboration), listen for opportunities to innovate, and to achieve higher customer satisfaction (i.e., differential customer value).
Efficiency adaptation	Self-reinforcing process by which teams increasingly collaborate across IT functions (intra-IT collaboration) and enhance their IS develop/delivery capabilities as they innovate and utilize new digital technologies to achieve better delivery performance.

Table 6-3. Mechanisms of DevOps Adaptation.

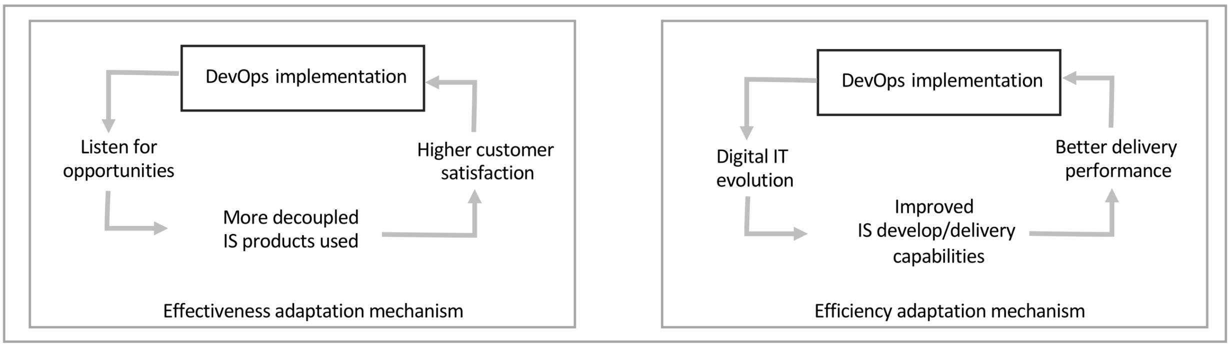

Figure 6-2. Effectiveness adaptation mechanism and efficiency adaptation mechanism.

6.4.4.1 Effectiveness Adaptation Mechanism

As part of this causal loop, a DevOps implementation enables the team to "listen for opportunity", for example, while gathering user feedback based on data of user behavior from log files and click streams. This mechanism is morphing the developed and delivered IS products to become technically decoupled, and is similar to a product innovation by architectural change (Albert and Siggelkow, 2021). The morphing may include creation of new or rearranging of existing IS products (e.g., splitting the product into smaller components), either products for internal customers (e.g., a platform), or IS products for external customers on the market (Rindova and Kotha, 2001). Most importantly, this leads to a decoupling of IS products and IS product components, for example, by using microservices (Krancher *et al.*, 2018), which results in a modular architecture (Yoo *et al.*, 2010). Technically decoupled IS products based on modular architecture more effectively provide functionality to customers and thus lead to higher customer satisfaction. This considerably changes how DevOps is implemented. A good example for the effectiveness adaptation mechanism is the following vignette from the case of SoftwareDev, see Figure 6-3.

SoftwareDev had launched the car sharing IS product "ShareMe"[13] in 2010. It was a result of a grassroot initiative to foster practical product ideas of employees, and was created for internal usage to optimize mobility and share cars across employees on the way to/from their daily work. As the winning contribution, the early version of "ShareMe" was promoted to be an IS product of SoftwareDev, later commercialized and sold to another company in 2019.

[13] Name anonymized for confidentiality reasons.

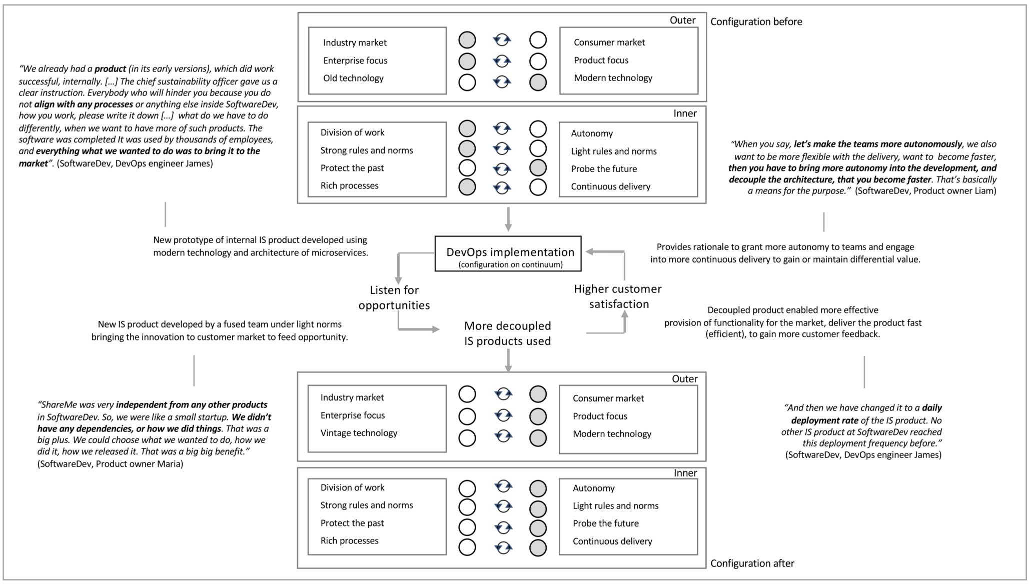

Figure 6-3. Effectiveness adaptation mechanism actualized at SoftwareDev.

As theorized above, the DevOps Continuum offers a set of characteristics, with typical flavors, describing the inner and outer environment of a team and its DevOps implementation. SoftwareDev saw a business opportunity in promoting this successful, internal product, and launching this new IS product (that had no dependencies to other products of SoftwareDev). The "effectiveness adaptation" mechanism was actualized. In its inner environment, while the mechanism was actualized, flavors changed from standard flavors of SoftwareDev to new flavors as a result of the morphing: "division of work" changed to "autonomy", "strong rules and norms" to "light rules and norms", and "rich processes" to "continuous delivery". In its outer environment, flavors changed from "enterprise focus" to "product focus" and from "industry market" to "consumer market". Although embedded in the organization, which still had their old, "classic" characteristic flavors, informants report that these specific contextual environment of "ShareMe" contributed to the further success of the IS product that was to effectively bring the new product to the public market. Besides team size and product type, autonomy was considered to be a contextual factor that positively affected the success of the IS product. Up to 20 persons autonomously worked on the product spanning business, development and operations activities. The cross-functional team integrated different perspectives of team members with their respective, specialized knowledge. Informants state that the team profited from very strong upper management commitment and its specific context being part of the sustainability line, not the regular product line, that led to degree of freedom that all corporate requirements and basic conditions were able to be questioned in order to move the product forward and deliver versions fast and in high quality. The DevOps team (with its context) was open for change and the team “probed the future”. With “ShareMe”, continuous delivery was applied, that led to new versions of the IS product shipped to customers on a daily base. This deployment frequency was new for SoftwareDev and considered to be a new blueprint for other IS products.

Optimizing efficiency was not a causal force driving the morphing, rather it was a generative result of the causal loop with the mechanism “effectiveness adaptation” being actualized. At that specific time, when the mechanism was actualized, the morphing connected to a considerably changed context. From the perspective of the entire organization, influenced by the team being (allowed to be) open for change and to “probe the future”, the morphing led to a “revolution” of the contexts of teams of the organization. In the example, instead of only transforming an old context to a new one for one specific team, it serves as a blueprint to change DevOps implementations across the entire organization. It starts a profound transformation in changing the environment of all teams of the entire organization how they develop and deliver their IS products. With a changed context after the mechanisms are actualized, the position of the DevOps team on the Continuum is changed, and thus the respective DevOps implementation is changed. Basically, the team was an autonomous unit comprising IS development, IS operations, and business employees. The actualized mechanism led to a movement on the DevOps continuum for SoftwareDev.

6.4.4.2 Efficiency Adaptation Mechanism

As part of the causal loop, a DevOps team uses any emergent digital technology as a technical means to develop IS products and deliver them to the user. The general evolution of IT (e.g., the availability of new digital components) “pushes”, and may lead to new capabilities of the team to develop and deliver its IS product more efficiently. Morphing conciliated significant changes and profound transformations to the IS product along with reconfiguration of resources, capabilities, and structures (Rindova and Kotha, 2001). As part of this change, new development and delivery capabilities emerge, for example, capabilities of how the IS product is developed and can be delivered to customers. Our findings to the first part of our research question indicated that “applied practices” is a characteristic of a team's inner environment. Based on digital IT evolution, the actualized mechanism improves the delivery performance, and thus makes development and delivery of the IS product more efficient. It moves the DevOps implementation to the right on the DevOps continuum.

A prominent example for the causal structure of this mechanism is the following case vignette from MovieStream, see Figure 6-4. MovieStream introduced a microservice architecture in 2011. As a successful streaming provider, with a strongly growing number of customers, MovieStream was faced with technical scaling challenges. Teams at MovieStream already implemented DevOps before the architecture was changed to microservices. Due to a monolithic IS product and different preferences and needs of teams and their respective delivery performances, some teams felt to be thwarted by other teams who in turn had different preferences of their delivery performances. A unified delivery performance, standardized across teams, rendered as a bottleneck for many teams. MovieStream broke up their team structure and introduced product teams, with each product team owning one or more microservices. Although these microservices are still part of the overall IS product (that is delivered to the customer), microservices are technically decoupled from each other and can thus be provided in new versions independent from other microservices (of other teams).

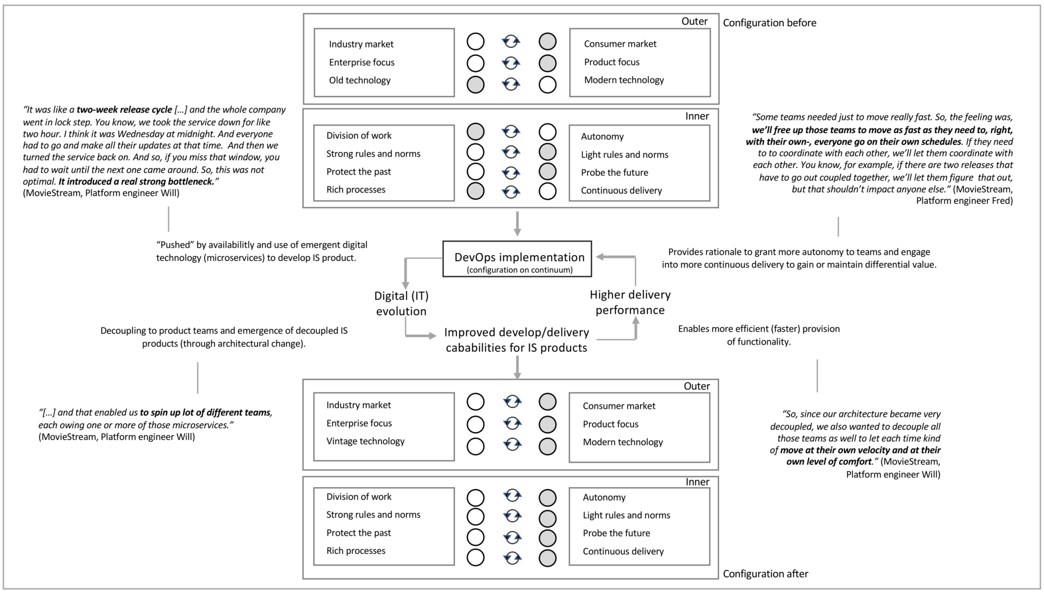

Figure 6-4. Causal structure synthesized with its components, and the efficiency mechanism actualized at MovieStream.

The improved develop and delivery capabilities allowed teams to proceed aligned with their local optimal delivery performance without affecting other teams. The mechanism "efficiency adaptation" was actualized: the "technology push" of microservices as part of the IS evolution morphed how the teams developed/delivered their IS products, leading to a better delivery performance. It was a revolution of how all teams at MovieStream implement DevOps. This specific example also demonstrates that both mechanisms are intertwined. On the one hand, "product focus based on microservices" is related to a different delivery performance (or success criteria) than "delivering the IS product based on a monolithic architecture". On the other hand, through creating new, decoupled IS products, customer satisfaction did increase, since new features could be provided faster and in better quality. As part of the causal loop, while mechanisms were actualized, the teams' context was changed and their positions on the DevOps continuum moved to the right.

Reviewing their individual environment and DevOps implementation, the used digital technology was "old" (monolithic IS product development), with division of work and rich processes (IS product could only be provided to customers in a monolithic way). The teams being open for change and (allowed to) probe the future, the DevOps implementation morphed to new IS develop/delivery capabilities, connected to new, decoupled IS teams, with an evolving, new context of autonomous teams, with continuous delivery processes, hand in hand with leveraged modern technologies, led to higher customer satisfaction and improved delivery performance.

6.5 Discussion

6.5.1 DevOps Implementations as Positions on the DevOps Continuum

Similar to companies who customize AISD (Fitzgerald et al., 2006), companies also tailor their respective DevOps implementations depending on their context (Davison and Martinsons, 2016). Our results show that these variations of DevOps implementations are gradual transitions on a DevOps continuum. We uncovered a broad set of characteristics and some typical flavors, each of them related to how DevOps is executed in a specific situational context. What we observed is that even in one organization many different DevOps implementations may exist. Following our analysis, one reason for this is that in one team, the context may be very different compared to the context of another team, both in one company. Informants in our cases articulated a strong importance of "success". While implementing DevOps, teams' (definition of) success evolves and changes. The characteristic "success" relates to the reasons why DevOps is implemented at all. While all cases operationalize their respective success differently, they work with defined goals and strive for success. Our findings suggest that teams of our sampling group A can be arranged on the right side and teams of group B on the left side of the DevOps continuum, and that teams on both sides of the continuum indeed work successfully – according to their individual measures. *Customer satisfaction* (DeLone and McLean, 1992) can be considered to be a *humanistic goal* whereas *delivery performance* can be considered to be an instrumental goal (Sarker et al., 2019). Teams need to find their place on the continuum, and their acceptable customer satisfaction and delivery performance.

For example, in the case of ManuFact with an industrial product and business-to-business context, key informants reported that the customers were dissatisfied with the high frequency of releases because users got frustrated about the constant changes in the user interfaces on a daily basis. Based on evidence from our cases, customer satisfaction (Bhattacherjee, 2001; Fowler and Highsmith, 2001) becomes a leading success criteria the more you move towards the right side of the continuum, replacing more siloed measures for success, either from the developmental side or from the operational side. Interestingly, this provides an explicit link to one of the most influential IS contributions (Urbach et al., 2009), the IS Success Model (ISSM) (DeLone and McLean, 1992). Customer satisfaction emphasizes the concept of "user satisfaction" (Urbach and Ahlemann, 2019) of the ISSM. Instrumental goals and humanistic goals are both holistic (spanning different organizational functions) and in its conjunction emphasize the sociotechnical nature of IS (Sarker et al., 2019). In this sense, DevOps can be seen as a bridge and an attempt as well as a means to reconcile the plethora of different approaches trying to measure IS success and ISD success, to focus on digital innovation, and maintain or reach differential value on the respective market.

6.5.2 DevOps Implementations as Transitions on the DevOps Continuum

We found that teams continually change the position of their DevOps implementation on the DevOps continuum through the actualization of the two mechanisms in order to align with their respective context in a given moment in time and to find the right balance between effectiveness and efficiency, including best-fit for team sizes, team boundaries, and any collaboration across teams (Hall *et al.*, 2018;

Larson *et al.*, 2022). Moreover, our data suggests that effectiveness adaptation ultimately has a clear tendency to lead to decoupled IS products, which target a specific customer group with a specific market offering and creates differential value. This underlines the importance of modularity and its positive benefits for creating configurations (Baldwin and Clark, 2000). In contrast, efficiency adaptation underlines the importance of applying modern practices and leverage digital technology to improve IS develop/delivery capabilities and enable the team to accomplish its tasks in the most efficient way (Bosch, 2019). Customer satisfaction (or effectiveness) and delivery performance (or efficiency) are intertwined: customer satisfaction contributes to delivery performance, and vice versa. For example, the customer may be satisfied by short mean time to repair (or dissatisfied by long mean time to repair), or the customer satisfaction is high since deployment frequency is high and thus new user feedback is gathered constantly and streamed back into improved quality and functionality of IS products. However, also the reverse can be true (if customers do not want to have constant and frequent changes in their IS products' functionality).

In our analysis influenced by CMO, "success" as the outcome of interest of DevOps implementations (Iannacci *et al.*, 2021) also serves as the outcome of a generative model of causation, with its intertwined characteristics making up the context of a team's DevOps implementation (see Figure C1 in Online Appendix C; Figure 6-7). The two identified mechanisms may be actualized, leading to observable events changing how the team implements DevOps. From a team perspective, this results in a changed context (a new position on the DevOps continuum) and influences both what is seen as success (the outcome of interest) and the actual instantiation of success – in effect a self-referential loop.

Our analysis of the teams' environments, and especially what is perceived as "success", offers a thought-provoking lens on the foundational concepts effectiveness and efficiency. On the one hand, DevOps bridges to effectiveness, because continuous delivery and automatically gathered insights derived from usage of the IS product on production systems support the teams to adjust, extend, or remove current IS functionality, or even find entirely new business opportunities – providing a valuable outcome for customers. On the other hand, DevOps is a means to develop and deliver a product in an efficient way, for example, by continuously delivering new versions of the IS product quickly, with lower costs and fewer errors. A DevOps implementation may well start with a focus on intra-IT alignment and the goal of becoming more efficient (efficiency adaptation), which requires automation (continuous integration, testing, and delivery). However, automation is much easier with decoupled IS products, using independent components – which may lead to effectiveness adaptation.

Our study also aligns with the finding of Sia *et al.* (2016) who reported that organizations following digital business strategies must be "ambidextrous" by being able to pursue two disparate things (in our study: effectiveness and efficiency) at the same time (Birkinshaw and Gibson, 2004; March, 1991), and that the opposing elements form part of the same continuum (Cao *et al.*, 2009; Werder and Heckmann, 2019). Following this line of argumentation, our study draws attention to the role of context in DevOps (Davison and Martinsons, 2016) and supports the finding of Werder and Heckmann (2019) that the specific context is important for ambidexterity: contextual ambidexterity relies on organizational context to help an organization achieve two seemingly conflicting goals (Birkinshaw and Gibson, 2004; Werder and Heckmann, 2019), in our case effectiveness and efficiency. Furthermore, the ambidextrous nature of DevOps has exploitative and explorative elements (Werder and Heckmann, 2019), with exploitation associated with efficiency and exploration associated with effectiveness and innovation (Gregory *et al.*, 2015; March, 1991; Sia *et al.*, 2021; Werder and Heckmann, 2019; Xue *et al.*, 2012).

6.5.3 DevOps in Relation to Fusion of Business and IT

To answer our third research question, how does DevOps relate to the fusion of business and IT, we integrate our empirical findings and theoretically discuss them on an abstract analytical level. Figure 6-5 summarizes this general theoretical elaboration and contribution.

Fundamentally, to digitally innovate, we propose that a team responsible for an IS product has two objectives: (1) produce an IS product that is valuable to the customer (i.e., be effective), and (2) increase, or at least sustain, their ability to do so (i.e., be efficient). We suggest that IS products are unique in that ignoring the latter (2) will rapidly endanger the former (1). It is about being both effective (digitally innovating and delivering valuable IS products) while continuously improving one's ability to do so efficiently (enabling to continuously deliver IS products). To be able to constantly innovate, efficiently provide effective IS products, gain and maintain differential value, and thus maximize IS success, organizations implement DevOps, with leveraged modern technology and collaborating organizational units within cross-functional teams of members of IS development, IS operations, and business (Bharadwaj *et al.*, 2013). IS products and the required delivery capability are designed (Woodard *et al.*, 2013), to digitally innovate (Bharadwaj *et al.*, 2013; Sia *et al.*, 2016, 2021), and to optimize IS success (DeLone and McLean, 1992).

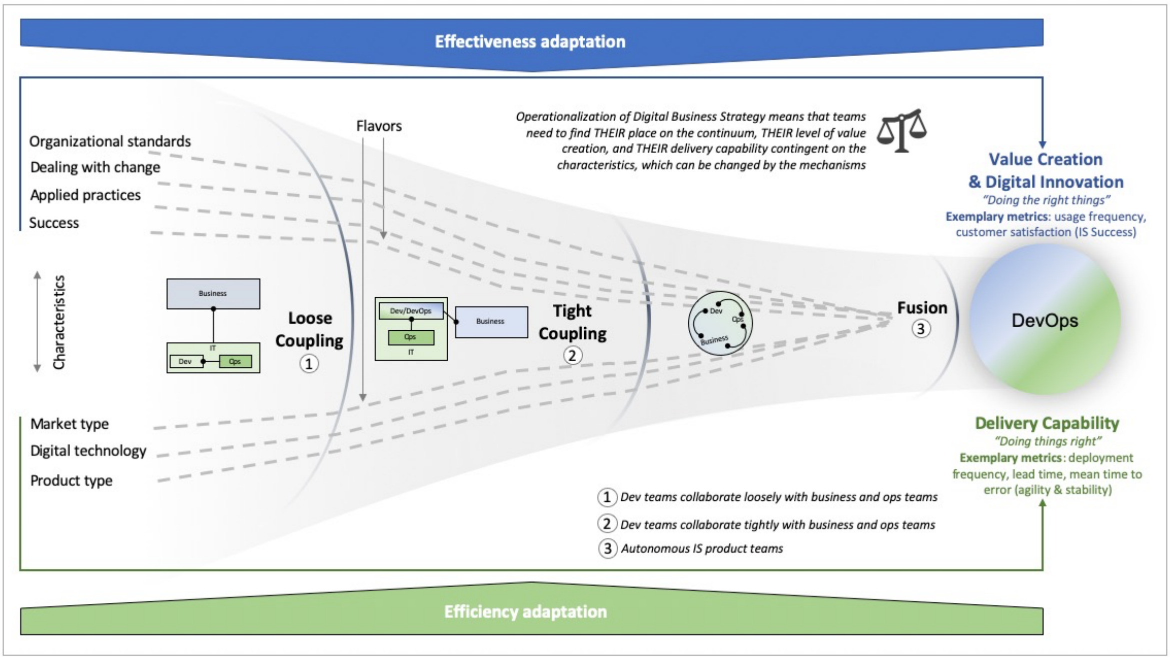

Figure 6-5. A DevOps implementation as a sum of characteristics with their specific flavors at a given time. Actualized mechanisms of effectiveness (driving innovation) and efficiency lead to a change of the team's position on the continuum.

To efficiently deliver an effective IS product, teams must continually find the position on the DevOps continuum that is the best-fit for the team to implement DevOps in its context to achieve respective IS success. Our theorizing unfolds that a transition on the DevOps continuum also entails a changed form of collaboration between the different functions. DevOps implementations are gradual transitions on the continuum between two edges: mostly detached functions of business and IT on the left, full coalescence between functions with autonomous teams fully responsible for development and operational concerns during the entire lifecycle of the IS product on the right, and setups with functions of business and operations immersed in the development function based in the middle of the continuum. We reinterpret that most case studies by Wiedemann *et al.* (2020) essentially can be positioned in the middle of the continuum.

Integrating with and confirming suggestions from Orton *et al.* (1990), an alignment between functions may be based on a *loose coupling* of connected teams, with the IT (with each subunits development and operations) and business as clearly identifiable distinct subunits, which are responsive to each other but with clearly defined responsibilities. The alignment may also be based on a *tight coupling* with distinct and responsive functions of IT and business (Hüttermann, 2021), with many nuances in-between (e.g., representatives of business or operations immersed inside the development team or vice versa), with strong collaboration between functions and a shift of responsibilities to the team. The alignment may also be based on a *fusion* of functions of business and IT functions within one distinct team that acts autonomously and thus does not require to be responsive to other subunits (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013; Overby *et al.*, 2006; El Sawy, 2003). All nuanced forms of collaboration are executed DevOps implementations that we find in practice. These DevOps implementations are informed by the situational and contextual environment of the team. For example, if a team must align with strict organizational norms for division of work between functions because of regulatory requirements or work regulations (24x7 monitoring) or an enterprise-focused business market, or parts of operations and business are immersed into the development team, but the team is faced with old technology, the IS products cannot be decoupled technically, or a centralized technical platform must be used, this will hardly lead to fused teams unless the context changes – and fused teams may not be necessary. However, based on our cases, we find that the emergence of decoupled IS products tends to lead to the gradual emergence of autonomous, independent IS product teams, which ultimately can lead to fusion. According to most informants in the studied cases, the more autonomous a team is, the more flexible it can react based on gathered information and changing context (effectiveness adaptation). Interrelated with their flavors of other characteristics (e.g., being open for change, leveraging modern technology, product focus), the team thus fuses business and IT, and is at the same time decoupled from other teams and their respective IS products and release frequency, since teams may have different incentives and leverage different digital resources, to deliver new versions of their IS product faster (or slower), with more (or less) risk.

To summarize, our findings suggest that teams operationalize digital business strategy through a composition of characteristics including collaboration across functions and leveraging modern technology (Bharadwaj *et al.*, 2013; Henry Lucas *et al.*, 2013). Finding the right balance between effectiveness and efficiency for a specific context is key to digitally innovate contingent on the faced situation, to gain or maintain differential value for the customer, and thus maximize IS success. Our finding that (IS product) teams are the point of departure leveraging digital technology to maximize IS success delivers new insights about the role of DevOps and how teams operationalize digital business strategy to stay successful on competitive markets.

We highlight two limitations stemming from our research design choices. First, inherent to case-study research, the DevOps continuum as well as the causal arguments are grounded in the situation of the cases and their respective variety of industries. We carefully decided which industries and organizations to include in our research. The continuum and the mechanisms were derived in the setting of these industries. However, we expect to observe the same or very similar flavors and mechanisms in other industries. Future research may study other DevOps implementations, across industries, relate them to the continuum, and access and expand the suggested model with its characteristics, and how this relates to digital business strategy and digital innovation. Second, as described above, we deliberately focused on established, mature companies. By contrast, it would be interesting to challenge our findings and observe DevOps in young start-ups and early ventures. Although we also examined teams that acted similar to start-ups, this could further refine our insights. We expect the same continuum and mechanisms, leading to the same theoretical elaboration, although start-ups and early ventures arguably focus more on capitalizing opportunities (i.e., effectiveness) instead of optimizing for efficiency.

6.6 Conclusion

Our study adds a timely understanding of the emerging DevOps phenomenon to the academic field. First, it identifies characteristics that influence DevOps implementations. Characteristics, with their flavors, lead to DevOps implementations that can be positioned on a broad range that makes up the DevOps continuum. This model supports explaining existing and predicting future DevOps implementations. Second, we propose that the actualization of our two mechanisms effectiveness adaptation and efficiency adaptation leads to a new or the change of an existing DevOps implementation. We uncovered the importance of "success" as both a characteristic of the team's inner environment as well as the outcome of interest for a changed DevOps implementation, and the role of DevOps as a funnel to link effectiveness and efficiency contingently. Third, our research contributes to existing concepts and theories. We find that DevOps is an attempt as well as a means to reconcile the plethora of different approaches trying to measure IS success. We find that DevOps is a vehicle for organizations to operationalize digital business strategies (through collaboration between functions and utilizing technology, along with characteristics of the DevOps continuum), to continually digitally innovate, and to maximize success. We are convinced that the results of this study help to further conceptualize and unpuzzle DevOps, and through the theoretical reasoning also contribute to understanding and theoretical grounding. For practitioners, particularly because of the empirical, multiple-case study design, we provide valuable guidance for initiatives toward DevOps, and unpack that a DevOps implementation can be operationalized differently depending on the specific context of the team, to efficiently provide effective IS products. We encourage other researchers to continue to study DevOps implementations and to create supporting or contrasting theory.

6.7 Appendices of Paper IV

This chapter contains the appendices of this paper IV. For the current review process at ISR, I uploaded the appendices to the ISR author system.

Case Site* / Sources	Group A				Group B				
	SoftwareDev	**MovieStream**	**TransAct**	**FinePrint**	**ManuFact**	**AutoBank**	**CashFlow**	**InsureMe**	**Total**
Interviews	6	3	5	2	4	4	2	4	30
Key informants	DevOps engineers (Luis, James), Product owners (Liam, David, Maria), Site reliability manager (Lucas)	Platform engineers (Olivia, Ben, Henry)	Tribe lead (Paul), Chapter leads (Finn, Ahmed), Dev engineer (Emil), Product owner (Emma)	Engineering director (Akari), Platform engineer (Frank)	DevOps lead architects (Adam, Finn, Bob), Dev manager (Lea)	Dev lead (Clara), Ops lead (Hugo), IT manager (Jack), Service manager (Pablo)	Chief digital transformation officer (Santiago), Application domain officer (Mia)	Digital transformation manager (Rodrigo), IT manager (Lewis), IT Project manager (Oscar), Coach (Nora)	
Obser-vations	-	-	-	-	-	2400 minutes	-	-	2400 minute
Audio	401 minutes	170 minutes	308 minutes	77 minutes	285 minutes	172 minutes	105 minutes	270 minutes	1788 minute
Documents	16	10	5	-	4	3	4	4	46
Document types	Product docs, Blog posts, Video presentations, Git repo, API docs	Tech blogs; Video presentations; Technical documents	Tech videos, Team model description		Product documentation, Projject documentation, Tech blog post, Presentation	KPI dashboard, Kanban card, Mission statement	Tech video, API documentation	Tech video, Project documentation, Project presentation	

* *Names anonymized for confidentiality reasons*

Table 6-4. Summary of Data Sources (Table A1).

A2. Interview protocol

We started our interviews with entry questions about the background of the informant, then discussed the transformation towards DevOps and DevOps characteristics, asking for goals, practices, tools and any effect DevOps has on complexity, communication, or knowledge management. We closed interviews asking what next steps are planned in the team's DevOps journey, asked for anything the informant wanted to add, and for a recommendation for a colleague to talk to.

Level 1: Questions verbalized to specific interviewees.
1. Background and context
1.1 Could you please quickly introduce yourself and explain what your role is? <probe for role, experience, skills>
1.2 In what team or group do you work, and what is its size and purpose?
1.3 How would you describe the organizational context of the team you are working in? Are you part of a product or service team, a line or project organization, or something different?
1.3 Can you please summarize in one or two sentences your understanding of DevOps?
1.4 Does DevOps play any role in your daily work?

2. Transformation towards DevOps
2.1 Could you please describe when the adoption and use of DevOps started in your organization, as far as you know? <probe for critical incidents/events; ask for specific examples of meetings, announcements, policies etc.>
2.2 Could you please describe why the adoption and use of DevOps started, as far as you know?
2.3 Could you please describe how the adoption and use of DevOps started, as far as you know? <probe for examples of events, initiatives, projects etc.>
2.4 Is the DevOps initiative part of a bigger initiative, e.g. digital transformation?
2.4.1 What is the bigger initiative and how does it relate to DevOps?
2.4.2 Ask if the bigger initiative is "digital transformation": How does the DevOps initiative align with classic goals a digital transformation typically has, particularly opening and quickly addressing new business opportunities?
2.4.3 Ask if the bigger initiative is "digital transformation": Do you see any relationship of DevOps to Machine Learning?
2.5 What challenges did you encounter while implementing DevOps and how did you address them? Can you briefly summarize a concrete past challenge?
2.6 What is the current state of your DevOps initiative? Could you please briefly describe if, and if yes, how DevOps has changed the way of your daily work? <probe for specific examples>
2.7. Is everybody applying DevOps? Who is, who is not? <probe for whys>
2.7.1 Are there any teams working differently?
2.7.2 Do you experience any significant boundaries or friction points in your daily work e.g. rooted in hand-overs of work units?

3. DevOps
3.1 Can you please name some main characteristics of your DevOps initiative? <probe, eg., „How does that manifest in your daily work?">
3.2 What do you think are the main differences of DevOps compared to Agile? Did you in your team apply Agile before or in conjunction with DevOps?
3.3 Do you use the term goal or do you use a different word to describe the defined result you aim to successfully achieve?
3.4 What in your opinion and experience makes up DevOps regarding goals?
3.4.1 … what personal goals do you have in your daily work?
3.4.1.1 … can you please name some goals?
3.4.1.2 … what are the origins of these goals?
3.4.1.3 … how are these goals measured? How is work prioritized?
3.4.2 … what goals do you have as a team?
3.4.2.1 … can you please name some goals?
3.4.2.2 … what are the origins of these goals? How is work prioritized?
3.4.2.3 … how are these goals measured?
3.4.3 Are there any possible conflicts based on these goals? Does any other team may have conflicting goals? If yes, can you please elaborate on a specific example?
3.5 What makes up your DevOps implementation regarding practices? Can you summarize some practices you as a team use?
3.6 What makes up your DevOps implementation regarding technology, particularly platforms and tools?
3.6.1 … are platforms and tools important for your team in order to implement DevOps? Why are they important?
3.6.2 … can you please list some of the important platforms and tools you work with?

4. Changes compared to former approaches

4.1 What is the main difference of your work today, now aligned with DevOps, compared to how you worked before?

4.2 Compared to the approach used before, what changes do you experience now, while implementing DevOps, to

4.2.1 … the organization as a whole, as a result of applying DevOps?

4.2.2 … the roles? Which roles do you have in your team?

4.2.3 … the processes and software development and delivery methods?

4.3 What aspects did improve in your daily work while aligning with DevOps? <probe for examples, eg.g. „You just mentioned X. Can you give a recent. Example for X in your daily work?“)>

4.4 What aspects did not improve in your daily work while aligning with DevOps? <probe for examples>

4.5 What effect do you see on communication while applying DevOps?

4.5.1 How do you communicate in your team and across teams?

4.5.2 Does everybody always talk to everybody?

4.6 In your working environment, what relationship does DevOps have to complexity? Is complexity a topic that has momentum in your working context? Why?

4.7 In your working environment, what relationship does DevOps have to knowledge management? Is knowledge management a topic that has momentum in your working context? Why?

5. Closing

5.1 What next steps are planned in your team’s DevOps journey?

5.2 Anything more to add we did not talk about and is important for you to mention?

5.3 Can you recommend a colleague I can talk to a bit? (in case of snowball sampling)

Level 2: Questions about each case, which present line of inquiry.

Why did the organization decide to implement DevOps?

How did the organization implement DevOps? (focus on development/delivery teams!)

What challenges did the teams encounter while implementing DevOps?

What was the achieved result of implementing DevOps?

What did change with DevOps?

What are effects on teaming, communication and complexity?

What are the origins of success criteria? How does the company name the goals it aims to achieve? How do they address conflicts of goals?

What are the next steps on the DevOps journey?

Online Appendix B

We analyzed within-case and cross-cases without any a priori hypotheses. We started by crafting thick descriptions for each case and then relied on iterative sequences of open, axial, and selective coding to generate emergent themes and higher-level concepts. Particularly, we did not begin our analysis with any provisional codes; rather, we were guided by the flow the informants offered while asking exploratory questions. We always took special attention to remain open to alternative explanations, which helped us to understand the context-sensitivity of DevOps implementations. Our model emerged through continuously cycling between analyzing the data and unfolding literature.

Figure B1 (Figure 6-6) describes the coding process of this first part with exemplary illustrations. We broke down the data to discrete parts, and examined and compared them with other codes (Saldaña, 2016). Combined with pattern coding, we summarized segments of data and results, and identified what we call "flavors of the continuum characteristics", for example, *continuous delivery*. We then applied axial coding to identify the dominant codes and to explore theoretical relations. This led to categories that form our *characteristics*. *Applied practices* and *success* are example categories produced in this iterative step. Theoretical coding synthesized the categories to create the emerging model. Example concepts that resulted from this step are *inner environment* and *outer environment* making up the core category *environment*, which in turn suggested that a specific DevOps implementation is highly context-sensitive. In addition, our analysis relied on using memos, whiteboard sketching, tables, and graphs (Miles *et al.*, 2020). While closing the cross-case analysis, we finalized the main findings as well as the resulting model of the DevOps continuum (cf. Section 4.1) as a blend of case evidence, prior research, and stand-alone logic (Eisenhardt and Graebner, 2007).

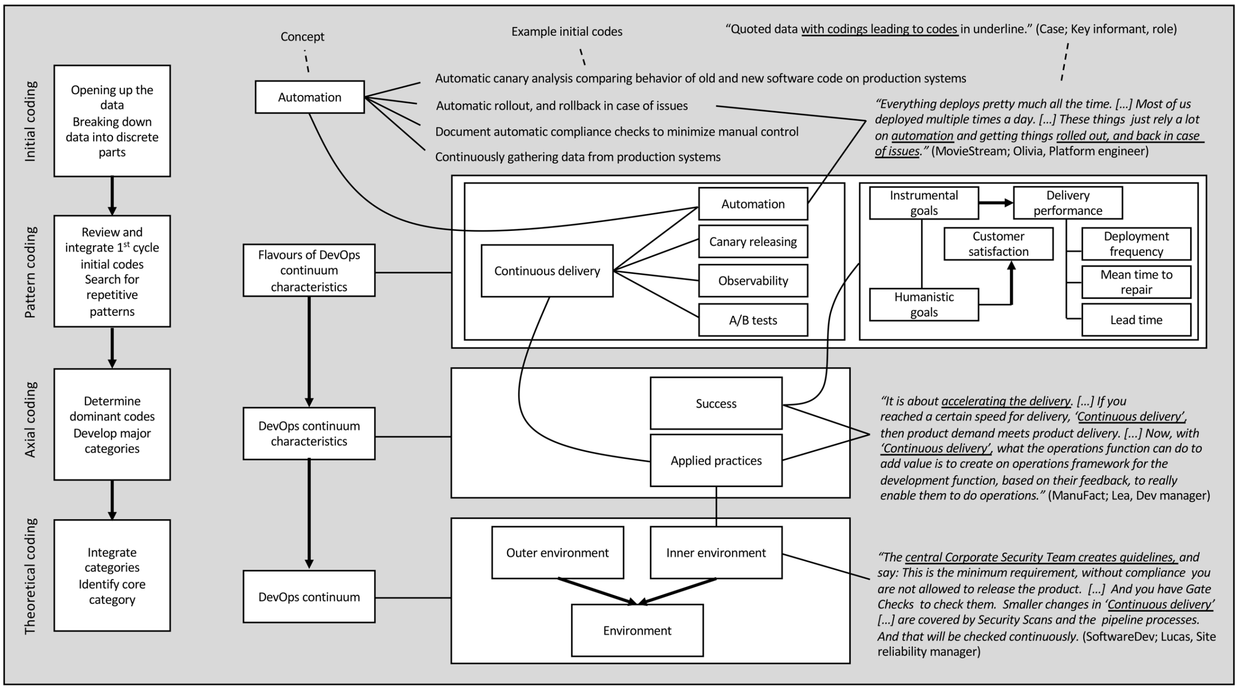

Figure 6-6. Coding process for the first part of our research question, with illustrations (excerpt) (Figure B1).

Online Appendix C

To theorize the underlying causal structure, our data analysis was influenced by the configurational framework of *Context, Mechanisms, and Outcome (CMO)* (Bygstad *et al.*, 2016; Henfridsson and Bygstad, 2013; Meyer *et al.*, 2007; Pawson and Tilley, 1997). The principles of critical realism suggested by Wynn and Williams (2012) guided our qualitative inquiry to answer our second research question. Table C1 (Table 6-5) summarizes how we implemented the framework and its principles.

We theorize the DevOps continuum as the configurational context of a team's DevOps implementation (Davison and Martinsons, 2016). We relocate the characteristic "success" to simultaneously also be the outcome of the CMO framework. In other words, a team (that implements DevOps) changes its position on the DevOps continuum by actualized mechanisms to optimize its success to achieve or maintain differential value for customers. Success is the "outcome of interest" (Iannacci *et al.*, 2021). Figure C1 (Figure 6-7) provides a graphical representation of the analytical process[14].

---

[14] Our interpretation may differ from those of other researchers. For example, we theorize that the changed context (in the moment a mechanism is actualized) is still the context (in the CMO sense), not the outcome of causation (that is the success), since success is the outcome of interest from a team's perspective. Hence, we concretized the general CMO framework by the special case of a context change during actualization.

The analytical findings of the preceding theorizing step to answer the first research question (see Section 4.1 “The DevOps Continuum”) provided the structure and context. The settings of teams (cf. Bygstad (2017)) served as the structure, with the team being the unit of analysis and the point of departure of our theorizing[15], with contingent paths leading to specific outcomes. Then, we engaged in the process of taking an empirical observation and hypothesizing mechanisms (Henfridsson and Bygstad, 2013) that generatively feed on themselves (Bygstad *et al.*, 2016; Huang *et al.*, 2017) (“self-reinforcing”), explaining these specific outcomes (Mingers *et al.*, 2013; Sayer, 2000).

We then applied a process coding procedure to discover revelatory key events (Van de Ven and Poole, 1990; Miles *et al.*, 2020; Saldaña, 2016). For each case, we focused on expressive key events and arranged them on a timeline. Some events were identified a priori (e.g., publicly available business decisions), whereas others emerged from the data (e.g., the founding of specific teams). This helped us to establish a timeline of key events that occurred over time. Figure C2 (Figure 6-8) shows an exemplary visual representation of key events at MovieStream together with illustrations of the coding process. Figures C3-C10 (Figure 6-9 to Figure 6-16) provide the timelines for all cases based on revelatory events.

Next, to analytically explain the causal structure of DevOps implementations, we gathered an understanding of which key events are generated by which mechanisms emerging from context, and identified and assessed mechanisms leading to specific outcomes. We reflected on the dynamics of DevOps implementations by theoretically interweaving the key events of the eight heterogeneous case sites. Using retroduction as a “thought trial” (Weick, 1989), we analyzed the interplay between the different DevOps implementation(s) at each case site with their respective contextual characteristics to identify the underlying mechanisms. We applied theoretical chaining from the previously identified outcomes of contingent paths to causes and vice versa (Henfridsson and Bygstad, 2013). Our results suggest that the outcome of DevOps implementations follows from two mechanisms acting in a given context (cf. Section 4.2 “Causal Structure of DevOps Implementations”).

---

[15] Delanda (2006) points out that the smallest entity (the team is our unit of analysis) can also be a convenient one to find a departure point for a bottom-up model in a micro-macro relation between the entity and its setting (cf. Coleman (1994)).

Understanding the causal composition and the dynamics of DevOps configurations, compared to alternatives such as “adaption by learning”, “adaptation by exploring”, “adaptation by morphing”, “adaptation by scaling”, “adaption by sense-making of new key employees”, and several taken from animality (“mimesis”, “scallops”), we found that the two mechanisms effectiveness adaptation and efficiency adaptation are the most dominant with the most explanatory power.

Principle	Description of Principle	Summary of Our Data Analysis and Principle Implementation
Explication of Events	Identify and abstract the events being studied, usually from empirical experiences, as a foundation what really happened in the underlying phenomena.	We used an initial coding combined with a process coding procedure to discover key events. We arranged the identified key events on a timeline for each case (see timelines as part of Online Appendix D).
Explication of Structure and Context	Identify components of social and physical structure, contextual environment, along with relationship among them.	We used our previous findings for structure and context (see Section 4.1 “The DevOps Continuum”). We identified that the DevOps continuum is the context for DevOps implementations, and that all teams work towards success operationalized by humanistic goals (customer satisfaction) and technical goals (delivery performance). Success is the characteristic that links to effectiveness and efficiency. It describes the contingent paths (see Section 4.2 “DevOps as a Funnel to Link Effectiveness and Efficiency”).
Retroduction	Identify and elaborate upon power/tendencies of structure that may have interacted within a specific context to generate explicated events.	We analytically integrated the previously offered findings for structure and context with the previously identified outcome of contingent. We applied theoretical chaining from outcome to causes and vice versa, in order to identify and assess mechanisms that led to the observable events. Teams across cases listened for opportunity to provide new or decoupled products in order to improve customer satisfaction (the “effectiveness” mechanism), and utilized modern digital technology (for example, digital could components) to improve their IS develop/delivery capabilities, in order to improve their delivery performance (the “efficiency” mechanism). We synthesized the causal structure based on the configurational framework of context, mechanisms, and outcome, leading to a framework of causal structure of DevOps implementations. The DevOps continuum as the context of DevOps implementations is morphed by actualized mechanisms leading to new flavors of the Continuum (the outcome). It thus moves the position of the team and its DevOps implementation to a different position on the continuum (see Section 4.3 “Causal Structure of DevOps Implementations”).
Empirical Corroboration	Ensure that proposed mechanisms have causal power and that they have better explanatory power than alternatives.	We iterated across candidate mechanisms and compared their explanatory power with the power of alternative mechanisms to determine the dominant mechanisms having most explanatory power.
Triangulation	Employ multiple approaches to support causal analysis based on a variety of data	We re-utilized the benefits of our multiple-case study design and its tactics to emphasize

	types and sources, analytical methods, and theories.	triangulation. Specifically, by investigating eight cases, we were able to abstract and assess our findings across cases.

Table 6-5. Summary of Application of Principles of Critical Realism for Case Study Research, based on (Williams and Karahanna, 2013; Wynn and Williams, 2012) *(Table C1).*

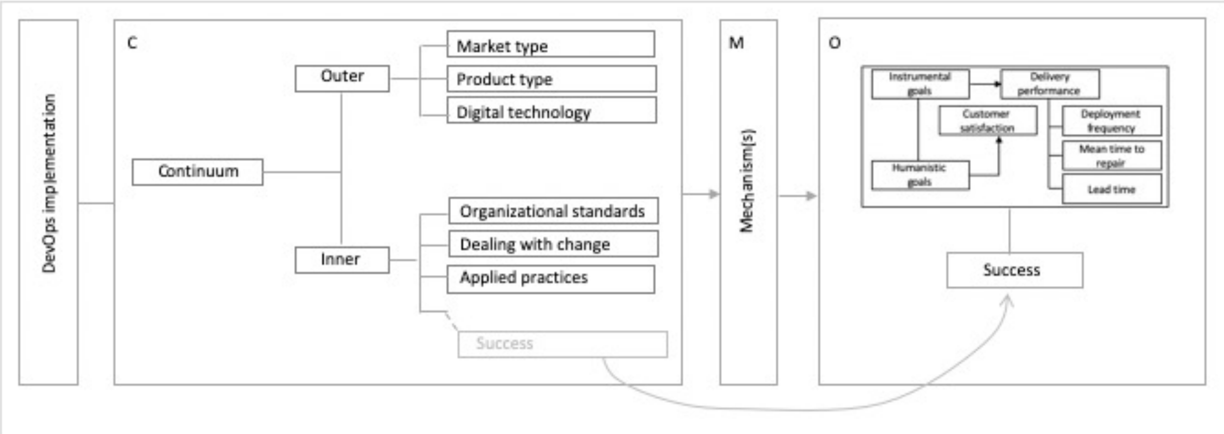

Figure 6-7. Success is a characteristic of the continuum, and the "Outcome of Interest" (Figure C1).

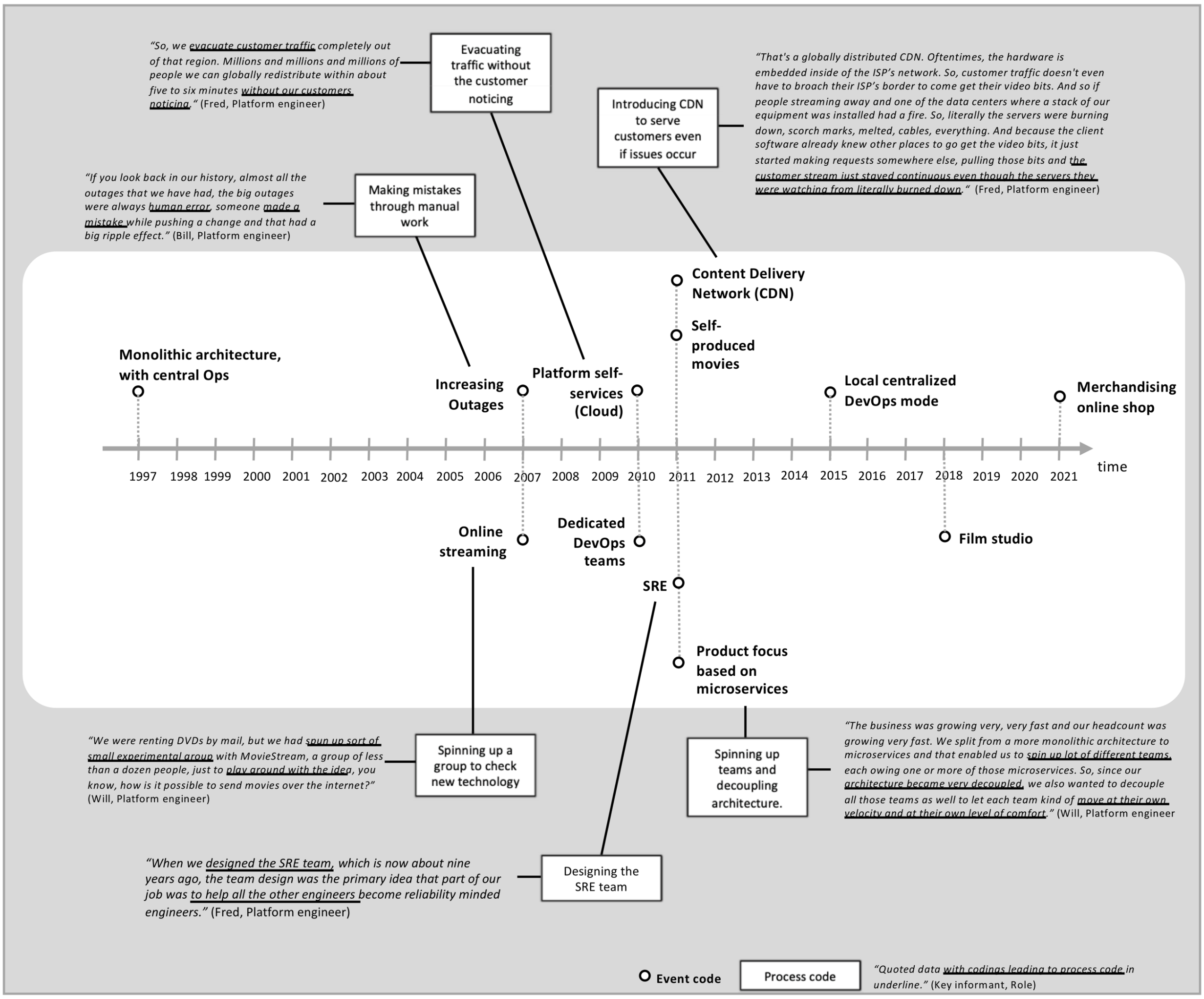

Figure 6-8. Coding process (excerpt) to identify the key events resulting in the chronological timeline for MovieStream (Figure C2).

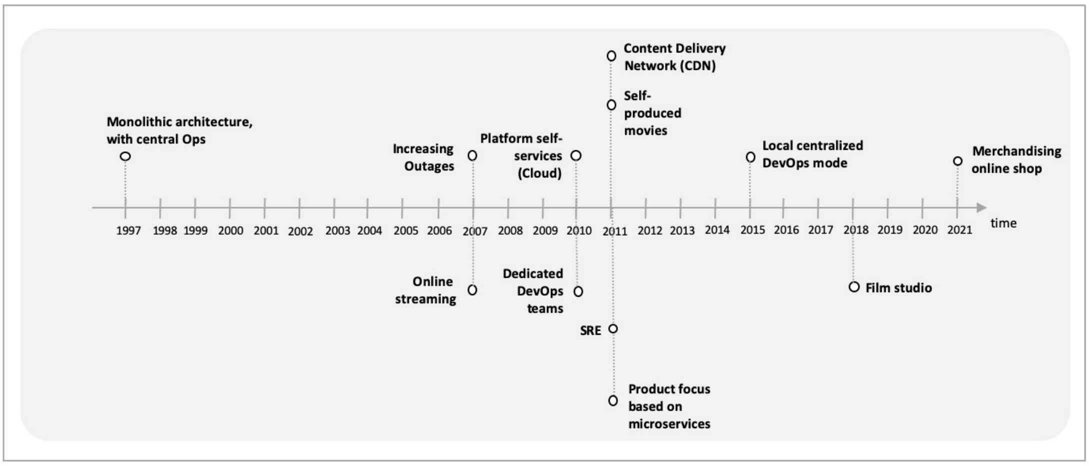

Figure 6-9. Timeline MovieStream, with example, revelatory events (Figure C3).

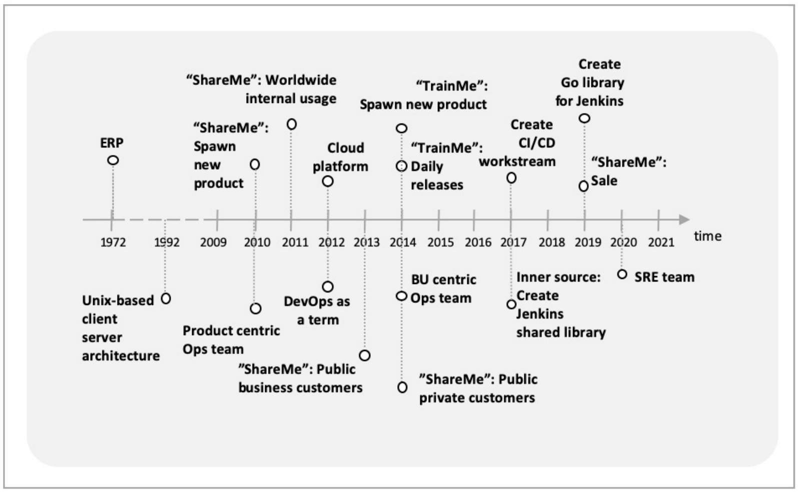

Figure 6-10. Timeline SoftwareDev, with example, revelatory events (Figure C4).

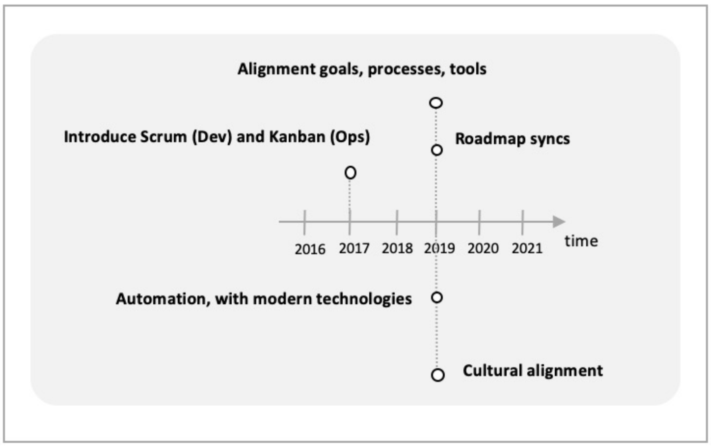

Figure 6-11. Timeline AutoBank, with example, revelatory events (Figure C5).

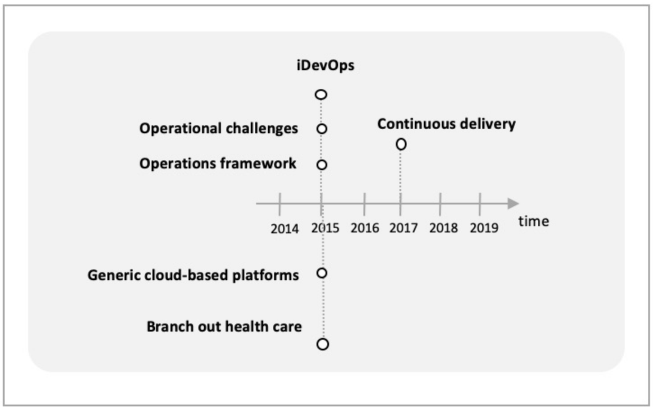

Figure 6-12. Timeline ManuFact, with example, revelatory events (Figure C6).

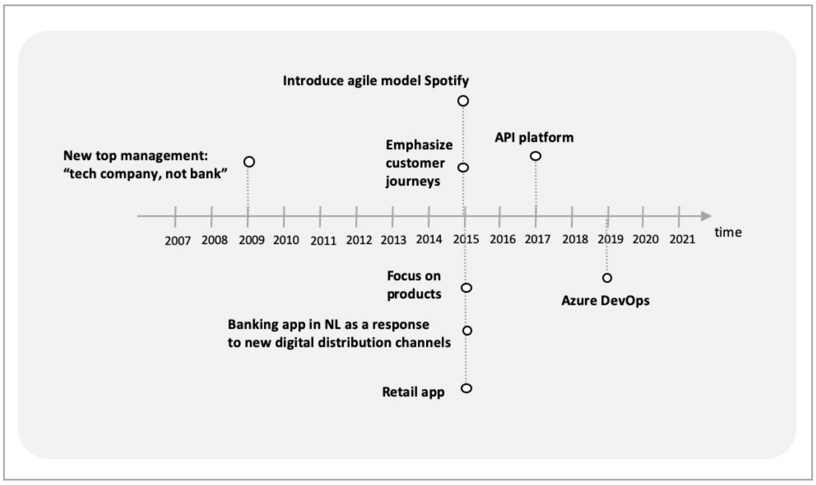

Figure 6-13. Timeline TransAct, with example, revelatory events (Figure C7).

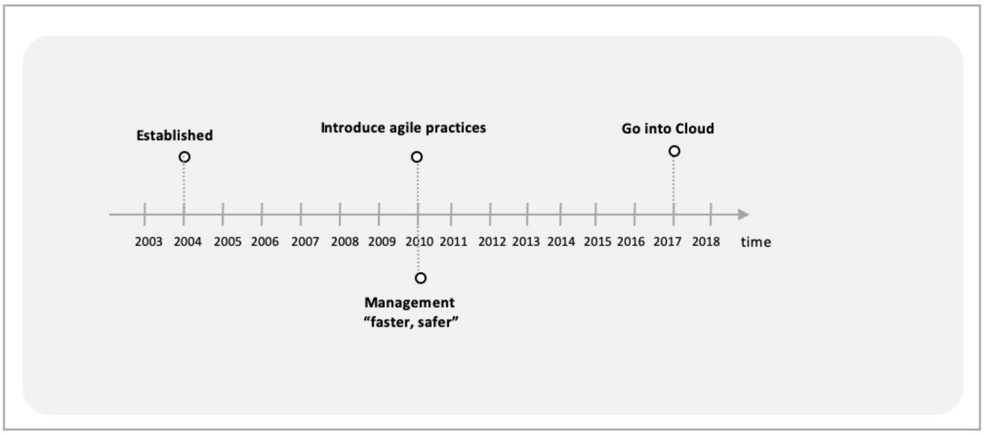

Figure 6-14. Timeline FinePrint, with example, revelatory events (Figure C8).

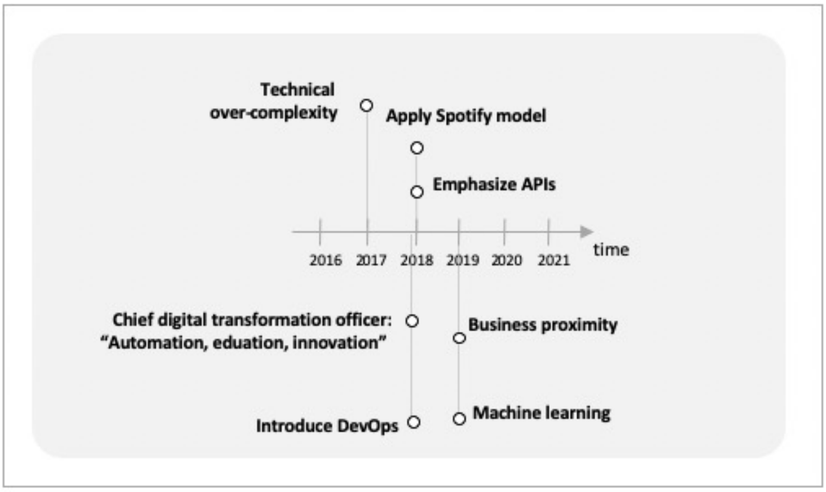

Figure 6-15. Timeline CashFlow, with example, revelatory events (Figure C9).

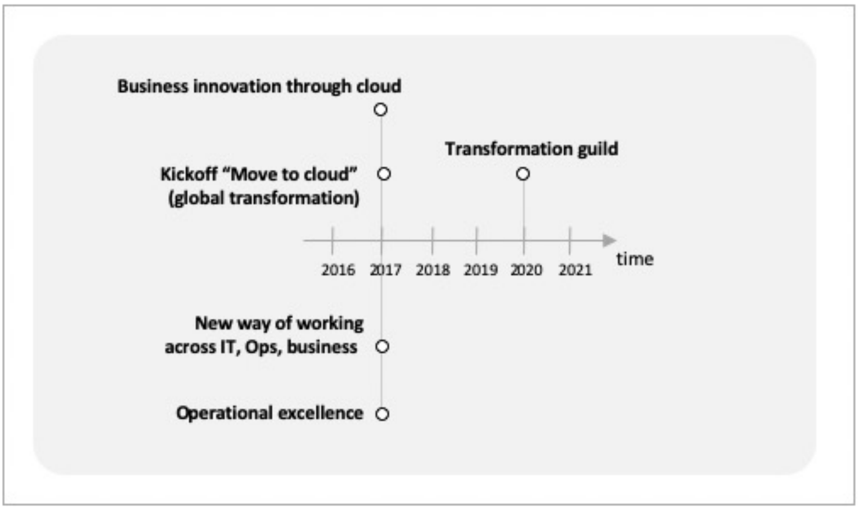

Figure 6-16. Timeline InsureMe, with example, revelatory events (Figure C10).

Online Appendix D

Case / Characteristic	Organizational Standards	Dealing with Change	Applied Practices	Success
SoftwareDev	Both, light and strong compliance requirements, depending on team-in-focus: • Light corporate compliance requirements: Usage of security stages in the delivery pipeline are not mandatory. They can be deactivated. • Strong corporate compliance requirements: Gate checks are implemented for larger changes. Teams need approvals from central teams. *"That is a job for*	Both protect the past and probe the future, depending on team-in-focus: • Protect the past: for some teams, team members, criticize and question the need for changes: *"I'm still scetical about an ERP running in the cloud. I worked in another business for 20 years, [...] and if I remember changes on an ERP system,*	Both, continuous delivery and process rich, depending on the team-in-focus: • Continuous delivery: some teams have highly automated tasks, and deliver releases very quickly. For example, based on an "Inner Source" approach, a shared Jenkins code library is continuously updated and can always be used in	Both, humanistic and instrumental goals: Blend of both customer satisfaction and delivery performance. Delivery performance is the main goal to positively affect customer satisfaction. However, it may also negatively affect customer

Case / Characteristic	Organizational Standards	Dealing with Change	Applied Practices	Success
	our infrastructure experts, who hold the running infrastructure in a secure state. [...] We use highest DLS-standards." (Site reliability engineer Lucas) Both, corporate rigor with classic division of work and autonomous team setup with strong ownership, depending on team-in-focus: • Corporate rigor with classic divison of work: *"There is a dedicated operations team responsible for database updates, installations. The development is not doing this, this is doing a separate team."* (Product owner Liam) • Autonomous team setup, strong ownership: Many teams have their own Jenkins build server provided by a centralized infrastructure team. *"I would say that most teams now are doing their own operations with the support of the platform team"*. (Product owner Maria)	*we had huge efforts for documentation and testing.*" (Product owner Luis) • Probe the future: other teams and team members perceive change as an opportunity to learn and to understand customer needs: "*We have to evolve continuously. And we must watch the market for new trends.*" (Product owner Liam)	its most recent versions. Product owner for CI/CD tools exist. Compliance is automated: *"We must confirm to over 200 formal criteria (defined by company, customer, country regulations) to bring a change to production. [...] What has changed is the perception, that those things can be automated"*. (DevOps engineer James) • Process rich: other teams have rich processes. For example, for a product, there are defined maintenance windows: *"That is determined by contract. Tuesday and Thursday, early in the morning, maintenance operations on production can be done."* (Product owner David)	satisfaction if product demand is not aligned with product delivery: *"We aim to deliver every day. But the customer is not happy to detect new features every day, [...] we are now delivering weekly, and it turns out, that this is an acceptable frequency also for the customer."* (Product owner David)
MovieStream	Light corporate compliance requirements: No strict audit process exists. *"We provide guard rails, not gates."* (Platform engineer Henry) Autonomous team setup with strong ownership: Teams have a lot of freedom and autonomy as regards decision-making and action-taking: *"There is a large freedom and responsibility culture which means that teams are very free in choosing how they do things exactly."* (Platform engineer Olivia)	Probe the future: Teams improve their development practices and tools continuously as part of an ongoing effort. This also includes the socio-technical part: e.g., the platform team supports product teams to improve while addressing production incidents: *"We also cover the sociotechnical side of thinks. [...] How does a good and successful on call team take care of the people side of things"?* (Platform engineer Henry)	Continuous delivery: Product teams can use platform and central libraries to keep up the flow. They deploy multiple times a day to production. As part of continuous delivery, modern approaches are used: *"We have a lot of resilience tooling that can be used during canary processes."* (Platform engineer Henry)	Primarily humanistic goals: Customer satisfaction has top priority (e.g., usage of "login" and "play" functionality, and hiding of technical issues). Instrumental goals contribute to humanistic goals.
Transact	Strong corporate compliance requirements: Compliance is very important. A lot of out-of-the-box compliancy is part of a platform including	Probe the future: Learning is essential: *"It's a journey with learning, lessons learned and failures, lots of failures, and step*	Continuous delivery: It is considered to be important to constantly deliver business value, as a product owner states	Both, humanistic and instrumental goals: blend of both, customer satisfaction and

Case / Characteristic	Organizational Standards	Dealing with Change	Applied Practices	Success
	control whether necessary approvals exist. *"The platform will give you a lot of heavy lifting, and a lot of out of the box compliancy."* Tribe lead Paul Both, rigor with alternative division of work and autonomous team setup with strong ownership, depending on team-in-focus: • Corporate rigor with classic division of work, with colloboration of devs and ops: *"For us, an ops engineer should not be somebody who accepts the work of dev but instead somebody who automates the checks for them."* IT area lead *"We definitely also still have operations in the sense that and we have people that mainly focus on operations"*. (Dev engineer Emil) • Autonomous team setup, strong ownership: Teams are decoupled via interfaces. *"We did the transformation three years ago through the API platform. So, everything is a REST API."* (Dev engineer Emil) Teams are organized as autonomous squads.	*by step going further."* Tribe lead Paul	*"Short cycles, small portions, but adding business value every time, this really helps in customer satisfaction"* (Product owner Emma)	delivery performance. *"There is a DevOps pipeline and we build it for all engineers so that they can deliver software, that's the main goal for our end customer"*. (Chapter lead Finn)
FinePrint	Light corporate compliance requirements: It is possible to flexibly adopt rules: *"The pain was following the rules and it turned out that the rules weren't necessary. And so, we removed those rules"* (Platform engineer Frank) Autonomous team setup, strong ownership: Teams own their product and how it is delivered. They may flexibily introduce security scanning and testing into automatic delivery processes. The platform team is responsible for enabling product teams: *"The team owns that codebase, own their deployment pipeline, […],*	Probe the future: Informants report a constant culture of learning, and of development engineers getting involved in operational concerns: *"Because they are still learning. It is definitely a culture of getting involved when there's a problem in production"*. (Platform engineer Frank)	Continuous delivery: Codified and automated delivery steps. *"I was part of a spearhead in front of a lot of awesome, amazing people to help move that into continuous delivery. And now we release that piece of software five times a day"*. (Platform engineer Frank)	Humanistic goals: Customer satisfaction has top priority, and instrumental goals contribute. *"Everything we do, we have to look at the customer. And whether that's our employees considering our external customers and how we want to make them successful. Or whether it's an internal team looking at how we make other teams*

Case / Characteristic	Organizational Standards	Dealing with Change	Applied Practices	Success
	whether they want to introduce security scanning or different types of testing […]. And the platform is responsible for giving them the primitives to be able to assemble those sorts of things." (Engineering director Akari)			*successful."* (Engineering director Akari)
ManuFact	Strong corporate compliance requirements: "*DevOps aims an established environment for ensuring a stringent operational quality and regulatory compliance of industry software and software intense systems." (*DevOps lead architect Bob) Both, corporate rigor with classic division of work and autonomous team setup with strong ownership, depending on team-in-focus: • Corporate rigor with classic division of work. Ops engineers and dev engineers with classic responsibilities. Partly ops is "in the field", i.e. users of the IS products check compliancy and operate. Compliancy is important: *"We have projects, where we, together with the customer, collaborate, to reach a lead time of change of two weeks, including FTA compliancy."* (DevOps lead architect Bob) • Autonomous team setup with strong ownership: Ops team provides an operations framework based on feedback of the dev teams. They enable the dev teams to themselves operate their IS products. *"The operations team provides the operational framework, to enable the dev teams to operate their services".* (Dev manager Lea)	Probe the future: Accelerate and close the feedback loop to users, in order to learn from feedback, and to continuously push product innovation, including applying changes to existing IS products running in the field: *"That's called brownfield digitalization. Many facilities do exist. That's an invest, you cannot imagine."* (DevOps lead architect Bob)	Continuous delivery: Lot of automation across the entire development and delivery in order to enable continuous delivery. Continuously gather user feedback to improve the products. *"We have a closed feedback loop. That is actually the clou. […] not only the path to the customer, but also the way back, then I can really learn, and the gathered data, […], I can learn from them, and again stream in improvements at the start, in order to continuously improve".* (DevOps lead architect Finn)	Both, humanistic and instrumental goals, depending on team-in-focus. *"We use software delivery performance metrics, such as lead time, cycle time and failure recovery. […] And we use classic metrics, e.g. ISO-25010, because availability is important for us. Not 99.99, that's simple. Instead we often have five nines or six nines." (*DevOps lead architect Bob) *"In the plannings, we aim to focus on the (valuable) outcome, instead of just producing features, and to produce those features as output".* (Dev manager Lea) *"Product demand is equal to product delivery. This is the continuous delivery effect."* (Dev manager Lea)
AutoBank	Strong corporate compliance requirements: As a bank, work is aligned with the "dual control	Protect the past: People are sceptical regarding new technologies because they use and	Process rich: A release calendar schedules the deliveries of IS	Instrumental goals are the top priority. Strong alignment with

Case / Characteristic	Organizational Standards	Dealing with Change	Applied Practices	Success
	principle" to ensure a chain of responsibilities with approvals. Corporate rigor with classic division of labor: The development team is strongly focusing on the development, the operations team is strongly focusing on the operations of the IS products, with distinct approaches: *"Scrum for the development, Kanban for operations. To align this with each other, this is really a strong challenge."* (Ops lead Hugo)	know existing technology for 20 years. New technologies may lead to new sources of failures in the daily work and staff's goal is to not do any errors: *"We need creative heads that motivate others to escape from the mass" (* IT manager Jack)	products. The function of IS development is driven by quantity of functionality, IT operations by operational quality of IS products, both measured and incentivized by monetized compensation.	classic metrics of production of features (development) and stability of IS product (downtimes, errors during early life of new versions of IS product). *"If we have 100 hand-overs from dev to ops, in a specific time interval, only three are allowed to provoke an error in production".* (Ops lead Hugo)
CashFlow	Strong corporate compliance requirements: As a bank, work is aligned with the "dual control principle" to ensure responsibilities with approvals. Corporate rigor with classic division of labor: Ops personnel has more permissions on specific systems than dev staff. Specific roles exist for system responsibility. *"We define the roles and permissions of roles. This means, we also define the roles, technically."* (Application domain officer Mia)	Protect the past: Informants report that it is hard to include all colleagues to continuously learn. *"With education, we want to convey innovation, particularly around digital business models."* (Chief digital transformation officer Santiago) Courses with learning paths are provided, however, some colleagues can not follow with the speed of continuous learning. *"[…] and in the courses, you have some people participating, they are struggling, and you know, that will be very hard for them."* (Chief digital transformation officer Santiago)	Continuous delivery: Frequent delivery of small, decoupled IS products, reyling on APIs, to test business models. *"We try to test new business models, by bringing small products to the market quickly."* (Chief digital transformation officer Santiago)	Both, humanistic and instrumental goals, depending on team-in-focus. The top management defines goals to bring out products quickly. *"Our COO wants that".* (Chief digital transformation officer Santiago)
InsureMe	Corporate rigor with classic division of labor: Organization is split into global IT and local IT, with global IS products (e.g. ERP) and local IS products. Central teams provide platforms, development teams can autonomously work inside the provided guard rails and classic job roles. Platform ensures compliancy: *"DevOps teams work autonomously. […] But we have configured, that they are not able to use any*	Protect the past: Preference to organize and schedule all steps before. Work that indicates changes tends to be postponed: "*I'm only intrinsically motivated if I can apply those new learned things. I do not learn French if I will never speak French*" (Coach Nora) Transformation guild to support teams to transform to new cloud-based practices: *"Each*	Both, continuous delivery and process rich, depending on the team-in-focus: • Continuous delivery: Some teams have highly automated tasks, and deliver new versions of IS products very quickly • Process rich: Tend to organize and schedule up front and all steps in	Both, humanistic and instrumental goals, depending on team-in-focus. Goals are distributed to teams: *"We work on distribute metrics to teams, that teams can define their own metrics, their own goals. […] We don't predefine goals, because teams are too*

Case / Characteristic	Organizational Standards	Dealing with Change	Applied Practices	Success
	unapproved services of our cloud provider." (Digital transformation manager Rodrigo) Division of work due to work regulations and scaling considerations: *"I think only the biggest teams run their 24x7 monitoring and incident management themselves, because it does not pay off."*. (Digital transformation manager Rodrigo)	*individual team would be unable to totally manage the organization of the transformation itself."* (IT project manager Oscar)	advance. *"It happens that it is preferred to do planning weeks, months more, just to avoid to make wrong decisions"*. (Coach Nora)	*different."* (IT manager Lewis)

Table 6-6. Examples of Characteristics of Inner Environment (Table D1).

Case / Characteristic	Market Type	Product Type	Digital Technology
SoftwareDev	Both, industry and consumer markets, depending on the team-in-focus: • Industry markets: ERP systems, for use in enterprises. • Consumer markets: For example, an IS product for sports management for clubs (that typically have no own IT) and an IS product for car pooling used by people sharing their cars.	Product diversity, with both simple products and complex, critical products, depending on the team-in-focus: • Car pooling is a simple IS product. • ERP: complex IS product. • Sports management: complex; IS product integrates diverse functionality to manage sports clubs.	Both, old and modern, depending on the team-in-focus: • Old: classic, settled technology such as ERP. *"It is a big challenge to run an ERP in the cloud, and that it works correctly"*. (Site reliability engineer Lucas) • Modern: Cloud-based, with an own cloud system (SoftwareDev provides a cloud solution) and usage of other cloud providers. *"Our own cloud product is not the leading one any more"*. (Product owner David) Cloud technology is associated with faster delivery. *"And then we said, we want to more enagage with Cloud. [...] we want to develop and deliver faster."* (Product owner Liam)
MovieStream	Consumer markets: The IS product is for consumers who as end users stream movies using the Internet. Industry markets: The newly created branch "film studio" will create movies, and thus vertically integrates with "streaming".	Streaming is a simple IS product: Although technically very complex in the back-end, it focuses on a simplified functionality for the front-end. For the newly created branch "film studio", this is different: *"But when they (outages) happen, right, it's unfortunate. But you know, we're not a hospital, we are not a financial company, right? We're providing entertainment. When it happens in the studio, if some piece of technology breaks there, it has the possibility of slowing or stopping that production."* (Platform engineer Henry)	Modern: Cloud, distributed. *"That is a global distributed Content Delivery Network. [...] Customer traffic doesn't even have to broach their ISP's border to come get their video bits. [..] On the backend side, we operate primarily out of three AWS global regions"*. (Platform engineer Henry)
Transact	Consumer markets: IS products are for consumers	As products of a bank, products are medium-	Modern: Cloud as a strategic platform. *"We decided to go to Azure DevOps*

Case / Characteristic	Market Type	Product Type	Digital Technology
	as end users who operate their banking using the Internet.	complex to complex. Breakdown to smaller sub-products minimizes complexity.	*because they have data centers all over the world. And also Microsoft, because we are a bank."* (Chapter lead Finn) Entire firm is considered to be a tech company with banking license. *"Our chairmain has also said that we are a tech company with a banking license, then the most important enabler for the banking business is tech."* (Tribe lead Paul)
FinePrint	Consumer markets: IS products offers functionality for consumers as end users to print goods based on paper, e.g. business cards.	As a product to print goods (such as business cards), the IS product is relatively simple.	Diverse modern technologies are used: *"We have some stuff on virtual machines in the cloud. We have some stuff on an old Kubernetes cluster and we have some stuff on a new Kubernetes cluster"*. (Platform engineer Frank) Technology is also distributed: *"Because it is a physical manufacturing company, […] we also have physical points of presence. We have the warehouses where the products get manufactured. And so we have infrastructure running in those locations as well"*. (Engineering director Akari)
ManuFact	Industry markets: IS products are used by business users (e.g. in power plants, hospitals, or trains). After reaching speed threshold, further acceleration is not necessary: *"Competiting companies are definitely not faster than us. […] our industry is different."* (Dev manager Lea)	Product diversity, with complex products. IS products typically run on infrastructure of clients, where they have to pass quality assurance and installation routines: *"Big customers want new versions, but before they are used, they are buffered onsite at the customer to run compliance checks. Afterwards they are deployed to facilities.“ (*DevOps lead architect Adam)	Both, Old and modern, depending on the team-in-focus: • Old: classic, settled technology, existing facilities, "brownfield". • Modern and distributed, heterogenous: *"We have everything from cloud, to on-prem, to edge, then different execution platforms, and of course: how do I come (with my software) into the hospital or into the nuclear power plant or into the train."* (DevOps architect.) *"In our case, we take data from the hospital, and send them into the cloud"*. (Dev manager Lea)
AutoBank	Industry markets: IS products offer finance functionality for car sellers as end users. Company is seen as depending on two market types, financial and car.	As products of a bank, products are semi-complex to complex.	Old as well as modern development technologies. Vintage (mainly manual) delivery technologies.
CashFlow	Consumer markets: IS products are for consumers as end users who can operate their bankings using the Internet.	As products of a bank, products are semi-complex to complex. Breakdown to smaller sub-products minimizes complexity.	Both, vintage and modern, depending on the team-in-focus: • Old: classic, settled technology. Some teams have highly automated tasks, and deliver new versions of IS products very quickly • Modern: Cloud. *"Everybody has to develop in an AWS-enabled way. It must run in the cloud, that is the common denominator"*. (Application domain officer Mia)
InsureMe	Consumer markets: IS products with functionality to offer and manage	As products of an insurance company, products are complex. Breakdown to	Both, vintage and modern, depending on the team-in-focus:

Case / Characteristic	Market Type	Product Type	Digital Technology
	insurances for consumers as end users.	smaller sub-products minimizes complexity.	• Old: classic, settled technology. *“Central operations is done for our mainframe platforms. […] There, our old monolithic are running.“ (*Digital transformation manager Rodrigo) • Modern: Cloud: *“They can order everything by one-click and then use it themselves, either on AWS, or Microsoft Azure, or our own Open-PaaS environment”* Digital transformation expert. Cloud is strategically important to delivery efficiently, although it is not considered to be unique selling point for itself: *“Typically, an insurance company does not have any innovation pressure. […] You normally don’t gain any competitive advantage through technology.”* (Coach Nora)

Table 6-7. Examples of Characteristics of Outer Environment (Table D2).

References

Abrahamsson, P., Conboy, K. and Wang, X. (2009), "'Lots done, more to do': the current state of agile systems development research", *European Journal of Information Systems*, Vol. 18 No. 4, pp. 281–284.

Albert, D. and Siggelkow, N. (2021), "Architectural Search and Innovation", *Organization Science*, No. August, pp. 0–18.

Alt, R., Auth, G. and Kögler, C. (2021), *Continuous Innovation with DevOps*, Springer.

Alt, R., Marco, J., Thomas, L., Stephan, P., Urbach, N., Wunderlich, N. and Alt, R. (2020), "Software-Defined Business Implications for IT Management", *Business & Information Systems Engineering*, Springer Fachmedien Wiesbaden, Vol. 62 No. 6, pp. 609–621.

Anderson, D.J. and Reinertsen, D.G. (2010), *Kanban - Successful Evolutionary Change for Your Technology Business*, Blue Hole Press, Seattle.

Baccarini, D. (1999), "The Logical Framework Method for Defining Project Success", *Project Management Journal*, Vol. 30 No. 4, pp. 25–32.

Baham, C. and Hirschheim, R. (2021), "Issues , challenges , and a proposed theoretical core of agile software development research", No. December 2020, pp. 1–27.

Baiyere, A., Gupta, A., Grover, V., Woerner, S. and Lyytinen, K. (2018), "Digital 'x' - A new tune for IS research or old wine in new bottles?", *ICIS 2017: Transforming Society with Digital Innovation*, pp. 0–5.

Baiyere, A. i, Grover, V., Lyytinen, K.J., Woerner, S. and Gupta, A. (2023), "Digital ' x '— Charting a Path for Digital-Themed Research", *Information Systems Research*.

Baldwin, C.Y. and Clark, K.B. (2000), *Design Rules: The Power of Modularity*, MIT Press.

Balijepally, V. (2009), "Are Two Heads Better Than One for Software Development?", *MIS Quarterly*, Vol. 33 No. 1, pp. 91–118.

Bandara, W., Furtmueller, E., Gorbacheva, E., Miskon, S. and Beekhuyzen, J. (2015), "Achieving rigor in literature reviews: Insights from qualitative data analysis and tool-support", *Communications of the Association for Information Systems*, Vol. 37, pp. 154–204.

Baskerville, R.L., Myers, M.D. and Young. (2020), "DIGITAL FIRST: THE ONTOLOGICAL REVERSAL AND NEW CHALLENGES FOR INFORMATION SYSTEMS RESEARCH", *MIS Quarterly*, Vol. 44 No. 2, pp. 509–523.

Basle Committee on Banking Supervision. (1997), "Core Principles for Effective Banking Supervision", available at: https://www.bis.org/publ/bcbs30a.pdf (accessed 13 August 2021).

Beck, K. (1999), "Extreme Programming Explained.", Addison Wesley, Boston.

Beck, K., Cockburn, A., Fowler, M. and Schwaber, K. (2001), "Agile Manifesto", available at: https://agilemanifesto.org/iso/en/manifesto.html (accessed 11 April 2023).

Benbasat, I., Goldstein, D.K. and Mead, M. (1987), “The Case Research Strategy in Studies of Information Systems”, *MIS*, No. September.

Berente, N. (2020), “Agile development as the root metaphor for strategy in digital innovation”, *Handbook of Digital Innovation*, 1st ed., Edward Elgar Publishing Ltd, Northampton, pp. 83–96.

Berente, N. and Yoo, Y. (2012), “Institutional Contradictions and Loose Coupling: Postimplementation of NASA’s Enterprise Information System”, *Information Systems Research*, Vol. 23 No. 2, pp. 376–396.

Bharadwaj, A., Sawy, O.A. El, Pavlou, P.A. and Venka. (2013), “Digital Business Strategy: Toward a Next Generation of Insights”, *MIS Quarterly*, Vol. 37 No. 2, pp. 471–482.

Bhaskar, R. (2008), *A Realist Theory of Science*, Routledge, New York.

Bhattacherjee, A. (2001), “Understanding Information Systems Continuance: An Expectation-Confirmation Model Motivation for the Study”, *MIS Quarterly*, Vol. 25 No. 3, pp. 351–370.

Birkinshaw, J. and Gibson, C. (2004), “Building Ambidexterity into an Organization”, *MIT Sloan Management Review*, No. June 2004.

Bosch, J. (2019), “From efficiency to effectiveness: Delivering business value through software”, *Lecture Notes in Business Information Processing*, Vol. 370 LNBIP, pp. 3–10.

Bosch, J. and Olsson, H.H. (2021), “Digital for real: A multicase study on the digital transformation of companies in the embedded systems domain”, *Journal of Software: Evolution and Process*, Vol. 33 No. 5, pp. 1–25.

Brinkmann, S. and Kvale, S. (2014), *InterViews*, 3rd ed., SAGE Publications, London.

vom Brocke, J., Simons, A., Riemer, K., Niehaves, B. and Plattfault, R. (2015), “Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information System Research”, *Communications of the Association for Information Systems*, Vol. 37, pp. 205–224.

Brown, S. and Eisenhardt, K.M. (1997), “The Art of Continuous Change: Linking Complexity Theory and Time-Paced Evolution in Relentlessly Shifting Organizations”, *Administrative Science Quarterly*, Vol. 42 No. 1, pp. 1–34.

Burgelman, R.A. (1983), “A Process Model of Internal Corporate Venturing in the Diversified Major Firm”, *Administrative Science Quarterly*, Vol. 28 No. 2, pp. 223–244.

Burgess, R.G. (1984), *In the Field: An Introduction to Field Research*, Unwin Hyman Ltd, London.

Bygstad, B. (2017), “Generative innovation: A comparison of lightweight and heavyweight IT”, *Journal of Information Technology*, Vol. 32 No. 2, pp. 180–193.

Bygstad, B. and Munkvold, B.E. (2011), “In Search of Mechanisms. Conducting a Critical Realist Data Analysis”, *International Conference on Information Systems*, No. 2011.

Bygstad, B., Munkvold, B.E. and Volkoff, O. (2016), “Identifying generative mechanisms through affordances: A framework for critical realist data analysis”, *Journal of Information Technology*, Vol. 31 No. 1, pp. 83–

96.

Cao, L., Mohan, K., Xu, P. and Ramesh, B. (2009), “A framework for adapting agile development methodologies”, *European Journal of Information Systems*, Vol. 18 No. 4, pp. 332–343.

Cao, Q., Gedajlovic, E., Zhang, H., Science, S.O., Ambidexterity, O., Aug, J., Cao, Q., *et al.* (2016), “Unpacking Organizational Ambidexterity: Dimensions , Contingencies , and Synergistic Effects”, *Organization Science*, Vol. 20 No. 4, pp. 781–796.

Carroll, N., Conboy, K. and Wang, X. (2023), “From Transformation to Normalisation: An Exploratory Study of Large-Scale Agile Transformation”, *Journal of Information Technology*.

Cassell, C. and Symon, G. (2012), *Qualitative Organizational Research: Core Methods and Current Challenges*, SAGE Publications, London.

Chakravarty, A., Grewal, R. and Sambamurthy, V. (2013), “Information Technology Competencies, Organizational Agility, and Firm Performance: Enabling an Facilitating Roles”, *Information Systems Research*, Vol. 24 No. 4, pp. 976–997.

Chandler Jr., A.D. (2018), *Strategy and Structure: Chapters in the History of the Industrial Enterprise*, Reprint., Forgotten Books.

Charmaz, K.C. (2006), *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*, SAGE Publications, London, England.

Chow, T. and Cao, D.-B. (2008), “A survey study of critical success factors in agile software projects”, *Journal of Systems and Software*, Vol. 81, pp. 961–971.

Coleman, J.S. (1994), *Foundations of Social Theory*, Harvard University Press.

Colquitt, J.A. and George, G. (2011), “From the Editors: Publishing in AMJ - Part 1: Topic Choice”, *Academy of Management Journal*, Vol. 54 No. 3, pp. 432–435.

Conboy, K. (2009), “Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development”, *Information Systems Research*, Vol. 20 No. 3, pp. 329–354.

Creswell, J.W. and Creswell, D.J. (2018), *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, edited by 5th, SAGE, London.

Danermark, B., Ekström, M. and Karlsson, J.C. (2019), *Explaining Society: Critical Realism in the Social Science.*, 2e ed., Routledge, Abingdon.

Davis, F.D. (1989), “Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology”, *MIS Quarterly*, Vol. 13 No. 3, pp. 319–340.

Davison, R.M. and Martinsons, M.G. (2016), “Context is king! Considering particularism in research design and reporting”, *Journal of Information Technology*, Vol. 31 No. 3, pp. 241–249.

Debois, P. (2008), “Agile Infrastructure & Operations”, *Agile 2008 Toronto*, available at: https://www.jedi.be/presentations/agile-infrastructure-agile-2008.pdf (accessed 11 April 2023).

Dekleva, S.M. (1992), “The Influence of the Information Systems Development Approach on Maintenance”, *MIS Quarterly*, Vol. 16 No. 3, pp. 355–372.

Delanda, M. (2006), *A New Philosophy of Society: Asemblage Theory And Social Complexity*, Continuum.

DeLone, W.H. and McLean, E.R. (1992), “Information systems success: The quest for the dependent variable”, *Journal of Management Information Systems*, Vol. 3 No. 1, pp. 60–95.

DeLone, W.H. and McLean, E.R. (2003), “The DeLone and McLean Model of Information System Success: A Ten-Year Update”, *Journal of Management Information Systems*, Vol. 19 No. 4, pp. 9–30.

DeLone, W.H. and McLean, E.R. (2016), *Information Systems Success Measurement*, *Foundation and Trends in Information Systems*, Vol. 2, now Publishers Inc., Boston.

DevOpsDays. (2009), “Devopsdays Ghent 2009”, available at: http://www.devopsdays.org/events/2009-ghent/ (accessed 11 April 2023).

Diegmann, P., Dreesen, T., Binzer, B. and Rosenkranz, C. (2018), “Journey Towards Agility : Three Decades of Research on Agile Information Systems Development”, *International Conference on Information Systems*, pp. 1–17.

digital.ai. (2022), “16th Annual State of Agile Report”, available at: https://info.digital.ai/rs/981-LQX-968/images/AR-SA-2022-16th-Annual-State-Of-Agile-Report.pdf (accessed 11 April 2023).

Dikert, K., Paasivaara, M. and Lassenius, C. (2016), “Challenges and success factors for large-scale agile transformations: A systematic literature review”, *Journal of Systems and Software*, Elsevier Inc., Vol. 119, pp. 87–108.

Dingsøyr, T., Bjørnson, F.O. and Schrof, J. (2023), “A longitudinal explanatory case study of coordination in a very large development programme: the impact of transitioning from a first- to a second-generation large-scale agile development method”, *Empirical Software Engineering*, Empirical Software Engineering, Vol. 28 No. 1.

Dingsøyr, T., Nerur, S., Balijepally, V. and Moe, N.B. (2012), “A decade of agile methodologies: Towards explaining agile software development”, *Journal of Systems and Software*, Vol. 85, pp. 1213–1221.

Dubé, L. and Paré, G. (2003), “Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations”, *MIS Quarterly*, Vol. 27 No. 4, pp. 597–635.

Dybå, T. and Dingsøyr, T. (2008), “Empirical studies of agile software development: A systematic review”, Vol. 50, pp. 833–859.

Dybå, T. and Dingsøyr, T. (2009), “What do we know about agile software development?”, *IEEE Software*, Vol. 26 No. 5, pp. 6–9.

Eisenhardt, K.M. (1989), “Building Theories from Case Study Research”, *Academy of Management Review*, Vol. 14 No. 4, pp. 532–550.

Eisenhardt, K.M. and Graebner, M.E. (2007), “Theory Building From Cases: Opportunities And Challenges”, *Academy of Management Journal*, Vol. 50 No. 1, pp. 25–32.

Eisenhardt, K.M. and Piezunka, H. (2011), “Complexity theory and corporate strategy”, *The SAGE Handbook of Complexity and Management*, SAGE, pp. 506–523.

Fitzgerald, B., Hartnett, G. and Conboy, K. (2006), “Customising agile methods to software practices at Intel Shannon”, *European Journal of Information Systems*, Vol. 15 No. 2, pp. 200–213.

Fitzgerald, B. and Stol, K.J. (2017), “Continuous software engineering: A roadmap and agenda”, *Journal of Systems and Software*, Elsevier Ltd., Vol. 123, pp. 176–189.

Forsgren, N. (2018), *Accelerate. Building and Scaling High Performing Technology Organizations.*, IT Revolution.

Forsgren, N., Clay, P.F., Wang, X. and Durcikova, A. (2016), “The integrated user satisfaction model: Assessing information quality and system quality as second-order constructs in system administration”, *Communications of the Association for Information Systems*, Vol. 38 No. 1, pp. 803–839.

Fowler, M. and Highsmith, J. (2001), “The Agile Manifesto”, *Software Development*, Vol. 9 No. 8, pp. 28–32.

Frefer, A., Mahmoud, M., Haleema, H. and Almamlook, R. (2018), “Industrial Engineering & Management Overview Success Criteria and Critical Success Factors in Project Management”, *Industrial Engineering & Management*, Vol. 7 No. 1, pp. 1–6.

Furr, N., Ozcan, P. and Eisenhardt, K.M. (2022), “What is digital transformation? Core tensions facing established companies on the global stage”, *Global Strategy Journal*, No. April, pp. 1–24.

Gall, M. and Pigni, F. (2021), “Taking DevOps mainstream: a critical review and conceptual framework”, *European Journal of Information Systems*, Taylor & Francis, Vol. 00 No. 00, pp. 1–20.

Gerow, J.E., Grover, V., Thatcher, J. and Roth, P.L. (2014), “Looking Toward the Future of IT-Business Strategic Alignment through the Past”, *MIS Quarterly*, Vol. 38 No. 4, pp. 1159–1186.

Glaser, B.G. (1978), *Theoretical Sensitivity*, Sociology Press.

Glaser, B.G. and Strauss, A.L. (1967), *The Discovery of Grounded Theory: Strategies for Qualitative Research*, de Gruyter, Chicago, Ill.

Google. (2019), “Accelerate: State of DevOps 2019”, available at: https://services.google.com/fh/files/misc/state-of-devops-2019.pdf (accessed 11 April 2023).

Google. (2022), “Accelerate: State of DevOps Report 2022”, available at: http://services.google.com/fh/files/misc/2022_state_of_devops_report.pdf (accessed 11 April 2023).

Graebner, M.E. and Eisenhardt, K.M. (2004), “The Seller’ s Side of the Story: Acquisition as Courtship and Governance in Syndicate in Entrepreneurial Firms”, *Administrative Science Quarterly*, Vol. 49 No. 3, pp. 366–403.

Greenwood, R., Oliver, C., Lawrence, T.B. and Meyer, R.E. (2020), *The SAGE Handbook of Organizational Institutionalism*, SAGE Publications, London.

Gregor, S. (2017), “The Nature of Theory in Information Systems”, *MIS Quarterly*, Vol. 30 No. 3, p. 611.

Gregory, R.W., Keil, M., Muntermann, J. and Mähring, M. (2015), “Paradoxes and the Nature of Ambidexterity in IT Transformation Programs”, *Information Systems Research*, Vol. 26 No. 1.

Grodal, S., Anteby, M. and Holm, A.L. (2021), “Achieving Rigor in Qualitative Analysis: The Role of Active Categorization in Theory Building”, *Academy of Management Review*, Vol. 46 No. 3.

Gupta, V., Kapur, P.K. and Kumar, D. (2017), “Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling”, *Information and Software Technology*, Vol. 92, pp. 75–91.

Hall, K.L., Vogel, A.L., Huang, G.C., Serrano, K.J., Rice, E.L., Tsakraklides, S.P. and Fiore, M. (2018), “Gaps on Collaboration in Science”, *American Psychologist*.

Hanseth, O. and Modol, J.R. (2021), “The dynamics of architecture-governance configurations: An assemblage theory approach”, *Journal of the Association for Information Systems*, Vol. 22 No. 1, pp. 130–155.

Heeager, L.T. (n.d.). “Introducing Agile Practices in a Documentation-Driven Software Development Practice: A Case Study”, *Journal of Information Technology Case and Application Research*, Vol. 14 No. 1, pp. 3–24.

Hemon-Hildgen, A., Lyonnet, B., Rowe, F. and Fitzgerald, B. (2020), “From Agile to DevOps: Smart Skills and Collaborations”, *Information Systems Frontiers*, Vol. 22 No. 4, pp. 927–945.

Hemon-Hildgen, A., Rowe, F. and Monnier-Senicourt, L. (2020), “Orchestrating automation and sharing in DevOps teams: a revelatory case of job satisfaction factors , risk and work conditions”, *European Journal of Information Systems*, Taylor & Francis, Vol. 29 No. 5, pp. 474–499.

Henderson, J.C. and Venkatraman, N. (1999), “Strategic alignment: Leveraging information technology for transforming organizations”, *IBM Systems Journal*, Vol. 32 No. 1.

Henfridsson, O. and Bygstad, B. (2013), “The Generative Mechanisms of Digital Infrastructure Evolution”, *MIS Quarterly: Management Information Systems*, Vol. 37 No. 3, pp. 907–931.

Henry Lucas, J., Agarwal, R., Clemons, E.K., Sawy, O.A. El and Weber, B. (2013), “Impactful Research on Transformational Information Technology: An Opportunity to Inform New Audiences”, *MIS Quarterly*, Vol. 37 No. 2, pp. 371–382.

Holmqvist, M. and Pessi, K. (2006), “Agility through scenario development and continuous implementation: A global aftermarket logistics case”, *European Journal of Information Systems*, Vol. 15 No. 2, pp. 146–158.

Hong, W., Thong, J.Y.L., Chasalow, L.C. and Dhillon, G. (2011), “User Acceptance of Agile Information Systems: A Model and Empirical Test”, *Journal of Management Information Systems*, Vol. 28 No. 1, pp. 235–272.

Horlach, B., Drews, P., Drechsler, A., Schirmer, I. and Böhmann, T. (2020), “Reconceptualising Business-IT Alignment for Enabling Organisational Agility”, *Proceedings of the European Conference on Information Systems (ECIS)*, pp. 1–17.

Huang, J., Henfridsson, O., Liu, M.J. and Newell, S. (2017), “Growing on Steroids: Rapidly Scaling the User Base of Digital Ventures Through Digital Innovation”, *MIS Quarterly*, Vol. 41 No. 1, pp. 301–314.

Humble, J. and Molesky, J. (2011), “Devops: A Software Revolution in the Making?”, *Cutter IT Journal*, Vol. 24 No. 8, p. 6.

Hummel, M., Rosenkranz, C. and Holten, R. (2013), “The role of communication in agile systems development: An analysis of the state of the art”, *Business and Information Systems Engineering*, Vol. 5 No. 5, pp. 343–355.

Hüttermann, M. (2012), *DevOps for Developers*, Apress, New York.

Hüttermann, M. (2021), “The DevOps Continuum: Walking the Shadowy Bridge from Information Systems Development to Operations”, *ECIS*.

Iannacci, F., Simeonova, B. and Kawalek, P. (2021), “Investigating the determinants of inter-organizational information sharing within criminal justice: A context-mechanism-outcome approach”, *Journal of Information Technology*.

Ives, B., Olson, M.H. and Baroudi, J.J. (1983), “The Measurement of User Information Satisfaction”, *Communications of the ACM*, Vol. 26 No. 10, pp. 785–793.

Kahre, C., Hoffmann, D. and Ahlemann, F. (2017), “Beyond business-IT Alignment - Digital Business Strategies as a Paradigmatic Shift: A Review and Research Agenda”, *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 4706–4715.

Kaiser, A.K. (2023), *Reinventing ITIL® and DevOps with Digital Transformation*, 2nd ed., Apress, New York.

Karlsson, F. and Agerfalk, P. (2009), “Exploring agile values in method configuration”, *European Journal of Information Systems*, Vol. 18 No. 4, pp. 300–316.

Khan, L.R. and Sarker, R.A. (2002), “An optimal batch size for a JIT manufacturing system”, *Computers and Industrial Engineering*, Vol. 42, pp. 127–136.

Kohli, R. and Melville, N.P. (2019), “Digital innovation: A review and synthesis”, *Information Systems Journal*, Vol. 29 No. 1, pp. 200–223.

Kotlarsky, J. (2007), “Re-Engineering at Lecroy Corporation: The Move to Component-Based Systems”, *Journal of Information Technology*, Vol. 22 No. 4, pp. 32–41.

Krancher, O., Luther, P. and Jost, M. (2018), “Key Affordances of Platform-as-a-Service: Self-Organization and Continuous Feedback”, *Journal of Management Information Systems*, Vol. 35 No. 3, pp. 776–812.

Krey, M. (2022), “DevOps Adoption: Challenges & Barriers”, *Proceedings of the 55th Hawaii International Conference on System Sciences*, pp. 7297–7309.

Larson, L.E., Harris-watson, A.M., Carter, D.R., Dechurch, L.A. and Zaccaro, S.J. (2022), “Staying Apart to Work Better Together: Team Structure in Cross-Functional Teams”, *Academy of Management Discoveries*.

Lee, G. and Xia, W. (2005), “The ability of information systems development project teams to respond to business and technology changes: A study of flexibility measures”, *European Journal of Information Systems*, Vol. 14 No. 1, pp. 75–92.

Lee, G. and Xia, W. (2010), “Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility”, *MIS Quarterly*, Vol. 34 No. 1, pp. 87–114.

Levy, Y. and Ellis, T.J. (2006), “A systems approach to conduct an effective literature review in support of information systems research”, *Informing Science Journal*, Vol. 9, pp. 181–211.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., *et al.* (2004), “Agile software development in large organizations”, *Computer*, Vol. 37 No. 12, pp. 26–34.

Locke, K. and Golden-Biddle, K. (1997), “Constructing opportunities for contribution: Structuring intertextual coherence and ‘Problematizing’ in organizational studies”, *Academy of Management Journal*, Vol. 40 No. 5, pp. 1023–1062.

Luftman, J., Papp, R. and Brier, T. (1999), “Enablers and Inhibitors of Business-IT Alignment”, *Communications of the Association for Information Systems*, Vol. 1 No. March.

Luftman, J. and Zadeh, H.S. (2011), “Key information technology and management issues 2010-2011: An international study”, *Journal of Information Technology*, Nature Publishing Group, Vol. 26, pp. 193–204.

Luz, W.P., Pinto, G. and Bonifácio, R. (2019), “Adopting DevOps in the real world: A theory, a model, and a case study”, *Journal of Systems and Software*, Vol. 157 No. July.

Lwakatare, L.E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H.H., Bosch, J. and Oivo, M. (2016), “Towards DevOps in the embedded systems domain: Why is it so hard?”, *Proceedings of the Annual Hawaii International Conference on System Sciences*, Vol. 2016-March, pp. 5437–5446.

Lwakatare, L.E., Kuvaja, P. and Oivo, M. (2016), “An exploratory study of DevOps: Extending the dimensions of devops with practices”, *ICSEA 2016 : The Eleventh International Conference on Software Engineering Advances*, No. August, pp. 91–99.

Mangalaraj, G., Mahapatra, R. and Nerur, S. (2009), “Acceptance of software process innovations- The case of extreme programming”, *European Journal of Information Systems*, Vol. 18 No. 4, pp. 344–354.

March, J.G. (1991), “Exploration and Exploitation in Organizational Learning”, *Organization Science*, Vol. 2 No. 1, pp. 71–87.

Markus, M.L. and Robey, D. (1988), “Information Technology and Organizational Change: Causal Structure in Theory and Research”, *Management Science*, Vol. 34 No. 5, pp. 583–598.

Maruping, L.M. and Matook, S. (2020), “The evolution of software development orchestration: current state and an agenda for future research”, *European Journal of Information Systems*, Taylor & Francis, Vol. 29 No. 5, pp. 443–457.

Maruping, L.M., Zhang, X. and Venkatesh, V. (2009), “Role of collective ownership and coding standards in coordinating expertise in software project teams”, Vol. 18 No. 4, pp. 355–371.

Matook, S., Lee, G. and Fitzgerald, B. (2021), “Information Systems Development”, *MISQ Research Curations*, available at: https://www.misqresearchcurations.org/blog/2021/12/7/information-systems-development (accessed 11 April 2023).

Meyer, A.D., Tsui, A.S., Hinings, C.R., Academy, T., Dec, N. and Meyer, A.D. (2007), “CONFIGURATIONAL APPROACHES TO ORGANIZATIONAL ANALYSIS”, Vol. 36 No. 6, pp. 1175–1195.

Miles, M.B., Hubermann, A.M. and Saldaña, J. (2020), *Qualitative Data Analysis. A Methods Sourcebook.*, 4th ed., SAGE Publications, London.

Mingers, J., Mutch, A. and Willcocks, L. (2013), “Critical Realism in Information Systems Research”, *MIS Quarterly: Management Information Systems*, Vol. 37 No. 3, pp. 795–802.

Mishler, E.G. (1986), *Research Interviewing: Context and Narrative*, Harvard University Press.

Mishra, A. and Otaiwi, Z. (2020), “DevOps and software quality: A systematic mapping”, *Computer Science Review*, Elsevier Inc., Vol. 38, p. 100308.

Morton, P. (2006), “Using Critical Realism to Explain Strategic Information Systems Planning”, *Journal of Information Technology Theory and Application*, Vol. 8 No. 1, pp. 1–20.

Mukumbang, F.C. (2023), “Retroductive Theorizing: A Contribution of Critical Realism to Mixed Methods Research”, *Journal of Mixed Methods Research*, Vol. 17 No. 1, pp. 93–114.

Myers, M.D. (2009), *Qualitative Research in Business & Management*, SAGE Publications, London.

Nambisan, S. (2003), “INFORMATION SYSTEMS AS A REFERENCE DISCIPLINE FOR NEW PRODUCT DEVELOPMENT”, *MIS Quarterly*, Vol. 27 No. 1, pp. 1–18.

Nambisan, S., Lyytinen, K., Majchrzak, A. and Song, M. (2017), “Digital Innovation Management: Reinventing Innovation Management Research in a Digital World”, *MIS Quarterly*, Vol. 41 No. 1, pp. 223–238.

Nambisan, S., Lyytinen, K. and Yoo, Y. (2020), *Handbook of Digital Innovation*, 1st ed., Edward Elgar Publishing Ltd, Northampton.

Nokia Corporation. (2023), “Our history.”, available at: https://www.nokia.com/about-us/our-history/ (accessed 27 March 2023).

Oliver, L.W. (1987), “Research Integration for Psychologists: An Overview of Approaches”, *Journal of Applied Social Psychology*, Vol. 17 No. 10, pp. 860–874.

Olsson, H.H. and Bosch, J. (2020), “Going digital: Disruption and transformation in software-intensive embedded systems ecosystems”, *Journal of Software: Evolution and Process*, Vol. 32 No. 6, pp. 1–24.

van Oorschot, K.E., Sengupta, K. and Van Wassenhove, L.N. (2018), “Under Pressure: The Effects of Iteration Lengths on Agile Software Development Performance”, *Project Management Journal*, Vol. 49 No. 6, pp. 78–102.

Orton, J.D., Weick, K.E., Orton, J.D. and Weick, K.E. (1990), “Loosely Coupled Systems: A

Reconceptualization", *Academy of Management*, Vol. 15 No. 2, pp. 203–223.

Osmundsen, K. and Bygstad, B. (2021), "Making sense of continuous development of digital infrastructures", *Journal of Information Technology*.

Overby, E., Bharadwaj, A. and Sambamurthy, V. (2006), "Enterprise agility and the enabling role of information technology", *European Journal of Information Systems*, Vol. 15 No. 2, pp. 120–131.

Pawson, R. and Tilley, N. (1997), *Realistic Evaluation*, SAGE Publications, London.

Peppard, J., Galliers, R.D. and Thorogood, A. (2014), "Information systems strategy as practice: Micro strategy and strategizing for IS", *Journal of Strategic Information Systems*, Elsevier B.V., Vol. 23 No. 1, pp. 1–10.

Peppard, J. and Ward, J. (2004), "Beyond strategic information systems: Towards an IS capability", *Journal of Strategic Information Systems*, Vol. 13 No. 2, pp. 167–194.

Petter, S., DeLone, W.H. and McLean, E.R. (2008), "Measuring information systems success: Models, dimensions, measures, and interrelationships", *European Journal of Information Systems*, Vol. 17 No. 3, pp. 236–263.

Petter, S., DeLone, W.H. and McLean, E.R. (2013), "Information Systems Success: The Quest for the Independent Variables", *Journal of Management Information Systems*, Vol. 29 No. 4, pp. 7–61.

Poole, M.S., Van de Ven, A.H., Dooley, K. and Holmes, M.E. (2000), *Organizational Change and Innovation Processes*, Oxford University Press, New York.

Poppendieck, M. and Poppendieck, T. (2003), *Lean Software Development*, Addison Wesley, Boston.

Qumer Gill, A., Loumish, A., Riyat, I. and Han, S. (2018), "DevOps for information management systems", *VINE Journal of Information and Knowledge Management Systems*, Vol. 48 No. 1, pp. 122–139.

Rafi, S., Yu, W., Akbar, M.A., Mahmood, S., Alsanad, A. and Gumaei, A. (2021), "Readiness model for DevOps implementation in software organizations", *Journal of Software: Evolution and Process*.

Rajlich, V. (2006), "Changing the Paradigm of Software Engineering", *Communications of the ACM*, Vol. 49 No. 8, pp. 67–70.

Ramasubbu, N., Bharadwaj, A. and Tayi, G.K. (2015), "Software Process Diversity: Conceptualization, Measurement, and Analysis of Impact on Project Performance", *MIS Quarterly*, Vol. 39 No. 4, pp. 787–807.

Recker, J., Holten, R., Hummel, M. and Rosenkranz, C. (2017), "How Agile Practices Impact Customer Responsiveness and Development Success: A Field Study", *Project Management Journal*, Vol. 48 No. 2, pp. 99–121.

Reichertz, J. (2007), "Abduction: The Logic of Discovery of Grounded Theory", in A., B. and Charmaz, K.C. (Eds.), *The SAGE Handbook of Grounded Theory*, SAGE, London, pp. 214–228.

Reynolds, P. and Yetton, P. (2015), "Aligning business and IT strategies in multi-business organizations", *Journal of Information Technology*, Vol. 30 No. 2, pp. 101–118.

Rindova, V.P. and Kotha, S. (2001), “Continuous ‘Morphing’: Competing through Dynamic Capabilities, Form, and Function”, *Academy of Management Journal*, Vol. 44 No. 6, pp. 1263–1280.

Saldaña, J. (2016), *The Coding Manual for Qualitative Researchers.*, SAGE Publications, London.

Sandberg, J. (2014), *Digital Capability: Investigating Coevolution of IT and Business Strategies*, Department of informatics, Umeå university.

Sandberg, J. and Alvesson, M. (2011), “Ways of constructing research questions: Gap-spotting or problematization?”, *Organization*, Vol. 18 No. 1, pp. 23–44.

Sandberg, J., Holmström, J. and Lyytinen, K. (2020), “Digitization and phase transitions in platform organizing logics: Evidence from the process automation industry”, *MIS Quarterly: Management Information Systems*, Vol. 44 No. 1, pp. 129–153.

Sarker, S., Chatterjee, S., Xiao, X. and Elbanna, A. (2019), “The sociotechnical axis of cohesion for the IS discipline: Its historical legacy and its continued relevance”, *MIS Quarterly: Management Information Systems*, Vol. 43 No. 3, pp. 695–719.

Sarker, S., Xiao, X. and Beaulieu, T. (2013), “Qualitative Studies in Information Systems: A Critical Review and Some Guiding Principles”, *MIS Quarterly*, Vol. 37 No. 4.

Sarker, S., Xiao, X., Beaulieu, T. and Lee, A.S. (2018), “Learning from first-generation qualitative approaches in the is discipline: An evolutionary view and some implications for authors and evaluators (part 1/2)”, *Journal of the Association of Information Systems*, Vol. 19 No. 8, pp. 752–774.

El Sawy, O.A. (2003), “The IS Core IX: The 3 Faces of IS Identity: Connection, Immersion, and Fusion”, *Communications of the Association for Information Systems*, Vol. 12 No. November.

El Sawy, O.A., Malhotra, A., Park, Y.K. and Pavlou, P.A. (2010), “Seeking the configurations of digital ecodynamics: It takes three to tango”, *Information Systems Research*, Vol. 21 No. 4, pp. 835–848.

Sayer, A. (2000), *Realism and Social Science*, Sage, London, UK.

Schwaber, K. and Beedle, M. (2002), *Agile Software Development with Scrum*, Prentice Hall, Upper Saddle River.

Sharp, J. and Babb, J. (2018), “Is Information Systems Late to the Party? The Current State of DevOps Research in the Association for Information Systems eLibrary”, *Twenty-Fourth Americas Conference on Information Systems*, pp. 1–8.

Sia, S.K., Soh, C. and Weill, P. (2016), “How DBS Bank Pursued a Digital Business Strategy”, *MIS Quarterly Executive*, Vol. 15 No. 2, pp. 105–121.

Sia, S.K., Weill, P. and Zhang, N. (2021), “Designing a Future-Ready Enterprise: The Digital Transformation of DBS Bank”, *California Management Review*, Vol. 63 No. 3, pp. 35–57.

Siau, K., Woo, C., Storey, V.C., Chiang, R.H.L., Chua, C.E.H. and Beard, J.W. (2022), “Information Systems Analysis and Design: Past Revolutions, Present Challenges, and Future Research Directions”, *Communications of the Association for Information Systems*, Vol. 50 No. 1, pp. 835–856.

Siau, K., Yoanna Long and Ling, M. (2010), “Toward a Unified Model of Information Systems Development Success”, *Journal of Database Management*, Vol. 21 No. 1, pp. 80–101.

Strauss, A. and Corbin, J. (2015), *Basics of Qualitative Research*, SAGE Publications, London.

Tallon, P. (2007), “A process-oriented perspective on the alignment of information technology and business strategy”, *Journal of Management Information Systems*, Vol. 24 No. 3, pp. 227–268.

Tallon, P. and Pinsonneault, A. (2011), “Competing Perspectives on the Link Between Strategic Information Technology Alignment and Organizational Agility: Insights From a Mediation Model”, *MIS Quarterly*, Vol. 35 No. 2, pp. 463–486.

Tarafdar, M. and Gordon, S.R. (2007), “Understanding the influence of information systems competencies on process innovation: A resource-based view”, *Journal of Strategic Information Systems*, Vol. 16 No. 4, pp. 353–392.

Tilson, D., Lyytinen, K. and Sørensen, C. (2010), “Digital infrastructures: The missing IS research agenda”, *Information Systems Research*, Vol. 21 No. 4, pp. 748–759.

Todnem By, R. (2005), “Organisational change management: A critical review”, *Journal of Change Management*, Vol. 5 No. 4, pp. 369–380.

Urbach, N. and Ahlemann, F. (2019), *IT Management in the Digital Age*, Spinger.

Urbach, N., Smolnik, S. and Riempp, G. (2009), “The State of Research on Information Systems Success”, *Business & Information Systems Engineering*, Vol. 1 No. 4, pp. 315–325.

Urquhart, C. (2013), *Grounded Theory for Qualitative Research*, SAGE Publications, London.

Van de Ven, A.H., Angle, H.L. and Poole, M.S. (Eds.). (2000), *Research on the Management of Innovation*, Oxford University Press, New York.

Ven, A.H. Van de and Drazin, R. (1984), “The Concept of Fit in Contingency Theory”, *The Strategic Management Research Center*.

Van de Ven, A.H., Polley, D.E., Garud, R. and Venkataraman, S. (2008), *The Innovation Journey*, Oxford University Press, New York.

Van de Ven, A.H. and Poole, M.S. (1990), “Methods for Studying Innovation Development in the Minnesota Innovation Research Program”, *Organization Science*, Vol. 1 No. 3 (Special Issue), pp. 313–335.

VersionOne. (2018), “VersionOne 12th Annual State of Agile Report”, available at: https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report (accessed 9 April 2019).

Volberda, H.W., van der Weerdt, N., Verwaal, E., Stienstra, M. and Verdu, A.J. (2012), “Contingency fit, institutional fit, and firm performance: A metafit approach to organization-environment relationships”, *Organization Science*, Vol. 23 No. 4, pp. 1040–1054.

Volkoff, O. and Strong, D.M. (2013), “Critical Realism and Affordances: Theorizing IT-Associated Organizational Change Processes”, *MIS Quarterly: Management Information Systems*, Vol. 37 No. 3, pp. 819–834.

Wang, G., Henfridsson, O., Nandhakumar, J. and Yoo, Y. (2022), “Product Meaning in Digital Product Innovation”, *MIS Quarterly*, Vol. 46 No. 2, pp. 947–976.

Wang, X., Conboy, K. and Cawley, O. (2012), “‘Leagile’ software development: An experience report analysis of the application of lean approaches in agile software development”, *Journal of Systems and Software*, Elsevier Inc., Vol. 85 No. 6, pp. 1287–1299.

Webster, J. and Watson, R.T. (2002), “Analyzing the Past to Prepare for the Future: Writing a Literature Review”, *MIS Quarterly*, Vol. 26 No. 2, pp. xiii–xxiii.

Weick, E. (1989), “Theory Construction as Disciplined Imagination”, *Academy of Management Review*, Vol. 14 No. 4, pp. 516–531.

Weick, K.E. and Quinn, R.E. (1999), “Organizational Change and Learning”, *Annual Review of Psychology*, Vol. 50.

Werder, K. and Heckmann, C.S. (2019), “Ambidexterity in Information Systems Research: Overview of Conceptualizations, Antecedents, and Outcomes”, *Journal of Information Technology Theory and Application*, Vol. 20 No. 1, pp. 23–52.

Wetherbe, J.C. and Frolick, M.N. (2000), “Cycle Time Reduction: Concepts and Case Studies”, *Communications of the Association for Information Systems*, Vol. 3.

Whetten, D.A. (1989), “What Constutes a Theoretical Contribution”, *Academy of Management Review*, Vol. 14 No. 4, pp. 490–495.

Wiedemann, A. (2017), “A New Form of Collaboration in IT Teams - Exploring the DevOps Phenomenon”, *21st Pacific Asia Conference on Information Systems, {PACIS} 2017, Langkawi, Malaysia, July 16-20, 2017*, p. 82.

Wiedemann, A., Forsgren, N. and Wiesche, M. (2019), “Research for Practice : The DevOps Phenomenon”, *Communications of the ACM*, Vol. 62 No. 8, pp. 44–49.

Wiedemann, A. and Wiesche, M. (2018), “Are you ready for DevOps? Required skill set for DevOps teams”, *Twenty-Sixth European Conference on Information Systems (ECIS2018)*.

Wiedemann, A., Wiesche, M., Gewald, H. and Krcmar, H. (2020), “Understanding how DevOps aligns development and operations: a tripartite model of intra-IT alignment”, *European Journal of Information Systems*, Taylor & Francis, No. 00, pp. 1–16.

Wiedemann, A., Wiesche, M., Thatcher, J.B. and Gewald, H. (2019), “A Control-Alignment Model for Product Orientation in DevOps Teams–A Multinational Case Study”, *Fortieth International Conference on Information Systems (ICIS2019)*, pp. 1–17.

Williams, C.K. and Karahanna, E. (2013), “Causal Explanation in the Coordinating Process: A Critical Realist

Case Study of Federated IT Governance Structures", *MIS Quarterly: Management Information Systems*, Vol. 5 No. 3123, pp. 548–554.

Woodard, C.J., Ramasubbu, N., Tschang, F.T. and Sambamurthy, V. (2013), "Design Capital and Design Moves: The Logic of Digital Business Strategy", *MIS Quarterly: Management Information Systems*, Vol. 37 No. 2, pp. 537–564.

Wynn, D. and Williams, C.K. (2012), "Principles for Conducting Critical Realist Case Study Research in Information Systems", *MIS Quarterly*, Vol. 36 No. 3, pp. 787–810.

Wynn, D. and Williams, C.K. (2020), "Recent advances and opportunities for improving critical realism-based case study research in IS", *Journal of the Association for Information Systems*, Vol. 21 No. 1, pp. 50–89.

Xu, P. and Ramesh, B. (2007), "Software process tailoring: An empirical investigation", *Journal of Management Information Systems*, Vol. 24 No. 2, pp. 293–328.

Xue, L., Ray, G. and Sambamurthy, V. (2012), *Efficiency or Innovation: How Do Industry Environments Moderate the Effects of Firms' IT Asset Portfolios?*, *MIS Quarterly: Management Information Systems*, Vol. 36.

Yin, R.K. (2018), *Case Study Research and Applications*, SAGE Publications, London.

Yoo, Y. (2013), "The tables have turned: How can the information systems field contribute to technology and innovation management research?", *Journal of the Association for Information Systems*, Vol. 14 No. 5, pp. 227–236.

Yoo, Y., Henfridsson, O. and Lyytinen, K. (2010), "The new organizing logic of digital innovation: An agenda for information systems research", *Information Systems Research*, Vol. 21 No. 4, pp. 724–735.

Zachariadis, M., Scott, S. and Barrett, M. (2013), "Methodological Implications of Critical Realism For Mixed-Methods Research", *MISQ*, Vol. 37 No. 3, pp. 855–879.

Affidativ
according to § 8 paragraph 3 of the doctoral degree regulations of
February 17, 2015

I hereby affirm in lieu of oath that I have prepared the submitted work independently and without using any other than the specified aids. Statements, data and concepts taken directly or indirectly from other sources are identified with reference to the source. In the selection and evaluation of the following material, the persons listed below helped me in the manner described, for a fee / free of charge (underline as appropriate):

- Earlier/different versions of literature reviews were conducted by student assistants: Philipp Scheider, Dennis Keles, Thomas Rohrsdorfer, Maike Walzel, Felix Urban, Simon Plum, Niklas Bartelt, Daryna Enns, Diana Geibel, Imke Schwenke.
- Transcription of a part of the conducted interviews: abtipper.de.

Other people, in addition to the co-authors listed in the introduction to the work, were not involved in the content-related preparation of the present work. In particular, I did not make use of the paid help from mediation or advisory services. Nobody has received direct or indirect monetary benefits from me for work that is related to the content of the submitted dissertation.

The thesis has not yet been submitted to another examination authority in the same or a similar form, either in Germany or abroad.

I assure you that to the best of my knowledge, I have told the pure truth and have not concealed anything.

I confirm that the submitted electronic version corresponds in full to the submitted printed version.

I am aware of the criminal liability of a false affidavit, namely the threat of punishment according to § 156 StGB with an imprisonment of up to three years or with a fine in the case of intentional commission of the act or according to § 161 Paragraph 1 StGB up to one year imprisonment or fine in the case of negligent commission.

Köln, 04-12-2023

Place, Date Signature

Michael Hüttermann
-Curriculum Vitae-

Herderstr. 10, 51147 Köln, Germany
michael@huettermann.net
http://huettermann.net
+49 (0) 173 5373629

Born 06-08-1972, in Bottrop, Germany

Professional Experience

06/1991-present	Different roles in IT. Since 2006 supporting enterprises as freelancing DevOps consultant.

Milestones

1982-1988	Gerhart-Hauptmann-Realschule, Dorsten
1988-1991	Gymnasium Petrinum, Dorsten
1991-1994	Apprenticeship “Management assistant in data processing”, 3M Germany, Borken
06/1994-09/1994	IT expert, 3M Germany, Borken
10/1994-09/1995	Bundeswehr Wehrdienst, FmBtl. Coesfeld
10/1995-12/2000	Study of Wirtschaftsinformatik, University Paderborn; Dipl.-Wirt.Inf. 12-19-2000 (Master’s degree in business informatics) “Konzeption und prototypische Realisierung eines XML-basierten Repositories für eine konstruktionsbegleitende Kalkulation eines Automobilzulieferers”
02/2001-03/2003	IT expert, Lycos Europe, Gütersloh
04/2003-06/2006	IT expert, REWE-Informations-Systeme, Köln
06/2006-present	Freelance consultant supporting organizations implementing DevOps
10/2018-2023	External PhD student, at Cologne Institute for Information Systems (CIIS), University of Cologne. Topic: “DevOps”

Books

DevOps for Developers, Apress, 2012, ISBN13 978-1430245698
Agile ALM, Manning, 2011, ISBN13 978-1935182634
Fragile Agile, Hanser, 2010, ISBN13 978-3446422582
Agile Java-Entwicklung in der Praxis, O’Reilly, 2008, ISBN13 978-3897214828

Publications

Google Scholar https://scholar.google.de/citations?user=iNvFt2iIHHAJ
ResearchGate https://www.researchgate.net/profile/Michael-Huettermann