Association for Information Systems

AIS Electronic Library (AISeL)

ICIS 2019 Proceedings

IS Development and Implementation

Coexisting Plan-driven and Agile Methods: How Tensions Emerge and Are Resolved

Isabel Laux
Ludwig-Maximilians University, laux@bwl.lmu.de

Coexisting Plan-driven and Agile Methods: How Tensions Emerge and Are Resolved

Completed Research Paper

Isabel Laux
Ludwig-Maximilians-Universität München
Ludwigstrasse 28,
80539 Munich, Germany
laux@bwl.lmu.de

Johann Kranz
Ludwig-Maximilians-Universität München
Ludwigstrasse 28,
80539 Munich, Germany
kranz@lmu.de

Abstract

Fast changing products, processes, and services caused by digital technologies require organizations to adopt agile methods after having used plan-driven approaches for decades. Adopting agile methods only to software development, can lead to a challenging coexistence of methods. To date, little empirical understanding exists with regard to the difficulties that emerge when organizations introduce agile teams in plan-driven environments. Consequently, we investigate the coexistence of agile and plan-driven methods and study its impacts. We conducted an exploratory multiple case study of four organizations and draw from adaptive structuration theory to study how agile methods are adopted on team level to an environment of deeply entrenched plan-driven methods. We find that this coexistence causes several tensions between agile and plan-driven teams (i.e., budgeting, knowledge, planning, process, responsibility, and cultural tension). Further, we reveal how organizations and teams overcome these tensions with balanced and blended resolutions.

Keywords: agile method, software development, coexistence, tension, resolution

Introduction

Numerous examples in industries as diverse as media, financial services, or retail show how fast digital technologies have changed products, processes, and services. As digital technologies can often be quickly copied, borrowed, or substituted by competitors, competitive advantage is short lived. Thus, while in the industrial age, first-movers could exploit advantages gained by being the first to introduce an innovation, in digital business, fast-movers outpace competitors by learning and reacting fast (Venkatraman 2017). Driven by these rapid, frequent, and discontinuous changes, evermore companies adopt agile software development methods to stay in sync with rapidly changing markets (Rigby et al. 2018). In the past, stable environments permitted the rigidities of plan-driven methods, like heavy upfront project planning (e.g., requirements engineering and specification) and thus, a long implementation period (Bick et al. 2017; Hekkala et al. 2017). Today, accelerated product development abilities and high operating flexibility are the predominant competitive advantages to persist in an ever-changing environment (Conboy 2009). Thus, today we can observe an increasing number of businesses using plan-driven methods across industries adopt agile methods (Hekkala et al. 2017; Hobbs and Petit 2017).

Agile methods such as Scrum or Kanban (Anderson 2010; Schwaber and Beedle 2002) involve various characteristics and practices (e.g., iterative cycles or higher customer involvement) that help increase time-to-market and create products right to customer needs (Hekkala et al. 2017). However, by adopting agile methods to an environment in which plan-driven methods have been used, hybrid approaches emerge that combine both methods as an immediate overhaul of established practices is seldom feasible (Bick et al. 2017). Empirical evidence emphasizes that hybrid approaches either integrate agile and traditional plan-driven methods advantageously (Cooper and Sommer 2016) or lead to a challenging coexistence of both methods (Hekkala et al. 2017; Vinekar et al. 2006). Research suggests that retrieving a balanced control portfolio (e.g. creating an effective set of various types of control) is crucial as the adoption evokes changes in control types (Mahadevan et al. 2015). Another major challenge is a lack of experience with agile methods (Hekkala et al. 2017). Hobbs and Petit (2017) highlight that significant adaptations of established methods are required on project and organizational level. To date, research has paid little attention to the difficulties that emerge on team level when teams that have used plan-driven approaches for a long time adopt agile methods leading to a coexistence of agile and plan-driven methods. This is surprising as both methods build upon fundamentally opposing characteristics (e.g., process-centric vs. people-centric) and roles (e.g., project manager vs. scrum master) (Mahadevan et al. 2015). Thus, adopting agile methods involves major changes regarding deeply entrenched routines that occur due to the simultaneous use of plan-driven and agile methods in collaborating teams. Consequently, we believe that it is important to augment our understanding of the unique challenges arising from teams using coexisting methods and examine which problems arise and how they are addresses.

In this study, we analyze the challenges arising from the collaboration of teams using coexisting plan-driven and agile methods. Considering the team level enables us to investigate the individual teams' processes profoundly and understand how the use of either plan-driven or agile methods challenges the workflow as they collaborate. By studying the collaboration on the team level, we seek to unearth emerging problems and solutions between teams. Further, by analyzing the coexistence of agile and plan-driven methods, we investigate the process (i.e. social interaction) of the adoption of agile methods to plan-driven environments and study its impacts. We investigate the adoption of agile methods only to SD where the methods originate from. However, all other teams continue working with plan-driven methods and hence the agile mindset only slowly arrives at other teams. We draw from adaptive structuration theory (AST) (DeSanctis and Poole 1994) to study how agile methods are adopted on team level to an environment of deeply entrenched plan-driven methods (i.e. sources of structure). We examine the appropriation of both structures to reveal how it affects the social interaction between agile teams and teams following a plan-driven model (Cao et al. 2009). We use the Grounded Theory Method (GTM) as we seek to generate new insights about coexisting methods and build on existing theory while analyzing empirical data (Glaser and Strauss 1967).

To study the phenomenon in depth, we conducted an exploratory multiple case study of four organizations that adopted agile methods after having used plan-driven methods for decades. We found that the coexistence of plan-driven and agile methods in companies causes several tensions (i.e., budgeting, knowledge, planning, process, responsibility, and cultural tension) between agile and plan-driven teams. Further, we reveal how organizations and teams overcome these tensions with balanced and blended resolution approaches. Our study makes two contributions to literature on agile methods. First, we unearth tensions that emerge from the collaboration of plan-driven and agile teams. We show that to resolve these tensions, organizations either accelerate (blended resolution approaches) or inhibit (balanced resolution approaches) the adoption of agile methods. Second, we show that the emerging team-level tensions lead to the introduction of more integrated, cross-functional approaches such as DevOps. Our findings suggest that the adoption of agile methods in software development (SD), after a prior use of plan-driven methods, is a multi-layered phenomenon that requires fundamental rethinking of characteristics, practices, and roles.

The remainder of this paper is structured as follows. In the next section, we present the theoretical background that continuously informed our research during data collection and analysis through constant comparison between the literature and our data. Following that, we explain our research method and elaborate on our findings per case. After an analysis and discussion of our cross-case results, the paper concludes by highlighting our contribution as well as future research ideas.

Theoretical Background

Hybrid Methods vs. Coexisting Methods

Hybrid software development approaches encompass elements of plan-driven methods (e.g., waterfall or V-model) and agile methods (e.g., Scrum or Kanban) (Cooper and Sommer 2016). A hybrid approach can combine elements of two or more different SD methods such as different *characteristics* (e.g., bureaucratic processes with high formalization of plan-driven methods combined with cooperative social action between individuals of agile methods), *practices* (e.g., extensive design phase of plan-driven methods combined with short increments of agile methods), or *roles* (e.g., plan-driven project managers combined with agile scrum masters and product owners) (Bick et al. 2017). By intentionally selecting both agile and plan-driven elements to combine the methods' beneficial practices, organizations shape a specific hybrid approach along the continuum between both methods as illustrated in Figure 1 (Bick et al. 2017; Nerur et al. 2005).

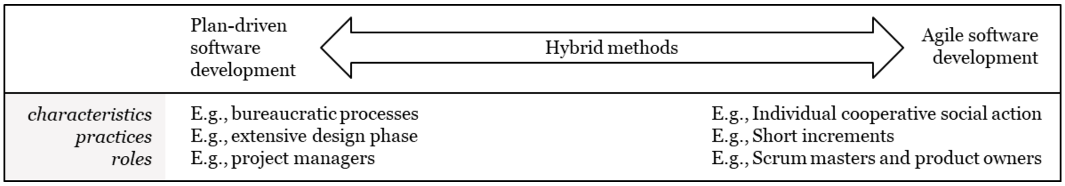
Plan-driven software development
Hybrid methods
Agile software development
characteristics
practices
roles
E.g., bureaucratic processes
E.g., extensive design phase
E.g., project managers
E.g., Individual cooperative social action
E.g., Short increments
E.g., Scrum masters and product owners

Figure 1. Continuum of Plan-driven and Agile Software Development Methods (Bick et al. 2017; Nerur et al. 2005)

The combination of agile and plan-driven elements leads to a coexistence of both methods (Vinekar et al. 2006) which can complement each other (Cooper and Sommer 2016). However, the coexistence can also cause tensions between the diverse or even opposing elements of plan-driven and agile methods (Hekkala et al. 2017; Lewis 2000). Tensions are defined as "elements that seem logical individually, but inconsistent, even absurd when juxtaposed" (Smith and Lewis 2011, p. 382). In general, tensions arise as a result of two contrasting yet contradictory management styles (Lewis 2000). Especially the team level, opposing beliefs of agile and plan-driven methods collide when teams collaborate with each other (Moe et al. 2010). For instance, agile methods encourage teams to self-organize their work, whereas teams using plan-driven approaches are used to following orders (Hoda and Murugesan 2016). By studying tensions between opposing elements of plan-driven and agile methods, we focus our consideration on a coexistence of methods (Vinekar et al. 2006) rather than on beneficial combinations (i.e. hybrid approaches) (Bick et al. 2017).

Prior literature studied the coexistence of methods by considering the change in control when introducing agile methods and observing a hybrid-like control (Mahadevan et al. 2015). Other research examined the acceptance of hybrid methods (Schlauderer et al. 2015) or the tailoring criteria that influence the adoption of an agile method (Campanelli et al. 2018). We look beyond the application of the methods individually and examine the circumstances of their coexistence in general. Thus, we focus on the collaborations of multiple teams that use different plan-driven and agile methods in coexistence.

Adaptive Structuration Theory

Adaptive structuration theory (AST) proposes an approach for the study of social aspects of group processes. Social aspects determine technology used by groups and social structures influenced by group choices (Chin et al. 1997). AST examines the mutual influence of technology and social processes to analyze how groups adapt to structures and in turn how technologies influence groups (DeSanctis and Poole 1994). Originating from structuration theory (Giddens 1987), AST examines the interplay of *structuration*, the "process of bringing rules, resources, and other structures into action" and *appropriation*, defined as the application of structures in a distinct context (Cao et al. 2009, p. 334). Prior research on agile methods has applied AST in various ways to understand the role of social structures in SD. Cao et al. (2009) studied the adaptation of agile methods on project and organizational level. Extending their research Cao et al. (2013) identified the traditional IT funding process as a difficulty on project level when adopting agile methods. Fuchs (2019) studied the adaptation of agile methods by organizational features caused by their interplay and vice versa. Primarily, AST has been used for team level research, for instance to analyze virtual teams (e.g. Charlier et al. 2016). In our study, we focus again on the adoption of agile methods on team level. Thus, we consider team collaboration and study the effects of a coexistence of plan-driven and agile methods.

We use AST to examine team structures and roles of teams using coexisting methods. The coexistence of plan-driven and agile methods arises from an organization's adoption agile methods. By drawing from AST, we seek to understand how groups organize themselves and how different forms of social interaction are encouraged. We aim at generating new insights about collaboration in an environment of coexisting plan-driven and agile methods (DeSanctis and Poole 1994). Figure 2 depicts the framework based on DeSanctis and Poole (1994).

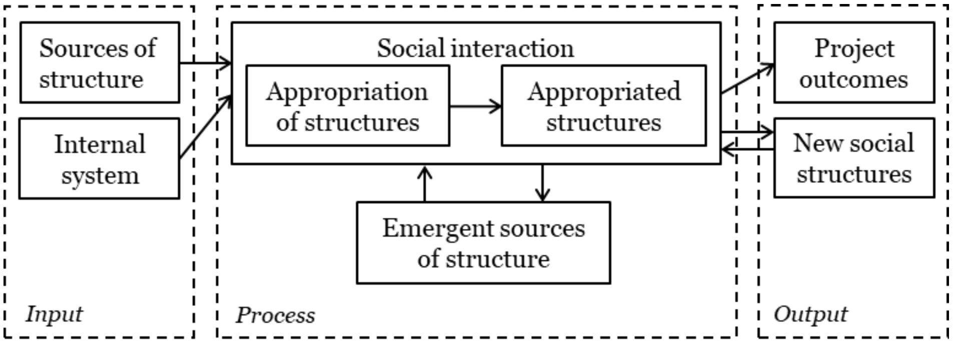

Figure 2. Adaptive Structuration Theory Model (based on DeSanctis and Poole 1994)

The AST model consists of three elements: input, process and output. The input phase comprises the sources of structure and the team's internal system. The sources of structure of our model are plan-driven and agile methods. The *structure of plan-driven methods* is characterized by an inflexible, sequential process (structural features) that "produces solutions to problems that [may] have already changed"(Austin and Devin 2009, p. 463). Plan-driven projects require formal hierarchies, clearly documented processes (*plan-driven spirit*), and maintain planning discipline to avoid risks (Nerur et al. 2005). The *structure of agile methods* comprises values and inherent rules and capabilities. In contrast to plan-driven methods, they emphasize flexibility and autonomy. To encourage autonomy, authority is delegated to the team by fostering self-organization (Moe et al. 2010; Nerur et al. 2005). Agile methods are characterized by an increased responsiveness to changing requirements, a closer interaction with customers, and an iterative development approach (*agile spirit*) (Anderson 2010; Schwaber and Beedle 2002). The *team's internal system* describes the interaction patterns on team-level. For instance, the team members' experience of working together or with agile methods or their individual style of leadership (DeSanctis and Poole 1994).

The process phase describes the social interaction between teams with different sources of structure (e.g., plan-driven or agile methods) that generate *emergent sources of structure* and affect the *social interaction* in turn. Social interaction describes the *appropriation of structures* that lead to *appropriated structures*. Appropriation refers to the implementation of applied social structures in a specific context (DeSanctis and Poole, 1994). AST suggests four characteristics of appropriation: *appropriation moves, faithfulness of appropriation, instrumental uses*, and *attitude* toward appropriation. Appropriation moves describe how teams adopt structures (e.g., how the agile method is used within the team). Faithfulness describes the consistency of the structure with its spirit (e.g., the extent to which the agile method is consistent with common guidelines). Instrumental use specifies the purpose of appropriation (DeSanctis and Poole 1994). Finally, attitude toward appropriation refers to the "perceptions and feelings of users toward the structure" (Cao et al. 2013, p. 193). The adaptation leads to *appropriated structures* that are being promoted by social interactions (e.g., the decision to tailor methods to the agile teams' needs). *Emergent sources of structure* arise from applying and rebuilding structures. For instance, resemblance against agile methods impedes social interaction processes (e.g., teams refuse to adapt to agile methods).

The third aspect of the model describes the output of the social interaction process. Social interaction results in *project outcomes*, related to schedule, budget or scope. By rebuilding existing structures, *new social structures* emerge that may incorporate sources of structure or the internal system (e.g., introduce new leadership styles) and influence social interaction (DeSanctis and Poole, 1994).

Study Design

Case Selection

We use an exploratory, multiple-case study approach to analyze the coexistence of plan-driven and agile methods emerging in collaboration between agile and non-agile teams. A multiple-case design allows us to conduct a cross-case pattern search and is suited to answer 'how' and 'why' questions by investigating a phenomenon in its real life context (Yin 2009). Based on Patton (1990) we selected our cases purposefully and information-rich, meaning that the cases yield to an in-depth understanding of our research area. Key selection criteria were that case companies had adopted agile methods within the last three years and had gained first experience with agile methods. Another important criterion was that only SD teams use agile methods while other teams use plan-driven methods.

The four companies acquired in the sample are located in the same geographical region. However, the firms differ in size, industry, and experience with agile methods. Project teams of our case companies use Scrum or Kanban as agile SD methods and the waterfall model as plan-driven method (see Table 1). Among the projects we analyzed, around 50% to 70% of involved team members used agile methods. We used theoretical sampling up to the point where theoretical saturation was reached to reduce sampling bias and to increase data coverage (Glaser and Strauss 1967).

Case	**FinanceCo**	**CommerceCo**	**E-commerceCo**	**MediaCo**
Founded	1890	1995	1999	1992
Employees	140,000	800	1,000	500
Agile SD method	Scrum	Scrum	Kanban and Scrum	Kanban
Plan-driven method	Waterfall	Waterfall	Waterfall	Waterfall
Number of teams	5 Scrum teams 2 IT operations teams 1 business team	3 Scrum teams 2 IT operations teams 2 business teams	4 Kanban teams 3 Scrum teams 1 business team	2 Kanban teams 1 business team
Interviewee position (Years of experience with agile methods)	#1 PO (<1) #2 SM (>3) #3 IT operations (<1) #4 Developer (>3) #5 CPO (>2)	#1 PO (>1) #2 IT operations (>1) #3 Developer (>1) #4 SM (>5) #5 Manager (>1)	#1 QA tester (>5) #2 Manager. (>2) #3 SRM (<1) #4 Developer (>1) #5 SDM (>2)	#1 SDM (>2) #2 Developer (>1) #3 Agile coach (>2) #4 Manager (<1) #5 IT Manager (>2) #6 SDM (<3)
Note. PO = product owner, SM = scrum master, CPO = chief product owner, QA = quality assurance, SRM = service request manager, SDM = service delivery manager				

Table 1. Case Descriptions

Data Collection and Analysis

We conducted semi-structured on-site face-to-face interviews with 21 interviewees that had detailed knowledge about the introduction and application of agile methods in SD. Based on the Grounded Theory Method (GTM), theoretical sampling guided us to select interview partners based on the analysis of previously conducted interviews (Glaser and Strauss 1967). The interviews lasted between 45 and 70 minutes. The questions were open-ended, allowing respondents to freely convey their experiences, views, and expressions of the socially complex contexts that involve agile method use (Yin 2009). All interviews were recorded, anonymized, and transcribed. The data was encoded using the software NVivo 11. We followed an interview guideline that was refined in the course of data collection. We interviewed members of at least two teams per organization. We started by asking rather general questions about interviewees' backgrounds and experience with plan-driven and agile methods. To avoid researcher bias, we encouraged the respondents to provide vignettes about personal and team-level experience of working with agile methods. Next, we concentrated our questions on the collaboration between teams. We soon realized that tensions could arise between teams of different methods. Thus, the arising tensions and respective resolutions became the focus of our interviews. With an increasing understanding of team collaboration and the resulting tensions, we chose to consult prior literature and began to take relevant concepts into account (Glaser and Strauss 1967). As GTM suggests a continuous interplay between data collection and analysis, we used the identified concepts and theories that informed our GTM data collection and analysis to link our results to the existing body of knowledge (Urquhart et al. 2010). Beyond interviews, our study includes internal (e.g., agile boards and internal communication) and external (e.g., annual reports) sources. The use of multiple data sources, data triangulation, allowed to increase internal validity and mitigate potential biases (Yin 2009).

For the data analysis, we utilize informed GTM to perform three rounds of coding (Corbin and Strauss 1990; Glaser and Strauss 1967). We chose an inductive research approach to study the coexistence of plan-driven and agile methods in collaborating teams. We used the Straussian coding approach (Corbin and Strauss 1990) and started with open coding to identify initial categories. In a second step, we applied axial coding to develop a deeper knowledge and relate categories to each other. By using selective coding, we identified a core category and related other categories. Throughout the process, we compared our data constantly to ensure rigorous coding and continuously consulted existing literature and theories to confirm our findings or to reveal possible conflicting explanations (Glaser and Strauss 1967). Consequently, our three-stage data collection and analysis process is provided.

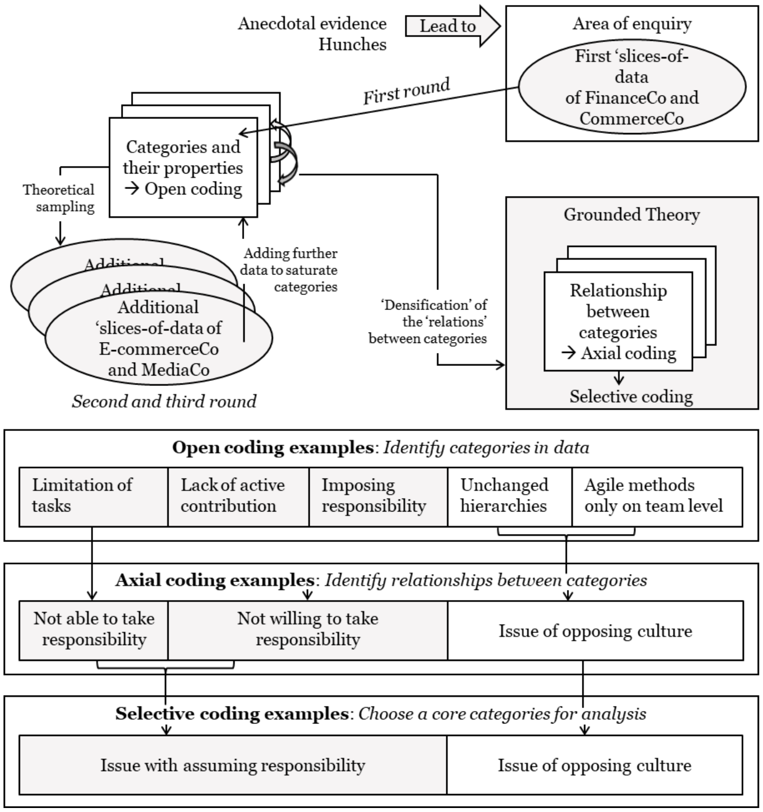
Anecdotal evidence
Hunches
Lead to
Area of enquiry
First 'slices-of-data of FinanceCo and CommerceCo
First round
Categories and their properties → Open coding
Theoretical sampling
Adding further data to saturate categories
Additional
Additional
Additional 'slices-of-data of E-commerceCo and MediaCo
Second and third round
'Densification' of the 'relations' between categories
Grounded Theory
Relationship between categories → Axial coding
Selective coding
Open coding examples: Identify categories in data
Limitation of tasks
Lack of active contribution
Imposing responsibility
Unchanged hierarchies
Agile methods only on team level
Axial coding examples: Identify relationships between categories
Not able to take responsibility
Not willing to take responsibility
Issue of opposing culture
Selective coding examples: Choose a core categories for analysis
Issue with assuming responsibility
Issue of opposing culture

Figure 3. Cycle of Data Collection and Analysis (Urquhart et al. 2010)

The *first stage* started with a small set of interviews at FinanceCo and CommerceCo. We sampled the organizations as they had recently adopted agile methods (one to two years prior to interviews). The companies allowed for an in-depth investigation of teams involved in the development process as key activities and interactions were established and reproducible. The interviews were open-ended and addressed the current state of the adoption. Our initial open coding revealed adaptations on team level resulting from the adoption of plan-driven to agile methods.

In the *second stage*, our initial understanding from the first round of interviews was grounded theoretically on previous literature. We used AST to further analyze and make sense of the data. By applying the theory, we discovered tensions emerging from the adoption of agile methods. AST was used as an explanation for our findings (Gregor 2006) but we remained open and allowed for further concepts to emerge throughout the process. We included E-commerceCo into the study, which had used Scrum and Kanban for three years, to enrich out data and to analyze a later stage of the adoption process. Although we still focused on comprehensive interviews, we revised our interview guideline to include specific questions about adaptations of methods, team structures, and emerging tensions. After comparing incidents with cross-organizational data to identify differences in the events, activities, and processes (Glaser and Strauss, 1967) we found similar emerging tensions and resolution strategies. At this stage, we learned that AST fits our conducted data.

In the *last round*, we extended our sample to include interviewees in management positions to achieve a clearer understanding of the collaboration between agile and plan-driven teams. We integrated a fourth company (MediaCo), which uses Kanban, into our sample to rule out method-related tensions and to generalize our findings. Figure 3 summarized our data collection and analysis process based on Urquhart et al. (2010) and indicates exemplary codes to illustrate the coding process.

Results

Within-Case Analysis

FinanceCo is an internationally operating finance and insurance company that *"introduced Scrum step-by-step to increase process transparency for all stakeholders [e.g., customers, management etc.]"* (chief product owner (CPO)). Before adopting agile methods, all teams involved in the development process (SD, business and IT operations) used a waterfall approach without any feedback loops. The organizational structure remained unchanged by the adoption of agile methods, meaning that the teams managing the company's customer platforms remained separate into business and IT teams which included SD and IT operations teams. The first agile team consisted of a product owner (PO), a scrum master (SM) and four developers. In terms of disciplinary responsibility, the IT team members (i.e. developers or the SM) reported to the leader of the SD team whereas the PO reports to the business team lead. In terms of functional responsibility, however, the agile team reports to the business team lead as the agile team is part of the business function. Starting with one agile team, FinanceCo scaled Scrum to five SD teams within two years. To coordinate all agile teams, a *"project manager holds a meeting of all scrum masters to control their teams' progress"* (SM). A CPO coordinates the POs and reports to management. After using Scrum for three years, FinanceCo combined the IT and business pillars into one and started using a DevOps approach.

CommerceCo is a teleshopping company operating in the EMEA region with a focus on selling physical goods. To keep up with the speed of their born digital competitors and to increase sales on their online platform, CommerceCo introduced Scrum to SD. Another reason to adopt agile methods was that, "*the job market [that] offers mainly developers with agile skill profiles that are unwilling to use the waterfall method to develop on the existing platform*" (manager). The company's structure consists of three departments with four SD teams. The adoption of agile methods began with one SD team which was extended to all four teams in one year. The two IT operations teams continued using the waterfall method. The agile teams consisted of a PO, a SM and three to five developers. All agile POs are part of the business team that is organized in a project-driven way using the waterfall model. To ensure a continuous customer experience, process managers support the agile teams on a technical level. A CPO coordinates all Scrum teams and acts as boundary spanner between departments and the management. Communities of practice meetings offer developers the possibility to exchange ideas on domain level (e.g., frontend design). By considering integrating SD and IT operations teams (i.e. DevOps teams), CommerceCo intends to avoid handover problems.

E-commerceCo operates a national e-commerce platform structured in business units according to six product domains. *"Each product domain engages an agile team using either Scrum or Kanban"* (PO). After having used the waterfall model for years, one team adopted Scrum and within three years, all remaining teams followed. Initially all agile teams used Scrum, but over time, four teams adopted Kanban because it seemed more appropriate to ensure continuous delivery outside of a sprint framework. As a result, four teams used Kanban and three teams applied Scrum. A Scrum or Kanban team consists of a product representative (PO or service request manager (SRM)) and four to seven developers. An SM joins the Scrum teams only upon request whereas Kanban teams do not have a service delivery manager (SDM). To comply with a platform approach, the holding company combines all products developed by subsidiary business units into an application portfolio. The platform approach minimizes necessary exchange between developers. POs meet on a weekly basis with the respective domain lead. Agile teams within a domain report to a team lead independent of their tasks and a technical lead supervises all developers. E-commerceCo follows a DevOps approach, where "*teams are responsible for maintaining the developed software*" (developer).

MediaCo is a national broadcaster that adopted Kanban two years ago to compete with the fast changing video-on-demand market. Before, MediaCo *"used the waterfall model with detailed requirements specifications and project planning for a long time"* (SRM). The company is divided into a business (e.g. marketing team) and an IT department. After the adoption, the structure remained unchanged. However, both SD teams located in the IT department adopted Kanban and formed agile teams. An SDM supports each agile team. A Kanban team consists of an SDM and six to eight developers. Due to limited resources, MediaCo uses the SDM as an SRM to mediate between the business and agile teams. The actual SRM is located in the marketing team that places requirements. MediaCo follows a DevOps approach meaning that Kanban teams deploy and maintain the developed software. With the ongoing use of Kanban, agile methods *"increasingly appeal to different teams [at MediaCo] and spread outside of SD (e.g. to accounting)"* (manager).

Cross-Case Analysis

In the following, we present the results of our cross-case analysis illustrated in Figure 4, starting with the input phase by describing the sources of structure and the internal system of our case companies. The process phase is the focus of our analysis. In this phase, we describe the tensions arising from coexisting plan-driven and agile methods (i.e. sources of structure) and elaborate on how these tensions were addressed by emergent sources of structure. We conclude by a brief description of the output phase.

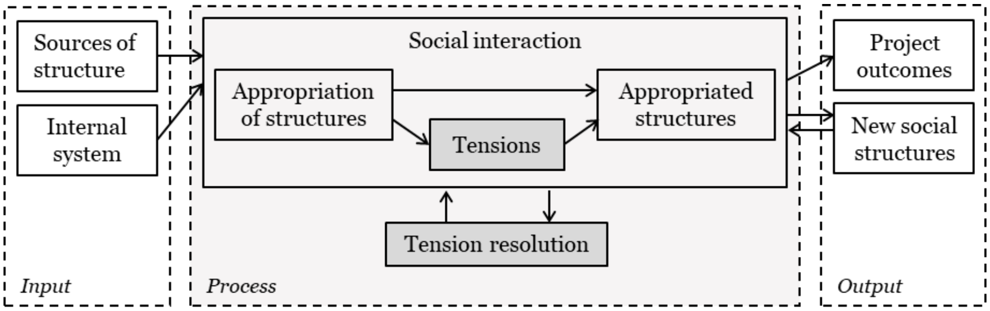

Figure 4. Tensions Arising in the Process Phase.

The input phase comprises plan-driven and agile methods as sources of structure and the internal system. Before the adoption, all organizations applied the waterfall model (plan-driven methods) for SD "*including a long requirements process and heavy documentation*" (manager at CommerceCo). Thus, the *structural features of plan-driven methods* were similar in all four cases including a hierarchical structure to exercise disciplinary control and a sequential plan to follow strict policies and procedures (*plan-driven spirit*). FinanceCo, CommerceCo and E-commerceCo initially implemented Scrum according to the guidelines provided by Schwaber and Beedle (2002). *"We have all Scrum artifacts in place. We have a planning meeting, a review and a retro and all team roles"* (developer at FinanceCo). MediaCo introduced Kanban also according to the method's guidelines: "*We started by following the principle 'start with what you do now' and introduced work in progress limits. We visualize our work with a Kanban board"* (SDM). Some teams at E-commerceCo adopted Kanban acknowledging the method's guidelines. Kanban was seen as more appropriate to ensure continuous delivery outside of a sprint framework. Thus, all companies introduced agile methods according to the *agile method structure*. Regarding the *agile spirit*, all organizations promote *"a failure culture with transparent processes and open feedback. All teams are open to agile methods and pursue customer-orientation by focusing on communication and iterative processes"* (MediaCo's SM). Concerning the *internal system*, the organizations rely on command-and-control leadership and had no (FinanceCo and CommerceCo) or little (E-commerceCo and MediaCo) experience with agile methods beforehand. None of the four companies provided extensive training. For example, only a single training day was provided to employees of CommerceCo and E-commerceCo. Later in the process, FinanceCo and CommerceCo hired developers with agile expertise to promote the adoption of agile methods.

We summarize the tensions and an exemplary codes in Table 2.

Tension (definition)	**Sample of interview statement**	**Revised code**
Budgeting (Issue of allocating and controlling budgets)	*“The business team is accountable for the whole budget spent. But often they struggle to decide what amount to allocate to which team.”*	Budget related issue
Knowledge (Lack of skills to complete autonomously)	*“The service request manager lacks knowledge to completely overview the process.”*	Incapability of completing a task autonomously
Process (Lack of compliance with agile processes)	*“Business people come to [the agile team] and prioritize tasks without consulting our product owner.”*	Plan-driven processes inhibit agile ones
Planning (Plan-driven planning opposes agile methods)	*“We argue about fixed timelines with the business team. They don’t understand that it’s impossible to plan the product in detail one year ahead.”*	Detailed plan for development
Responsibility (Unwillingness and inability to assume responsibility)	*“Developers are not allowed to merge their code into the master branch without having it checked by the IT operations team.”*	Issue with assuming responsibility
Cultural (Plan-driven culture inhibits agile structures)	*“Collaboration between business and agile team is difficult, they have a completely different point of view.”*	Issue of opposing culture

Table 2. Emerging Tensions and Illustration of Coding Scheme

Regarding the process phase, all case companies described that they had been using the waterfall method for decades. Besides the appropriation of plan-driven structures, the respective agile methods were appropriated according to common guidelines at all companies. All case companies introduced agile methods to running SD projects reflecting *appropriation moves*. For instance at MediaCo: *“When we introduced agile methods, we adapted our running project from Waterfall to Kanban”* (manager). SD teams adopted the principles and values of either Scrum (FinanceCo, CommerceCo and E-commerceCo) or Kanban (E-commerceCo and MediaCo) ensuring the *faithfulness of appropriation*. At FinanceCo: *“We enabled all teams to work autonomously and assigned PO and SM to each team”* (manager). At E-commerceCo: *“Each team, regardless of the method in use works on a defined product and collaborates closely with the customer representative”* (SRM). Besides the agile teams, business and IT operations teams continued using plan-driven methods and valued waterfall principles (e.g., detailed project planning and control). For instance, at CommerceCo: *“Although we use Scrum, we still organize every customer request as a new [plan-driven] project”* (PO). The organizations’ purpose of appropriation (*instrumental use*) was to leverage agile methods’ benefits, to resist competitive pressures, and respond to changing consumer needs. The overall *attitude* towards agile methods was mainly positive. The plan-driven project organization frustrated the majority of employees and caused them to welcome the adoption. However, also critical voices were expressed. *“The majority was looking forward to adopting Kanban. We had heard many positive things about agile methods. Others were skeptical about changing our way of working”* (developer, MediaCo).

After the appropriation of both sources of structure (plan-driven and agile methods), tensions arose because of the agile and plan-driven methods’ coexistence which were addressed by emergent sources of structure.

First, the coexistence of methods caused tensions regarding budgeting. All case companies kept allocating budgets in a traditional fashion (e.g., defining ex-ante budgets for all projects). The need for budgetary control caused a tension between the business team overseeing the project and the agile teams. The plan-driven business team was used to defining project plans with regular status meeting reports. After each iteration, the project management indicated the budget spent for development. However, the process of re-evaluating a product at the beginning of each iteration according to the remaining budget was new to the business team and caused fear of overrunning budgets. “*We have trouble allocating budgets to the development teams. We have one product budget bucket* for *the financial year. As several teams work on the* *same product, one issue is to allocate the necessary budget to the agile teams. We are used to sequential or independent projects with different teams. Now, all teams work on the same product"* (CPO, FinanceCo). At CommerceCo similar tensions emerged in regard to allocating budgets to the agile teams. At MediaCo another budgetary issue arose. MediaCo's SM stated: *"The business side seems to think 'despite agile methods you have to report the budget spent and notify us how projects can be capitalized'. And if a new project comes up, we have the issue of 'finding' any available budget. This rigid process limits our flexibility of developing new features".*

Thus, plan-driven budgetary control executed by the business team inhibited the development process and inhibited new projects. Only E-commerceCo avoided budgeting tensions by organizing the company in product divisions and allocation budgets separately to each division. The other companies had to find other ways to resolve budgeting tensions. MediaCo started experimenting with beyond budgeting, a more empowered and adaptive way of management than traditional command-and-control. However, the company still greatly relies on budgetary control executed by the business team as allocating budgets, especially for new projects, remains an issue. CommerceCo established the role of a CPO that coordinates agile and business teams. The CPO reports the budget spent for development to the business team in charge. By organizing weekly meetings, the CPO hands in reports on a regular basis. At FinanceCo, a weekly meeting organized by the project manager serves as a coordination mechanism. FinanceCo's SM stated: *"The scrum masters of all agile teams meet on a weekly basis. The project manager that leads the meeting asks about our progress and how many man-days we've spent so far."*

Second, a knowledge tension arose from the fact that several employees of different departments form an agile team. Before joining the agile team, some team members (e.g., POs of the business department or SMs of the IT department) performed tasks that covered only part of the development process. *"Before our product owner joined the team, he was only responsible for specifying a little part of the product. Now he governs whole features"* (developer, E-commerceCo). Thus, adopting agile methods, caused changes in team structures that affected agile team members because of lacking information necessary for everyday work. Similar to E-commerceCo, the PO of CommerceCo was appointed to supervise whole product features and was consequently unable to ensuring compliance to prior features and knowing interdependencies of all product features. The PO was incapable with assigning process-compliant tasks. "*The continuous customer experience of* existing *features must not be jeopardized by the introduction of new ones. The product owner is sometimes unable to make decisions autonomously because he lacks knowledge"* (manager, CommerceCo). Additionally, MediaCo's case revealed a knowledge tension for the SRM. At FinanceCo, the knowledge tension arose from incorporating requirements engineering and collecting customer requirements. With a highly complex development product, a single role could hardly combine the necessary knowledge to handle all requests autonomously. For instance, FinanceCo's CPO stated: *"We receive very high-level customer requirements and the product owner is not capable of translating high-level requirements into development tasks".* In summary, knowledge tensions occurred at all case companies. The knowledge tensions were resolved in two ways by our case companies. CommerceCo, for instance, established a new team called process managers that ensured a continuous customer experience. *"All product owners can ask a member of the process manager team for help if they worry about the customer journey,"* stated CommerceCo's manager. MediaCo, FinanceCo and E-commerceCo, resolved knowledge tensions through a regular weekly meeting with necessary stakeholders (e.g., PO and other members of the business teams) to address open issues on which team members lacked information. In the second year, FinanceCo reduced the meeting frequency to biweekly.

Further, a process tension occurred from the interference of plan-driven teams with agile processes. Rapid, continuous processes are key specifically for agile teams to manage their short iterative workflows without interruptions. Disturbing agile workflows would cause process disturbances beyond recall. By handing over software to the IT operations team for maintenance, this workflow was interrupted. At FinanceCo: *"Although we ourselves are responsible to integrate our code, the IT operations team denies the handover of software after a certain time which really impedes the process"* (SM). In contrast, the business team interfered with the agile process by overruling the PO's or SRM's responsibility of prioritizing tasks and the developers' ability to commit to defined work. For instance, at E-commerceCo: *"Oftentimes we have the issue to develop product features at the last minute. When the business team decides to include features in the next release by all means, we start to struggle"* (developer). Similarly, at MediaCo the business team regularly ignored the responsibility of the SRM and included features for the agile team to work on, with highest priority. In conclusion, the collaboration between both business and IT operations teams causes a process tension at FinanceCo, E-commerceCo and MediaCo. Being in the first year of its adoption of agile methods, CommerceCo organized processes still in a plan-driven way and experienced no process tensions. Our analysis revealed that the process tensions were resolved by introducing new coordination mechanisms or granting autonomy to the agile team. At FinanceCo: *"Enabling the agile team to merge their software independently into the master branch solved a huge impediment. The possibility to merge code without having it checked by IT operations makes the team less dependent on the IT operations team"* (SM). Similarly, at MediaCo the SDM manager coordinated deployments: *"We are just not allowed to check in code without telling our SDM. The business [team] is afraid we'll destroy the build"* (developer). On the other hand, the established agile roles (i.e. SM and SDM) helped to overcome the process tension. For instance at E-commerceCo *"it helped that our scrum masters are gate keepers in coordination with non-agile teams"* (developer).

Fourth, the adoption of agile methods caused a planning tension. Using the waterfall model entailed detailed planning of project milestones by business teams. The business team that received customer requirements was responsible for planning the product increments for a pre-defined launch date and for releasing this information to customers. The adoption of agile methods changed this detailed planning approach. Agile teams iteratively revised the development scope to increase transparency and to ensure a valuable software release for the customer after each iteration according to fixed parameters (e.g., budget and customer requirements). The unpredictability raised concerns about loss of control over the software development scope on the side of business teams. Thus, agile teams developed autonomously, they were supervised by business teams that considered the agile teams to have lost track of planning and lost sight of the communicated development scope. For instance at MediaCo: "*The business team only cares about the communicated roadmap. They focus on when we deliver what was promised and if we deliver the full scope"* (developer). Similarly, the PO at CommerceCo described that the business teams mainly focused on controlling the product scope and thus impeded continuous delivery: "*The business is mainly interested in facts. For example 'what's the feature scope' and 'when is the feature ready to launch'. They miss the point of being agile – developing a product with the highest customer value possible"*. Planning tensions occurred at all case companies which were addressed by making detailed roadmaps mandatory for agile teams. For instance at E-commerceCo, *"Roadmaps are inevitable for all agile teams. The business needs an annual overview of the software to come"* (SDM). Therefore, none of our examined case companies found a way to resolve the planning tension, but all of them stuck to plan-driven roadmaps. However, we found first approaches in FinanceCo and MediaCo to simplify the required information of a roadmap to reduce agile teams' workload and increase flexibility.

Another tension that emerged from the adoption was a responsibility tension. Agile methods entail that teams are empowered to ensure an autonomous end-to-end development process. However, as only SD teams adopted agile methods, non-agile business and IT operations teams remained accountable for parts of the SD process and impeded the agile team's end-to-end responsibility. Our analysis revealed that non-agile teams slowed down the development process because several mandatory approval process lied within their responsibility. At CommerceCo, for instance, "*IT operations is in charge of creating new git repositories, but they aren't fast enough and block our workflow"* (developer). Similarly, the IT operations team at FinanceCo impeded the agile teams from integrating their software into the system without having it checked by a member of the IT operations team. Besides distributed responsibilities across teams, agile teams refuse to assume responsibility in collaboration with non-agile teams. For instance, at MediaCo *"passing on software to the IT operations team is a major issue. Nobody feels responsible and often we lose a lot of time"* (SDM). Thus, our cases showed two kinds of responsibility tensions. First, a tension caused by a lack of responsibility granted to the agile teams (CommerceCo and FinanceCo) and second a tension resulting from a lack of responsibility assumed by the agile teams (MediaCo). To resolve the arising responsibility tensions, FinanceCo implemented a major organizational restructuring, merging the business and IT teams. By reorganizing teams and flattening hierarchies, employees developed a shared understanding and assumed responsibility for their actions instead of passing it on to others. The reorganization towards interdisciplinary teams fostered a more agile culture and helped the company embrace agile methods further. In contrast, the SDM at MediaCo distributed responsibilities among the team. *"We introduced the role of a story owner within the teams. Each developer is responsible alternately for a specific task and its handover to IT operations"* (SRM). At CommerceCo responsibility tensions could not be resolved. All attempts, such as inviting IT operations personnel to the retrospective meetings, have so far been unsuccessful as the IT operations team refuses to adapt to agile methods.

Finally, we found evidence of cultural tensions. The adoption of agile methods complicated the collaboration between agile SD and non-agile business and IT operations teams as they follow different values, referring to a people-centric approach when using agile methods and a process-centric approach when using plan-driven methods. In a collaboration of teams using different methods the opposing cultures clashed. *"Communication between business and agile teams is tedious. The business team sends emails and involves their team lead for mediation instead of just talk to us."* (SM, FinanceCo). Beyond different approaches of communication, the working style of agile teams at MediaCo and CommerceCo caused tensions. A developer of CommerceCo stated: "*The mindset is still a problem, processes between the agile and IT operations team are very bureaucratic and take time."* In summary, FinanceCo, MediaCo and CommerceCo experienced cultural tensions. E-commerceCo and MediaCo used broader measures to change the culture towards agile methods as MediaCo's SM indicates: *"We offer trainings for the entire workforce on a regular basis. It's still challenging for us to establish a stable agile culture."* On the other hand, FinanceCo and CommerceCo tried to hire personnel with agile skills and experiences to foster a bottom-up agile culture. Being in a later stage of the adoption, E-commerceCo experienced no tension in relation to culture. After using agile methods in SD for some time and intending to minimize interfaces between agile and non-agile teams, E-commerceCo and MediaCo implemented a DevOps approach. Also interviewees of FinanceCo and CommerceCo reported experiments with DevOps teams.

Regarding appropriated structures, agile teams decided to tailor the methods to their needs. For instance, at CommerceCo and E-commerceCo not every agile team has a dedicated SM. The companies introduced agile methods by sourcing SMs for each agile team but learned over time that team-independent SMs are sufficient. FinanceCo reduced the interval of a team's daily scrum to every other day as co-location served the purpose of regular communication.

Beyond the social interaction within the process phase of agile and non-agile teams, the output phase revealed changes in performance measures, such as quality, costs, and time (i.e., project outcomes). New social structures emerged as a consequence of the tensions occurring from the social interaction process. Our cases showed that resolving tensions lead companies to recognize hybrid methods (i.e., the beneficial combination of the plan-driven and agile methods (*sources of structure*)) as a potential resolution or adopted other agile methods. Two of our case companies, namely CommerceCo and MediaCo were considering combining agile and plan-driven methods. Whereas MediaCo embraced Kanban over time and thus decided against hybrid methods, CommerceCo saw advantages in this approach (e.g., to keep established processes).

Discussion

The key objective of this paper was to examine the collaboration of agile and non-agile teams (in the software development process). Based on data we gathered from 21 interviews and document analysis, we found six distinct tensions that led us to refine the initial AST model (DeSanctis and Poole 1994) as depicted in Figure 4. By adopting agile methods in SD, organizations find themselves in a situation where the SD team already adopted agile methods but other teams involved in the software development process (e.g., business and IT operations teams) stick to plan-driven methods, which leads to a coexistence of methods in collaboration between teams.

The collaboration of agile and non-agile teams causes tensions related to budgeting, knowledge, process, planning, responsibility and culture. All tensions impede the SD severely. By developing emerging sources of structure, the teams aim at resolving these tensions and ensure to continue the development process.
Our cases revealed that different types of resolutions emerged. According to Gregory et al. (2015), we use the terms *balancing* and *blending* to account for the types of resolution. Balancing refers to a "compromise among contrasting demands through coordination mechanisms and roles" (Gregory et al. 2015, p. 76). *Blending* on the other hand, is a harmonious solution of two contrasting demands in order to combine both. We regard a resolution as harmonious if it facilitates the adoption of agile methods. Balancing provides either coordination mechanisms (e.g., coordination meetings) or coordinating roles (e.g., the process manager ensures a continuous customer experience) that offer only an intermediate resolution for a tension. Balancing resolutions can be regarded as intermediate steps that bridge tensions and thus restore the development process. However, balancing tensions potential preserves plan-driven methods by reestablishing known routines that inhibit the agile method's full potential to resolve tensions permanently and promote the adoption, blended resolutions are necessary that originate from agile methods. Thus, balancing tensions

tend to preserve plan-driven methods, while blending tensions promote agile methods. Table 3 summarizes the tension resolutions and Figure 5 illustrates the process of resolving tensions.

Tension	Resolution approaches (Case)	Type of resolution
Budgeting	Beyond budgeting (MediaCo) Coordinating role (CommerceCo) Weekly exchange meeting (FinanceCo)	Blending Balancing Balancing
Knowledge	Coordinating role (CommerceCo) Weekly exchange meeting (FinanceCo, E-commerceCo, and MediaCo)	Balancing Blending
Process	Strengthening team autonomy (FinanceCo) Coordinating role (MediaCo) Strengthening the agile roles (E-commerceCo)	Blending Balancing Blending
Planning	No resolution	
Responsibility	Reorganization of the company (FinanceCo) Shared responsibility (MediaCo) No resolution (CommerceCo)	Blending Blending
Cultural	Individual trainings (MediaCo) Employ new personnel (FinanceCo and CommerceCo)	Blending Blending

Table 3. Tension Resolutions

Subsequently, we discuss the resolution approaches for the revealed tensions. First, we found budgeting tensions at three of our case companies (FinanceCo, CommerceCo, and MediaCo). The reasons for budgeting tensions were similar at all companies. The plan-driven and agile characteristics, control and flexibility are opposed in the course of budgeting. Originating from a plan-driven way of allocating budgets, all companies struggle with allocating one budget for different agile teams. Depending on the priority of a software feature, the flexibility of budget allocation for agile teams has to increase. By examining "specific factors (…) between agile and traditional approaches for IT funding" (p. 204) budgeting tensions were resolved as proposed by Cao et al. (2013). By experimenting with beyond budgeting (Lohan et al. 2010), MediaCo tried to find a blended resolution. However, strict budgetary control executed by the business team prevented the initiative. CommerceCo and FinanceCo resolved budgeting tensions with balanced resolutions. Both case companies established a coordination mechanism. While CommerceCo alienated the role of the CPO whose task is to coordinate the POs, FinanceCo organized a meeting with all SMs and a project manager to report progress milestones. As the blended resolutions resolved the tension intermediately but cannot serve as a permanent resolution the budgeting tension persisted.

Knowledge tensions occurred as personnel of the business team (e.g. the PO) and the SD team (e.g. the SM) assembled agile teams. Agile team members of different departments were not used to working together within the same team (Fitzgerald and Stol 2017). Knowledge tensions were resolved by exchanging necessary feature knowledge between the new agile role (i.e. the PO) and the business teams on a regular basis. MediaCo, FinanceCo and E-commerceCo attempted to spread knowledge and thus enable the customer representatives (e.g., the PO) to work autonomously by fostering face-to-face communication (Rolland et al. 2016). Sharing and spreading knowledge is a vital engine to promote agile methods (Strode et al. 2012) However, CommerceCo resolved the knowledge tension by establishing a team (i.e. process manager team) that supported the PO to ensure a continuous customer experience. Over time, the PO might learn from collaborating with the process manager but by forming a new team besides the agile team CommerceCo found a resolution that only bridges the tension. Adding a new team and thus dividing knowledge instead of enabling the PO is considered a balanced resolution attributed to plan-driven methods.

Regarding process tensions emerging from opposing agile and plan-driven practices, FinanceCo and E-commerceCo implemented blended resolutions by strengthening the team autonomy (in FinanceCo' case) and supported the role of the SM (in E-commerceCo's case). Fostering the role of the SM as gatekeeper revealed that the decision to disengage SM and SDM from teams was revoked. The SMs can be seen as an empowering agile leader (Andrias et al. 2018). Restricting the agile team in continuously integrating software autonomously prevented a complete implementation of agile methods at FinanceCo. Inter-team coordination without giving authority to the teams is a major challenge as the agile teams are unable of making decisions autonomously and lose speed (Dingsøyr et al. 2018). At MediaCo, process tensions between the agile and IT operations teams were resolved by using the SDM as coordinator. By establishing a balanced resolution, the tensions between teams persisted but were bridged. Thus, the coordinating role is a measure towards plan-driven methods. With regard to agile methods the team would have resolved the tension autonomously (Hoda and Murugesan 2016). Agile ways to resolve process tensions (e.g., granting autonomy to the agile team) foster agile methods by increasing their status on team level.

Responsibility tensions at FinanceCo and MediaCo were resolved by blended resolutions. FinanceCo started a reorganization of the company by merging IT and business pillars. The organization expected to integrate distributed knowledge and ease up coordination between teams engaged in development process. Whenever agile methods reach their limits they try to break open functional silos and lead to networked organizations that integrate knowledge and have the power of decision making across boundaries (Peppard 2018). From an organizational perspective, FinanceCo supported the adoption of agile methods by fostering cross-functional teams (Hoda and Murugesan 2016). MediaCo supported collective responsibility of the development team to achieve accountability of agile team members to each other (McHugh et al. 2012). By sharing responsibility in collaboration with a non-agile team (i.e. handing over software to the IT operations team), the agile team encouraged agile practices and started self-organizing its processes to become an autonomous agile team. By emphasizing self-organization "the roles played by individuals are (…) reshaped" (p. 358) to increase autonomy, more frequent interactions and greater participation in the decision making process (Vidgen and Wang 2009). Vidgen and Wang (2009) find that optimizing self-organization fosters agility. Responsibility tensions between agile and non-agile team at CommerceCo remained as the non-agile team (i.e. IT operations team) was not willing to adopt an agile mindset (Fuchs 2019).

Cultural tensions are recognized when adopting agile methods (Iivari and Iivari 2011). When adopting agile methods opposing characteristics cause cultural tensions that are especially challenging (Ramesh et al. 2017). MediaCo, CommerceCo and FinanceCo found blended resolutions for cultural tensions. Whereas MediaCo chose trainings for the entire workforce to boost the agile culture in a broader sense, FinanceCo and CommerceCo decided to establish the agile culture within the teams using agile methods by employing agile trained personnel. Thereby, both organizations intended to foster a bottom up agile culture that spreads across team boundaries. All organizations acknowledged cultural factors (e.g., an agile mindset) to be particularly important for the transition.

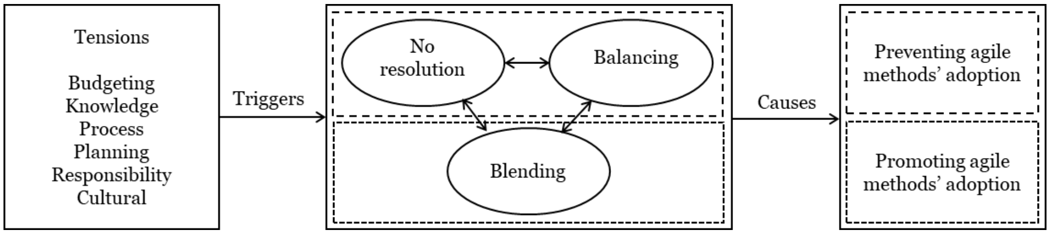

Figure 5. Process of Tension Resolutions.

A comparison of all companies revealed that balanced resolutions prevented the adoption of agile methods and blending resolutions promoted agile methods. In CommerceCo's case, we saw that tensions arising from the opposing orientations of plan-driven and agile methods were resolved more frequently by balanced solutions. The company in the first year of the adoption struggled with agile methods and felt more comfortable establishing intermediate (balanced) resolutions than to rely on agile methods. However, FinanceCo and MediaCo, further along the adoption process appreciated agile methods and tried to find blended resolutions whenever possible. By organizing according to their product domains, E-commerceCo in the third year of using agile methods managed to find resolutions exclusively beyond balancing. The planning tension was an obstacle for all organizations; however, they were unable to resolve it. In accordance with Fitzgerald and Stol (2017), we find that planning is performed according to a traditional plan-driven cycle and continuous planning is a key prerequisite for delivering software continuously. Our case companies failed to adopt continuous planning. Thus, Cooper and Sommer (2016) present the integration of traditional planning into an agile SD process as a future research idea. Finally, we conclude that replacing balanced with blended resolutions fosters an agile culture and spreads agile methods beyond SD (see Figure 5). All companies announced to adopt or implemented a DevOps approach to extend agility. By implementing or planning to implement a DevOps approach to solve interface issues between agile and non-agile teams, all four organizations indicate to resolve tensions by elevating the resolution to a higher level. The move towards DevOps responds to the enduring need to break functional silos. Using agile methods without DevOps is considered a paradigm with two separate functions (Hemon et al. 2018). The reorganization at FinanceCo revealed evidence for an implementation of a BizDev approach. By combining business, and SD operations teams, the company seeks to tighten the adoption of agile methods (Fitzgerald and Stol 2017).

Implications

Theoretical Implications

This study addresses theoretical and empirical gaps in the literature on coexisting methods. The findings of our study refine theory to explain an organization's adoption of agile teams in plan-driven environments which results in a coexistence of agile and plan-driven methods. This study contributes to the literature on agile methods in two ways. First, our study reveals tensions that emerge from a collaboration of plan-driven and agile teams. Building on AST (DeSanctis and Poole 1994), we examine the coexistence of methods as social interactions between agile and non-agile teams and unearth tensions between the appropriation of structures and affecting appropriated structures. We show that tensions are resolved by emergent sources of structures such as resolving the knowledge tension by establishing a coordination meeting. Thus, we extend knowledge on the occurrence of emergent sources of structure (Cao et al. 2009). Further, we find that the emergent sources of structure either accelerate (blended resolutions) or slow (balanced resolution) the adoption of agile methods. This is evident in the case of CommerceCo where mainly balanced resolution approaches are used that inhibit a promotion of agile methods. Second, besides revealing tensions emerging from a coexistence of agile and plan-driven methods, we show that the adoption of agile methods is a multilayered phenomenon which entails the reorganization of structures, roles, and processes. Our study indicates that enabling a close collaboration between teams is inevitable to achieve a complete transition. Thus, after resolving tensions emerging from a malfunctioning collaboration between agile and plan-driven teams, organizations are triggered to find more integrated, cross-functional approaches such as DevOps. Organization use DevOps as a way to extend agile methods and break down functional silos.

Managerial Implications

Managers can draw several insights from this study. Our results indicate that restructuring an organization to facilitate the adoption of agile methods is beneficial. By merging business and IT departments, emerging tensions from the adoption of agile teams in plan-driven environments are mainly resolved. FinanceCo's case indicates that a reorganization of plan-driven structures benefits the transition process. Moreover, E-commerceCo's organizational structure (i.e. separate product domains in business units) reveals that limiting interfaces between agile and non-agile teams boosts the adoption of agile methods (Jöhnk et al. 2017). Further, we find that balanced resolutions subvert the advantages of agile methods, especially flexibility and speed (Conboy 2009). Thus, choosing balanced resolutions yields not only to an impairment of the adoption of agile methods but weakens competitiveness of an organization. In times of rapid changing markets, the emphasis should be on guiding teams to blended resolutions towards agile methods.

Limitations and Further Research

In this study, we have proposed an extension of AST by including tensions based on the evidence of four case studies with organizations undergoing an adoption of agile methods in SD after using plan-driven methods for decades. By considering the adoption process, we are only able to observe the coexistence of methods for single point in time. Through the phase of data collection and analysis we ensured reliability and internal validity of our data. However, a longitudinal study, which we propose for future research, might reveal additional tensions. Further, generalizing our findings to other contexts that involve tensions emerging from collaboration should be done with caution, as our findings may be particularistic to use of specific tensions from plan-driven and agile methods, and to firm-specific factors. Nevertheless, generalization of the findings requires quantitative approaches that test our adapted AST model (Figure 4). Further research could adopt competing theoretical lenses besides AST that may lead to new additional insights regarding the resolution of tensions emerging from an adoption of agile methods.

References

Anderson, D. J. 2010. *Kanban: Successful Evolutionary Change for Your Technology Business*. Seattle, WA: Blue Hole Press.

Andrias, M. S., Sabine, M., and Richard, V. 2018. "Towards a Typology of Agile ISD Leadership.," *Proceedings of the 26th European Conference on Information Systems (ECIS)*, Portsmouth, UK.

Austin, R. D., and Devin, L. 2009. "Weighing the Benefits and Costs of Flexibility in Making Software: Toward a Contingency Theory of the Determinants of Development Process Design," *Information Systems Research* (20:3), p. 462.

Bick, S., Spohrer, K., Hoda, R., Scheerer, A., and Heinzl, A. 2017. "Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings," *IEEE Transactions on Software Engineering* (44:10), pp. 932-950.

Campanelli, A. S., Camilo, R. D., and Parreiras, F. S. 2018. "The Impact of Tailoring Criteria on Agile Practices Adoption: A Survey with Novice Agile Practitioners in Brazil.," *Journal of Systems and Software* (137), pp. 366-379.

Cao, L., Mohan, K., Ramesh, B., and Sarkar, S. 2013. "Adapting Funding Processes for Agile It Projects: An Empirical Investigation," *European Journal of Information Systems* (22:2), pp. 191-205.

Cao, L., Mohan, K., Xu, P., and Ramesh, B. 2009. "A Framework for Adapting Agile Development Methodologies," *European Journal of Information Systems* (18:4), pp. 332-343.

Charlier, S. D., Stewart, G. L., Greco, L. M., and Reeves, C. J. 2016. "Emergent Leadership in Virtual Teams: A Multilevel Investigation of Individual Communication and Team Dispersion Antecedents," *The Leadership Quarterly* (27:5), pp. 745-764.

Chin, W. W., Gopal, A., and Salisbury, W. D. 1997. "Advancing the Theory of Adaptive Structuration: The Development of a Scale to Measure Faithfulness of Appropriation," *Information Systems Research* (8:4), pp. 342-367.

Conboy, K. 2009. "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development," *Information Systems Research* (20:3), pp. 329-354.

Cooper, R. G., and Sommer, A. F. 2016. "The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity," *J. Prod. Innov. Manag* (33:5), pp. 513-526.

Corbin, J. M., and Strauss, A. 1990. "Grounded Theory Research: Procedures, Canons, and Evaluative Criteria," *Qualitative Sociology* (13:1), pp. 3-21.

DeSanctis, G., and Poole, M. S. 1994. "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory," *Organization Science* (5:2), pp. 121-147.

Dingsøyr, T., Moe, N. B., Fægri, T. E., and Seim, E. A. 2018. "Exploring Software Development at the Very Large-Scale: A Revelatory Case Study and Research Agenda for Agile Method Adaptation," *Empirical Software Engineering* (23:1), pp. 490-520.

Fitzgerald, B., and Stol, K.-J. 2017. "Continuous Software Engineering: A Roadmap and Agenda," *Journal of Systems and Software* (123), pp. 176-189.

Fuchs, C. 2019. "Adapting (to) Agile Methods: Exploring the Interplay of Agile Methods and Organizational Features," *Proceedings of the 52nd Hawaii International Conference on System Sciences*.

Giddens, A. 1987. *Social Theory and Modern Sociology*. Stanford, CA: Stanford University Press.

Glaser, B. G., and Strauss, A. L. 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, IL: Aldine.

Gregor, S. 2006. "The Nature of Theory in Information Systems," *MIS Quarterly* (30:3), pp. 611-642.

Gregory, R. W., Keil, M., Muntermann, J., and Mähring, M. 2015. "Paradoxes and the Nature of Ambidexterity in IT Transformation Programs," *Information Systems Research* (26:1), pp. 57-80.

Hekkala, R., Stein, M.-K., Rossi, M., and Smolander, K. 2017. "Challenges in Transitioning to an Agile Way of Working," *Proceedings of the 50th Hawaii International Conference on System Sciences*.

Hemon, A., Monnier-Senicourt, L., and Rowe, F. 2018. "Job Satisfaction Factors and Risks Perception: An Embedded Case Study of Devops and Agile Teams," *Proceedings of the 99th International Conference on Information Systems (ICIS)*, San Francisco, CA.

Hobbs, B., and Petit, Y. 2017. "Agile Methods on Large Projects in Large Organizations," *Project Management Journal* (48:3), pp. 3-19.

Hoda, R., and Murugesan, L. K. 2016. "Multi-Level Agile Project Management Challenges: A Self-Organizing Team Perspective," *Journal of Systems and Software* (117), pp. 245-257.

Iivari, J., and Iivari, N. 2011. "The Relationship between Organizational Culture and the Deployment of Agile Methods," *Information and Software Technology* (53:5), pp. 509-520.

Jöhnk, J., Röglinger, M., Thimmel, M., and Urbach, N. 2017. "How to Implement Agile It Setups: A Taxonomy of Design Options," *Proceedings of the 25th European Conference on Information Systems (ECIS)*, Guimarães, Portugal.

Lewis, M. W. 2000. "Exploring Paradox: Toward a More Comprehensive Guide," *Academy of Management Review* (25:4), pp. 760-776.

Lohan, G., Conboy, K., and Lang, M. 2010. "Beyond Budgeting and Agile Software Development: A Conceptual Framework for the Performance Management of Agile Software Development Teams," *Proceedings of the 31th International Conference on Information Systems (ICIS)* St. Louis, MO.

Mahadevan, L., Kettinger, W. J., and Meservy, T. O. 2015. "Running on Hybrid: Control Changes When Introducing an Agile Methodology in a Traditional" Waterfall" System Development Environment," *Communication of the Association for Information Systems* (36:5), pp. 77-103.

McHugh, O., Conboy, K., and Lang, M. 2012. "Agile Practices: The Impact on Trust in Software Project Teams," *IEEE Software* (29:3), pp. 71-76.

Moe, N. B., Dingsøyr, T., and Dybå, T. 2010. "A Teamwork Model for Understanding an Agile Team: A Case Study of a Scrum Project," *Information and Software Technology* (52:5), pp. 480-491.

Nerur, S., Mahapatra, R., and Mangalaraj, G. 2005. "Challenges of Migrating to Agile Methodologies," *Communication of the Association for Information Systems* (48:5), pp. 72-78.

Patton, M. Q. 1990. *Qualitative Evaluation and Research Methods*. Thousand Oaks, CA: SAGE Publications.

Peppard, J. 2018. "Rethinking the Concept of the Is Organization," *Information Systems Journal* (28:1), pp. 76-103.

Ramesh, B., Cao, L., Kim, J., Mohan, K., and James, T. L. 2017. "Conflicts and Complements between Eastern Cultures and Agile Methods: An Empirical Investigation," *European Journal of Information Systems* (26:2), pp. 206-236.

Rigby, D. K., Sutherland, J., and Noble, A. 2018. "Agile at Scale," *Harvard Business Review* (96:3), pp. 88-96.

Rolland, K., Dingsoyr, T., Fitzgerald, B., and Stol, K.-J. 2016. "Problematizing Agile in the Large: Alternative Assumptions for Large-Scale Agile Development," *Proceedings of the 37th International Conference on Information Systems (ICIS)*, Dublin, Ireland.

Schlauderer, S., Overhage, S., and Fehrenbach, B. 2015. "Widely Used but Also Highly Valued? Acceptance Factors and Their Perceptions in Water-Scrum-Fall Projects," *Proceedings of the 36th International Conference on Information Systems (ICIS)* Fort Worth, TX.

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

Smith, W. K., and Lewis, M. W. 2011. "Toward a Theory of Paradox: A Dynamic Equilibrium Model of Organizing," *Academy of Management Review* (36:2), pp. 381-403.

Strode, D. E., Huff, S. L., Hope, B., and Link, S. 2012. "Coordination in Co-Located Agile Software Development Projects," *Journal of Systems and Software* (85:6), pp. 1222-1238.

Urquhart, C., Lehmann, H., and Myers, M. D. 2010. "Putting the 'Theory'back into Grounded Theory: Guidelines for Grounded Theory Studies in Information Systems," *Information Systems Journal* (20:4), pp. 357-381.

Venkatraman, V. 2017. *The Digital Matrix: New Rules for Business Transformation through Technology*. Vancouver: LifeTree Media.

Vidgen, R., and Wang, X. 2009. "Coevolving Systems and the Organization of Agile Software Development," *Information Systems Research* (20:3), pp. 355-376.

Vinekar, V., Slinkman, C. W., and Nerur, S. 2006. "Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View," *Information Systems Management* (23:3), pp. 31-42.

Yin, R. K. 2009. *Case Study Research: Design and Methods*. Los Angeles: Sage.