Do As You Want Or Do As You Are Told?
Control vs. Autonomy in Agile Software Development

Tim Dreesen
University of Cologne
dreesen@wiso.uni-koeln.de

Thomas Schmid
University of Cologne
tschmid1@smail.uni-koeln.de

Abstract

Agile Software Development (ASD) projects still draw the attention of the research community. Agile methodologies promise to increase an ASD team's agility in such a way, that these teams are able to respond and react to changing user requirements. Existing studies on flexibility and autonomy in ASD projects, however, imply that these projects potentially can benefit from different elements of control. Our objective is to improve the understanding of how to enact control through agile practices, and how these practices affect either formal or informal control in ASD teams. Based on an extensive literature review, our study (1) provides an overview of adequate control-enacting agile practices and (2) compares the results with our empirical findings, derived from qualitative data.

1. Introduction

In today's software development practice the capability of rapid response to changing user requirements "has become increasingly critical for software development performance" [34]. To address this crucial need, different agile software development (ASD) approaches have emerged during the 1990s and 2000s [34], for example, Scrum [56] or eXtreme Programming (XP) [5], and are widely used in corporate settings. Whereas each ASD methodology may differ in terms of key principles and practices, they all have in common that they emphasize the importance of project teams that are empowered to make decisions, while the project manager's role has become rather team-supportive than team-directive [40]. Thus, although originally designed for small teams, ASD approaches are nowadays used even by large organizations, which tend to use scaling methodologies such as Scrum of Scrums or Scaled Agile Framework [62].

Despite the popularity of ASD methodologies, projects using ASD still fail. For example, 94% of all organizations surveyed by a recent industry survey use ASD methodologies, but only half of them assess majority of their agile projects successful [62]. The most often mentioned reasons of project failure are a lack of experience regarding the use of agile methodologies (41%), a company philosophy or culture contrary to core agile values (46%), and missing management support (38%) [63]. Other studies come to similar results and conclude that agile projects have more or less the same fail rate today as in 2001 [47]. So despite proponents' view of ASD approaches, they are clearly not a "silver bullet" in and of themselves, overcoming long-known problems in software development [15, 16]. Because of the high popularity and still increasing use of ASD methodologies in practice and the notable number of unsuccessful projects, there is a need of identifying issues and proposing solutions to contribute to the enhancement of the success rate of ASD projects.

An often-mentioned trade-off that is seldom investigated may hold the key to answering this problem. It is known that a key factor of effectively managing any kind of software development project is controlling the development process and its results [30, 48, 68]. ASD, however, is characterized by autonomously working teams, where this autonomy on the one hand enables them to respond to change but on the other hand, can be detrimental to the development process, for instance, when teams lose themselves in arguing how to tackle a problem rather than solving it [37]. Acknowledging this apparent conflict between control and autonomy, and taking into account that ASD projects can and do fail [47, 62], the question is in how far control and structure are needed in ASD projects, and how they can be applied with respect to the core principles of agile methodologies, especially to empower teams in decision making [40]. Only limited guidance exists on how ASD teams should be governed, especially in regards to the relationship between control and autonomy [37].

URI: http://hdl.handle.net/10125/50495
ISBN: 978-0-9981331-1-9

The goal of this research is to analyze common agile practices in ASD projects and, especially, to identify their impact on control and autonomy within ASD project teams. We agree with Wiener, Mähring, Remus and Saunders [68] that more research is needed on control enactment in IS. In this review, we focus on a specific project context, that is ASD. Hence the following research question guides our study: *"How can control be enacted in ASD projects through specific agile practices and how do they affect different types of control (i.e., formal and informal control) within an ASD team?"*.

To answer our research question, we conducted a structured and comparative literature review on control enacting practices within ASD projects, based on the guidelines of Webster and Watson [66] and Levy and Ellis [35]. We analyzed the existing literature on ASD projects and identified a total set of 29 control enacting practices related to particular control modes. To empirically validate the literature review's results, we investigated agile practice usage and their impact on control and autonomy within 8 different ASD student teams by conducting semi-structured interviews. Based on the review's results and on our qualitative findings, we conducted an in-depth comparison of these practices concerning their suitability to enact control. The result of our study is a comprehensive summary of control enacting practices suitable for ASD projects.

The remainder of the paper is structured as follows. The next section provides information on the theoretical background, specifically on control theory, which serves us a theoretical lens, and the relation of control to ASD approaches. Section three introduces our research design with a description of the literature review as well as our data collection and analysis approach. Section four explains the results of our research with a focus on comparing control enacting practices and their impact on formal and informal control according to control theory. Section five summarizes our findings, explains the limitations of the study, and provides guidance for future research. Finally, section six provides a brief conclusion.

2. Theoretical Background

ASD is not only a technical process, but a social process as well [3, 26, 50, 54, 55]. This is why ASD project leaders must choose appropriate methods for managing both [37]. An important aspect of the management process is the function of control [30]. Following Tannenbaum [58], we define control in a broader way "to refer to any process in which a person or group of persons or organization of persons determines, that is, intentionally affects what another person or group or organization will do [58]. We primarily rely on control theory by Kirsch [29, 30, 32], which serves us as a theoretical lens. Although particular ASD methodologies are not specifically addressed within control theory [10], Kirsch points out that organizations in dynamic, changing environments may change control approaches through an ASD project's lifecycle, resulting in the implementation of appropriate control types [29, 30]. Theory distinguishes formal control types such as input, behavior and outcome control from informal control types such as self-control and clan control as relevant to ASD teams [29]. Table 1 summarizes key control modes, which often are exercised in concert rather than in isolation, representing a so-called control portfolio [30].

Table 1: Summary of control modes following Kirsch [29] & Jaworski [27]

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 the development team in task execution [31, 53]. It is well known that traditional software development (SD) approaches rely heavily on formal control mechanisms [29-31, 46, 60]. By contrast, informal control potentially provides developers with discretion regarding how tasks are accomplished [23, 31, 37]. Generally, ASD methodologies rely more on informal controls rather than traditional formal controls [12]. Informal controls such as clan and self-control promise to enact autonomy, which is seen as an important antecedent for development teams being able to respond to changing user requirements [17, 37]. The exercise of clan control allows the team to identify important project goals and to determine how to attain them on their own [37]. The establishment of self-control is similar, but focusses on the individual instead on a group of individuals. Self-control defines “the extent to which an individual exercises freedom or autonomy to determine both what actions are required and how to execute these activities” [23].

ASD approaches view team autonomy as one of the essentials that affects agility [33, 34]. Prior literature provides various definitions of team autonomy and other closely related terms, for example, self-organization [9, 24], self-management [57], or team empowerment [33, 39]. Following Lee and Xia [34], 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 [34]. Thus, ASD approaches are often seen as a counter-balance to the more rigid, formal, and structured SD approaches [6].

Next to team autonomy, the enactment of control is closely linked to the establishment of task performance, which is defined as the degree to which a team achieves its goals and how well its outputs match the team’s mission [20, 72]. Although we find several empirical studies that analyze the direct effect of control and team task performance on ASD project outcomes such as product quality [18, 21, 36, 37, 51], results still remain ambiguous [11]. For example, in terms of product quality Maruping, Venkatesh and Agarwal [37] suggest that ASD project teams can benefit from the implementation of control modes, especially formal outcome control, to create an environment in which agile practices can provide autonomy whilst at the same time clear performance goals and structures exist. On the other hand, Harris, Collins and Hevner [21] propose emergent outcome control as a new concept to achieve a better product-market match, as they argue formal outcome control to be insufficient in agile environments. Emergent outcome control therefore uses scope boundaries and ongoing feedback to “define the allowable space for exploration” and “check on decision as they are made throughout the development process” [22]. Regarding informal controls, Cram, Brohman and Gallupe [12] argue that little research has investigated informal controls such as clan and self-control and their effects on outcomes (e.g., software product quality). This matches some of the findings of Wiener, Mähring, Remus and Saunders [68] who showed that earlier studies on control in IS produced inconclusive and partly contradictory results. For example, there is no consensus if informal control has a positive [23, 67] or negative impact [60, 61] on project outcomes.

3. Research Design and Method

In line with our overarching research question *“How can control be enacted in ASD projects through specific agile practices and how do they affect different types of control (i.e., formal and informal control) within an ASD team?”*), our project followed a three-step data analysis approach (see Figure 1).

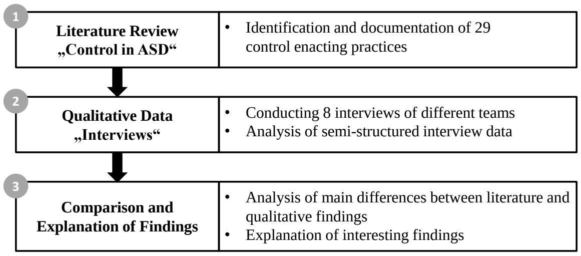
1
Literature Review
„Control in ASD“
Identification and documentation of 29 control enacting practices
2
Qualitative Data
„Interviews“
Conducting 8 interviews of different teams
Analysis of semi-structured interview data
3
Comparison and
Explanation of Findings
Analysis of main differences between literature and qualitative findings
Explanation of interesting findings

Figure 1: Analysis approach

First, we conducted a concept-driven and systematic literature review based on the approaches of Levy and Ellis as well as Webster and Watson [35, 66]. The review started with a keyword search on control within ASD projects in general and control enacting agile practices in ASD projects in particular, followed by a backward and forward search. To achieve high quality results, only journals and conference articles listed in the top MIS journals and conferences ranking provided by the VHB (http://vhbonline.org/vhb4you/jourqual/vhb-jourqual-3/teilrating-wi/) were used. We defined a single search string for the keyword search (see Figure 2) to identify relevant articles in databases like EBSCOhost, INFORMS or ProQuest. There was no restriction for the publishing year of the articles. All search results were examined regarding title, abstract, and keywords. Within the resulting set of papers, we further identified relevant articles for our project purpose (“in scope”) and dropped the others (“not in scope”). We subsequently proceeded with a reference, author and keyword backward search. Finally, a reference and author forward search identified our final set of articles for the data analysis phase. In total, our final set of articles consists of 28 articles on control in an agile environment. A brief summary of our literature search process can be found in Figure 2.

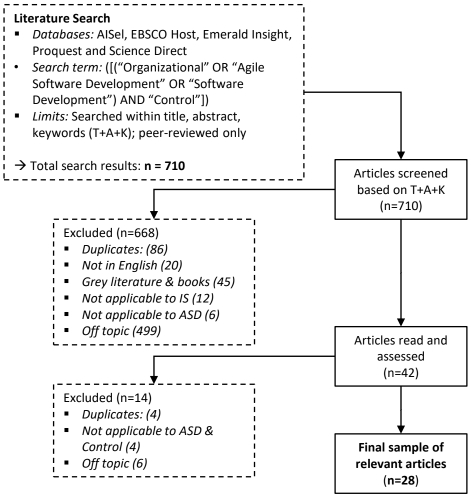
Literature Search
Databases: AISel, EBSCO Host, Emerald Insight, Proquest and Science Direct
Search term: ([("Organizational" OR "Agile Software Development" OR "Software Development") AND "Control"])
Limits: Searched within title, abstract, keywords (T+A+K); peer-reviewed only
→ Total search results: n = 710
Articles screened based on T+A+K (n=710)
Excluded (n=668)
Duplicates: (86)
Not in English (20)
Grey literature & books (45)
Not applicable to IS (12)
Not applicable to ASD (6)
Off topic (499)
Articles read and assessed (n=42)
Excluded (n=14)
Duplicates: (4)
Not applicable to ASD & Control (4)
Off topic (6)
Final sample of relevant articles (n=28)

Figure 2: Literature search process

Second, as part of our research design, we wanted to evaluate whether our findings of step 1 can be applied to practice by conducting semi-structured, one-to-one interviews with team members of 8 different development projects. One-to-one interviews allow gathering of rich data from people in different roles [45]. Furthermore, semi-structured interviews involve use of pre-formulated questions but allow improvisation for emerging topics during conversation. Each interview is based on an interview guide [71]. All teams consist solely of students, participating in development projects with different industry partners. All development teams made use of the agile methodology *Scrum*. Objective data such as logs, project schedules, code repositories have been accessed and analyzed as well as field observations were conducted.

The results of the first and second step are set down in two tables, describing our findings of the literature review as well as from our collected qualitative data. We used a concept matrix that is based on several categories to structure the presentation of the results. The approach allowed us to differentiate between practices that enable different types of control (or control modes). Based on the concept matrix as well as both result tables, we were able to perform step 3 in order to identify major findings and insights.

4. Results

4.1. Control in Agile Software Development

The literature revealed 29 associations between agile practices and the defined control modes (Table 2). Due to space restrictions, a complete set of literature references has been neglected but is available from the authors on request. The associated control modes are based on control theory by Kirsch (e.g. [29, 30, 32]) containing formal outcome- and behavior-control as well as informal self- and clan-control.

Table 2: Summary of agile practices and associated control modes in the literature

No.	Practice	Control Modes		#	References
1	Acceptance Testing	Formal	BC OC EOC	2	[22, 52]
2	Backlog prioritization / estimation	Formal	BC OC EOC	4	[11, 22, 36, 42]
3	Book clubs	Formal	BC	1	[19]
		Informal	SC	1	[19]
4	Burndown Chart	Formal	OC	4	[19, 21, 36, 42]
		Informal	CC	1	[19]
5	Code Review / Refactoring	Formal	BC OC EOC	3	[21, 51, 52]
		Informal	SC CC	2	[19, 51]
6	Coding Standards	Formal	OC	1	[70]
		Informal	CC	1	[37]
7	Collective Code Ownership	Informal	SC CC	4	[38, 51], [7, 14]
8	Continuous Integration	Formal	BC EOC	2	[21, 22]
		Informal	CC	1	[22]
9	Co-location of Team Members	Formal	EOC	1	[21]
10	Daily Stand-up	Formal	BC OC	2	[11, 44]
		Informal	SC CC	8	[2, 25, 36, 41, 43, 44, 59, 64]
11	Defect Reporting	Formal	OC	2	[11, 19]
		Informal	SC CC	1	[19]
12	Energized Work	Formal	BC	1	[22]
		Informal	CC	1	[22]
13	Incremental Design	Formal	BC EOC	1	[22]
14	Iterative Development	Formal	BC OC EOC	5	[18, 22, 28, 44, 52]

No.	Practice	Control Modes		#	References
15	Iteration Planning	Formal	BC	2	[36, 51]
		Informal	SC CC	6	[2, 36, 41, 43, 59, 64]
16	Iteration Retrospective	Formal	BC OC	2	[19, 36, 70]
		Informal	SC CC	5	[1, 36, 43, 59, 64]
17	Iteration Review	Informal	CC	1	[41]
18	Release Planning	Formal	OC	1	[36]
19	On-Site Customer	Formal	BC	2	[8, 18]
		Informal	CC	2	[18, 51]
20	Open Workspace	Formal	BC OC	1	[22]
		Informal	SC	2	[22, 64]
21	Pair Programming	Formal	BC EOC	1	[22]
		Informal	SC CC	4	[22, 37, 44, 70]
22	Planning Game	Formal	BC OC	1	[51]
		Informal	SC CC	1	[51]
23	Practice Guides	Formal	BC OC	1	[19]
24	Sit Together	Formal	EOC	1	[22]
		Informal	CC	1	[22]
25	Slack	Formal	BC EOC	1	[22]
26	Sustainable Pace	Informal	SC	2	[64, 69]
27	Unit Tests	Formal	OC	2	[19, 37]
		Informal	SC CC	1	[19]
28	User Stories	Formal	OC	4	[19, 22, 36, 52]
29	Whole Team	Formal	EOC	1	[22]
		Informal	CC	1	[22]

LEGEND: Control Modes: BC = Behavioral Control, CC = Clan Control, EOC = Emergent Outcome Control, OC = Outcome Control, SC = Self-Control

The results are not limited to a distinct agile methodology; thus, they comprise practices for methodologies like Scrum or XP. From a control mode perspective, we identified 17 practices affecting behavior control and clan control, followed by 15 practices that are suitable to enable outcome control. Emergent outcome control can be enacted through 11 of our identified practices, whereas only 12 practices are said to support self-control in ASD teams (Table 3). We found no evidence in literature regarding practices that might affect input control.

Table 3: Practices per control mode

Control Mode		Practices	#
Formal	*Input Control*	*None*	*0*
	Behavior control	1,2,3,5,8,10,12,13,14,15,16,19,20,21,22,23,25	17
	Outcome control	1,2,4,5,6,10,11,14,16,18,20,22,23,27,28	15
Informal	*Clan control*	4,5,6,7,8,10,11,12,15,16,17,19,21,22,24,27,29	17
	Self-control	3,5,7,10,11,15,16,20,21,22,26,27	12

Table 4 displays the results of the semi-structured interviews. We focused on an overall amount of eight distinct practices within qualitative data collection, as they imply to have effects on different types of control. These practices were chosen for two reasons: (1) the selected practices are supported by literature to enact different control modes and (2) the selected practices cover a broad range of control modes according to control theory [8, 30]. Consequently, we focused on practices of Scrum, XP, and custom hybrid approaches as they represent more than two-thirds of agile methodologies used in software projects [63].

Table 4: Agile practices associated to control modes based on empirical data

No.	Agile Practice	Control Mode		#	FREQ.
1	User stories	Formal	BC, OC	7	5,71
2	Iteration Retrospective	Formal	BC	2	4,28
		Informal	CC	5	
3	Burndown charts	Formal	BC, OC	4	4,14
		Informal	CC	3	
4	Pair programming	Informal	CC	7	4
5	Backlog prioritization	Formal	BC, OC	4	3,85
		Informal	CC	3	
6	Code reviews	Informal	CC	7	3,71
7	Daily standups	Formal	BC	2	3

		Informal	CC, SC	5	
8	Collective code ownership	Informal	CC	7	2,85

LEGEND: Control Modes: BC = Behavioral Control, CC = Clan Control, EOC = Emergent Outcome Control, OC = Outcome Control, SC = Self-Control; Freq.: frequency of usage (6 is high)

4.2. Comparison of Findings in Literature and Qualitative Data

Initially, we did not expect a high amount of control enacting agile practices specific to and dedicated for ASD. Within our total set of 29 practices, however, practices that focus on enabling formal control modes outnumber the overall amount of identified informal control enacting practices in ASD. The most frequently reported practices related to a particular type of formal control in ASD are, ordered by matches in literature, *iterative development*, *backlog prioritization/estimation*, *burndown chart* and *user stories*. Whereas *iterative development* and *backlog prioritization* seem to be suitable to enact behavior as well as outcome control, the usage of *burndown charts* and *user stories* in ASD are said to be applying outcome control only. On the other hand, the most frequently reported practices related to a specific type of informal control in ASD are *daily stand-up*, *iteration planning*, *iteration retrospective*, *pair programming* and *collective code ownership*. All these practices are suitable to foster both types of informal control, clan control as well as self-control.

Based upon the interview's results, all of the 8 agile practices could be assigned to control modes according to control theory. Only two practices could be assigned clearly, while the others were related more unambiguously. *Code reviews* and *pair programming* were both assigned to clan-control only. *User stories, retrospectives* and *collective code ownership* were said to support two different control modes. *Backlog prioritization, burndown charts* and *daily standups* even were associated to three different control modes. According to the results, self-control could only be enacted through *daily standups*, while almost every practice but *user stories* enact clan-control. 6 out of 8 practices were said to support behavior or outcome-control.

Table 4 also shows the frequency of usage of the same agile practices. The interviewees were told to rank agile practices on their frequency of usage inside the project they worked on. They could decide between a "0" that represents a non-existent usage or a scale from "1" to "6" with "1" representing the minimal level and "6" the maximal level of usage.

User stories, *retrospective* and *burndown charts* were used more frequently with a ranking between 4 and 5,71. A reason for the frequent usage of user stories is explained in the following quote.

> *"[…] the creation of user stories worked out quite well. Especially used for the initial planning to understand the whole requirements. What do they wish for and how will those requirements be developed? This was some kind of help for the whole team to understand what needs to be delivered in the future."*

It is noticeable that *user stories* were used by far the most with a frequent usage of 5,71 while *collective code ownership* got with 2,85 the lowest usage frequency. In contrast, *iteration retrospectives* ranked as the second important factor got a ranking of 4,28. Table 5 summarizes the overlapping and partially different results of step 1 and 2. With a focus on formal control, the review's results revealed that 23 out of 29 agile practices can be used to enact formal control. Similar results reflect our qualitative findings, 6 out of 8 practices are associated with formal control.

Table 5: Comparison of Control Mode Results

No.	Agile Practice	Interview results	SLR results
1	User stories	BC, OC	OC
2	Iteration Retrospective	BC, CC	BC, OC, CC, SC
3	Burndown charts	BC, OC, CC	OC, CC
4	Pair programming	CC	BC, CC, SC
5	Backlog prioritization	BC, OC, CC	BC, OC
6	Code reviews	CC	BC, OC, CC, SC
7	Daily standups	BC, CC, SC	BC, OC, CC, SC
8	Collective code ownership	BC, CC	CC, SC

LEGEND: Control Modes: BC = Behavioral Control, CC = Clan Control, OC = Outcome Control, SC = Self-Control

5. Discussion

5.1. Summary of Findings and Implications

Building upon our pre-defined research question, the main goal of this research project was the literature-based identification and empirical evaluation of suitable control practices for ASD. Based on the results described in Section 4, we were generally able to provide answers to our research question and enhanced our knowledge on control in ASD projects from both a theoretical as well as practical point of view:

(1) *Providing future research directions for control-enactment and the effect on structure (formal control) and autonomy (informal control) in ASD teams.* Despite the known importance of control on the quality of SD project outcomes [18, 21, 36, 37, 51], there is so far no focused literature review that sheds light upon the question how far control and structure are needed in ASD projects, and how they can be applied through agile practices, while providing team autonomy at the same time. Our study closes this gap by providing detailed results derived from our three-step research approach as well as future research directions based on the existing research on ASD teams.

Building upon our work, and especially based on the differentiation of formal and informal control in ASD, we are able to extend our understanding on how ASD teams can be governed, especially in regards to the relationship between control and autonomy. Our list of agile practices and their impact on particular control modes revealed several interesting findings related to the topic of control usage in such projects.

We identified within our review's results a set of 23 agile practices that can be linked with the enactment of formal control types such as outcome or behavior control. In contrast, we found only 20 agile practices suitable for fostering informal control types such as clan control and self-control. Whereas 12 practices are dedicated to formal control types, there are 3 practices that affect informal control types only. This is surprising, since the underlying principles of agile methodologies (e.g. team autonomy) resemble more informal control types e.g. self-control. Following the Agile Manifesto, principles like "The best architectures, requirements, and designs emerge from self-organizing teams" or "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done" [4] provide evidence that informal control types are seen as much more important compared to the more formal and rigid control types like outcome control to agile methodologies. In contrast, we found out that the usage of a certain sets of common agile practices, which can be seen as the method-in-action [13, 65], potentially enacts high amounts of formal control within an ASD project. This leads us to conclude, that the enactment of formal control, and thus, structure within ASD teams, is necessary as it acts as an important counter-balance to team autonomy. The following quote focuses on enacted formal control and how it can help to improve the overall project outcome. This can help to get a better understanding for the overall need of formal control in ASD projects.

> "We used daily standups as our meetings so we can discuss the progress of the project. Because we strictly performed these daily standups it was some kind of behavior controlling since everybody knows what you've done and what issues you are dealing with."

Moreover, we have recognized a lack of practices concerning the enactment of informal control types such as clan- and self-control. Although a lot of studies agree on the importance of team autonomy [33, 34] or team empowerment in decision making [40], our knowledge remains scarce about how to establish these principles in ASD teams. Our study provides first insights, that specific practices are well-suited to enact informal control. Especially the practice *daily stand-up*, having in sum most matches in literature, seems a very common enabler:

> "We are also clan-controlled. We try to see each other every day and do the daily standups. It makes me think if I did not see my team today, I need to call them tonight and show them what I did and didn't work on today. "

Summing up, our research project revealed that, despite our general knowledge on suitable control-enacting practices for ASD, the exact relationship between the governance of control and autonomy within ASD teams and ASD project success is still unknown. Hence, we would recommend to increase the IS communities' research endeavor on this important topic. This could be done for example by an evaluation of control within ASD projects based on in-depth case study research. By applying such research methods in this context, we could further increase our understanding of how to implement the right kind of control within ASD projects.

(2) *Providing a first overview of control-enacting practices for ASD projects in practice*. As already mentioned beforehand, our knowledge on suitable control-enacting practices for ASD projects in practice remains scarce (see Section 1). Our study is, by certain means, able to cover this gap by providing a first overview of suitable practices in terms of exercising different types of controls. This list of practices, including references, allows practitioners working in ASD projects to evaluate the existing practices for general suitability and implementation fit within their projects. Hence, based on our list, we are able to provide first insights for practical application, which need to be amended by future research projects on this topic (e.g. in-depth empirical

analysis of particular control practice suitability within different ASD projects settings).

5.2. Limitations and Future Research

While we were able to provide sufficient answers to our research question and enhance our knowledge on control in ASD projects, there are some limitations and corresponding future research directions that need to be acknowledged.

First, our research project considered relevant journals in the IS domain (based on the recommendations of the AIS and VHB) only. We did not take into account outlets, which focus for example on organizational control (in general) or cross-cultural studies. Hence, we cannot guarantee a complete analysis of the reference literature within our review. Nevertheless, due to the fact that ASD projects in particular are a phenomenon in the field of IS, we are quite sure, that our results are generalizable to a certain extend. However, we would recommend further literature reviews on this topic to even increase the coverage of the existing research on this topic.

Second, we need to address the topic of the broad perspective on *control* as a limitation of our research. By starting our literature review with a keyword search and also by following the guidelines of Levy, Ellis and Webster, Watson [35, 66] in regards to forward and backward search, we tried to incorporate all past studies. Nevertheless, within the data analysis, we partially identified incongruity of different control mode definitions. While, for instance, Harris et al. [21, 22] focus on the concept of *emergent outcome-control* as an alternate view on outcome-control in general, others still focus on the traditional outcome-control perspective closely related to classical control theory (e.g. [30, 49]). The different associations result in a lack of transparency on the overall associations between agile practices and control modes. Based on this limitation, we would recommend further research, which explicitly focus on the comparison of control modes according to control theory in the light of suitable control enacting practices.

Third, one important limitation is the lack of experience regarding agile methodology use and strict role definition of all interviewees. A clearly defined role interpretation is fundamental for the usage of agile methodologies. The following quote provides an example of an interviewee's comment that supports this argument. In particular, the comment highlights weaknesses in the team-design which, in turn, leads to an emphasized development-mentality across all team members.

> *"I am not the scrum master. We are all part of the development team, even the scrum master. We do have a scrum master but everyone including the scrum master is also a developer and thus, responsible for creating and delivering working software every day [...]"*

Furthermore, all the interviews were conducted with students, this means they generally lack experience compared to common employees working on an agile project. For example, 6 out of 8 interviewees worked on a project of this size and using an agile methodology for the first time. Thus, we recommend to extend future qualitative research to a wider field, comprising team participants such as senior developers, managers or certified scrum-masters on both, client and vendor site.

6. Conclusion

Uncertainty and changing user requirements in business and technology environments is ever-increasing. For companies, who want to stay competitive in SD, balancing control and autonomy to effectively deal with changing requirements has become an imperative, not an option. Given the complex relationships between control use and autonomy in ASD, project managers face difficult challenges in using control appropriately in ASD projects. While prior literature developed several frameworks to view control in ASD, little guidance is offered concerning which control modes are most efficient and how a control portfolio can be configured. Moreover, the body of knowledge lacks a comprehensive understanding on control enactment in general, e.g. how control and autonomy can be supported by utilizing agile practices. This research paper offers useful insights that are based on extant literature. Following Wiener et al. [68], our goal was to examine how to enact distinct types of controls through selected agile practices. The results suggest agile practices are able to potentially enact distinct types of control and thus, supports project manager to choose suitable practices for their project. The authors conclude that agile methodologies are most efficient, when combined with formal control rather than exclusively informal control, such as clan and self-control. Control and autonomy in ASD are often viewed as negatively correlated. However, this research suggests why ASD can be flexible and controlled at the same time.

7. Acknowledgements

The German Research Foundation (DFG) funded part of this study under record no. RO 3650/8-1.

8. References

[1] Babb, J., et al.: Embedding Reflection and Learning into Agile Software Development. IEEE Software 31, (2014),

[2] Babb, J., et al.: Embedding Reflection and Learning into Agile Software Development. IEEE Software 31, (2014), pp. 51-57

[3] Balijepally, V.: Collaborative Software Development in Agile Methodologies - Perspectives from Small Group Research. AMCIS 2005 Proceedings, (2005)

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

[5] Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change. Addison-Wesley Professional (2004)

[6] Berente, N., et al.: Rule Formation and Change in Information Systems Development: How Institutional Logics Shape ISD Practices and Processes. In: System Sciences (HICSS), 2015 48th Hawaii International Conference on, pp. 5104-5113.IEEE (2015)

[7] Cao, L., et al.: A framework for adapting agile development methodologies. European Journal of Information Systems 18, (2009), pp. 332-343

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

[9] Chow, T., Cao, D.-B.: A survey study of critical success factors in agile software projects. The Journal of Systems and Software 81, (2008),

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

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

[12] 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

[13] Fitzgerald, B.: The use of systems development methodologies in practice: a field study. Information Systems Journal 7, (1997), pp. 201-212

[14] Fitzgerald, B., et al.: Customising agile methods to software practices at Intel Shannon. European Journal of Information Systems 15, (2006),

[15] Fraser, S., Mancl, D.: No Silver Bullet: Software Engineering Reloaded. IEEE Software 25, (2008), pp. 91-94

[16] Fraser, S.D., et al.: "No silver bullet" reloaded: retrospective on "essence and accidents of software engineering". Companion to the 22nd ACM SIGPLAN conference on Object-oriented programming systems and applications companion. ACM, Montreal, Quebec, Canada (2007)

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

[18] Goh, J.C.-L., et al.: Developing the Agile IS Development Practices in Large-Scale IT Projects: The Trust-Mediated Organizational Controls and IT Project Team Capabilities Perspectives. Journal of the Association for Information Systems 14, (2013),

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

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

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

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

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

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

[25] Hoda, R., et al.: Self-Organizing Roles on Agile Software Development Teams. IEEE Transactions on Software Engineering 39, (2013), pp. 422-445

[26] 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),

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

[28] Karlström, D., Runeson, P.: Combining Agile Methods with Stage-Gate Project Management. IEEE Software 22, (2005), pp. 43-50

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

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

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

[32] Kirsch, L.J.: Deploying Common Systems Globally: The Dynamics of Control. Information Systems Research 15, (2004), pp. 374-396

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

[34] 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

[35] Levy, Y., Ellis, T.J.: A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. Informing Science 9, (2006), pp. 181-212

[36] 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),

[37] 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

[38] 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

[39] Maruping, L.M., Magni, M.: What's the Weather Like? The Effect of Team Learning Climate, Empowerment Climate, and Gender on Individuals' Technology Exploration and Use. Journal of Management Information Systems 29, (2012), pp. 79-115

[40] 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

[41] McHugh, O.: A Study of Clan Control in Agile Software Development Teams. (2011)

[42] McHugh, O., et al.: Using Agile Practces to Influence Motvaton within IT Project Teams. Scandinavian Journal of Information Systems 23, (2011), pp. 59-84

[43] McHugh, O., et al.: Agile Practices: The Impact on Trust in Software Project Teams. IEEE Software 29, (2012),

[44] Misra, S.C., et al.: Identifying some important success factors in adopting agile software development practices. The Journal of Systems and Software 82, (2009),

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

[46] Napier, N.P., et al.: IT project managers' construction of successful project management practice: a repertory grid investigation. Information Systems Journal 19, (2009), pp. 255-282

[47] Nguyen, D.S.: Success Factors That Influence Agile Software Development Project Success. American Scientific Research Journal for Engineering, Technology, and Sciences 17, (2016), pp. 172-222

[48] Orlikowski, W.J.: Integrated Information Environment or Matrix of Control? The Contradictory Implications of Information Technology. Accounting, Management and Information Technologies 1, (1991), pp. 9-42

[49] Ouchi, W.G.: A Conceptual Framework for the Design of Organizational Control Mechanisms. Management Science 25, (1979), pp. 833-849

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

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

[52] Ramesh, B., et al.: Ambidexterity in Agile Distributed Development: An Empirical Investigation. Information Systems Research 23, (2012), pp. 323-340

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

[54] Robinson, H., Sharp, H.: The social side of technical practices. Extreme Programming and Agile Processes in Software Engineering, pp. 100-108. Springer (2005)

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

[56] Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall PTR (2001)

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

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

[59] Tessem, B.: Individual empowerment of agile and non-agile software developers in small teams. Information & Software Technology 56, (2014), pp. 873-890

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

[61] Tiwana, A.: Systems Development Ambidexterity: Explaining the Complementary and Substitutive Roles of Formal and Informal Controls. Journal of Management Information Systems 27, (2010), pp. 87-127

[62] VersionOne: The 9th Annual State of Agile Survey. (2015), pp. 16

[63] VersionOne: The 10th Annual State of Agile Survey. (2016), pp. 16

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

[65] Wang, X., et al.: Assimilation of agile practices in use. Information Systems Journal 22, (2012), pp. 435-456

[66] Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: Writing a literature review. MIS Quarterly 26, (2002), pp. 11

[67] Wiener, M., et al.: The effective promotion of informal control in information systems offshoring projects. European Journal of Information Systems 24, (2015), pp. 569-587

[68] 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

[69] Wood, S., et al.: Successful extreme programming: Fidelity to the methodology or good teamworking? Information & Software Technology 55, (2013), pp. 660-673

[70] Xu, P.: Coordination In Large Agile Projects. The Review of Business Information Systems 13, (2009), pp. 29-43

[71] Yin, R.K.: Case Study Research: Design and Methods. SAGE Publications (2003)

[72] Zellmer-Bruhn, M., Gibson, C.: MULTINATIONAL ORGANIZATION CONTEXT: IMPLICATIONS FOR TEAM LEARNING AND PERFORMANCE. Academy of Management Journal 49, (2006), pp. 501-519