Widely Used but also Highly Valued? Acceptance Factors and Their Perceptions in Water-Scrum-Fall Projects

Completed Research Paper

Sebastian Schlauderer
Chair of Industrial Information Systems
University of Bamberg
sebastian.schlauderer@uni-bamberg.de

Sven Overhage
Chair of Industrial Information Systems
University of Bamberg
sven.overhage@uni-bamberg.de

Björn Fehrenbach
Chair of Industrial Information Systems
University of Bamberg
bjoern.fehrenbach@stud.uni-bamberg.de

Abstract

Agile methodologies like Scrum propose drastic changes with respect to team hierarchies, organizational structures, planning or controlling processes. To mitigate the level of change and retain some established processes, many organizations prefer to introduce hybrid agile-traditional methodologies that combine agile with traditional development practices. Despite their importance in practice, only a few studies have examined the acceptance of such methodologies, however. In this paper, we present the results of a qualitative study that was conducted at a Swiss bank. It uses Water-Scrum-Fall, which combines Scrum with traditional practices. Based on the Diffusion of Innovations theory, we discuss several acceptance factors and investigate how they are perceived. The results indicate that, compared to traditional development methodologies, some aspects of Water-Scrum-Fall bring relative advantages and are more compatible to the way developers prefer to work. Yet, there also exist potential acceptance barriers such as a restricted individual autonomy and increased process complexity.

Keywords: Agile software development, software development, qualitative research
Introduction

Today’s volatile business environments make it ever more important for information system development teams to be able to adjust the development process to quickly changing requirements and to deliver outcomes in a fast way. To successfully address this challenge, the agile manifesto proposes to build upon fundamentally different values and practices than traditional development methodologies (Beck et al. 2001). For instance, agile methodologies promote a close collaboration with the customer, self-organizing teams, and responsiveness to change over codified processes (Nerur and Balijepally 2007). Using the agile manifesto as a fundament, various agile development methodologies have emerged. Among them are methodologies that focus on agile practices for the programming steps such as Extreme Programming (XP) as well as more management-oriented methodologies such as Scrum. Studies indicate that agile methodologies are not only rapidly becoming popular, but that they can indeed enhance the flexibility of development teams (Fitzgerald et al. 2006; Mann and Maurer 2005; Schlauderer and Overhage 2013).

Meanwhile numerous companies have adopted agile methodologies such as Scrum or XP. However, most of them have adapted them to match company-specific peculiarities and properties (West 2011). Such adjustments oftentimes arise from cultural or organizational aspects, which are in conflict with agile principles. Depending on how severe these constraints are, the promised benefits of the agile paradigm might not be realized at all or only in a limited fashion. Accordingly, it is of crucial importance for companies to understand the limitations that might arise when adapting agile methodologies to such contexts. Current studies indicate that especially a combination of the traditional Waterfall methodology with Scrum is popular in practice. The so-called Water-Scrum-Fall methodology is even pronounced to be “the reality of agile for most organizations today” (West 2011). In this hybrid agile-traditional approach, steps from the Waterfall model alternate with steps from Scrum. Generally, the Waterfall model is used to define the upfront work, i.e. to plan the functionality of a development project, the budget, or other organizational aspects. Afterwards, Scrum is used to implement the software before the final steps of the development process, such as the delivery, are again carried out using the Waterfall methodology. Even though such hybrid agile-traditional methodologies might be the reality in many organizations today (West 2011), they are hardly investigated in literature. Related studies rather describe processes to guide the transition from traditional to agile methodologies such as Scrum. In these studies, hybrid agile-traditional methodologies are typically considered only as an intermediate state along the migration path. Up to now, only few empirical studies take into account that hybrid agile-traditional methodologies such as Water-Scrum-Fall can also be introduced as a permanent standard and examine, which factors are perceived as benefits or drawbacks. It consequently remains unclear in how far hybrid agile-traditional approaches such as Water-Scrum-Fall might limit the potential of purely agile methodologies and which critical success factors ought to be considered when employing such methodologies.

With the work at hand, we contribute to closing this literature gap. Using an exploratory study design and the Diffusion of Innovations Theory (Rogers 1995) as theoretical lens of analysis, we address the subsequent research questions: “Which factors determine the acceptance of hybrid agile-traditional methodologies?”; “Compared to traditional development methodologies, are these factors perceived as drawbacks or benefits?”. To answer the research questions and examine the use of such methodologies in a realistic setting, we conducted a qualitative study in which we examined the use of Water-Scrum-Fall in the banking sector. We decided to employ a qualitative setting as this is considered especially appropriate when the phenomenon under investigation is a new or evolving field in which little or no research has been conducted before (Yin 2014). The data was gathered at a large Swiss bank, in which Water-Scrum-Fall was used for over one year. Overall, we were able to study four different teams and to conduct semi-structured interviews with 17 team members. This setting accordingly allowed us to gain in-depth insights into the use of Water-Scrum-Fall in practice and to identify potential drawbacks and benefits. We deliberately decided to examine the use of Water-Scrum-Fall as it appears to be the most widely used hybrid agile-traditional methodology today (West 2011). Moreover, Water-Scrum-Fall projects affect various stakeholders as the methodology also provides guidance for the planning and tracking of projects while other methodologies mainly support the programming of systems (Fitzgerald et al. 2006). Finally, our research endeavor follows a call from Dyba and Dingsoyr (2008) to investigate concrete agile methodologies, which are also used in practice.

In the next section, we set the theoretical basis for the work at hand. In that section, we first describe the characteristics of Water-Scrum-Fall and explain differences to the Scrum methodology. Secondly, we introduce the Diffusion of Innovations Theory, which serves as theoretical lens of analysis. Thirdly, we

Thirty Sixth International Conference on Information Systems, Fort Worth 2015
discuss related approaches and highlight the research gap. Against this theoretical background, we describe the research method in section 3. In section 4, we analyze the gathered data. In section 5, we discuss the implications and limitations of our study. We conclude by summarizing our work and presenting future research directions in section 6. Supplementary interview results are depicted in the appendix.

**Theoretical Foundation**

In this chapter, we explain basic concepts that are important for our study and highlight the research gap.

**Water-Scrum-Fall**

Before describing the properties of Water-Scrum-Fall, we first depict the basic characteristics of the Scrum methodology and the Waterfall model, since they are the key elements of this hybrid approach. Scrum and agile methodologies in general set a polar opposite to traditional methodologies, such as the Waterfall model. The agile manifesto already emphasizes this in its first principle stating that the “highest priority is to satisfy the customer” (Beck et al. 2001). Whereas the Waterfall model follows a rigorous process management in which the development is planned upfront, Scrum pronounces flexibility and is based on an empirical process control. The Waterfall model typically divides the development tasks into multiple work packages and organizes them with milestones, for instance by using a work breakdown structure in order to plan the development work upfront. Consequently, the Waterfall model is said to be predictable, repeatable and to enable the optimization of processes (Boehm 2002). Assuming that requirements are likely to change throughout the project, Scrum instead advocates a planning from iteration to iteration and aims to cut “through complexity to focus on building software that meets business needs” (Schwaber and Beedle 2002). For the planning process, Scrum introduces three different levels: During the Release Planning, basic strategic decisions are being discussed. However, operational details are only addressed in the Sprint Planning and are only planned for the next Sprint, i.e. a development iteration of typically four weeks length (Schwaber and Sutherland 2015). The most detailed level of planning is the Daily Scrum, a daily 15-minute meeting in which the development teams discuss current tasks, issues and the project state.

Not only the project planning but also the project controlling is handled differently. In Waterfall projects, the developers typically return a percentage of completion for their tasks. This information can be used for milestone and status reports to estimate the overall percentage of completion. In Scrum projects, the current project state is discussed in the various meetings. Many Scrum teams use Burndown Charts, in which the outstanding work for the current Sprint is shown and constantly updated in the Daily Scrum meetings. Scrum furthermore handles the documentation differently than traditional methodologies. Valuing “working software over comprehensive documentation” (Beck et al. 2001), Scrum sets a contrast to the Waterfall model, in which the importance of documentation is generally emphasized and oftentimes regarded as an integral part of development projects. Scrum instead pronounces that knowledge is transferred due to the various team meetings and the increased communication. While this reduces the time developers need to spend for writing explicit documentations, it can also be a risk since knowledge can easier get lost if meetings are not taken seriously or if team members leave the development team.

Besides that, Scrum also promotes a different collaboration culture. Whereas in Waterfall projects a clear hierarchy exists with a project manager, who assigns the tasks to the team members and is responsible for the project success, Scrum promotes a flat hierarchy. Instead of a project manager, Scrum introduces the role of the Scrum Master who acts as “servant-leader”. In contrast to the Waterfall model, this role has no authority to prescribe the development team how to implement the requirements. Instead, they should rather coach the team and ensure that Scrum is done right (Schwaber and Sutherland 2015). Scrum team members consequently have a higher responsibility as they shall self-responsibly organize their work and take over the remaining tasks in the Daily Scrum meetings.

Moreover, Scrum introduces a specific role for the customer in the development process, namely the Product Owner. The Product Owner shall continuously participate in the meetings and work closely together with the development team. He or she shall thereby help the development team to identify and specify requirements, which are summarized and prioritized in the Product Backlog. Note that the Product Owner has the possibility to change, refine, or add new requirements throughout the whole project. This procedure shall ensure that the customers’ requirements are best possibly met and is emphasized as “one of the most valuable characteristics of agile methods” (Huo et al. 2004). Compared to that, the requirements
in Waterfall projects are typically fixed at the beginning of a project in a contract-like document. Accordingly, the customer has to exactly know upfront how the software shall look like and (s)he has to be able to communicate this to the development team. If customers realize that the initially determined requirements are not implemented in the intended way or if certain requirements are missing, a change-management process has to be initiated. Finally, Scrum aims to foster the learning process during the development project by introducing Sprint Review and Sprint Retrospective meetings. In that way, developers have the opportunity to obtain feedback, discuss measures that turned out to be successful, or to talk about potential problems right after the first Sprint. While developers discuss such lessons learned also in Waterfall projects, these meetings occur usually only once at the end of a project. Figure 1 (left) summarizes a Scrum cycle beginning with the requirements definition in the Product Backlog, the design stage including the various meetings and development steps as well as the delivery of a working piece of software, which should be the result of each Sprint.

As can be seen from the discussion above, Scrum introduces fundamental changes to the development environment in terms of the collaboration culture, team structure, communication, or project planning. Accordingly, a transition from the Waterfall model to Scrum is a cumbersome task in which companies have to accept significant changes. Several factors may hinder companies in so doing, however. Potential reasons could be a strict hierarchy in the company that contradicts Scrum practices, the need to have a fixed budget and an estimated outcome at the project beginning, or the goal to automatize development projects in an engineering approach (Komus 2014b). As a consequence, hybrid agile-traditional approaches such as Water-Scrum-Fall seem to be even more often used than purely agile methodologies and might be the reality for many organizations today (West 2011). Sometimes, hybrid agile-traditional approaches might only represent an intermediate step during the aspired migration to an agile methodology. Oftentimes, however, companies aim at deliberately bringing together the specific advantages of agile and traditional methodologies by uniting them. In such scenarios, hybrid agile-traditional methodologies such as Water-Scrum-Fall might present themselves as a permanent alternative to purely agile methodologies.
Acceptance Factors and Their Perceptions in Water-Scrum-Fall Projects

has specified his requirements at the beginning of the project, however, his/her options are somewhat limited compared to Scrum projects. The final steps of the Water-Scrum-Fall model, i.e. the software test and rollout, are again supported by the Waterfall model. During the testing step, the functional as well as non-functional properties of the software are assessed and compared to the requirements definition. While some of the functional requirements are already assessed by the Product Owner during the conducted Sprints, this is a more formal step, which determines the acceptance of the developed software as a whole. Finally, the software is distributed to the operating departments during the rollout step. The typical Water-Scrum-Fall process is summarized in Figure 1 (right).

**DOI Theory**

As the discussion shows, Scrum differs from traditional software development methodologies in various aspects. Even hybrid agile-traditional approaches, such as Water-Scrum-Fall, pose significant changes to the way the development of information systems is managed in a company. Rogers (1995) broadly defines an innovation as “an idea, practice, or object that is perceived as new by an individual or other unit of adoption”. From a theoretical perspective, we accordingly classify the innovative Water-Scrum-Fall process as a software process innovation (SPI, Conboy et al. 2007). To explain the assimilation of such innovations, innovation diffusion theories have been defined, which describe the assimilation process and the relevant factors that determine the assimilation. Amongst others, Gallivan (2001) introduced a model that aims to explain the process in which an innovation is assimilated by an individual, a group, or an organization. Consisting of six stages, this process begins with the initiation stage, in which the suitability of the innovation is assessed. In the adoption stage, the adapting unit decides whether the innovation shall be introduced. Within the adaption stage, the innovation is adjusted in a way that users can work with it best possibly. In this stage, the members of the adopting unit are also often trained in the use of the innovation. During the acceptance stage, the adopting units are committing to use the innovation, before its use becomes a regular and normal activity in the routinization stage. Finally, the innovation is used comprehensively in the infusion stage.

To identify factors that determine the assimilation of innovations in general, further theories have been formulated, such as the Diffusion of Innovations (DOI) Theory, the Technology Acceptance Model (TAM), or the Theory of Planned Behavior. All of these theories have repeatedly been used to explain the assimilation of software artifacts, information systems, or process innovations (Mustonen-Ollila and Lyytinen 2003; Riemenschneider et al. 2002). These theories can accordingly provide a starting point and a theoretical basis for the identification of specific factors in Water-Scrum-Fall projects that are perceived as potential benefits or drawbacks. All of these theories have many commonalities in terms of the factors that are examined. We decided to employ the DOI Theory as theoretical lens of analysis since it is broadly applicable and has been widely used in literature. The DOI Theory assumes that the individual adopters have a different willingness to assimilate an innovation and that, consequently, the users will assimilate an innovation at different times (Rogers 1995). It proposes five perceived attributes as generic factors that determine an individuals' willingness to assimilate an innovation (Karahanna et al. 2006; Rogers 1995):

Relative advantage: the extent to which it is viewed as better than the idea it supersedes.
Compatibility: the extent to which it is viewed as consistent with actual needs or preferred practices.
Complexity: the extent to which it is viewed as difficult to understand or use.
Trialability: the extent to which it is viewed as easy to experiment with.
Observability: the extent to which it is viewed as visible to others.

Relative advantage, compatibility, trialability, and observability positively affect the rate of acceptance, whereas complexity is negatively correlated. Research moreover found that trialability and observability do not have a significant influence regarding the individual willingness of users to accept and use a SPI (Mustonen-Ollila and Lyytinen 2003). According to Riemenschneider et al. (2002), there are two reasons for this observation: first, the magnitude of behavioral change caused by the adoption of a methodology is far greater than that caused by the adoption of a tool. Second, the degree to which the adoption of the target innovation has been mandated is much greater for methodologies than for tools. The acceptance of development methodologies is hence not so much a matter of accessibility or personal standing. Rather, it is a matter of relative advantage, compatibility, and complexity, which literature identified as significant determinants (Mustonen-Ollila and Lyytinen 2003; Riemenschneider et al. 2002; Tornatzky and Klein 1982). However, to comprehensively analyze the factors that influence the acceptance of Water-Scrum-Fall,
we nevertheless decided to examine all perceived attributes of innovations. Note that there is still an ongoing discussion about the conceptual difference between compatibility and relative advantage. Based upon the original definition of the concepts by Rogers (1995), some proponents argue that relative advantage encompasses compatibility. Yet, Karahanna et al. (2006) demonstrate that compatibility can be treated as a multidimensional construct of its own and describe its relation to the constructs of the TAM. Following their argumentation, we decided to consider compatibility as a separate construct in our study as well (Figure 2).

![Figure 2. Perceived attributes of innovations](image)

**Related Work**

To analyze the body of knowledge about hybrid agile-traditional methodologies and Water-Scrum-Fall in particular, we conducted a literature review based on the recommendations of Webster and Watson (2002). In the first step, we searched the databases of Google scholar, ACM Digital Library, IEEE Xplore, EBSCOhost, and Web of Science. Addressing our research questions, we specifically searched for the terms “Water-Scrum-Fall”, “hybrid agile development”, and “hybrid Scrum development”. We inspected the titles and abstracts of the resulting articles to sort out irrelevant articles. In the second step, we conducted backward and forward searches based on the articles that we perceived to be particularly relevant to identify further manuscripts (Webster and Watson 2002). Using a narrative review method, we then inspected the articles we had found in detail (King and He 2005).

With Scrum and agile development methodologies being primarily practitioner-driven, many of the promised advantages were not scientifically documented or proven for quite some time. Consequently, it was oftentimes criticized that the strength of empirical evidence about limitations and benefits of agile methodologies was low (Abrahamsson et al. 2009; Dyba and Dingsøyr 2008; Petersen and Wohlin 2009). In the last couple of years, this call for more empirical analyses, which evaluate agile software development methodologies in detail, was addressed by various researchers (e.g. Conboy 2009; Lee and Xia 2010; Schlauderer and Overhage 2013; Senapathi and Srinivasan 2012; Vijayasarathy and Turk 2012; Wang et al. 2012). This ongoing research stream mainly investigates the adoption of agile methodologies, their benefits and their limitations. Additionally, it is being evaluated how agile development methodologies can be adapted to best fit to an organizational context (Cao et al. 2009). However, there are only few articles that address hybrid agile-traditional approaches and their adoption. This is even more severe since hybrid agile-traditional approaches and specifically Water-Scrum-Fall are very popular in practice and seem to be even more often used than purely agile methodologies (Komus 2014a; West 2011).

Overall, we found only a small number of studies that address the use of hybrid agile-traditional methodologies such as Water-Scrum-Fall. Among them were mostly articles that describe such methodologies as part of a transition process from traditional software development to agile software development (Ayed et al. 2014). In these articles, the authors describe measures to achieve a smooth transition from Waterfall projects to Scrum projects or they explain which problems arise during the transition process. However, since various companies willingly decide to introduce Water-Scrum-Fall as a methodology of its own, e.g. for organizational reasons, it cannot only be viewed as a temporary solution. Instead, its potential benefits,
drawbacks, or limitations ought to be investigated. Next to articles focusing on the transitioning process, we also identified articles that propose specific metrics for the project controlling in hybrid development environments, such as Water-Scrum-Fall projects (Jinzenji et al. 2013; Mahadevan et al. 2015). While these articles are useful when implementing a project controlling in such environments, they nevertheless provide little insights into potential pitfalls or benefits of hybrid agile-traditional development methodologies.

Only few studies specifically investigate in how far traditional and agile methodologies, such as Scrum and the Waterfall model, are compatible and in how far mixed approaches could be beneficial (Aitken and Ilango 2013; Hayata and Jianchao 2011; Rahmanian 2014). Interestingly, the authors come to the conclusion that hybrid approaches could be “the appropriate solution” (Rahmanian 2014) and that there “is nothing really incompatible” between traditional and agile software development methodologies (Aitken and Ilango 2013). Rahmanian (2014) even sees a couple of benefits in the combination of Scrum and the Waterfall model. He argues that the upfront requirements specification will reduce the risk of confusion, that during the Scrum steps the implementation will speed up, and that the level of testing and acceptance in the final Waterfall steps will be higher (Rahmanian 2014). However, these studies argue on a purely theoretical basis. Accordingly, there is a need to empirically evaluate such findings and to examine in how far hybrid agile-traditional methodologies such as Water-Scrum-Fall are indeed successful in practice. So far, we have only identified one article that investigates such details (Bannink 2014). Yet, Bannink (2014) focuses on challenges that arise during the transition from Waterfall to Scrum. While they also observe some phenomena that they refer to be a “typical Water-Scrum-Fall situation” (Bannink 2014), they inspect such details of Water-Scrum-Fall only as an aside. It is hence not clearly apprehended yet if Water-Scrum-Fall can indeed provide benefits, which limitations this hybrid approach has, or which factors influence its acceptance. With the work at hand, we address these questions aiming to contribute to the closure of this research gap.

Research Method

As summarized in our research questions, we aim at gaining an understanding about the factors that determine the acceptance of the Water-Scrum-Fall methodology in practice. Furthermore, we want to identify if these factors are perceived as benefits or drawbacks in comparison to traditional, plan-driven software development approaches, which are often based on the Waterfall model. As shown in the previous discussion of related work, there is currently little confirmed knowledge about how hybrid agile-traditional methodologies such as Water-Scrum-Fall are perceived in practice. Accordingly, we adopted an exploratory research design. As suggested for new or evolving research fields, in which no or only little research has been conducted, a qualitative research design was chosen (Rubin and Babbie 2006). Doing so allowed us to gain in-depth insights into the perceptions of Water-Scrum-Fall in practice and to understand which factors facilitate or hinder the acceptance (Given 2008). In order to gain an understanding about which factors influence the acceptance of Water-Scrum-Fall, we decided to conduct interviews with experts on the subject. Generally, an expert is characterized as someone who has privileged knowledge about a certain topic (Bogner et al. 2009), in our case the use of Water-Scrum-Fall in development projects. As experts typically provide extensive insights of a domain, the number of interviewees can be rather low as long as they are selected in a way that they can indeed contribute to answering the research question (Bogner et al. 2009).

We decided to conduct semi-structured face-to-face interviews with the experts, since they are considered to be the superior data collection technique for interpretive investigations (Yin 2014). In particular, semi-structured interviews ensure comparable results by following a common guideline, yet they provide a greater breadth than other types of interviews by allowing the interviewer to adjust questions and to ask for explanations.

The interviews overall closely followed the recommendations of Myers and Newman (2007). The interview guideline consisted of two parts. In the first part, we asked for general demographic information, such as the departments the experts work for, the experience they have with Water-Scrum-Fall and with plan-driven approaches based on the Waterfall model, or how long they have been working in software development projects. In the second part, we asked about the factors that determine the acceptance of Water-Scrum-Fall and how they are perceived. Based on the DOI Theory as lens of analysis, we specifically asked which factors of Water-Scrum-Fall are perceived to be advantages or disadvantages in comparison to traditional, plan-driven development approaches. In this respect, we moreover wanted to know which aspects of Water-Scrum-Fall are perceived to be more or less compatible to the way the team members would prefer to work, and which factors reduce or increase the complexity of a development project. Finally,
we asked for relevant aspects regarding the observability and the trialability of Water-Scrum-Fall. For each factor that was mentioned by the interviewed experts, we further wanted to know how the factor was perceived, why this was the case, and how the factor influences the commitment of the interviewee.

We collected our data at a large Swiss bank, which began introducing Water-Scrum-Fall as a development methodology a few years ago. In 2015, the time of our study, the bank had already conducted several smaller projects, in which it used and slightly adapted the methodology to fit its organizational context. As the company had finished adapting Water-Scrum-Fall to its needs, it was in the acceptance stage with respect to the innovation assimilation process. The bank explicitly decided to use Water-Scrum-Fall for various organizational reasons and employed a typical Water-Scrum-Fall process, with the steps faithfully resembling the ones depicted in Figure 1 (right). While the project initiation, the requirements definition, the rollout, and maintenance steps were supported by the Waterfall model, the implementation step was supported by Scrum principles. The projects that we studied were so far the biggest in which Water-Scrum-Fall was used. They had a similar goal and addressed tasks in the SAP context, in which various applications had to be developed to support the migration of the system. The implementation stage overall took about twelve months. During this time, eight to ten Sprints were conducted in the four different projects that we surveyed. Accordingly, a Sprint on average was a bit longer than one month, which comes close to the Sprint duration suggested by Schwaber and Sutherland (2015). In the four projects, four Scrum teams were employed each supporting exactly one project. The four teams were supported and guided by three different Scrum Masters. Moreover, seven Product Owners handled the customer roles in the four projects throughout the project duration. The Product Owners were additionally supported by three team members who had a supporting role. This role was introduced by the company to guide the quality assurance and testing stages, yet it does not exist in the Scrum guidelines. In total, approximately 60 different team members were working in the four projects throughout the whole project. In our study, we interviewed 17 team members who came from all four projects. We deliberately decided to include different roles into our survey to better ensure the reliability of our results and to examine perceptions from multiple perspectives. Accordingly, we included seven interviewees who had the role of a Product Owner in their projects. Seven participants worked as developers and one worked as Scrum Master. Finally, we interviewed two participants who had a support role. As this role basically supports the Product Owner in his/her tasks, we decided to include it into the study. All of the interviews were conducted face to face in the interviewees’ offices. The interviews were recorded and transcribed for analysis afterwards.

Results and Discussion

The gathered demographic data shows that the 17 interviewees had profound experience with Scrum-based methodologies and the traditional Waterfall model. At the time of our study, they had been working with Scrum-based methodologies for 1.25 years and with the Waterfall model for 10 years on average. On a scale from (1) marginal knowledge to (4) expert knowledge, the participants ranked their expertise with Scrum-based methodologies at 2.7 and with the Waterfall model at 3.3 on average. The Scrum Master specified a more profound and longer experience with Scrum. Other than that, the results only varied slightly between the different groups (i.e. between the interviewed Product Owners, developers, and support roles). All in all, the analysis of the demographic results yields that we have chosen a homogenous sample of participants showing typical characteristics for an organization, which is in an early stage of the assimilation process.

Using the expert statements gathered during the second interview part, we searched for specific acceptance factors for the Water-Scrum-Fall methodology, on which the experts reached broad consensus. Accordingly, we focused our investigation on acceptance factors that were emphasized as important either by the majority of the interviewees or at least by the majority of the participants from one group. As a first step to analyze the expert statements, we used open coding techniques. Doing so allowed us to search for recurrent statements about potential acceptance factors in the data. We then grouped thematically related statements into topics in a way that each topic covers a specific acceptance factor (Miles and Huberman 1999). Relying on so-called in-vivo codes (Given 2008), each topic was labeled with the denomination that was predominantly used by the interviewees to describe the concept. In so doing, we identified several factors that influence the acceptance of the Water-Scrum-Fall methodology and were recurrently mentioned by the interviewed experts. We furthermore analyzed the segmented data to identify consistencies or distinctive statements regarding the perceptions of the identified factors and the causes behind the perceptions. Based on this analysis, we were able to gain an in-depth understanding of how the various acceptance factors are
perceived by the interviewed experts and why this apparently is the case. All in all, we identified 9 acceptance factors for Water-Scrum-Fall (F1-F9). In the following, we describe these factors and their perception in detail. For each factor, we first depict the most revelatory interview statements (which we faithfully translated into English language) and thereafter present our interpretation. We decided to keep the original expert statements and our conclusions separate to make our analysis traceable. We therefore inserted anchors in squared brackets into the interview statements which we then reference to justify our conclusions. To further substantiate our interpretations and to provide a complete picture of the interview results, we depict additional interview transcripts, which we examined during the data analysis, in the appendix.

**Relative Advantage**

With respect to the relative advantages and disadvantages of Water-Scrum-Fall in comparison to the traditional Waterfall model that was used before, the interviewed experts consistently pointed to three factors. Six Product Owners, six developers, and the interviewed Scrum Master emphasized the *meeting of requirements* (F1) as an important factor that influences the acceptance of Water-Scrum-Fall:

“In sum, the requirements are better met in Water-Scrum-Fall projects [1.1] because we have multiple reviews at the end of each Sprint. The identified faults can then instantly be addressed [1.2]” (Product Owner 1). “The requirements are better met [1.3]. This is due to the fact that developers and business users already communicate early in the process. The developers hence get a better understanding of the requirements [1.4]. In addition, the business users are involved into the review and acceptance process earlier than before [1.5]” (Product Owner 2). “You have to have a clear picture of the requirements. For that reason, I appreciate our mixture of traditional and agile processes with a dedicated requirements definition phase [1.6]. The subsequent development is governed by clear specifications of what has to be done. [...] I have participated in purely agile projects where this has been chaotically [1.7]” (Product Owner 6). “The requirements are better met since the business department has to directly accept all results [1.8]. It is also possible to flexibly adapt requirements during the process [1.9]. Altogether, this is much better than to wait until the moment of truth just to learn that your results do not work [1.10]” (Developer 4).

From the depicted statements, we can conclude that both the involved customers and developers feel the requirements to be met better in Water-Scrum-Fall projects compared to the traditional projects they had participated in before [1.1, 1.3, 1.8]. The better meeting of requirements is apparently perceived as a relative advantage that positively influences the acceptance of Water-Scrum-Fall [1.6, 1.10]. The better meeting of requirements seems to be caused by several measures, which partially come from the Scrum methodology and in parts stem from the traditional Waterfall model. With respect to the Scrum methodology, three measures were praised by the interviewees. First, the continuous acceptance testing of deliverables seems to provide an effective mechanism to find and sort out defects early in the process [1.2, 1.5, 1.8]. Second, the collaborative discussion of requirements appears to help clarifying the development goals for all involved parties [1.4]. Third, the possibility to flexibly develop and reprioritize requirements seems to provide an effective means to cope with unstable requirements [1.9]. With respect to traditional techniques, the interviewees pointed out that retaining a dedicated requirements definition phase provided an important means to clarify development goals and to facilitate the upfront planning [1.6, 1.7]. It hence seems to be the unique combination of agile and traditional practices that create the relative advantage. Judging from the interview results, this combination of practices might even be an advantage of Water-Scrum-Fall compared to purely agile methodologies, in which a detailed upfront requirements planning is not intended [1.7].

As a second factor, four Product Owners, six developers, the Scrum Master, and one member of the support staff described the *economic value* (F2) of the development project:

“The productivity is higher [2.1]. A developer is expected to flexibly take over any new task from the task-board when he completed his assignment. Idle times are so reduced and the resources are more efficiently used [2.2]. Each requirement undergoes a quality check during the review at the end of the Sprint. The results are hence of a better quality [2.3]” (Product Owner 4). “The project state is pretty transparent. Hence, it is easy to detect and minimize idle times [2.4]. Although the effort is higher at the same time, this might increase productivity [2.5]. [...] The quality of the results will probably be better due to the close collaboration that helps avoiding misunderstandings and mistakes [2.6]” (Product Owner 6). “As we develop incrementally and start with the top-priority requirements, we have readily usable software in any case. We don’t just have everything finished to 80% [2.7]. That is a big advantage [2.8]” (Developer 5).
The statements indicate that the interviewees consider the economic value in Water-Scrum-Fall projects to be higher than in the traditional projects they had been part of before [2.1, 2.3, 2.5, 2.6]. The increased economic value obviously is perceived as a relative advantage that positively influences the acceptance of Water-Scrum-Fall [2.8]. In part, the increased economic value is felt to come from an improved utilization of developer capacities [2.2, 2.4]. However, the experts perceive economic value to be a broader concept, which does not only encompass efficiency gains. Gains in effectiveness likewise seem to contribute to the perceived economic value [2.3, 2.6, 2.7]. The gains in efficiency and effectiveness seem to be caused by the usage of agile practices during development. In particular, the interviewees praised the flexible take-over of tasks instead of a static assignment of work packages [2.2], the continuous collaboration of all stakeholders [2.4], and the incremental development of ready-made software according to prioritized requirements [2.7]. Traditional practices in contrast were not specifically mentioned as beneficial by the interviewees.

As a third factor, two Product Owners, four developers, and one member of the supporting staff discussed the time to market (F3):

“I think that individual features might be available sooner than in traditional projects [3.1]. At the time of development, however, there is no deployment concept yet [3.2]” (Product Owner 3). “Overall, I think, the development is faster [3.4]” (Developer 3). “The business department will get individual functionalities sooner, and according to their prioritization [3.5]” (Developer 5). “In contrast to traditional projects, we start developing without much warm-up. Hence, we deliver results sooner [3.6]” (Developer 6).

From the statements, we can conclude that the experts have a mixed attitude regarding the time to market in Water-Scrum-Fall projects. On the one hand, they seem to perceive the time to delivery as shorter than in the traditional projects they had participated in before [3.1, 3.4, 3.6]. The shorter time to delivery seems to result from the agile practices that are also responsible for the higher economic value [3.5, 3.6]. On the other hand, the interviewees emphasize that ready-made software features cannot be used until the rollout phase has been completed [3.2]. However, the rollout phase belongs to the traditional, sequential part of Water-Scrum-Fall and is only carried out after the development has been entirely completed. The early use of ready-made software increments hence seems to be made difficult by the traditional roll-out practices of the Water-Scrum-Fall methodology. As the perception of the time to market remains inconclusive, no assumption about its effect on the acceptance of the Water-Scrum-Fall methodology can be made.

**Compatibility**

Regarding the compatibility of the Water-Scrum-Fall methodology to preferred work practices or actual needs of the team members, the majority of the interviewees emphasized three factors. Five Product Owners, six developers, and two members of the support staff described transparency (F4) as a central factor:

“I like the increased transparency [4.1]. It corresponds to my favorite way of working [4.2]” (Product Owner 2). “There is more transparency in Water-Scrum-Fall projects [4.3]. The requirements are communicated more clearly [4.4] and you can see the current project state anytime by looking at the Burndown Chart [4.5]” (Product Owner 4). “I am working in a project with a huge impact. For me, it is hence important to have an overview anytime [4.6]. With the agile methodology this is possible. There is more transparency [4.7]” (Developer 3). “The meetings and the charts make the project status transparent [4.8]. Therefore, I find Water-Scrum-Fall much more useful than the Waterfall model [4.9]” (Developer 5).

The depicted statements indicate that the experts find the transparency in Water-Scrum-Fall projects to have increased compared to the traditional projects they had worked in before [4.1, 4.3, 4.7]. The increased transparency appears to be more compatible to the way the team members prefer to work [4.2, 4.6]. Obviously, this perception facilitates the acceptance of the Water-Scrum-Fall methodology [4.1, 4.9]. The increased transparency seems to stem from the agile practices that are part of Water-Scrum-Fall. In particular, the interviewees welcomed the practices to publicly document the progress [4.5, 4.8] and to discuss the progress [4.4, 4.8]. Traditional practices were not mentioned as potential causes for the increased transparency by the interviewees.

Six Product Owners, three developers, and the Scrum Master furthermore referred to the communication (F5) as an important factor:

“The agile philosophy strengthens the communication [5.1]. People talk with each other much more, they approach one another if there is a problem and learn to rely on each other [5.2]” (Product Owner 6). “The
business department and the developers do not live in separate worlds anymore. You get a better understanding of why something is or is not done [5.3]. This creates mutual trust and makes things easier [5.4]” (Product Owner 2). “I enjoy interacting more closely with the other team members [5.5]. In the daily meetings, you get to know what they are doing and where the problems are [5.6]” (Product Owner 4). “You talk with each other, which is a huge benefit [5.7]. I have never seen business departments and developers interact so closely in traditional projects [5.8]” (Scrum Master).

From the interview results, we can conclude that the experts perceive the communication in Water-Scrum-Fall projects to be better than in the traditional projects they had worked in before [5.1, 5.5, 5.8]. Obviously, the better communication is more compatible to the way the team members prefer to work [5.4, 5.7]. This apparently facilitates the acceptance of the Water-Scrum-Fall methodology [5.2, 5.5]. The increased communication obviously is caused by the agile practices that are part of the Water-Scrum-Fall methodology [5.1]. In particular, the interviewees praised the daily meetings [5.6] and the collaborative development [5.2, 5.7]. The agile practices seem to facilitate the utilization of the different skills that the team members bring into the project [5.8]. They also contribute to the emergence of mutual understanding and trust [5.2, 5.3, 5.4]. Traditional practices in contrast were not mentioned by the interviewed experts.

Two Product Owners and four developers finally discussed the self-responsibility (F6) as a central factor:

“Having a global project management helps to create and stay focused on a long-term plan. But the project management should not interfere with my daily work. I want to be self-responsible for my work [6.1]. The same is true for the Scrum Master. He should guide the project, but he is not my boss. The team should be the boss. In this project, we still need to work on this prerogative [6.2]” (Developer 2). “I understand that the project management has to accurately know how many requirements have been implemented and how much budget is left. However, we also receive quite strict specifications of how we have to do things. Consequently, we cannot decide for ourselves anymore [6.3]. We could be more productive and agile if we had the chance to act self-responsibly [6.4]” (Developer 6).

From the depicted statements, we can conclude that the experts still perceive the self-responsibility to be limited in Water-Scrum-Fall projects [6.2, 6.3]. The limited self-responsibility seems not to be compatible with the way the team members prefer to work [6.1, 6.4]. The limited self-responsibility seems to result from the traditional practices to steer development activities by building upon a strict control hierarchy and work specifications, which are generally part of the Water-Scrum-Fall methodology [6.2, 6.3]. The intensive use of such practices seems to limit the agility of the team members [6.4]. In particular, they seem to be in conflict with the team members’ wish to organize their daily work more autonomously [6.1]. Apparently, this negatively influences the acceptance of the Water-Scrum-Fall methodology [6.2].

**Complexity**

Regarding the complexity of Water-Scrum-Fall, the interviewees consistently highlighted two factors. Five Product Owners and five developers emphasized the process complexity (F7) as an essential criterion:

“The organization of the project is much more complex [7.1]. On the one hand, we have the institutionalized procedures of Scrum such as the daily meetings, the Sprint reviews etc. [7.2]. On the other hand, we have a global project management with quality standards and reporting procedures [7.3].” (Product Owner 1). “The working process is fine-grained and consists of smaller tasks that are easier to implement by the developers [7.4]. However, I think that the global view gets lost. Composing all the small things to see if a requirement has been completely covered is difficult [7.5].” (Product Owner 2). “We have a non-agile global project management with which we only communicate through the Scrum Master. This indirect communication is not productive because misunderstandings occur [7.6].” (Product Owner 4). “You can start developing quickly, which is positive [7.7]. However, you also have to plan every Sprint, which is negative [7.8]. I am not sure if the resulting process is less complex [7.9].” (Developer 6).

The interview results suggest that the experts have a mixed attitude regarding the complexity of the development process [7.1, 7.4, 7.9]. To some extent, the agile practices contained in the Water-Scrum-Fall methodology seem to be responsible for this mixed attitude. On the one side, the interviewees seem to find the complexity of individual development tasks to be lower than in the traditional projects they had worked in before [7.4, 7.7]. A reason for this seems to be that the tasks are more fine-grained than the work-packages in traditional projects [7.4]. On the other side, additional effort is required to coordinate and integrate the fine-grained development tasks [7.2, 7.5]. In addition, each Sprint has to be planned individually as
there exists no detailed upfront planning for development tasks [7.8]. Besides agile practices, the experts also blame traditional practices to increase complexity. Particularly, they criticize the multi-level hierarchy that is typical for traditional projects. Above all, they criticize the indirect communication with the project management level [7.6] and the additional controlling layer [7.3]. As the perception of the process complexity varies, no assumption on the consequences for the acceptance of Water-Scrum-Fall can be made.

Besides the process complexity, five Product Owners, three developers, and the Scrum Master highlighted the required discipline (F8) to be a key factor for the acceptance of the Water-Scrum-Fall methodology:

“There is such a hectic rush at the end of each Sprint. You have to participate in acceptance tests and at the same time you have to plan the next Sprint. I personally lack the required coolness to take this easy [8.1]” (Product Owner 2). “The negative point is [8.2]: I personally did not expect so much effort that is required again and again [8.3]. In each Sprint you have this effort on top of your daily work in the business department” (Product Owner 3). “I often feel extremely under pressure because I have tight time schedules and nevertheless have to help others when they come to discuss problems [8.4]. Water-Scrum-Fall is more complex because you collaborate with so many people [8.5]. You have to have a very good time management” (Product Owner 7). “You are much more responsible for your task [8.6]. You have to commit yourself and live for the project” (Developer 2). “You have quite some overhead like the daily meetings. Although they only last for 15 minutes, they interrupt your flow of work. I think, we’d be better off having such meetings only every two days or so [8.7]” (Developer 3).

The interview results indicate that the stakeholders find the required discipline to be higher than in the traditional projects they had worked in before [8.1, 8.3, 8.5]. Obviously, this perception negatively affects their acceptance of the Water-Scrum-Fall methodology [8.1, 8.2, 8.7]. The higher level of required discipline seems to be caused by the agile practices that are contained in the Water-Scrum-Fall methodology. Particularly, the participants seem to find it challenging to continuously live up to their tasks [8.1, 8.3], to participate in the institutionalized daily meetings [8.7], and to closely collaborate with others while still staying focused on their own tasks [8.4]. In addition, it seems to be a burden that the agile team members are more responsible for the success of the project than in traditional development endeavors [8.6]. Traditional practices were in contrast not blamed to increase the required discipline.

**Trialability and Observability**

We also asked if factors regarding the trialability and observability have an influence on the acceptance of Water-Scrum-Fall. In response, five Product Owners, six developers, and both members of the supporting staff emphasized the learnability (F9) as important:

“I found it easy to get used to the procedures of Water-Scrum-Fall [9.1]” (Product Owner 2). “Water-Scrum-Fall can easily be learned even during the project [9.2]” (Product Owner 4). “Once you have read the concept and participated in a Sprint, you already understand the method. It is not complicated at all [9.3]” (Developer 1). “Of course it needs some warm-up, but after one or two Sprints you get it. It is easy to motivate people because Water-Scrum-Fall lets you get rid of the downsides of the Waterfall model [9.4]” (Developer 5). “There are only few rules and principles [9.5]. If you are open for new things, it is going to be a wonderful experience” (Developer 6). “It is fun to learn [9.6]” (Supporter 2).

The statements indicate that the experts find Water-Scrum-Fall to be easily learnable [9.1, 9.2, 9.3]. The perception seems to positively influence the acceptance [9.4, 9.6]. The learnability apparently is facilitated by two mechanisms: first, there are comparatively few practices to study [9.5]. Secondly, the practices seem to be helpful to overcome weaknesses of traditional development methodologies [9.4].

Besides this factor, the majority of the interviewees found trialability and observability to be irrelevant:

“There is no personal motivation to work in a Water-Scrum-Fall project. Development fashions come and go. It is just one of them” (Product Owner 5). “Such factors are not important. There is no Aha experience in Water-Scrum-Fall projects” (Developer 1). “There might be a publicity for the project manager, but not for a team member” (Developer 3). Only one expert had a different opinion: “The methodology is defined by the project manager. But I like trying things and hence volunteered for the project” (Supporter 2).
**Discussion**

Taking a broad expert consensus as a basis, we identified nine perceived factors (F1-F9) that determine the acceptance of Water-Scrum-Fall. The interview results moreover provide rich insights into the context of the factor perceptions, which allows us to investigate the reasons behind the individual perceptions.

The results indicate that the stakeholders find the development according to the Water-Scrum-Fall methodology to bring *relative advantages* compared to traditional, plan-driven projects based on the Waterfall model. The team members apparently perceive the *requirements to be better met* in Water-Scrum-Fall projects (F1). Our findings indicate that the Water-Scrum-Fall approach might indeed be able to retain one of the central goals behind the creation of Scrum, which is “building software that meets business needs” (Schwaber and Beedle 2002). In Scrum projects, a better requirements meeting shall mainly be achieved by giving customers the opportunity to constantly add, change, or reprioritize requirements (Schwaber and Beedle 2002). While this opportunity was indeed mentioned as helpful in our study, it apparently did not play a central role in the examined projects. The experts rather praised the more intensive communication, which helped clarifying requirements for both sides, and the continuous acceptance testing, which allowed identifying faults early during the implementation. Notably, the experts also appreciated the traditional up-front planning and specification of requirements during the preceding requirements definition stage, which apparently provided a more stable and clarified basis to begin the implementation. Compared to purely agile methodologies, where the lacking support for the up-front planning and conceptualization of requirements is sometimes criticized (Schlauderer and Overhage 2013), this observation might hence indeed point to a relative strength of the Water-Scrum-Fall methodology (Rahmanian 2014). The study results moreover suggest that the stakeholders perceive the *economic value* of Water-Scrum-Fall projects to be higher (F2) than that of traditional projects. To some extent, the perception seems to come from efficiency gains such as an improved utilization of developer capacities. The findings hence indicate that the Water-Scrum-Fall methodology might help to reduce the development costs, which is another general goal of agile methodologies (Fitzgerald et al. 2006). However, the stakeholders perceived economic value to be a broader concept that also encompasses goals such as a higher effectiveness. Our findings confirm that the economic value of agile methodologies should not solely be assessed in terms of efficiency gains (Conboy 2009). Besides these advantages over traditional projects, however, the interview results also indicate that the *time to market* is found to be important, but not significantly reduced in Water-Scrum-Fall projects (F3). While the time to delivery seemingly is lower, implemented software features cannot be used until the global rollout phase has been completed. Compared to agile methodologies, which reduce the time to market by delivering ready usable software early in the process (Beck et al. 2001), these findings unveil a relative weakness of the Water-Scrum-Fall methodology.

Compared to the traditional Waterfall model, the study results suggest that the stakeholders view the Water-Scrum-Fall methodology to be *more compatible* to the way they prefer to work. Specifically, they seem to perceive the *transparency* as higher (F4). In development projects, transparency is essential to facilitate the collaboration and to control the project state (Fitzgerald 1997). To increase the transparency of the development state, Scrum builds upon Burndown Charts, daily meetings, and the continuous inspection of working software after each Sprint (Schwaber and Beedle 2002). Apparently, the Water-Scrum-Fall methodology successfully takes over these techniques to institutionalize transparency and so retains one of the frequently mentioned advantages of Scrum (Schwaber and Sutherland 2015). Compared to traditional projects, the interview results furthermore suggest that the team members find the *communication* to be better in Water-Scrum-Fall projects (F5). An immediate collaboration of the different stakeholders is crucial to facilitate the usage of their individual skills and strengths (Beath and Orlikowski 1994; Sawyer et al. 2008). In Scrum projects, collaboration is promoted by establishing an immediate, informal communication among all stakeholders (Schwaber and Sutherland 2015). Judging from the interview statements, this principle can effectively be established in Water-Scrum-Fall projects, too. However, the experts criticized that there is still much left to be desired regarding the actually aspired level of *self-responsibility* in Water-Scrum-Fall projects (F6). To facilitate the practice of agile principles such as collaboration, communication, flexibility, and transparency, agile methodologies like Scrum democratize the management of the team and allow developers to organize themselves (Schwaber and Sutherland 2015). The study results show that this principle is obviously not easily transferrable to Water-Scrum-Fall projects, which make use of traditional, hierarchical management and strict controlling processes. The experts even reported that the project management was tempted to misuse agile principles such as the transparency of
the project state to more closely control developers, thereby further limiting their creativity and confidence. We were hence not able to confirm that there “is nothing really incompatible” between traditional and agile software development methodologies (Aitken and Ilango 2013). As self-organization and flat hierarchies are important facilitators for agile practices (Beck et al. 2001), our findings rather point to a potential weakness of the Water-Scrum-Fall methodology that needs to be carefully addressed in order to successfully put agile principles into action.

In comparison to traditional projects, the interview results moreover indicate that projects using the Water-Scrum-Fall methodology are found to be more complex. First of all, the experts were not convinced that the process complexity indeed drops (F7). While they found the individual tasks to be less complex because of their finer granularity, they felt overburdened with having to constantly manage the planning process and to coordinate the achieved results. The interview results moreover indicate that the hierarchical management structure, which is typical for Water-Scrum-Fall projects, introduces an additional layer of indirection and complexity. Compared to Scrum, which aims at introducing a “management and control process that cuts through complexity” (Schwaber and Beedle 2002), our findings might hence point to another comparative weakness of the Water-Scrum-Fall methodology. In comparison with traditional projects, the experts additionally found the required discipline to be higher (F8). Judging from the expert statements, both the required on-site availability and the increased responsibility for project decisions were felt as burdens. Seen through the lens of the DOI theory, especially the factors describing the complexity might pose a barrier to acceptance. While the Water-Scrum-Fall methodology obviously is easy to learn (F9), our observations raise questions with respect the long-term sustainability of the hybrid agile-traditional development approach.

Implications

Based on the collected empirical data, we identified nine factors that influence the acceptance of the Water-Scrum-Fall methodology. The results of our exploratory study deepen our understanding of the factors that determine the acceptance and diffusion of software process innovations. In particular, we present innovative findings that help explaining the acceptance of hybrid development methodologies, which combine traditional and agile practices. The presented findings have implications for academia and practice.

For academia, the study results yield a theory on the factors that influence the acceptance of hybrid development methodologies, which combine traditional and agile practices. To our best knowledge, it is the first theory that addresses this intermediate class of software process innovations. Our theory builds upon the DOI theory and expands its generic constructs with specific concepts that explain the acceptance of the Water-Scrum-Fall methodology (Figure 3). The findings of our study suggest that the perceived attributes of innovations as introduced by the DOI theory can also be used to explain the acceptance of hybrid agile-traditional development methodologies. In line with the literature on the diffusion of software process innovations, we found that relative advantage, compatibility, and complexity seem to be primarily determining the acceptance (Agarwal and Prasad 1998; Mustonen-Ollila and Lyytinen 2003; Riemenschneider et al.)
As is to be expected in a mandatory usage context, the influence of trialability and visibility on the acceptance seems to be comparably limited. With the provided theory, we do not only inform research on the diffusion of software process innovations, however. The results of our study also inform research on agile methodologies and their practices. While the results of our study only represent an initial understanding of the subject, they already provide insights in how far agile and traditional practices can be combined and where frictions might occur. The results of our research also call for a more in-depth examination of the effects of individual agile practices. For instance, it ought to be investigated more closely if individual practices such as self-responsibility, collaboration, or communication might be used independently of each other or if they only work in concert. The results of such research endeavors would lead to a deepened understanding of agile practices and their (independent) usability in customized methodologies.

For practice, the presented acceptance factors and perceptions give information about the potential opportunities and threats that accompany the introduction of hybrid agile-traditional methodologies such as Water-Scrum-Fall. Our findings suggest that, compared to the use of the Waterfall model, Water-Scrum-Fall can bring relative advantages such as a better meeting of requirements and a higher economic value. The results furthermore indicate that Water-Scrum-Fall better satisfies the needs of the team members for transparency and communication. However, our findings also show that intermediate approaches such as the Water-Scrum-Fall methodology will possibly not leverage the full potential of agile methodologies, for instance with respect to the time to market. The results of our study moreover suggest that the complexity increases with the introduction of Water-Scrum-Fall although agile methodologies usually promise the opposite. The perceived complexity might propose a barrier to the sustainability of the methodology and therefore should be monitored in the long term. We finally observed that in hybrid agile-traditional methodologies the degree of individual autonomy is quickly being restricted to an extent that agile practices are compromised. This potential danger ought to be carefully monitored throughout the project.

We have taken several precautions to ensure the validity and reliability of our findings. To ensure the reliability of our findings, we have carefully chosen a representative case, in which the Water-Scrum-Fall methodology was faithfully implemented. We moreover followed established qualitative research procedures to avoid distortions when designing and conducting our study. To ensure the internal validity of our findings, we firstly decided to use data triangulation. Accordingly, we interviewed owners of different roles and checked their statements for consistency. Secondly, we have taken care that all of the interviewees have sufficient expertise both in the Waterfall model and the Water-Scrum-Fall methodology to avoid a selection bias. To assess construct validity, we have analyzed the data for consistency and discrepancies. The analysis revealed that we obtained highly convergent data that quickly led to theoretical saturation. Nevertheless, there exist several limitations in the light of which the results have to be interpreted. In particular, the internal validity of our results is still limited because we have only interviewed members of Water-Scrum-Fall projects so far. It could further be strengthened by contrasting the statements with those of members of traditional or purely agile development teams. Furthermore, the external validity of our results is limited as we so far only studied the projects of one company in which Water-Scrum-Fall, a specific hybrid agile-traditional methodology, was used. The generalizability of our results is hence limited in both respects. The expressive power of our results might furthermore be criticized because of the comparably small size of the selected sample. As we obtained highly convergent data, however, we deem the results nevertheless to be useful (Given 2008). When adopting an exploratory, qualitative approach to investigate a new field, researchers are typically not yet looking for generalization from a sample to a population. They are rather looking for plausibility and logical reasoning through the development of a theory, which is to be verified in other settings later on (Given 2008). This is the type of contribution that we wanted achieve in our study.

**Conclusions**

Although hybrid agile-traditional methodologies such as Water-Scrum-Fall meanwhile belong to the widely used development approaches in practice, research so far has rarely investigated which factors determine their acceptance and how these factors are perceived. In this paper, we have presented the results of an exploratory study, in which we investigated how team members perceive hybrid agile-traditional methodologies using Water-Scrum-Fall as an example. Using the DOI theory as a lens of analysis, we identified several acceptance factors and analyzed their perception in detail. Based on our findings, we
refined the generic constructs of the DOI theory and provided a dedicated theory to explain the acceptance of hybrid agile-traditional methodologies.

The identified factors provide a conceptual basis to support additional efforts, in which the benefits and drawbacks of hybrid agile-traditional methodologies ought to be examined more closely. Future research particularly ought to validate and extend our findings in different contexts. To find out if the identified drawbacks may indeed pose acceptance barriers or if they are dominated by the perceived benefits, future research should also investigate how the identified factors impact the acceptance in comparison to each other. Apart from such efforts, future research should also examine the interrelationship and dependencies of agile practices more closely. The results of such endeavors might provide valuable insights in how far agile practices can be independently adapted, composed, or combined with traditional development practices to form customized, individual methodologies according to the specifics and peculiarities of the adopting unit.

Agile methodologies have primarily been studied as a distinctively novel approach to software development so far. In practice, however, agile methodologies are frequently combined with traditional approaches (Komus 2014b). Likewise, examining their potential for such a form of hybridization ought to become a research priority to unveil the true potential of agile methodologies. With the acceptance factors and their perceptions presented in this paper, we hope to provide a starting point for such endeavors.

Appendix: Supplementary Interview Statements

In the following, we depict additional interview results that corroborate our interpretations and further strengthen the empirical evidence. To increase readability, we decided to separate these repetitive statements from the already depicted results as they do not convey new information.

Regarding the meeting of requirements (F1), we also gathered the following statements: “For me, the biggest advantage is that you always have testable project states in between [A1.1]. So you can more quickly spot any faults [A1.2]” (Product Owner 5). “You realize much more quickly if a requirement is not met [A1.3]” (Product Owner 7). “The positive thing is that you try to develop something in much shorter cycles and that you are able to identify faults much sooner than if you develop for 4-5 months and start tests only afterwards [A1.4]” (Developer 2). “Requirements are better met because they are more fine-grained and you have a dedicated acceptance test for each requirement [A1.5]” (Developer 3). “There is much more certainty due to a better understanding of the requirements by the business department and the developers [A1.6]. You are able to identify faults much earlier, not only at the end of the project [A1.7]” (Developer 5). “The early acceptance at the end of each Sprint is very helpful to better meet the requirements [A1.8]” (Developer 6). “When the developers communicate with the business department during the Sprints, it becomes much clearer what the customer wants [A1.9]. I think we do more precisely what the customer wants [A1.10]” (Scrum Master).

These statements corroborate our claim that the requirements are perceived to be better met [A1.3, A1.5, A1.10] and that this perception positively affects the acceptance of Water-Scrum-Fall [A1.1, A1.4]. The additional statements also point to the continuous acceptance testing of deliverables [A1.2, A1.4, A1.5, A1.7, A1.8] and the collaborative discussion of requirements [A1.6, A1.9] as effective mechanisms.

With respect to the economic value (F2), we received the following additional statements: “I would say that Water-Scrum-Fall is more expensive due to the administrative and collaborative efforts throughout the process. Thanks to the continuous testing of the results, the quality also is much better, however [A2.1]. From a customer’s perspective, the effort pays off [A2.2]” (Developer 2). “You are never on hold and have to search for work. You can just take the next free task and start to work on another requirement [A2.3]. That is one of the biggest advantages [A2.4], although I don’t think that the costs are reduced due to the additional efforts spent into daily meetings etc.” (Developer 3). “You communicate a lot with the others. Everyone knows your status and you have much less idle times [A2.5]” (Supporter 2).

The statements strengthen our claim that the seemingly increased economic value is found to be a relative advantage of Water-Scrum-Fall [A2.2, A2.4]. The statements also confirm that the use of the agile practices mentioned in section 4 lead to gains in efficiency [A2.3, A2.5] and effectiveness [A2.1].

Concerning the transparency (F4), we moreover collected the following answers: “Personally, I find the institutionalized transparency very valuable [A4.1]” (Product Owner 1). “I like that you always know the
Ensuring the acceptance of the agile process is the big challenge. The communication is stronger. A, we collected the following additional statements:

- More personal,
- It is difficult to live,
- The individual tasks are easier to achieve

Instead, we have to explain ourselves all the time. This limits our agility and easily creates a feeling of distrust [A6.3]. “Personally, I have an aversion against too much control [A6.4]. And you have a lot of control in this project [A6.5]” (Product Owner 1).

They corroborate our claim that the self-responsibility is found to be limited in Water-Scrum-Fall projects [A6.6.6, A6.5] and that the limited self-responsibility seems not to be compatible with the way the team members prefer to work [A6.4]. Apparently, this compromises a team’s agility and negatively influences the acceptance of the Water-Scrum-Fall methodology [A6.1, A6.3].

With reference to the complexity (F7), the following additional statements were recorded: “I think that the planning effort is underestimated. It takes a significant amount of time before you know what to do in a Sprint [A7.1]” (Product Owner 7). “The individual tasks are easier to achieve [A7.2]. However, there is a significant overhead generated by controlling measures such as the daily meetings [A7.3]” (Developer 1). “I think that the process complexity increases because you have to plan every Sprint (what to do, how long it will take etc.) [A7.4]. [...] I find the present hierarchy to be rather counterproductive. It is important to get the project manager into the boat. Otherwise, it is likely that the information will not be transmitted as intended, which causes frictions [A7.5]” (Developer 3).

These statements corroborate our claim that the experts have a mixed attitude regarding the complexity of the development process [A7.2, A7.4]. They also support our impression that the agile practices of Water-Scrum-Fall simultaneously reduce [A7.2] and increase [A7.3, A7.1, A7.4] complexity. Finally, there is further support for our assumption that the traditional practices additionally introduce complexity [A7.5].

Concerning the required discipline (F8), we also collected the following statements: “For me, a meeting every day is too much. I skipped some of the meetings as I had to finish requirements specifications. I had to prioritize and disregarded some meetings as less important [A8.1]” (Product Owner 1). “I find the daily meetings to be an inconvenience, but you have to attend [A8.2]. Communication is an important factor” (Product Owner 6). “It is difficult to live some Water-Scrum-Fall practices [A8.3]. The bureaucracy is a big problem, especially for the people who have to be the visionaries and ought to stay on top of things [A8.4]” (Developer 4). “Ensuring the acceptance of the agile process is the big challenge [A8.5]. If the people join in and support the new principles, then it is no problem” (Scrum Master).

The additional statements support our claim that the required discipline is perceived as higher [A8.4] and that this perception negatively affects the acceptance of the Water-Scrum-Fall methodology [A8.2, A8.3, A8.5]. The statements also confirm that the interviewees may find it difficult to continuously participate in all meetings [A8.1, A8.2] and to be mainly responsible for the project success [A8.4].
References


