The Impact of Modes, Styles, and Congruence of Control on Agile Teams: Insights from a Multiple Case Study

Abstract

Agile software development (ASD) strongly relies on social interaction and teamwork. Team processes and agile practices adopted by team members play an important part for the outcome of software development projects. Agile practices promise teams to be able to respond to change by granting them autonomy. Existing studies, however, imply that these projects can benefit from different elements of control. Our objective is to improve our understanding of how to enact control in agile teams and how these control mechanisms influence team autonomy and team performance. In this paper, we present our findings from four case studies conducted within two insurance companies and two software development firms. We found that it is not a question of 'what' controls should be exercised, but rather 'how' controls are implemented in practice. Our results prompt to the need for further studies on control mechanisms in ASD.

1. Introduction

In the almost two decades since the publication of the Agile Manifesto [2], *agile software development (ASD)* approaches have emerged as a dominant paradigm [18]. The capability of responding rapidly to changing user requirements promoted by ASD "has become increasingly critical for software development performance" [26]. Whereas each ASD method may differ in terms of emphasis on key principles or suggested practices for action, they all have in common that they emphasize the importance of *project teams*, which are empowered to make decisions on their own by ASD, while the project manager's role has become rather team-supportive than team-directive [30]. As a result, one key characteristic that is often considered in order to determine if a project team is being 'agile' is the principle of *team autonomy* – providing individual team members and groups the power to self-organize and the discretion of self-direction [32]. However, extant research paints an ambiguous picture of team autonomy's impact on team behavior and outcomes. On the one hand, team autonomy has been observed to inhibit productivity and performance in the context of project teams [24]; on the other hand, it has been identified as a key factor enabling teams to respond to change and thus enabling them to perform in environments where business needs continuously evolve over time and the whole ASD process is a "moving target" [26, 28, 51].

The linkage between team autonomy and team behavior as well as outcomes respectively is further influenced and complicated by the question of *control* – understood broadly to mean "any process in which a person or group of persons or organization of persons determines […] what another person or group or organization will do" [48]. As this definition suggests, the exercise of control necessarily implies certain limits on the ideal of team autonomy. Yet, research suggests that control leads to better task performance within a team [14, 56], even in ASD contexts [15, 23, 38], for instance, by aligning team members and increasing team cohesion [38], having a positive effect on such performance measures as software quality [28].

In sum, only limited guidance exists on how ASD teams should be governed with regard to the relationship between control and team autonomy [28]. It is not clear how much team autonomy and how much control are needed, and what the fitting balance between both is. This is especially the case in an ASD context [8]. Accordingly, we follow recent calls [52] for further research on balancing the enactment of control and team autonomy in ASD [3, 49], the interplay between different ways of enacting control [33, 38, 53], and their relationship to team autonomy [9] and team performance [23]. Consequently, the central research question guiding our study is:

> *How does the enactment of control embodied in agile practices influence team autonomy and team performance of project teams?*

In pursuing this question, we build on both the long-standing insights of control theory [e.g., 21] and recent research on control in the context of information systems development (ISD) projects [52]. As far as we know, there are no studies that address control and agile practices, specifically focusing on the balance between control and team autonomy. Integrating these perspectives, we propose a model to investigate the influence of control on agile teams, which aims to improve the ASD process and its outcomes. Specifically, we build on existing literature to suggest that agile practices are likely to enact different control modes and therefore have a direct effect on team performance and team autonomy. Moreover, we propose that, aside from the direct exercise of different types of control, different control styles and degrees of control congruence influence the behavior of agile teams and outcomes.

In the following, we give an overview of related work and our theory development. This is followed by a description of the cases and the research methods. Subsequently, we present the results of our analysis. Finally, we discuss our results, implications, and limitations.

2. Related Work

2.1 Agile Software Development

ASD is an umbrella term for a variety of distinct methods, such as Scrum and eXtreme Programming (XP) [e.g., 40, 46], which collectively emphasize an iterative development model, close collaboration between stakeholders, and a lightweight approach to documentation. One common feature that characterizes these methods is that they grant more flexibility and autonomy to an ASD project team. In ASD, the overall development process is not planned and scheduled upfront by an all-powerful project manager; progress is made in small iterative phases, with decisions taken by the team [18].

In a business environment where available technologies, market structures, and customer preferences change rapidly, ASD approaches have been shown to enable teams to react to emergent needs in a timely manner [5, 41]. When teams decide to apply ASD methods, key *agile practices* and principles have to be considered [19, 37, 45]. Examples of agile practices from XP are pair programming (all production code is written with two programmers at one machine) or collective code ownership (anyone can change any code anywhere in the system at any time). Popular Scrum practices include daily scrums (a daily stand-up meeting in which all project participants briefly review the status of their work) or user stories (a method to define broad requirements while enabling creativity) [16, 50].

While ASD places an emphasis on autonomous and self-organizing teams [2], and while many agile practices support a self-organized and self-governing team [26], control is nevertheless enacted [15, 23, 38].

2.2 Control Enactment

Within our research, we define control broadly to mean “any process in which a person or group of persons or organization of persons determines […] what another person or group or organization will do” [48]. We primarily rely on control theory by Kirsch [21, 22] and focus on extensions of the expanded theoretical framework of IS project control [52], which serve us as theoretical lenses.

Although particular ISD methods are not specifically addressed within control theory [6], Kirsch points out that organizations in dynamic, changing environments may change control approaches over the course of an ISD project’s lifecycle, resulting in the implementation of appropriate control types [21, 22]. With respect to ISD teams, theory distinguishes *formal control modes*, such as input, behavior, and outcome control, from *informal control modes*, such as self-control and clan control [21]. Table 1 summarizes key control modes, which often are exercised in concert rather than in isolation, representing a so-called control portfolio [22].

Table 1: Summary of control modes following Kirsch [21] & Jaworski [20]

Control Mode		Characteristics
Formal	*Input Control*	Measurable actions prior to implementation of an activity e.g. recruitment, training programs or manpower allotments.
	Behavior control	Emphasizes behaviors, processes and procedures that must be followed, and offering rewards contingent on the adherence to the prescriptions.
	Outcome control	Involves outlining project goals, and offering rewards contingent on their accomplishment. Emphasizes outputs regardless of the process used.
Informal	*Clan control*	Socializes team members into sets of valued norms. Emphasizes reinforcement of acceptable behaviors through shared rituals and experiences.
	Self-control	Provides autonomy to individuals to determine what actions are required and how to execute them. Emphasizes self-regulation of goals and self-monitoring of progress.

The exercise of formal control provides guidance and structure, which assist an ISD team in task execution [23, 42]. It is well known that traditional ISD approaches rely heavily on formal control mechanisms [21-23]. By contrast, informal control potentially provides developers with discretion with regard to how tasks are accomplished [17, 23, 28, 49]. Informal controls such as clan and self-control promise to enact autonomy, which is seen as an important antecedent for responding to changing user requirements [9, 28]. The exercise of clan control allows the development team to identify important project goals and to determine how to attain them on their own [28]. The exercise of self-control similarly enables flexibility in pursuit of objectives, focusing on the role of the individual rather than that of the group. Self-control represents “the extent to which an individual exercises freedom or autonomy to determine both what actions are required and how to execute these activities” [17].

While most studies focus on controlling portfolio configuration (“what” control modes are used), few studies investigate “how” controls can be put into practice [12, 49] – the enactment of control. *Control enactment* can be defined as the interaction between a controller (the person exercising control) and a controllee (the target of control), or in other words, the way in which the controller puts different modes of control into practice [52].

Building on this understanding, we see *control style* as a relevant concept for our context, which can be defined “as the manner in which the interaction between the controller and the controllee is conducted” [52]. Related literature distinguishes between two contradictory control styles – authoritative and enabling [1, 12]. An *authoritative control style* is employed if strict behavioral compliance is desired, granting the controllee less discretion in how control is enacted [52]. An *enabling control style*, on the other hand, is used to achieve compliant behavior while granting flexibility in decision making to deal with uncertainties in daily work procedures [1, 42].

Moreover, with regard to “how” controls can be put into practice, we consider the concept of *control congruence* as another important element of control enactment in ASD [35, 52]. Control congruence can be understood as the “level of agreement” and “degree of understanding” between a controller’s and controllee’s perceptions of distinct controls [35]. The level of agreement regarding the appropriateness of controls is also called “evaluational congruence”, whereas the degree of (a shared) understanding is known as “communicational congruence” [35]. Thus, control congruence may influence the quality of the whole control enactment process [52].

2.3 Autonomy and Team Performance in Agile Teams

Flexibility and adaptiveness in ASD approaches is reflected in the concept of *team autonomy* [25, 26]. Prior literature provides various definitions of team autonomy and other closely related concepts, including self-organization [18], self-management [47], and team empowerment [25]. Following Lee and Xia [26], we define team autonomy “as the degree of discretion and independence granted to the team in scheduling the work, determining the procedures and methods to be used, selecting and deploying resources, hiring and firing team members, assigning tasks to team members, and carrying out assigned tasks” [26].

Next to team autonomy, the enactment of control is closely linked to the establishment of *team performance*, which is defined as the degree to which a team achieves its goals and how well its outputs match the team’s mission [14, 56]. Although a variety of empirical studies analyze the effects of control and team performance on project outcomes [15, 27, 28, 38], results remain ambiguous, especially for ASD [7]. For example, ASD project teams can benefit in terms of product quality from the implementation of certain control modes (especially outcome control) to create an environment in which agile practices can engender autonomy while clear performance goals and structures are maintained [28]. On the other hand, Harris, Collins and Hevner [15] argue that formal outcome control is insufficient in agile environments and propose the concept of emergent outcome control as a way to achieve a better product-market match.

3. Theory Development

In light of the inconclusive and partly contradictory results regarding control and the limited extant evidence concerning how control influences an ASD project team, we propose a theoretical model that conceptualizes the interrelationship between control-enacting agile practices and control styles, control congruence, team autonomy, and team performance (see Figure 1). From a control-enactment perspective, we include control modes (in the form of control-enacting practices), control style, and control congruence as independent variables in our research model.

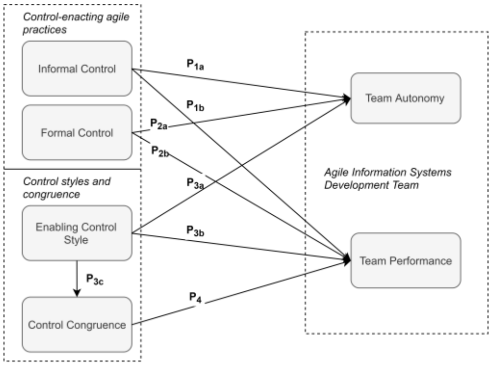

Figure 1: Proposed research model

We suggest that different types of control can be exercised through different agile practices, that is, the method-in-action and generative rules, which are adapted to fit an ASD team's specific context [18]. While some extant research has mapped agile practices to either formal or informal control modes [e.g., 16, 38], conclusive determinations are challenging. Based on the results of an extensive structured literature review (anonymous for review), a total set of 29 agile practices were identified. This analysis included exploring the correspondence to specific control modes. Although most agile practices defy a straightforward classification by control mode, a subset of these practices offer clear indications of formal and informal control modes in their enactment. It should be mentioned that no practice could be identified that addresses input control. Consequently, input control is neglected in the following. Table 2 provides an excerpt of control modes embodied in agile practices.

Table 2: Control modes embodied in agile practices (excerpt)

Agile Practice	Control Modes	References
Backlog prioritization / estimation	BC, OC	[27]
Burndown charts	CC, OC	[13, 27]
Code Reviews / Refactoring	BC. CC, OC, SC	[16, 38]
Collective Code Ownership	CC, SC	[29, 38]
Daily Scrum / stand-up	BC, CC, OC, SC	[7, 27]
Pair Programming	BC, CC, SC	[16, 28]
User stories	OC	[16, 27]
LEGEND: BC = Behavioral Control, CC = Clan Control, OC = Outcome Control, SC = Self-Control		

Next, team autonomy is an important dependent variable in our model, which describes the extent to which a team is granted discretion and independence (e.g., in scheduling the work or carrying out tasks) [26] or is restricted through control [39]. We also suggest to use team performance as a dependent variable, which is defined as the degree to which a team achieves its goals and how well its outputs match the team's mission [14, 56].

We now discuss propositions that link the concepts in our research model. In line with recent arguments from control theory [52], we assert the need for greater consideration of the question of control-enactment – that is, how software project leaders are able to put distinct configurations of control portfolios into practice. Regarding the effects of formal and informal controls, several studies find that informal control usage provides high levels of autonomy in managing assigned work tasks – for example, by enabling the team to determine objectives, tasks, and monitoring activities to achieve project goals [23, 42]. Moreover, informal controls have been found useful in promoting effectiveness, and recent studies emphasize their performance-enhancing effect in the context of specific ISD projects [4, 49]. In particular, the use of self-control provides developers with discretion regarding how tasks are accomplished [17, 23].

For example, self-controlling team members are able to align their resources and choose methods for goal achievement without relying on the project leader to do so [17, 28]. Clan control can be promoted by establishing a collaborative culture within the team, allowing the controller to create an environment where the controllee has freedom to make use of her own skills and knowledge in order to accomplish certain tasks, leading to better team performance [4, 11]. Consequently, we propose:

P1: *Greater use of informal controls positively impacts (a) team autonomy and (b) team performance.*

Other studies find that formal controls "limit the team's autonomy" [39] by overemphasizing work formalization [e.g., 42]. For example, routine team progress reports and strict adherence to schedules and task assignments may hinder a team's effectiveness, as teams frequently turn to managers instead of solving problems on their own [39, 43]. Emphasizing functional specialization puts a manager in the position of controlling most decision making, leading to decreasing team autonomy [9]. On the other hand, formal controls provide some degree of guidance and structure, which supports the execution of tasks [42]. Such controls may provide clear directions and predefined workflows on how to perform certain tasks [23] or recommend proven techniques or practices (e.g., user stories), which in turn positively affect team performance [42]. Hence:

P2: *Greater use of formal control negatively impacts (a) team autonomy, while it positively affects (b) team performance.*

As authoritative and enabling control styles can be seen "as end points on a continuum" [52], we follow Remus, Wiener, Saunders, Mähring and Kofler [42] and focus on an enabling style in our model. An enabling control style has two main characteristics, "repair" and "transparency" [1]. Together, both features establish an environment for the controllee that is characterized by feedback, involvement in the control configuration, and some degree of freedom to "deviate from controller prescriptions […] in order to respond to real-work contingencies" [42]. Additional exchange of knowledge, regular feedback, and close collaboration between controller and controllee leads to increasing team performance [1, 42]. Conversely, a lack of information exchange and feedback mechanisms associated with an authoritative style lead to decreased team performance [3].

We also suggest that an enabling control style increases team autonomy. An enabling style is likely to promote informal controls (such as clan control), which in turn positively affect team autonomy (see P1) [52]. This may be due to the repair and transparency characteristics, which allow for better knowledge exchange and continuous feedback loops [1, 52]. Both features are also able to promote evalutaional and communicational congruence. Beside the direct positive effect on team performance (see P3b) our research indicates a mediation between the variables control style, control congruence and team performance, where control congruence represents the mediator variable. An enabling control style might avoid communication breakdowns, conflict and resistance behaviors which in turn will have a positive effect on team performance (see P4) [35, 53]. Thus, we propose:

P3: *Greater degrees of an enabling control style positively affect (a) team autonomy, (b) team performance, and (c) control congruence.*

Past studies indicate team members' misunderstandings, poor relationships, and conflicts as negatively influencing the overall performance [e.g., 36]. The question arises how congruent values can be generated between controller and the controllee. For example, if controllers are able to establish evaluational congruence, this might be an useful instrument for obtaining feedback about the attempted control mechanisms. This might even "[…] help to foster a climate in which disagreements can be discussed constructively and in turn boost team motivation" [35]. Moreover, communicational congruence can be used to check communication mechanisms against their effectiveness, leading to transparency within the whole team and ensuring that both controller and controllee speak a common language in terms of objectives and tasks to be done to achieve these goals [35].

Consequently, we argue that a high level of control congruence has a positive impact on team performance, as it contributes significantly to the quality of the controls adopted and avoids negative socio-emotional effects such as decreased job satisfaction [35, 50]. Hence:

P4: *Greater degrees of control congruence positively affect team performance.*

4. Research Design and Method

In order to test the relationships between the different concepts, we conducted an *embedded, multiple-case study* of eight teams in four projects in four companies, following a positivist and explanatory approach [34, 54] (Table 3).

All investigated organizational units are based in Germany. We selected the cases following a theoretical sampling logic. Two of the cases, Apocorp and Dominsur, are set in large insurance companies - one of which is active internationally (Apocorp) and one nationally (Dominsur). As the banking and insurance industry is regarded as more traditional and therefore conservative [10], we expected a comparatively high degree of hierarchies and more (formal) control within the two organizations. The two other cases, Unidevelop and Softac, are medium-sized software development companies. By comparison, we expect both Unidevelop and Softac to have a setting with significantly flatter hierarchies and less (formal) control. Based on the differences we therefore expect to observe different characteristics of the control portfolio as well as the control styles exercised, and thus different results.

Apocorp and Dominsur both are in the process of organizational transformation initiatives, which started in both cases a little over a year ago on 2018. With the adoption and use of ASD methods, both companies have set themselves the goal of (a) digitizing the product portfolio and (b) achieving a better time-to-market for these products. All teams of both companies are working according to elements of the Kanban and Scrum methods.

In contrast, Unidevelop and Softac are both familiar with the use of ASD methods for a longer period. Softac already has many years of experience in the field of ASD, but in comparison to Unidevelop also has extensive knowledge of non-agile methods (e.g., waterfall model or extended V-model) for software development. This is due to the fact that Unidevelop is a rather young company, which exclusively uses ASD methods for software development. Unidevelop claims to develop software in an agile way to a high degree. The employees report to be very satisfied with the everyday (agile) routines. A slightly different picture emerges at Softac. The employees stated that they still see some potential here to further advance the "agile way of working". It happens that new processes are introduced and rituals are tried out in order to achieve an even better time-to-market. Table 3 provides a short summary of the cases.

We followed established guidelines for data collection and analysis [31, 44, 55]. We collected data from various data sources and with different data collection methods. Data sources such as work descriptions were used to identify relevant interview participants. Semi-structured interviews and project documentation (e.g., burndown charts, work environments) were used to generate data and to put data into context. We interviewed both project managers and project workers, allowing for triangulation of sources.

Table 3: Cases and informants

	Apocorp[1]	**Dominsur[1]**
Industry	Insurance	Insurance
Size	Large, international company	Large, national company
Teams / Inter-viewees	3 teams, 12 interviews including a project manager, a product owner, a scrum master, developers and agile coaches	3 teams, 12 interviews, including two project managers, a product owner, a scrum master, developers and agile coaches
	Unidevelop[1]	**Softac[1]**
Industry	Software Development	Software Development
Size	Small to medium size, national	Medium size, international
Teams / Inter-viewees	One team, 4 interviews including a project manager, a scrum master and developers	One team, 4 interviews including two project managers and developers

[1] *company names are anonymized for confidentiality purposes.*

Administrative documents, work descriptions, interview transcripts, and field notes were collected in a case study database. We collected data from July 2018 to November 2018 while conducting 32 face-to-face interviews at the organizations' site.

Our guidelines were derived from extant literature. The interviews lasted about 60 minutes and were recorded and transcribed. The guideline was not shared with the interviewees and we only used it as a checklist and outline. The aim was to encourage the interviewees to provide a narrative of their experiences as freely as possible.

Two researchers coded the data independently. We applied different coding strategies and techniques [44]. Within our two-step coding process we started to identify and refine our proposed constructs by means of pattern coding, developing major themes from our data [31, 44]. These codes are capable to "identify an emergent theme" and therefore are helpful for "grouping those summaries into a smaller number of sets, themes, or constructs" [31]. The theoretical lenses of the expanded theoretical framework of IS project control [52] and control theory [21, 22] served as guidelines in providing initial seed codes.

Within the second coding step, we aimed at identifying statements in the conducted interviews to support or reject our propositions by using hypothesis coding [44]. Once again, the above mentioned theoretical lenses of the second coding step served as guidelines for coding the interview data.

5. Findings

Table 4 presents the identified control enactment concepts that we observed in each of the different cases. It should be mentioned that the codes ECS and ACS (enabling and authorative) represent the construct "control style" as well as CC and EC (communicational and evaluational) represent the construct "control congruence". In all cases, control was exercised through managers (including top management) and scrum masters (controller). We distinguish between three different degrees, describing to what extent ("high", "moderate", "low") certain controls could be identified, control styles have been used, and to which there existed control congruence between controllers and controllees. These degrees were derived from the clarity of the statements made and their occurrence. For example, a high degree exists if more than half of the informants have made a clear statement and vice versa, a low degree exists if no or few informants have made statements or these were not conclusive.

In sum, all cases reveal different patterns with respect to the ways in which (a) control is enacted and (b) how these controls impact the team. First, we found evidence that in all cases different formal and informal controls are enacted. For example, in all cases top management was responsible for aspects such as team composition, the allocation of resources (e.g., the design of workspaces), or trainings (*input control*) [20, 21] as well as for the instruction to use an ASD method, putting emphasis on processes and procedures that must be followed by these teams (*behavior control*) [22].

In a direct comparison with our two insurance industry cases, we found that both Unidevelop and Softac tend to use fewer practices that address formal control. There were certain guidelines regarding the applicability of agile practices, but in general the teams of all companies could also decide in part which practices they would like to use. So far, research has been able to note the promotion of both formal controls (such as behavior or outcome control) as well as informal controls (such as clan control and self-control) through agile practices [e.g., 16, 28, 38]. For example, the usage of the agile practice "user stories" can be seen as a formal control, as "they are a documented set of requirements (goals) to be achieved by development” [13].

Table 4: Control enactment concepts observed

Code	Apocorp	Dominsur	Unidevelop	Softac
FC	High	High	Moderate	Moderate
IC	Moderate	Moderate	High	Moderate
ECS	Moderate	High	High	High
ACS	Moderate	Low	Low	Low
CC	Moderate	High	High	High
EC	Low	High	High	High
LEGEND: FC = formal control, IC = informal control, ECS = enabling control style, ACS = authoritative control style, CC = communicational congruence, EC = evaluational congruence				

All of the practices introduced in table 2 have been used by the teams across all cases, however, it should be noted that in all cases an *enabling control style* was applied. That is, we were able to identify the two characteristics of an enabling control style, “repair” and “transparency”. However, we found that the degree of an enabling control style in case of Apocorp is significantly lower than in other cases, as we have even found evidence of an authoritative control style on closer examination, which is also due to a lack of a repair as well as transparency feature of an enabling style:

> *"The team is managed with a rather ‘strict hand’ as far as the method is concerned! Um...that means there is less need-oriented adaptation of the process model”*

A similar scenario can be observed when looking at the concept of control congruence. Although in all cases there is a common understanding between the controller and the controllee (*communicational congruence*), Apocorp indicates a deficit in the appropriateness of some control mechanisms (*evaluational congruence*). For example, most of the interviewees of Apocorp observed or reported “resistances” within the team regarding the mandatory usage of agile practices:

> *"...oh God, not a retro again, it eats time, it eats capacity, I can't go on working then and really don't see the benefit.”*

While the identification of specific control enactment concepts is important, the more substantial question is how these concepts relate to each other and how this influences team mechanisms such as team autonomy or team performance. Table 5 summarizes to what extent we found evidence of how control influences team autonomy and team performance. Informants of Unidevelop and Softac mostly reported that they already feel autonomous within their teams. For example:

> *“The team itself has also been given a great deal of freedom from the management level. This means that from the very beginning it was up to the team to develop (software) what they thought was the right thing to do.”*

In contrast, the informants of Apocorp and Dominsur felt somehow restricted in their daily working routines:

> *“Well, I think they could be more autonomous and free, but they don't use it.” (Apocorp)*

> *“The degree of flexibility we have here helps. And I say 20 percent more flexibility, I think would help even more.” (Dominsur)*

From a control mode perspective, we found evidence across all cases that formal control is seen to have a positive effect on team performance.

> *“You need a certain amount of control to be able to keep the whole process under control and assess the process. Especially when it comes to meeting deadlines. Improving quality may also be another example. You must have a healthy level of control and freedom” (Unidevelop)*

Similarly, we found – compared to formal control – a slightly weakened evidence for having informal control positively influencing team performance:

> *“Self-organization promotes motivation, communication and success (of a team)” (Dominsur)*

> *“Yeah, that's for sure. That's why we as a team decided*

Table 5: Relations between control enactment concepts and agile teams

Code group	Codes	Apocorp	Dominsur	Unidevelop	Softac
Enabling control style (ECS)	…increases control congruence (P3c)	(x)	X	X	(x)
Team Autonomy (TA)	…is increased by informal control (P1a)		(x)	(x)	(x)
	…is decreased by formal control (P2a)		(x)	(x)	(x)
	…is increased by an enabling control style (P3a)	X	X	X	X
Team Performance (TP)	…is increased by informal control (P1b)	(x)	X	X	(x)
	…is increased by formal control (P2b)	(x)	X	X	X
	…is increased by an enabling control style (P3b)	X	X	X	X
	…is increased by communicational congruence (P4)		X	X	X
	…is increased by evaluational congruence (P4)	(x)	X	X	
LEGEND: X marks a clearly and frequently identified code, while (x) marks a less clearly identified code					

> *back then that we would control all the pull requests from someone else, which means that another pair of eyes would look over it." (Unidevelop)*

The effects of formal and informal control modes on team autonomy, however, could only be identified with few and mostly less clearly codes. Enabling control styles, on the other hand, could clearly be identified to positively influence both, team autonomy and team performance. For example:

> *"This means that from the very beginning it was up to the team to develop what they thought was the right thing to do. This means that if we think that something is somehow beneficial, then we don't have to ask anybody, we can simply implement it. In the sense that we have complete freedom and as long as the result is right, everything is good." (Unidevelop)*

> *"Now, we have even (as people in charge) consciously taken back some of ourselves and have simply tried to rely on the self-healing powers and self-responsibility of the team, to simply try it out. That actually worked quite well!" (Dominsur)*

Finally, we see support in three of four cases regarding the positive influence of control congruence on team performance. Regarding a shared understanding of controls a developer of Dominsur argues:

> *"That we still somehow speak a uniform language and not everyone else advises us in the team. Therefore, I would say a bit of a success factor, that it is important that we find a common line, that we develop common views on things [...] that is just somehow important."*

Combining all statements, we were able to support some of our propositions based on the four cases. While much support was given to propositions P1b, P2b, P3a, P3b, P3c and P4, we found only less evidence for support of propositions P1a and P2a.

6. Discussion

Building upon our pre-defined research question, the main goal of this research project was to shed light upon the question of how to control ASD project teams, taking into account the extended control empowerment concept, and explaining the impact on project teams in terms of team autonomy and team performance in ASD. Based on our results, we were generally able to provide answers to our research question and enhanced our knowledge on control in ASD teams from both a theoretical as well as practical point of view.

First, although the influence of formal and informal controls through agile practices on team autonomy remains obscured, we can state that such controls enabled by agile practices have a positive impact on team performance. This may be due to the fact that our case observations found only moderate levels of informal controls (except Unidevelop), which are said to provide high levels of autonomy in managing assigned work tasks [e.g., 23, 42]. In the case of Apocorp and Dominsur, these moderate levels of informal control can be explained by both firms still being in the process of an agile transformation and and adoption. Thus, both are still largely characterized by hierarchies, structure, and formal processes. Only Unidevelop, as a young company, seems to rely entirely on informal mechanisms. Nevertheless, the positive influence of control on team performance is in line with the results reported in the literature. Regarding informal controls, the use of self-control allows team members to align their resources and to choose methods for goal achievement without involvement of the project leader [17, 28], the use of clan control establishes an environment where the controllee has freedom to make use of her own skills and knowledge in order to accomplish certain tasks, leading to better team performance [4, 11]. Formal controls, on the other hand, provide some degree of guidance and structure, which supports the execution of tasks and leads to better team performance [42].

Second, control styles seem to play an important role in the remission of control portfolios and have a significant impact on ASD project teams. We see two implications in those cases where both characteristics of an enabling control style (i.e., repair and transparency) have been clearly identified: (a) a frequent presence of an enabling control style reduces the likelihood of an authoritative control style, and (b) an enabling control style promotes a shared understanding (communicational congruence) and an increased perceived appropriateness (evaluational congruence) of the controls enacted [33, 35]. While (a) can be explained by the fact that both control styles are two endpoints of a continuum [e.g., 1], (b) needs a closer look.

We presume that both characteristics of an enabling control style generally have a positive influence on control congruence. The repair characteristic, on the one hand, may contribute to a generally better understanding, especially of the controls enacted, done through the establishment of regular feedback mechanisms [12]. On the other hand, the transparency feature of an enabling control style provides the "big picture" [52], which in turn might lead to both an increased shared understanding of the rationale of controls and increased perceived appropriateness of controls.

Third, we argue the concept of control congruence to be important when control is exercised within ASD project teams. Our results show that in three of our four cases, a high level of control congruence had a positive impact on team performance. This is also consistent with the results of recent studies, which indicate control congruence to positively influence team performance, as it contributes significantly to the quality of the controls adopted and avoids negative socio-emotional effects such as decreased job satisfaction [35, 50].

The main limitation of our study lies in its – by design – limited research method. We therefore call for replication of our study in different contexts, with organizations of different sizes, industries, countries, and overall agility. While our qualitative method enabled us to go into more detail and explicitly deal with context, this also limits the reliability of our findings to a certain extent. By including quantitative methods and by replicating our study with a quantitative or mixed methods approach, future research could further improve the reliability of our findings. Another limitation lies in the selection of participants. While all major roles of each team were interviewed, we did not conduct interviews with each and every team member. It is likely that perceptions of controls, styles, or congruence varies. The final limitation is the influence of social desirability bias, as it is generally more socially desirable to report success rather than failure. We tried to minimize the social desirability bias emerging from our questions. However, due to the clear favor of success over failure, social desirability bias was still likely to emerge from questions during our interviews.

7. Conclusion

In this paper, we explained the interplay between control modes, control styles and control congruence and the resulting influence on autonomy and performance within ASD teams. We gave an overview over the findings of recent decade’s research on control in ASD and conducted qualitative research across four cases from two different industries. Further, we discussed implications for both theory and practice. Limitations were discussed as well as avenues for future research to further improve agile.

8. References

[1] Adler, P.S., Borys, B.: Two types of bureaucracy: Enabling and coercive. Administrative Science Quarterly 41, (1996), pp. 61

[2] Beck, K., et al.: Manifesto for Agile Software Development. (2001) http://www.agilemanifesto.org

[3] Choudhury, V., Sabherwal, R.: Portfolios of control in outsourced software development projects. Information Systems Research 14, (2003),

[4] Chua, C.E.H., et al.: Enacting Clan Control in Complex Projects: A Social Capital Perspective. MIS Quarterly 36, (2012), pp. 577-601

[5] Conboy, K., Fitzgerald, B.: Agile Drivers, Capabilities, and Value: An Over-Arching Assessment Framework for Systems Development. In: Desouza, K.C. (ed.) Agile Information Systems: Conceptualization, Construction, and Management, pp. 207-221. Elsevier, Massachusetts, USA (2007)

[6] Cram, W.A., Brohman, M.K.: Beyond Modes: a New Typology of ISD Control. In: Internation Conference on Information Systems (ICIS), (2010)

[7] Cram, W.A., Brohman, M.K.: Controlling information systems development: a new typology for an evolving field. Information Systems Journal 23, (2013),

[8] Cram, W.A., et al.: Information Systems Control: A Review and Framework for Emerging Information Systems Processes. Journal of the Association for Information Systems 17, (2016), pp. 216

[9] Gerwin, D., Moffat, L.: Withdrawal of team autonomy during concurrent engineering. Management Science 43, (1997), pp. 1275-1287

[10] Gomber, P., et al.: On the Fintech Revolution: Interpreting the Forces of Innovation, Disruption, and Transformation in Financial Services. Journal of Management Information Systems 35, (2018), pp. 220-265

[11] Gopal, A., Gosain, S.: The Role of Organizational Controls and Boundary Spanning in Software Development Outsourcing: Implications for Project Performance. Information Systems Research 21, (2010), pp. 960-983

[12] Gregory, R.W., et al.: Control Balancing in Information Systems Development Offshoring Projects. MIS Quarterly 37, (2013), pp. 1211-1237

[13] Gregory, T., et al.: Chains of Control in Agile Software Development. AMCIS 2013 Proceedings, (2013)

[14] Hackman, J.R.: The design of work teams. In: Lorsch, J.W. (ed.) Handbook of organizational behavior, pp. 315-342. Prentice Hall, Englewood Cliffs, NJ (1987)

[15] Harris, M.L., et al.: Control of Flexible Software Development Under Uncertainty. Information Systems Research 20, (2009), pp. 400-420

[16] Harris, M.L., et al.: Controls in Flexible Software Development. Communications of the Association for Information Systems 24, (2009),

[17] Henderson, J.C., Lee, S.: Managing I/S Design Teams: A Control Theories Perspective. Management Science 38, (1992), pp. 757-777

[18] Highsmith, J., et al.: Agile Software Development: The Business of Innovation. Computer 34, (2001), pp. 120-123

[19] Hummel, M., et al.: The Role of Social Agile Practices for Direct and Indirect Communication in Information Systems Development Teams. Communications of the Association for Information Systems 36, (2015),

[20] Jaworski, B.J.: Toward a Theory of Marketing Control: Environmental Context, Control Types, and Consequences. Journal of Marketing 52, (1988), pp. 23-40

[21] Kirsch, L.J.: The management of complex tasks in organizations: Controlling the systems development process. Organization Science 7, (1996), pp. 1-21

[22] Kirsch, L.J.: Portfolios of Control Modes and IS Project Management. Information Systems Research 8, (1997), pp. 215-240

[23] Kirsch, L.J., et al.: Controlling Information Systems Development Projects: The View from the Client. Management Science 48, (2002), pp. 484-499

[24] Langfred, C.W.: Too much of a good thing? Negative effects of high trust and individual autonomy in self-managing teams. Academy of management journal 47, (2004), pp. 385-399

[25] Larman, C.: Agile and Iterative Development: A Manager's Guide. Pearson Education (2003)

[26] Lee, G., Xia, W.: Toward Agile: An Integrated Analysis of Qualitative and Quantitative Field Data on Software Development Agility. MIS Quarterly 34, (2010), pp. 87-115

[27] Mahadevan, L., et al.: Running on Hybrid: Control Changes when Introducing an Agile Methodology in a Traditional "Waterfall" System Development Environment. Communications of the Association for Information Systems 36, (2015),

[28] Maruping, L.M., et al.: A Control Theory Perspective on Agile Methodology Use and Changing User Requirements. Information Systems Research 20, (2009), pp. 377-400

[29] Maruping, L.M., et al.: Role of collective ownership and coding standards in coordinating expertise in software project teams. European Journal of Information Systems 18, (2009), pp. 355-371

[30] McAvoy, J., Butler, T.: The role of project management in ineffective decision making within Agile software development projects. European Journal of Information Systems 18, (2009), pp. 372-383

[31] Miles, M.B., Huberman, A.M.: Qualitative data analysis : an expanded sourcebook. Sage Publications, Thousand Oaks (1994)

[32] Moe, N.B., et al.: Team Autonomy in Large-Scale Agile. In: Proceedings of the 52nd Hawaii International Conference on System Sciences, (2019)

[33] Murungi, D., et al.: Control and emotions: Understanding the dynamics of controllee behaviours in a health care information systems project. Information Systems Journal (2019), pp. 1-25

[34] Myers, M.D.: Qualitative research in business & management. SAGE, London (2013)

[35] Narayanaswamy, R., et al.: The Impact of Influence Tactics in Information System Development Projects: A Control-Loss Perspective. Journal of Management Information Systems 30, (2013), pp. 191

[36] Nelson, R.R.: Project retrospectives: Evaluating project success, failure, and everything in between. MIS Quarterly Executive 4, (2005), pp. 5

[37] Pelrine, J.: On Understanding Software Agility: A Social Complexity Point Of View. Emergence : Complexity and Organization 13, (2011), pp. 26-37

[38] Persson, J.S., et al.: Agile distributed software development: enacting control through media and context. Information Systems Journal 22, (2011), pp. 411-434

[39] Piccoli, G., et al.: Virtual teams: team control structure, work processes, and team effectiveness. Information Technology & People 17, (2004),

[40] Poppendieck, M., Poppendieck, T.: Lean Software Development: An Agile Toolkit. Addison-Wesley Longman, Amsterdam (2003)

[41] Recker, J., et al.: How agile practices impact customer responsiveness and development success : a field study. Project management journal. 48, (2017),

[42] Remus, U., et al.: Control Modes Versus Control Styles: Investigating ISD Project Control Effects at the Individual Level. ICIS 2016 Proceedings, (2016)

[43] Robey, D., et al.: Situated learning in cross-functional virtual teams. Technical Communication 47, (2000), pp. 51-66

[44] Saldaña, J.: The coding manual for qualitative researchers. SAGE, London (2016)

[45] Sarker, S., Sahay, S.: Understanding Virtual Team Development: An Interpretive Study. Journal of the Association for Information Systems 4, (2003),

[46] Schwaber, K.: Scrum Development Process. In: Conference on Object-Oriented Programming Systems, Languages, and Applications, pp. 117-134, (1995)

[47] Sharp, H., Robinson, H.: An Ethnographic Study of XP Practice. Empirical Software Engineering 9, (2004), pp. 353-375

[48] Tannenbaum, A.S.: Control in Organizations: Individual Adjustment and Organizational Performance. Administrative Science Quarterly 7, (1962), pp. 236-257

[49] Tiwana, A., Keil, M.: Control in Internal and Outsourced Software Projects. Journal of Management Information Systems 26, (2009), pp. 9-44

[50] Tripp, J.F., et al.: Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign. Journal of the Association for Information Systems 17, (2016), pp. 267-307

[51] Vidgen, R., Wang, X.: Coevolving Systems and the Organization of Agile Software Development. Information Systems Research 20, (2009), pp. 355-377

[52] Wiener, M., et al.: Control configuration and control enactment in information systems projects: review and expanded theoretical framework. MIS Quarterly 40, (2016), pp. 741-789

[53] Wiener, M., et al.: The View From the Top–How Senior Executives Exercise Control Over Information Systems Projects to Enhance Performance. European Conference in Information Systems (ECIS), pp. pp. 1423-1438, Guimarães, Portugal (2017)

[54] Yin, R.K.: Case Study Research: Design and Methods. SAGE Publications Inc., Thousand Oaks, CA, USA (2003)

[55] Yin, R.K.: Case Study Research: Design and Methods. SAGE Publications, Thousand Oaks, CA, USA et al. (2003)

[56] Zellmer-Bruhn, M., Gibson, C.: Multinational Organization Context: Implications for Team Learning and Performance. Academy of Management Journal 49, (2006), pp. 501-519