Hybrid Software and System Development in Practice: Waterfall, Scrum, and Beyond

Marco Kuhrmann
Clausthal University of Technology
kuhrmann@acm.org

Philipp Diebold
Fraunhofer IESE
philipp.diebold@iese.fhg.de

Jürgen Münch
Reutlingen University
j.muench@computer.org

Paolo Tell
IT University Copenhagen
pate@itu.dk

Vahid Garousi
University of Luxembourg
garousi@svv.lu

Michael Felderer
University of Innsbruck
Michael.Felderer@uibk.ac.at

Kitija Trektere
Dundalk Institute of Technology
kitija.trektere@dkit.ie

Fergal McCaffery
Dundalk Institute of Technology
fergal.mccaffery@dkit.ie

Oliver Linssen
FOM University of Applied Sciences
oliver.linssen@fom.de

Eckhart Hanser
DHBW Lörrach
hanser@dhbw-loerrach.de

Christian R. Prause
German Aerospace Center
christian.prause@dlr.de

ABSTRACT

Software and system development faces numerous challenges of rapidly changing markets. To address such challenges, companies and projects design and adopt specific development approaches by combining well-structured comprehensive methods and flexible agile practices. Yet, the number of methods and practices is large, and available studies argue that the actual process composition is carried out in a fairly ad-hoc manner. The present paper reports on a survey on hybrid software development approaches. We study which approaches are used in practice, how different approaches are combined, and what contextual factors influence the use and combination of hybrid software development approaches. Our results from 69 study participants show a variety of development approaches used and combined in practice. We show that most combinations follow a pattern in which a traditional process model serves as framework in which several fine-grained (agile) practices are plugged in. We further show that hybrid software development approaches are independent from the company size and external triggers. We conclude that such approaches are the results of a natural process evolution, which is mainly driven by experience, learning, and pragmatism.

CCS CONCEPTS

• **General and reference → Surveys and overviews**; • **Software and its engineering → Software development methods**; *Software organization and properties*; *Designing software*; *Software development techniques*; *Programming teams*;

KEYWORDS

Agile software development; software process; hybrid development approaches; survey

ACM Reference format:
Marco Kuhrmann, Philipp Diebold, Jürgen Münch, Paolo Tell, Vahid Garousi, Michael Felderer, Kitija Trektere, Fergal McCaffery, Oliver Linssen, Eckhart Hanser, and Christian R. Prause. 2017. Hybrid Software and System Development in Practice: Waterfall, Scrum, and Beyond. In *Proceedings of International Conference on Software System Process, Paris, France, July 5–7, 2017 (ICSSP'17),* 10 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Software development is diverse, and companies have to adopt to new technologies and markets quickly. According to Brooks [2], there is no "Silver Bullet" in software development. Hence, software engineers are on the quest for suitable development approaches, yet facing a huge variety of contextual factors influencing the definition of appropriate development processes [4, 14, 26]. To address these factors as well as the increasing number of domains for which software has become a vital part, different software and system development methods have been proposed. These methods implement different philosophies, e.g., stage-gate or plan-based, iterative-incremental or lean, and range in their lifecycle coverage from small task-specific practices (e.g., daily stand-up) to large and comprehensive process frameworks (e.g., the family of V-shaped processes). Furthermore, software development has become key to system development and, therefore, increasingly addresses safety-critical and reliable systems, such as Automotive software, Aerospace systems, or Medical Devices. These domains add standards, norms, and regulations to software and its development.

Hybrid Software Development Approaches. We aim to study hybrid software development approaches (short: hybrid approaches), which we define as follows: *A hybrid software development approach*

ICSSP'17, Paris, France

is any combination of agile and traditional (plan-driven or rich) approaches that an organizational unit adopts and customizes to its own context needs (e.g., application domain, culture, processes, project, organizational structure, techniques, technologies, etc.).

Problem Statement. In 2011, West et al. [24] coined the term "Water-Scrum-Fall" and hypothesized that hybrid development methods will become the standard. A systematic review by Theocharis et al. [20] aimed at collecting evidence to confirm West's claim. They revealed a gap in literature: while research on agile software development is rich, traditional processes are widely ignored in recent research. Hence, data (and evidence) on combination patterns and contextual factors driving the creation of hybrid approaches is missing, e.g., how do standards affect the use of agile methods, or do company size and industry sector matter?

Objective. The goal of our research is to close this gap and to collect data to help determining combination patterns, i.e., which development approaches are used in practice and how are these approaches combined in company- or project-specific development approaches. We further aim to identify context factors influencing the creation of hybrid approaches. We also aim to investigate which software development problems motivate hybrid approaches and if hybrid approaches help solving such problems.

Contribution. The paper at hand presents results from the HELENA[1] study. Differently from the study by Theocharis et al. [20], HELENA is an internationally conducted survey that aims at collecting data to study the use of hybrid approaches. Based on the analysis of 69 responses, we present a list of development approaches as used in practice. We analyze these development approaches for patterns, and test our data for different context attributes. Based on cluster analyses, we identified five major combination patterns. Furthermore, we present the data confirming that using hybrid approaches has become mainstream, and that their use is independent from company size, industry sector, and external standards.

Outline. The remainder of the paper is organized as follows: Section 2 provides an overview of related work. Section 3 describes the research design, and Section 4 presents the results. Finally, we conclude the paper in Section 5.

2 RELATED WORK

There is a long history of studies on the use of software processes. An important step was a special issue of IEEE Software in 2003 in which several authors started collecting knowledge about process use in general and combining processes in particular. For instance, Cusumano et al. [5] surveyed 104 projects and found many projects using and combining different development approaches. Jones' findings [13], which are based on an analysis of data from approx. 12,000 projects, indicate a certain diversity in the development methods used. Despite the variety observed, a pattern seems to be recurrent. Neill and Laplante [18] found in their study that approx. 35% of the participants use the Waterfall model, yet, projects also use evolutionary/incremental approaches—even within particular lifecycle phases. Starting in 2005, a series of independently conducted studies in Germany investigated, among other things, the use of software processes from different perspectives [3, 11, 15] and showed numerous development approaches applied and combined with each other. Recently, Garousi et al. [12] and Vijayasarathy and Butler [23] showed that "classic" approaches like the Waterfall model are (increasingly) combined with agile/lean development approaches. These studies confirm the observation that a huge bandwidth of processes exist and that they are combined with each other.

In 2011, West et al. [24] coined the term "Water-Scrum-Fall" to describe that very combination pattern, which was studied by Theocharis et al. [20]. Their major finding is that few data is available about combined process use in general. Moreover, Theocharis et al. [20] found an extensive knowledge base on agile software development, e.g., Dybå and Dingsøyr [9] or the continuously updated *State of Agile Survey* [22]. However, missing so far is a big picture due to the lack of information about traditional processes.

This missing big picture motivates the research presented in the paper at hand. The goal is to study the use of (hybrid) software development processes in general with respect to the development context (e.g., industry sector, company size) and the different constraints companies face (e.g., standards, norms, and regulations). This paper thus fills a gap in literature by (i) contributing to the body of knowledge on process use, but (ii) providing a more holistic perspective covering traditional and agile/lean approaches alike.

Table 1: Overview of the research questions to be investigated in the HELENA study.

	Research Questions
RQ_1	*Which development approaches are used in practice?* This question aims to study the development approaches for software and system development used in practice.
RQ_2	*How are the different development approaches combined?* This question aims to study how the different development approaches are combined, i.e., which classes of development approaches are present and if there are any observable usage patterns.
RQ_3	*How do external standards trigger hybrid approaches?* This question aims to study whether external standards trigger companies to implement hybrid approaches, i.e., to provide a process that fulfills compliance requirements as well as provides sufficient freedom to developers.

3 RESEARCH DESIGN

Section 3.1 presents research questions and hypotheses, followed by a description of the instrument in Section 3.2. Data collection and analysis procedures are presented in Section 3.3 and 3.4. Section 3.5 describes the procedures to increase the validity of our results.

3.1 Research Questions and Hypotheses

Our research is driven by accepting West's "Water-Scrum-Fall" hypothesis, and we accept that hybrid approaches have become reality and shape todays software system development [20]. Therefore, this research aims at studying what a hybrid approach is after all, how and why those approaches are developed, and whether hybrid approaches fulfill the expectations of practitioners. For this, we define the fine-grained research questions in Table 1. In addition, we aim to study the context factors *company size* and *external triggers*

[1] HELENA: Hybrid dEveLopmENt Approaches in software systems development

Table 2: The HELENA questionnaire lists the questions and question groups (conditional questions for the different paths are omitted in this table). The table lists the question scales, and if applicable the number of options and free-text options.

No.	Group	Question	Scale	#opt	FT
1.	M	What is your organization's size?	SC	5	
2.	M	What is the main business area of your company?	MC	7	✓
3.	M	Do you participate in distributed software projects?	SC	4	
4.	M	In which country are you personally located?	FT		
5.	M	In which application domain are you most frequently involved?	MC	17	✓
6.	M	Which role are you most frequently assigned to?	SC	9	✓
7.	M	In your projects, a software failure potentially can: *(option list)*	MC	8	✓
8.	PU	Does your company define a company-wide standard process for software and system development?	SC	3	
9.	PU	Which of the following development approaches and practices do you use?	MC	40	✓
10.	PU	Do you combine different development approaches?	YN		
11.	PU	For the following standard activities in your projects, please indicate to which degree you carry out these activities in a more traditional or more agile manner. *(option list comprises 11 categories)*	LI		
12.	PU	What is the main motivation for this particular combination of development approaches?	FT		
13.	PU	How were the combinations of development approaches in your company developed?	MC	3	✓
14.	PU	How do you select your project-specific development approach?	MC	6	✓
15.	PUS	Which external standards are implemented in your company?	FT		
16.	PUS	Why have you implemented the aforementioned standards?	MC	3	✓
17.	PUS	How is the compliance of the development process assessed?	MC	5	✓
18.	PUS	Does agility challenge the implementation of the standards you have to apply?	YN		✓
19.	PUL	Is your development approach continuously improved?	SC	5	✓
20.	PUL	What is your motivation for implementing an improvement program?	MC	5	✓
21.	PUL	Is your company, unit or project certified?	YN		✓
22.	PUL	What are the goals of your improvement program? *(option list comprises 12 categories)*	LI		
23.	E	Based on your experience, please rate the following statements: *(option list comprises 6 categories)*	LI		
24.	E	Based on your experience, can you name problems occurred regarding your current process and your current application domain?	FT		
25.	E	Do you have any further comments or issues not addressed so far?	FT		

Legend for scales: YN=yes/no, SC=single choice, MC=multiple choice/select, LI=5-item Likert scale, FT=free text.
Legend for groups: M=metadata, PU=process use, PUS=process use and standards, PUL=process use in the lifecycle, E=experience.

Table 3: Hypotheses and variables (ref. to questions (Q) from Table 2 and hypothesis (H) in which the variable is used).

Hypotheses		
H1: The use of hybrid approaches depends on the company size.		
H2: The use of external standards depends on the company size.		
H3: The use of hybrid approaches depends on external standards.		
H4: The use of hybrid approaches depends on external triggers.		
Variable	**Q**	**H**
$cs \in \{1 \ldots 5\}$ (company size)	1	H1, H2
$ha_{use} \in \{0, 1\}$ (hybrid approach use)	10	H1, H3, H4
$std_{use} \in \{1 \ldots 3\}$ (ext. standard use)	8	H2, H3
External trigger:	16	H4
$et = \begin{cases} 2 & \text{if } trig_{ext} = \text{true} \\ 1 & \text{if } trig_{prj} = \text{true} \vee trig_{pol} = \text{true} \\ 0 & \text{otherwise} \end{cases}$		

to investigate drivers for implementing hybrid approaches. Table 3 presents the hypotheses defined for this purpose.

3.2 The Survey Instrument

To answer the research questions, we developed the questionnaire shown in Table 2. The questionnaire is grounded in several input sources: the Success Study [3] and the IOSE[2] study [11] (both from Germany) built the foundation on which the *3ProcSurvey* [15] was developed. These studies together with our literature review [20] form the basis for the HELENA study. The questionnaire developed for HELENA is mainly grounded in [15], but also borrows from the annual *State of Agile Survey* [22], which we used to enrich the collection of development approaches (the *3ProcSurvey* only contained a limited list). The study as such targets practitioners and aims at (i) collecting quantitative data regarding the (general) process use and (ii) collecting qualitative data on experiences. The context is further set by constraints originating from external standards, norms, and regulations to be applied to organizations as a whole as well as to projects. Hence, the questionnaire covers organization-, project- and personal experience levels.

The questionnaire was developed incrementally to increase the validity of the instrument. In the first stage, the questionnaire from the *3ProceSurvey* was used and analyzed for reusable assets. In the second step, the questionnaire was initially crafted and tested in two external organizations[2] with in total 15 subjects [16].

After the trial phase, the questionnaire was finalized and published. The final questionnaire consists of max. 25 questions (excluding conditional selectors; Table 2) of which: seven questions aim at collecting metadata; seven questions aim at collecting information about general process use; four questions to study process use in the context of standards, norms, and regulations; four questions to investigate the use of processes and standards in the process lifecycle; and, three questions to gather general experiences. The questionnaire was designed to be manageable within 10–20 minutes, depending on the actual path through the questionnaire.

[2]The German Aerospace Center (DLR) and FOM University of Applied Sciences for Economics and Management. Researchers from both institutions were involved in the questionnaire's development and quality assurance before running the tests locally at their institutions.

3.3 Data Collection Procedure

The first stage of the HELENA survey was accepting answers from May to June 2016. A simple questionnaire design based on Google Forms was used. We opted for a convenience sampling strategy [25], and posted the survey to a number of mailing lists of IT clusters and networks, and we used LinkedIn, Twitter, Facebook, Xing, and ResearchGate to promote the survey within the relevant communities. Since one of our major goals in this first and exploratory stage was gaining broad visibility, we intentionally sacrificed the ability to calculate response rates (Section 3.5).

3.4 Data Analysis Procedure

To analyze the data, we utilized several methods to provide answers to the research questions. For all research questions, we use descriptive statistics, e.g., to provide tables and charts for process use and process selection. Furthermore, we applied cluster analysis and hypothesis testing to analyze our data set. To answer RQ_2, we applied a multi-staged cluster analysis: As a first step, we applied an *Affinity Propagation Clustering* (AP; [10]) to search for general structures in the result set. In the second step, we applied a *Spectral Clustering* (SC; [19]) to split the result set in two subsets: Af_{high} and Af_{low} to sort those development approaches with high affinity and those with little/no affinity. Af_{low} was excluded from further analyses. In the third step, we applied AP(Af_{high}) to determine the number of centers c in Af_{high} for further analysis, and applied SC(Af_{high}, c) to determine the final clusters. In addition, we applied SC(Af_{high}, 2) to investigate, whether the two "opposite worlds", i.e., traditional and agile, can be constructed from the result set. Finally, to answer RQ_3 and to test the hypotheses H1-4, we applied the Pearson's χ^2 test, with $p \leq 0.05$ to find (no) support for the respective hypotheses.

3.5 Validity Procedures

To improve the validity and to mitigate risks, we implemented different measures: First, our research is grounded in previously conducted studies, in particular [11, 15] that provided a basic set of questions. To find a set of development approaches of interest, we also ground our research in lists of approaches (e.g., offered by the *State of Agile Survey* [22]), which have been combined to complement the findings of our previously conducted research. A design group consisting of three researchers developed the initial version of the instrument (Sect. 3.2). Two more researchers performed a quality assurance and conducted an external test of the questionnaire to test the general feasibility [16]. Finally, the data analysis was handed over to another team of two researchers.

Another risk is related to the data collection strategy: Since one of the main goals of the study is to build a quantitative basis for our research, we accepted the risk of loosing full control in terms of sampling, response rate, and so forth, and—for HELENA's first stage—we intentionally opted for an open call for participation (convenience sampling; [25]) to maximize the number of datapoints.

4 RESULTS

We present the results from the survey. We start with an overview of the study population in Section 4.1, before we present the findings according to the research questions. Finally, we discuss our findings in Section 4.5.

Table 4: Overview of the company size and the roles participants have (n=69).

	Micro	Small	Medium	Large	Very Large	Σ	%
Project/Team Manager	4	3	5	4	3	19	27.5
Architect	3	2	–	1	2	8	11.6
Tester	1	1	1	2	3	8	11.6
Product Manager/Owner	3	1	1	1	1	7	10.1
Quality Manager	1	–	2	2	2	7	10.1
Developer	–	5	–	1	–	6	8.7
Other	–	1	1	1	3	6	8.7
Analyst/Req. Engineer	–	1	–	1	1	3	4.3
Trainer/Coach	–	–	1	–	2	3	4.3
Scrum Master	–	–	1	1	–	2	2.9
Σ	12	14	12	14	17	**69**	
%	17.4	20.3	17.4	20.3	24.6		**100**

4.1 Study Population

In total, we received 69 complete responses[3] from approx. 15 countries. Responses regarding the company size cover all categories, i.e., the result set contains answers ranging from micro-sized companies to very large companies (Table 4). Furthermore, approx. 2/3 of the respondents state that they work in a distributed fashion, in particular: 18.8% work distributed in the same country and 20.3% in the same region, and 26.1% work globally distributed. As found in previous studies [15], (globally) distributed work has become reality for companies of all sizes. Participants were asked for the roles that they are most frequently assigned to. Table 4 shows the outcomes and relates the roles to the company size. Project/team manager is the most frequently stated role, followed by architects, testers, product managers/owners, and developers. Another 8.7% was categorized as "other" (e.g., safety managers, compliance managers, and C-level managers).

Figure 1 provides an overview of the industry sectors in which the participants are active. In total, the survey returned 167 selections, i.e., several participants are engaged in multiple sectors. The figure shows approx. 29% of the participants are engaged in "Web Applications and Services", followed by "Medical Devices and Health Care", and "Public Sector/Public Contracting". Among the sectors categorized as "Other", participants named "Energy", "Traffic Management Systems", or "Industrial Control Systems".

4.2 RQ1: Development Approach Use

The first research question aims to study the use of different development approaches in general. As described in Sect. 3.2, the participants were presented a list of 40 development approaches to select—including a free text field. Figure 2 shows the participants' selection. In total, 729 selections have been made by the 69 participants. The figure shows (i) a ranking of the different approaches based on the frequency and that (ii) the participants use the different approaches in combination.

[3] A code book and raw data are available from: https://goo.gl/MK0mYZ Note that the raw data is available in an anonymized from only.

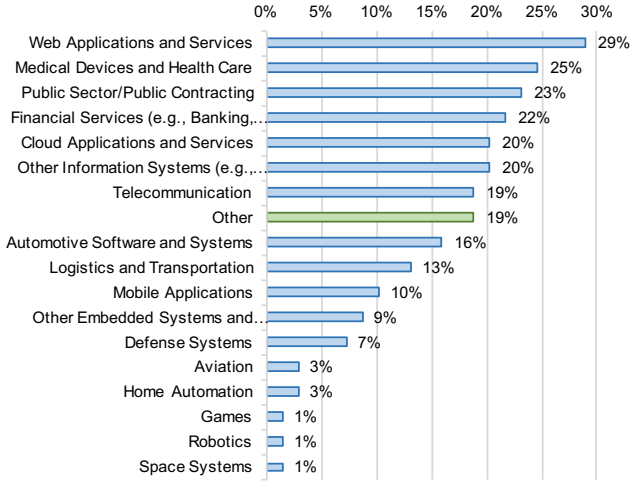

Figure 1: Overview of the industry sectors as stated by the participants (n=69).

To further analyze the result set, we defined the two main categories *method* and *practice*, and within each category, we used the three sub-categories *traditional*, *agile*, and *both*. The categorization is based on the definitions provided by Diebold and Zehler [8]. Figure 2 shows the resulting classification of the different approaches (main categories marked along side the label, sub-categories color coded), and shows that—starting with 20% share—the participants use development approaches of all kinds, i.e., traditional like the Waterfall model, agile like Scrum, and generic (both) approaches, such as code reviews. Among others, Figure 2 shows that more than a half of the participants (53.6%) implement Scrum and more than a third (34.8%) implement a Waterfall/Phase Model. That is, West's claim that the "Water-Scrum-Fall" will become reality for software system development [24] is confirmed by our result set. This large-scale combination is complemented by a number of small practices, which also supports our claim that practices have become the building blocks of process customization [20].

4.3 RQ2: Combination of Development Approaches

The second research question aims at studying the way of implementing combinations of different development approaches. For this, we categorized the different development approaches (Section 4.2) and, based on this categorization, analyzed the dataset from different perspectives.

4.3.1 Process Use in Projects (Self-Evaluation and Company Size). To analyze how the different approaches are combined and used in practice, we analyzed the result set from two perspectives: the first perspective is given by a self-evaluation of the participants on how they implement a set of given project disciplines. In the second perspective, we analyze the process use in the context of the different company sizes.

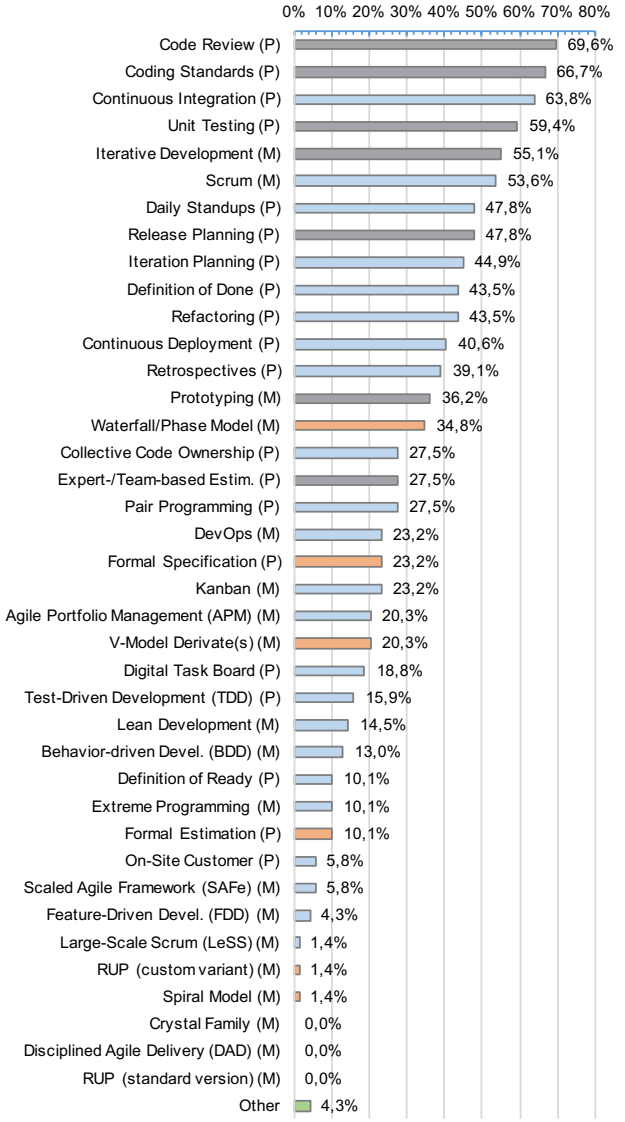

Figure 2: Overview of development approaches used in practice (n=69; M: method, P: practice; colors: agile: blue, traditional: red, generic: grey).

SWEBoK-based Self-Evaluation. The analysis concerning the implementation of the different project disciplines is structured according to the SWEBoK [1]. The SWEBoK defines 11 disciplines addressing the different project lifecycle phases[4], e.g., project management, requirements engineering, architecture and design, and implementation/coding. In the questionnaire, we asked the participants to decide[5] whether they implement a discipline more traditionally or more agile.

[4]Since we are interested into the project (core) business, disciplines addressing cross-cutting concerns, e.g., economics were not included in the questionnaire.
[5]We did not provide a definition of what to consider traditional or agile. The researchers performed the categorization during the data analysis using a pre-defined schema [8].

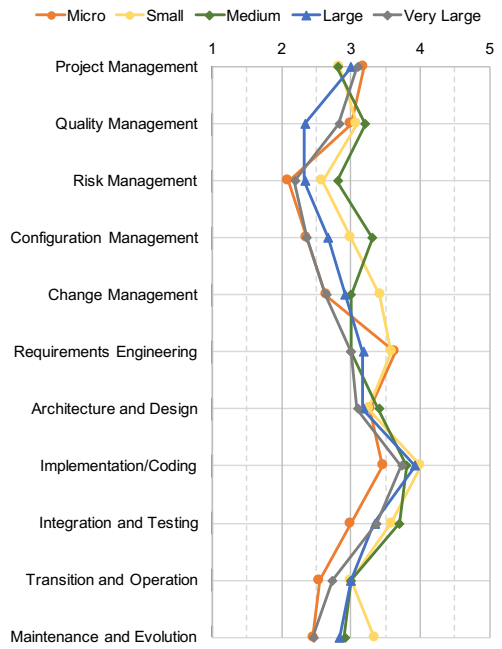

Figure 3: Participant rating on the implementation of the SWEBoK disciplines in their projects (1=fully traditional to 5=fully agile; n=56).

Figure 3 shows how the participants rate their way of implementing the SWEBoK disciplines using the averaged ratings grouped by company size. The results show that most of the disciplines are implemented in a balanced way, i.e., companies aim at combining the "best of both worlds". Deviations can only be observed in *Risk-* and *Configuration Management*, which have a tendency towards a more traditional implementation and, on the other hand, *Implementation/Coding* and *Integration/Testing*, which are implemented in a more agile way. The figure also shows that the development-related trend towards agile software development and testing is present regardless of the company size. Furthermore, a trend towards more traditional processes for larger companies cannot be observed; not even for "classic" management disciplines.

Does Company Size Matter? To confirm the trend observed in Figure 3, we normalized the selected development approaches per company size. Table 5 shows that (on average) companies' processes are composed of 5.5% traditional and 16.0% agile methods; agile practices account for 40.0%, and generic (both) methods and practices account for 9.1% and 25.7% respectively. That is, companies combine traditional and agile approaches regardless of the company size. Similar to the company size in Table 5, we analyzed the use in relation to the industry sector (Figure 1). The outcome is the same, i.e., companies combine different development approaches regardless of the industry sector.

4.3.2 Process Use (Quantitative Analysis). For a deeper analysis, we created a coincidence matrix, which is also the basis for Table 5. This matrix contains a "head count" of the pairwise combination of

Table 5: Use of development approaches (in % incl. mean, mean absolute deviation; categorized, company size; normalized by selected approaches *n*).

Comp. Size	Method			Practice				
	T	A	B	T	A	B	Other	n
Micro	6.6	15.4	13.2	5.5	33.0	25.3	1.1	91
Small	3.2	16.7	7.7	0.6	47.4	24.4		156
Medium	4.9	16.2	9.2	1.4	40.8	26.8	0.7	142
Large	4.3	16.0	7.4	3.7	40.5	27.6	0.6	163
Very Large	8.5	15.8	7.9	5.1	38.4	24.3		177
Mean	5.5	16.0	9.1	3.3	40.0	25.7	0.8	
MAD	1.6	0.3	1.7	1.8	3.5	1.2	0.2	

Legend: T=Traditional, A=Agile, B=Both.

Table 6: Low-affinity cluster excluded after applying first spectral analysis with two centers.

Development Approach	#
Rational Unified Process (custom variant)	1
Rational Unified Process (standard version)	0
Spiral Model	1
Disciplined Agile Delivery (DAD)	0
Feature-Driven Development (FDD)	3
Crystal Family	0

approaches, e.g., 28 participants state that they combine *Scrum* and *Continuous Integration*. Using this combination matrix and as described in Section 3.4, we performed the AP and SC cluster analyses. The first AP analysis resulted in six clusters, which revealed development approaches with little or no affinity to other approaches. Therefore, we split the result set into two subsets Af_{high} and Af_{low} using an SC analysis with two centers. Table 6 summarizes the approaches that were sorted into Af_{low}, and which we excluded from further analyses.

Running the AP analysis on the subset of the remaining 33 development approaches in Af_{high} resulted into five centers and, therefore, we applied the trained SC algorithm again searching for clusters around the five centers. The resulting clusters are illustrated in Table 7 with sizes ranging from four to eight elements. Given the categorization (traditional/agile) of the approaches (Figure 2) all five clusters show a mixture of traditional, agile, and generic (both) methods and practices, e.g., *Cluster 5-4*. However, remarkable is *Cluster 5-5*, which shows a collection of approaches as one would expect from a practically applied Scrum adaptation [7].

Therefore, in order to investigate if we are able to construct the two "extremes" (i.e., agile and traditional approaches) from our result set, we executed the SC algorithm again with two centers to split the result set. Table 8 shows the outcome: while on the one hand, the algorithm isolated elements characterizing an "almost pure" Scrum-centered development process (*Cluster 2-2*), one the other hand, it was not able to construct a pure traditional cluster. In fact, *Cluster 2-1* represents mixture of development approaches from all categories.

Table 7: High-affinity clusters of development approaches after applying spectral analysis with five centers.

Cluster 5-1	Cluster 5-2	Cluster 5-3	Cluster 5-4	Cluster 5-5
Size=4	Size=5	Size=8	Size=8	Size=8
XP	V-Model Derivate(s)	Iterative Development	Waterfall/Phase Model	Scrum
SAFe	APM	Continuous Deployment	BDD	Collective Code Ownership
Formal Estimation	Lean Development	Continuous Integration	DevOps	Daily Standups
On-Site Customer	Definition of Ready	Iteration Planning	Kanban	Definition of Done
	Expert-/Team-based Estimation	Unit Testing	TDD	Digital Task Board
		Retrospectives	Prototyping	Pair Programming
		Release Planning	Formal Specification	Refactoring
		Coding Standards	LeSS	Code Review

Legend: XP: Extreme Programming; SAFe: Scaled Agile Framework; APM: Agile Portfolio Management; BDD: Behavior-driven Development; TDD: Tesst-Driven Development; LeSS: Large-Scale Scrum.

Table 8: High-affinity clusters after applying spectral analysis with two centers.

Cluster 2-1	Cluster 2-2
Size=17	Size=16
V-Model Derivate(s)	Scrum
Waterfall/Phase Model	Iterative Development
APM	Collective Code Ownership
BDD	Continuous Deployment
DevOps	Continuous Integration
Extreme Programming	Daily Standups
Kanban	Definition of Done
LeSS	Iteration Planning
Lean Development	Pair Programming
SAFe	Refactoring
Prototyping	Retrospectives
Formal Estimation	Expert-/Team-based Estim.
Formal Specification	Code Review
Definition of Ready	Coding Standards
Digital Task Board	Unit Testing
On-Site Customer	Release Planning
TDD	

4.3.3 Process Use (Institutionalization). Finally, we study the way the development approach is composed and used in projects. Therefore, we asked the participants for company-wide policies, development of their particular development approach, and decision-making in regard to the process selection. Table 9 shows that more than the half of the participants (52.2%) state to be obliged to follow a standard process. Another 20.3% have a standard at the level of business units, and the remaining 27.5% state to select the actual process on demand.

That is, more than 3/4 of the participants state to have rules concerning the process use. After collecting the data regarding the development approaches used (cf. Section 4.2), we asked the participants if they combine the different approaches within projects[6], and 56 participants stated to explicitly combine different development approaches. We asked the participants how their particular combination of the different development approaches was developed, and a majority of 83.9% (Table 10) stated that the development approach emerges from experience and learning from past projects.

[6]Rationale: Accepting that companies might run projects for different clients, different methods might be known at the company level, but there is the possibility of an exclusive use on a per-project or per-client basis, i.e., multiple approaches but no combined use. Table 2, question 10 addresses this situation.

Table 9: Does the company have a company-wide standard process for software development? (n=69)

Option	#	%
All projects are operated according to the same (customized) standard process(es)	36	52.2
Each business unit has its own approaches, which all projects of this unit have to follow	14	20.3
Each project can individually select the process to be used	19	27.5

Table 10: How was the particular combination of development approaches created? (n=56)

Option	#	%
Planned in a process improvement program	11	19.6
Evolved from past projects over time	47	83.9
Situation-specific	13	23.2

Table 11: Overview of the actual process selection and tailoring in particular projects. (n=56)

Option	#	%
Practices and methods are selected in the project on demand	30	53.6
Practices and methods are selected according to customer demands	14	25.0
A project manager tailors the process in the beginning of a project	19	33.9
Project-specific process selection and tailoring follows defined rules	20	35.7
Project-specific process selection and tailoring is supported by tools	13	23.2
The process is not tailored at all	12	21.4
Other	2	3.6

Table 12: Results of question 19: Is your development approach continuously improved? (n=69)

Option	#	%
Yes, SPI as part of the development process	26	37.7
Yes, explicit SPI based on standards[a]	11	15.9
Yes, explicit SPI (non-standardized/custom)	9	13.0
Yes, sporadic activities, but SPI program	9	13.0
No, no SPI program implemented (at all)	13	18.8
Other	1	1.4

[a]: with standards, we refer to CMMI, ISO/IEC 15504 and similar

Finally, we were interested in the decision-making process, i.e., how are development approaches selected and by whom. The results summarized in Table 11 show that more than the half of the participants (53.6%) state selecting the methods and practices in the project on demand. Furthermore, 35.7% state that the project-specific tailoring follows defined rules and that a project manager carries out the tailoring in the beginning of a project (33.9%). Hence, this shows that standards exist and that they are (at least initially) applied to a project. However, during the project, the configuration of methods and practices might change.

As shown in Table 10, we asked participants, whether their companies implement a Software Process Improvement (SPI) program to investigate if the development approach used emerges from a planned and/or controlled procedure. Table 12 shows that only 15.9% have implemented an SPI program based on standardized approaches like ISO/IEC 15504 or CMMI, but the majority states implementing an in-process improvement (37.7%) or no SPI program at all (18.8%).

4.4 RQ3: Triggers for Hybrid Approaches

While the first two research questions are concerned with studying which development approaches are used and how they are combined with each other, the third research question aims to investigate the triggers for creating a hybrid approach. For this, we argued that external standards, norms, and regulations are major triggers introduced by the increasing complexity of software-intensive systems, especially by emerging industry sectors engaged in Cyber-physical systems (see also the hypotheses in Table 3). In this section, we study standards and whether standards foster (or even enforce) the creation of hybrid development approaches.

4.4.1 Use of Standards. We asked the participants, if they have to adhere to external standards (and to which ones), and if agility is a challenge in the light of those standards. Figure 4 illustrates the result as a word cloud. The participants named 45 different standards and standards-related methods of which ISO 9001 was the most frequently mentioned (12 mentions), followed by ISO 27001 (5 mentions), and ISO 13485 (4 mentions). The summary in Figure 4 shows a considerable variety of standards, and Table 13 shows whether using standards, norms, and regulations challenges the companies in implementing agility.

In total, 41 participants stated external standards, norms, and regulations relevant for their companies—even micro-sized companies have to adhere to standards. The column "Challenged by Agility?"

Figure 4: Summary of the standards used (n=41, 45 named standard/standard-related methods).

Table 13: Do standards, norms, and regulation challenge the implementation of agility? (n=41)

Company Size	Standards		Challenged by Agility?			
			Agile: Yes		Agile: No	
	#	%	#	%	#	%
Micro	4	33.3	2	16.7	2	50.0
Small	7	50.0	3	21.4	4	57.1
Medium	6	50.0	3	25.0	3	50.0
Large	11	78.6	4	28.6	7	63.6
Very Large	13	76.5	11	64.7	2	15.4

Table 14: Results of the Pearson's χ^2 (with $p \leq 0.05$).

Hypothesis	χ^2	df	p	No/Support
H1	4.2932	4	0.3678	No support
H2	8.5227	4	0.0742	No support
H3	1.9453	1	0.1631	No support
H4	2.1484	2	0.3416	No support

of Table 13 refers to the 41 participants stating that they have to use standards and shows that 23 out of 41 participants (approx. 53%) are challenged and that the very large companies seemingly face the biggest problems. The remaining 18 participants state the opposite, i.e., implementing agility is not considered a problem, but fulfilling the requirements of the standards is.

4.4.2 What Triggers Hybrid Approaches? Finally, we aim to investigate the triggers in more detail. In Table 3, we formulated four hypotheses to test. Table 14 presents the results from the tests using Pearson's χ^2. The table shows that we found no support for any of the hypotheses. In particular, we have no evidence for H1, i.e., the use of hybrid approaches does not depend on the company size. As also shown in Table 5, our data suggests that companies use hybrid approaches—regardless of their size. We have no evidence to support H2, i.e., the obligation to use external standards does not depend on the company size. Our data (Table 13) suggests that standards have become relevant to companies of all kinds. Data also suggests that the creation and use of hybrid approaches is not triggered by external standards. That is, we have no evidence supporting H3. Finally, we also have no evidence for H4, i.e., the use of hybrid approaches does not depend on external triggers. Especially for H3 and H4, we can find rationale in the open questions. For instance, Table 10 shows 83.9% of the participants stated that the hybrid approach has emerged from experience. A qualitative

analysis of the open questions reveals that 18 participants state the current hybrid approach being a result of pragmatically applied development approaches that evolved over time.

In our dataset there was no proof of dependance for any of the hypotheses. Hence, we conclude that the use of hybrid approaches can be considered independent of the company size (also of the industry sector, cf. Section 4.3.1), and external standards do not trigger the creation and use of hybrid approaches. Moreover, our data suggests that a hybrid approach is a result of a natural evolution of the different development approaches used by companies. But, even though the data presented supports this claim, a deeper investigation remains subject to future work.

4.5 Discussion

This study is grounded in observations made in Theocharis et al. [20]. In particular, this study aimed at collecting data to allow for closing a gap identified, namely missing data about process use in general including agile *and* rich processes. In addition—and in the light of modern software system development—we added an extra dimension by including standards, norms, and regulations to our research.

In this context, the HELENA study produced a list of development approaches as used in practice (Table 1, RQ_1). Compared to related studies, e.g., [5, 11, 15, 22, 23], our results show a good match. Traditional as well as agile methods and practices are present, and the trends emerging from our data are in line with previous studies, e.g., [15, 23]. West et al. [24] named this trend "Water-Scrum-Fall", and our results support West's claim that the "Water-Scrum-Fall" has become reality. For instance, based on the descriptive analysis of our categorized data, we could show that companies tend to implement a balanced software development approach that includes traditional as well as agile elements (Figure 3). The overall tendency shows risk management and configuration management implemented in a more traditional way, while the activities around requirements engineering, implementation, integration, and testing tend to be implemented in a more agile fashion.

Our data shows no evidence supporting the claim that the implementation of hybrid approaches dependents on the company size (Table 5 and H1) or on the industry sector (Table 1, RQ_3). In the detailed analysis, we also found no indication that standards or external triggers drive the development of hybrid approaches (Section 4.4.2, H2-4). However, our data also shows that companies are active in business areas enforcing requirements to adhere to standards on the companies, and that notably implementing agility in such standards-driven environments challenges companies (Table 13). While several industry studies (e.g., [17, 21]) argue that hybrid approaches are caused by a reluctance of the management to buy-in agile, we argue that implementing hybrid approaches is also an attempt to address multiple challenges, such as balancing management and developer expectations regarding the development process or implementing (rigorous) standards by, at the same time, keeping high levels of flexibility. A specific challenge to be addressed by hybrid approaches is scalability of agile methods. For instance, Murphy et al. [17] found Scrum to be considered the most favorite development approach at Microsoft. However, the suitability of agile methods for large projects was considered critical (substantial communication effort and overhead, reluctance of the management to accept the agile approach). This finding is supported by Melo et al. [6], who found the management-related agile practices to be either adopted to a large extent or completely rejected, while development-centered practices have become well-accepted.

From our data, we conclude that hybrid approaches can be considered a good compromise that helps balancing the needs of different stakeholder groups. As for instance found in [6, 17], management has different requirements and expectations concerning the development process than developers. A cluster analysis (Sect. 4.3.2, Tables 7 and 8) shows that hybrid approaches include development approaches from both worlds (traditional and agile). This combination addresses the needs of managers (more traditional methods and frameworks to support "classic" management tasks) and developers (freedom to select those practices best fitting the respective context). Tables 7 and 8 show that hybrid approaches are not limited to combinations of traditional and agile methods. Moreover, especially Table 8 shows that even agile methods are not implemented by the book. Rather, different practices are combined to address practical needs (Table 1, RQ_2; see also [7]). To a large extent, such combinations are developed on a per-project basis (Table 11 and [15]) and are continuously improved within the projects (Table 12). The inspection of the pair-wise coincidence matrix shows that the fine-grained development-related practices (e.g., unit testing, code reviews, pair programming, and retrospectives) are extensively combined with each other. In our previous study [20], we claimed that practices have become the major building blocks of process customization. A claim that is supported by the study at hand.

4.6 Threats to Validity

Despite the rigorous development procedure of the survey instrument (Sect. 3.2), still, our study faces some threats to validity, which we discuss in the following.

Internal Validity. The internal validity might be threatened by the questionnaire as such. To increase the internal validity, we used questionnaires from previously conducted studies (e.g., [3, 11, 15]) as reference for the instrument development. Furthermore, we conducted an iterative validation phase, and internal and external reviews to increase the internal validity via researcher triangulation.

External Validity. The external validity might be threatened by low number of participants, the participants' self-reporting, and the limited number of regions included in the study, which might affect the generalizability of our results. To increase the external validity, external reviews and trails with industry practitioners were conducted prior to the study's launch. Furthermore, results were compared with previous studies to find a reference for data interpretation. However, in to order to generalize the results, further research in more regions is necessary.

5 CONCLUSION

This paper presents findings from the HELENA project with which we study the use of *hybrid software development approaches*. An internationally conducted survey provided 69 complete responses from which we extracted a list of software development approaches used in practice. We categorized and analyzed the processes used

and found hybrid approaches to be widely used in practice. Our study revealed that hybrid approaches have become mainstream and are used by companies regardless of company size and industry sector. While standards, norms, and regulations challenge companies, even in regulated domains, companies adopt agile methods. An empirical analysis confirmed that there is no evidence to claim that the development and use of hybrid approaches are triggered by company size or external standards. Hybrid approaches used in practice today emerge from pragmatic process selection and evolve over time. The cluster analysis supports West's "Water-Scrum-Fall" hypothesis by showing that combinations of development approaches follow a pattern in which a traditional process serves as framework refined by (multiple) fine-grained practices. We further argue that individual practices, rather than large methods, have become the building blocks for process customization.

Limitations. The main limitation of the study presented in the paper at hand is the population. The data reported and analyzed is mostly coming from participants that are either located or involved in projects within Europe. Furthermore, the selected sampling strategy was a convenience sampling. That is, the aim has been to collect as many data points as possible with little regards to neither controlling the response rate nor the distribution over, e.g., industry sector and roles. Industry sectors and roles cover a broad spectrum, but are not evenly distributed. These aspects limit the generalizability of our result set. However, at this point, our analysis does not yet attempt to provide a generalizable and complete picture. Rather, the present study aims to improve our understanding of the software system development approaches in practice, yet calls for future work to eventually allow for generalizability.

Future Work. Future—already ongoing—steps of the HELENA project are in line with the aforementioned limitations. The survey instrument (Table 2) is receiving a revision (refined scope based on findings obtained in the first stage) before initiating the second stage of the project. Furthermore, by growing the international network (more than 20 countries from various continents) for a second stage, the study is no longer limited to the European context. Finally, due to expected larger data base, it will be possible, e.g., to develop statistical models grounded in evidence, test further hypotheses such as multivariate analyses, and, eventually, create focused research groups to study specific areas if interest more thoroughly, e.g., via interviews.

REFERENCES

[1] Pierre Bourque and Richard E. Fairley (Eds.). 2014. *Guide to the Software Engineering Body of Knowledge, Version 3.0.* IEEE Computer Society, Washington, DC, USA.

[2] Frederick P. Brooks. 1987. No Silver Bullet Essence and Accidents of Software Engineering. *IEEE Computer* 20, 4 (1987), 10–19.

[3] Ralf Buschermöhle, Heike Eekhoff, and Bernhard Josko. 2006. *Success – Erfolgs- und Misserfolgsfaktoren bei der Durchführung von Hard- und Softwareentwicklungsprojekten in Deutschland.* BIS-Verlag der Carl von Ossietzky Universität Oldenburg, Oldenburg.

[4] Paul Clarke and Rory V. O'Connor. 2012. The Situational Factors That Affect the Software Development Process: Towards a Comprehensive Reference Framework. *Information and Software Technology* 54, 5 (May 2012), 433–447. DOI: https://doi.org/10.1016/j.infsof.2011.12.003

[5] Micheal Cusumano, Alan MacCormack, Chris F. Kemerer, and Bill Crandall. 2003. Software development worldwide: the state of the practice. *IEEE Software* 20, 6 (Nov 2003), 28–34. DOI: https://doi.org/10.1109/MS.2003.1241363

[6] Claudia de O. Melo, Viviane Santos, Eduardo Katayama, Hugo Corbucci, Rafael Prikladnicki, Alfredo Goldman, and Fabio Kon. 2013. The evolution of agile software development in Brazil. *Journal of the Brazilian Computer Society* 19, 4 (2013), 523–552. DOI: https://doi.org/10.1007/s13173-013-0114-x

[7] Philipp Diebold, Jan-Peter Ostberg, Stefan Wagner, and Ulrich Zendler. 2015. What Do Practitioners Vary in Using Scrum? In *International Conference on Agile Software Development (XP 2015).* Lecture Notes in Business Information Processing, Vol. 212. Springer, Cham, 40–51.

[8] Philipp Diebold and Thomas Zehler. 2016. *The Right Degree of Agility in Rich Processes.* Springer International Publishing, Cham, 15–37. DOI: https://doi.org/10.1007/978-3-319-31545-4_2

[9] Tore Dybå and Torgeir Dingsøyr. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50, 9–10 (2008), 833 – 859. DOI: https://doi.org/10.1016/j.infsof.2008.01.006

[10] Brendan J. Frey and Delbert Dueck. 2007. Clustering by Passing Messages Between Data Points. *Science* 315, 5814 (Feb 2007), 972–976.

[11] Martin Fritzsche and Patrick Keil. 2007. *Kategorisierung etablierter Vorgehensmodelle und ihre Verbreitung in der deutschen Software-Industrie.* Research Report (in German) TUM-I0717. Technische Universität München.

[12] Vahid Garousi, Ahmet Coşkunçay, Aysu Betin-Can, and Onur Demirörs. 2015. A survey of software engineering practices in Turkey. *Journal of Systems and Software* 108 (2015), 148 – 177. DOI: https://doi.org/10.1016/j.jss.2015.06.036

[13] Capers Jones. 2003. Variations in software development practices. *IEEE Software* 20, 6 (Nov 2003), 22–27. DOI: https://doi.org/10.1109/MS.2003.1241362

[14] Georg Kalus and Marco Kuhrmann. 2013. Criteria for Software Process Tailoring: A Systematic Review. In *International Conference on Software and Systems Process (ICSSP).* ACM, New York, NY, USA, 171–180.

[15] Marco Kuhrmann and Daniel Méndez Fernández. 2015. Systematic Software Development: A State of the Practice Report from Germany. In *International Conference on Global Software Engineering (ICGSE).* IEEE Computer Society, Washington, DC, USA, 51–60.

[16] Marco Kuhrmann, Jürgen Münch, Philipp Diebold, Oliver Linssen, and Christian R. Prause. 2016. On the Use of Hybrid Development Approaches in Software and Systems Development: Construction and Test of the HELENA Survey. In *Proceedings of the Annual Special Interest Group Meeting Projektmanagement und Vorgehensmodelle (PVM) (Lecture Notes in Informatics (LNI))*, Vol. P-263. Gesellschaft für Informatik (GI), Bonn, 59–68.

[17] Brendan Murphy, Christian Bird, Thomas Zimmermann, Laurie Williams, Nachiappan Nagappan, and Andrew Begel. 2013. Have Agile Techniques been the Silver Bullet for Software Development at Microsoft. In *International Symposium on Empirical Software Engineering and Measurement (ESEM).* IEEE, Washington, DC, USA, 75–84.

[18] Colin J. Neill and Philip A. Laplante. 2003. Requirements engineering: the state of the practice. *IEEE Software* 20, 6 (2003), 40–45. DOI: https://doi.org/10.1109/MS.2003.1241365

[19] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an Algorithm. In *Proceedings of the International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS).* MIT Press, Cambridge, MA, USA, 849–856. http://dl.acm.org/citation.cfm?id=2980539.2980649

[20] Georgios Theocharis, Marco Kuhrmann, Jürgen Münch, and Philipp Diebold. 2015. Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices. In *International Conference on Product Focused Software Development and Process Improvement (Lecture Notes in Computer Science)*, Vol. 9459. Springer, Cham, 149–166.

[21] John F. Tripp and Deborah J. Armstrong. 2014. Exploring the Relationship between Organizational Adoption Motives and the Tailoring of Agile Methods. In *Hawaii International Conference on System Sciences (HICSS).* IEEE Computer Society, Washington, DC, USA, 4799–4806. DOI: https://doi.org/10.1109/HICSS.2014.589

[22] VersionOne. 2006-2014. State of Agile Survey. Available from: http://www.versionone.com/agile-resources/more-resources/blogs/. (2006-2014).

[23] Leo R. Vijayasarathy and Charles W. Butler. 2016. Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Software* 33, 5 (Sept 2016), 86–94. DOI: https://doi.org/10.1109/MS.2015.26

[24] Dave West, Mike Gilpin, Tom Grant, and Alissa Anderson. 2011. *Water- Scrum-Fall Is The Reality Of Agile For Most Organizations Today.* Technical Report. Forrester Research Inc.

[25] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in Software Engineering.* Springer-Verlag, Berlin Heidelberg.

[26] Peng Xu and Balasubramaniam Ramesh. 2008. Using Process Tailoring to Manage Software Development Challenges. *IT Professional* 10, 4 (July 2008), 39–45. DOI: https://doi.org/10.1109/MITP.2008.81