Plan-Driven Approaches Are Alive and Kicking in Agile Global Software Development

Marcelo Marinho[a,b], John Noll[c,b], Ita Richardson[b], Sarah Beecham[b]

[a] Federal Rural University of Pernambuco (UFRPE) Department of Computer Science (DC) Recife, PE, Brazil
Email: marcelo.marinho@ufrpe.br
[c] University of East London University Way, London, E16 2RD, UK
Email: j.noll@uel.ac.uk
[b] Lero, the Irish Software Research Centre University of Limerick, Ireland
Email: ita.richardson,sarah.beecham@lero.ie

Abstract—

Background: **Agile methods are no longer restricted to small projects and co-located teams. The last decade has seen the spread of agile into large scale, distributed and regulated domains. Many case studies show successful agile adoption in GSD, however, taken as a whole, it remains unclear how widespread this trend is, and what form the agile adoption takes in a global software development (GSD) setting.**

Aims: **Our objective is to gain a deeper understanding of how organisations adopt agile development methods in distributed settings. Specifically we aim to plot the current development process landscape in GSD.**

Method: **We analyse industrial survey data from 33 different countries collected as part of the HELENA project that explored the wider use of hybrid development approaches in software development. We extract and analyse the results of 263 surveys completed by participants involved in globally distributed projects.**

Results: **In our sample, 72% of globally distributed projects implement a mix of both agile and traditional approaches (termed 'hybrid'). 25% of GSD organisations are predominantly agile, with only very few (5%) opting for traditional approaches. GSD projects that used only agile methods tended to be very large.**

Conclusions: **Globally Distributed Software Development (and project size) is not a barrier to adopting agile practices. Yet, to facilitate project coordination and general project management, many adopt traditional approaches, resulting in a hybrid approach that follows defined rules.**

Index Terms—**Global Software Development; Agile software development; software process; hybrid development approaches; GSD.**

I. INTRODUCTION

Agile methods have become common in software development organizations around the world. Initially, the methods were used for the development of small, co-located projects. However, in recent years many large organizations have made the transition from traditional, plan-driven waterfall-type methods towards agile methods [1]. According to the largest reoccurring survey on agile adoption, the State of Agile Survey [2], 86% of respondents had at least some distributed teams adopting agile practices. While this survey is not scientific, it indicates that a significant number of large global organizations are adopting agile methods.

VersionOne [3] identified two reasons as to why companies use agile methods: some believe in the values and principles of the manifesto, and others see agile as best practice. However, Boehm and Turner [1] suggest that projects should find a 'sweet spot' combining a mixture of traditional and agile methods. According to their certainty/ambiguity puzzle, business management still demands accurate and complete long-term estimates of projects and tasks, whereas development teams have to deal with uncertainty during the project [4]. Previous research on development methods advise that methods need to be adapted to the work context [5, 6]. Hence, software development has to balance the need for planning and controlling and, at the same time, for flexibility and speed [7, 8].

The many development approaches adopt different philosophies (e.g. agile, traditional) and specific features (e.g. plan-driven, iterative-incremental). Such approaches comprise either practices or comprehensive process frameworks. Furthermore, software development has become vital in every industry sector and must adhere to domain-specific standards, norms, and regulations. In this regard, practitioners have begun developing so-called hybrid development approaches.

Acknowledged as one of the trends of the 21st century, globalization significantly changed many industries, including and, in particular, software development. Many companies foster global software development (GSD) to benefit from cheaper, faster, and better software development [9]. However, GSD has traditionally followed a plan-driven approach, where tasks are allocated according to where they appear in the software development lifecycle [10, 11]. The belief that agile methods, which were mainly used for small projects and co-located teams, cannot be used in GSD no longer stands [12, 13]. Agile methods tend to rely on informal processes and regular face-to-face communication to facilitate coordination, whereas distributed software development relies on formal mechanisms [12].

In this paper, we aim to highlight the extent to which GSD projects are adopting hybrid approaches. We take a deeper look at those few organizations that do not avail of plan driven approaches to manage their software development. We also uncover what drives organizations to change, adapt or merge

agile practices. This study seeks to determine whether it is possible to adopt agility in GSD projects without the support of traditional plan-driven approaches. The paper is based on the HELENA Survey [14], an international study on the use of hybrid development approaches in software development around the world.

This paper is organized as follows: in Section Section II we introduce the background to the problem and define our research questions. Section Section III describes the method used. In Sections Section IV and Section V we present the results and discuss their implications and limitations. Section Section VI presents conclusions and future research directions.

II. BACKGROUND

A. Global Software Development

Global software development (GSD) promises cost effectiveness, access to large multi-skilled workforces, reduced time to market, and the opportunity to follow critical-path tasks around the clock [15]. These deciding factors, among others, have made GSD a daily reality in today's IT organizations, even though development environments are more complex and exhibit several challenges: physical distance, multiple time zones, the loss of 'teamness,' culture differences, strategic issues, process differences, knowledge management and technical challenges [16, 17].

In particular, GSD is normally characterized by stakeholders from different national and organizational cultures, located in separate geographic locations and time zones, using different information and communication technologies to collaborate. Such conditions usually result in major problems in relation to team communication, coordination and collaboration [18]. Furthermore, key project-specific, team-specific, and distance-specific contextual factors may also impact team effectiveness. These include, for example, the nature of the contract, the application domain, the volatility of requirements, project personnel, site distribution, team experience, and temporal, geographical, and socio-cultural distance between partners and sites [19].

Traditionally, GSD follows a plan-driven, structured, waterfall approach, where tasks are allocated according to where they appear in the software development lifecycle [10, 11]. By contrast, agile practices are considered capable of mitigating the challenges faced by GSD, as reported by several studies [12, 13, 20–22]. However, Akbar and Safdar [6] find that the complexity of GSD and need for agility meant that organizations transitioning from co-located to distributed development were more inclined to adopt and tailor agile methods.

B. Agile and Traditional Development Approaches

Agile software development refers to a set of iterative and incremental software engineering methods that advocate an 'agile philosophy' captured in the Agile Manifesto [23]. While mostly repackaging and re-branding previously well-known and appreciated software development practices, the agile movement can be considered an alternative to traditional software development methods [24]. Agile development methods were believed to best suit small, co-located teams, and their success with small teams has inspired their use in large-scale software development [12] as reflected by the many scaled agile frameworks that have emerged over recent years [25]. However, fundamental assumptions regarding agile development are challenged when applying the methods at a large scale [26].

Agile methods were designed to accept and efficiently manage change [27]. It has been shown that agile methods have improved the satisfaction of both customers and developers. However, there is evidence that agile methods may not be a good fit for large undertakings [28], as they require more coordination and heavier methodologies than smaller projects [29].

Nerur et al. [30] describe the fundamental premise behind traditional methods as software that is wholly specifiable and should be developed with precise and comprehensive planning. Agile methods, on the other hand, consider that software can be built through on-going planning, improvement, and testing, based on fast feedback and change. Learning and adaptation should be embraced [24]. Adapting the approach to the context will require balance in some areas [5, 6].

In plan-driven development, the architecture is defined before implementation and testing, whereas the architectural design appears as a result of on-going clarification in purist agile development. Traditionally, to construct a large software system developed by many teams, it is pivotal for the architecture to be agreed upon and communicated, adding to the bureaucracy and overhead associated with traditional methods. Indeed, GSD calls for different practices when it comes to architectural design, even when applying agile methods [31].

C. Hybrid Development Approaches

Kuhrmann et. al. [7] proposed the following definition: '*A hybrid software development approach is any combination of agile and traditional (plan-driven or rich) approaches that as organizational unit adopts and customizes to its own context needs (e.g., application domain, culture, process, project, organizational structure, techniques, technologies, etc.)*'

Some researchers argue that combining a mixture of traditional and agile practices is the best way to manage a project [1]. Boehm and Turner [32] mention that a changing world needs agile and plan-driven development methods. They characterize 'house plots', where approaches are most likely to succeed, and they identify five critical dimensions, one being that a balanced strategy should be established to achieve a successful combination of agile and planned approaches.

Diebold and Zehler [33] examine the process of combining agile and traditional software development methods. They distinguish between revolutionary and evolutionary approaches, which differ in the methods order of occurrence. The authors defend the coexistence of both methods in software development projects.

Dingsoyr et al. [34] summarize 12 pieces of advice they collected from 12 Scrum teams, stating that methods need to be adapted to needs during the program lifecycle. They argue that, while there is much good advice in frameworks such as the Scaled Agile Framework and Large-Scale Scrum, frameworks eventually limit organizations. It is thus preferable to combine agile and traditional approaches in global projects.

West et al. [8] coined the term 'Water-Scrum-Fall' and hypothesized that hybrid development methods would become standard. They also provided evidence that development approaches are indeed used in combination (e.g. agile and traditional). Previous research on the use of development methods suggests that methods need to adapt to the work context [35].

III. METHOD

The present study is part of a series of inquiries that have been conducted on data from a large international study on the use of hybrid development approaches, called HELENA: '(H)ybrid d(E)ve(L)opm(EN)t (A)pproaches in software systems development. The overall research project employs a mixed research method, with elements of both quantitative and qualitative research [36]. The project aims to provide a strong, empirically based assessment of the current state-of-practice in software and systems development [7].

HELENA was designed as an international research endeavour. The first stage prepared the data collection and tested the survey instruments. The second stage involved mass data collection conducted by an international consortium that comprised more than 60 partners from over 30 countries. More details can be found on the official website[1]. The results of the second stage were published at the 3rd HELENA Workshop. Further, all the authors of this article are involved in the stages of the study. The second stage results will fuel the third stage of the project by focusing follow-up in-depth research pertaining to outcomes of the second stage.

In this paper we use a subset of this data. The HELENA survey asked questions on how software organizations develop their software and what processes are used [7]. All published studies [37–43] share the same general theme and use the same data sample, although they have different scope, variables, and designs. Our study is concerned with hybrid approaches in the context of GSD. A section of the survey was designed to shed light on current approaches to GSD in two different situations: what development approaches to GSD projects are used in practice, and which different development approaches are combined in practice.

Within the dataset of 691 completed responses[2] [14], there were 263 companies that answered either option 3 or option 4 to the question: 'Is the project or product your answer refers to carried out in a (globally) distributed manner?' with options:

- 1 = No;

---

[1]HELENA Survey: https://helenastudy.wordpress.com/
[2]In total, the survey yielded 1,467 answers of which 691 participants completed the questionnaire [14].

TABLE I
RESEARCH QUESTIONS OVERVIEW

Research Questions
RQ1   What development approaches do GSD projects use?
RQ2   How do GSD projects combine development approaches (if at all)?
RQ3   Do combined ('hybrid') development approaches tend toward agile or traditional development?

- 2 = Yes, nationally (same country);
- 3 = Yes, regionally (same continent);
- 4 = Yes, globally

A. Research Questions

In this paper we focus solely on those 263 globally distributed organizations, and we defined the research questions listed in Table I.

B. Survey Instrument

HELENA used an online survey [44, 45] to collect data from practitioners about the development approaches they use in their projects.

1) Instrument Development: The HELENA research team used a multi-stage approach to develop the survey instrument. Initially, three researchers developed the questionnaire and tested it with 15 German practitioners to evaluate the agreement. Based on the feedback, a team of eleven researchers from across Europe revised the survey and translated it to English. A first revised questionnaire public test (referred to as "Stage 1") that included 25 questions was conducted in 2016 in Europe. This public test yielded 69 data points, which were analysed and used to initiate the next stage of the study.

In Stage 2, the research team was expanded to 75 researchers from around the world. Also, the questionnaire was revised to improve structure, scope, relevance, question accuracy, value ranges for variables, and significance of the topics included. The revised questionnaire was translated and made available in English, German, Spanish, and Portuguese. Further details of the instrument are presented in [14])

2) Instrument Structure: The final survey consisted of 38 questions divided into five parts: Demographics (10 questions), Process Use (13 questions), Process Use and Standards (5 questions), Experiences (2 questions) and Closing (8 questions). However, because some questions may not be relevant depending on answers to previous questions, the survey responses do not always contain 38 answers [14].

C. Data Collection Procedure

The survey was promoted through personal contacts of the participating researchers, through posters at conferences, and through posts to mailing lists, social media channels [46]. The target population was members of IT clusters and networks, and we used LinkedIn, Twitter, Facebook, Xing, and ResearchGate to promote the survey within the relevant communities. A web survey with different kinds of scale questionnaires and demographic information collection was distributed to the target population.

TABLE II
HELENA QUESTIONNAIRE LIST (CONDITIONAL QUESTIONS FOR THE DIFFERENT PATHS ARE OMITTED IN THE TABLE)

No.	Group[1]	Question	Scale[2]	opt
1.	M	What is your company's size in equivalent full-time employees (FTEs)?	SC	5
2.	M	Do you participate in globally software project?	SC	4
3.	M	In which country are you personally located?	FT	
4.	M	What is the major role you have in this project?	SC	9
5.	M	What is the size of the project?	SC	5
6.	M	How many years of experience do you have in software and systems development?	SC	5
7.	PU	Do you combine different development approaches?	YN	
8.	PU	For the following standard activities in the project or product development, please indicate to which degree you carry out project activities in a more traditional or more agile manner.	LI	
9.	PU	How were the combinations of development approaches in your company developed?	MC	3
10.	PU	How was your project-specific development approach defined?	MC	6
11.	PU	Which of the following frameworks and methods do you use?	RT	7
12.	PU	What are the overall goals that you aim to address with your selection and combination of development approaches?	MC	2
13.	PU	To what degree did the combination of approaches help you to achieve your goals?	RT	10

[1]*Legend for groups: M=metadata, PU=process use, E=experience.*
[2]*Legend for scales: YN=yes/no, SC=single choice, MC=multiple choice/select, LI=5-item Likert scale, RT= Rating: 7 point scale.*

Potential biasing factors in the results include those that are common to self-selected written surveys with convenience sampling: response bias (i.e. people who responded may have different characteristics from those who did not), coverage errors (i.e. the representation of participants may not be balanced across different sub-communities), and item bias (i.e. some questions may have been skipped intentionally or unintentionally).

D. Data Analysis Procedure

We utilised several methods to provide answers to the research questions. For all research questions, we used descriptive statistics, to provide tables and charts for process use and selection.

To provide a context we analysed the types of companies involved (size, how many sites, what countries are represented, and roles of respondents). We also analysed (a) what types of development are used in GSD and (b) how the different development approaches are combined.

To further analyze the result set, we used the three categories: *traditional*, *agile*, and *generic*, based on the definitions provided by Kurhman and colleagues [7]. We classified the following approaches as agile: Scrum, Safe, Lean, LESS, Nexus, XP, Kanban, DevOps, ScrumBam, Crystal, DSDM and Feature-driven development (FDD). Similarly, we classified the following approaches as traditional: Waterfall, Spiral Model, V-Model, RUP, PRINCE2 and SSADM. Other approaches – Iterative development, Domain-Drive Design (DDD), Model Driven Architecture (MDA), Team Software Process (TSP), Personal Software Process (PSP) – were classified as generic, since the approach does not fit into either the agile or traditional category. While Iterative development is a key feature of agile development approaches [47], some traditional approaches (RUP and the Spiral Model, for example) also incorporate iterations [47]; consequently, we followed Kuhrmann et al. [7] and classified Iterative development in the generic category.

Many projects used combinations of methods in different categories. In light of this, we assigned such projects to a

TABLE III
NUMBER OF PROJECTS BY CATEGORY.

Class	Num	%
Agile	65	25.0
Traditional	5	2.0
Hybrid	189	72.0
Other	4	1.0
Total	**263**	**100**

category according to the following rules:

- Agile + Generic = Agile;
- Traditional + Generic = Traditional;
- Agile + Traditional = Hybrid.

These rules are based on the observation that "generic" processes could be either agile or traditional depending on context, which in this case is determined by the other methods used, which are either agile or traditional. Each organization was categorised based on the approaches they stated they used when answering the survey.

The thirteen questions from the HELENA survey which we analysed are listed in Table II. If a company stated that it combined different development approaches (question 7), we then looked at questions 8, 10 and 12 to confirm whether this was indeed the case. Of the 263 global companies, 204 selected 'Yes' to question 7. In checking the other answers from these 204 respondents, we were able to establish that companies used Hybrid approaches (Traditional + Agile) in 189 projects that were detailed in the survey (see Table III). Additionally, through undertaking this exercise, we verified that the answer given to question 7 is due to how the respondents interpreted 'combination'. Many respondents indicated the use of multiple agile or traditional methods; it is possible that respondents interpret this as 'combining' methods, even when they are in the same category.

IV. RESULTS

In this section, we summarize the results of our investigation. We first show descriptive statistics describing the

TABLE IV
PROJECT LOCATION (*n=263*, FROM 33 COUNTRIES)

Continent/Country	Number
Africa (2)	
Algeria	1
Uganda	1
America (85)	
Argentina	28
Brazil	6
Canada	4
Chile	2
Colombia	1
Costa Rica	25
United States	18
Uruguay	1
Asia (24)	
Armenia	1
China	4
Christmas Island	1
India	9
Japan	1
Pakistan	4
Saudi Arabia	1
Turkey	3
Europe (141)	
Austria	10
Denmark	17
Estonia	4
Finland	4
Germany	49
Ireland	10
Italy	1
Poland	1
Portugal	4
Russia	3
Spain	14
Sweden	8
Switzerland	13
United Kingdom	3
Oceania (11)	
New Zealand	11

- Not used = we do not know or we do not know if we use it or we never use it;
- Moderate = we rarely use it or we sometimes use it;
- Extensive = we often use it or we always use it;

The participants state that, among agile approaches, the most frequently used framework is Scrum, followed by Kanban. Iterative development was the most used by the GSD projects that responded to this study. However, a waterfall approach was the most commonly used traditional framework.
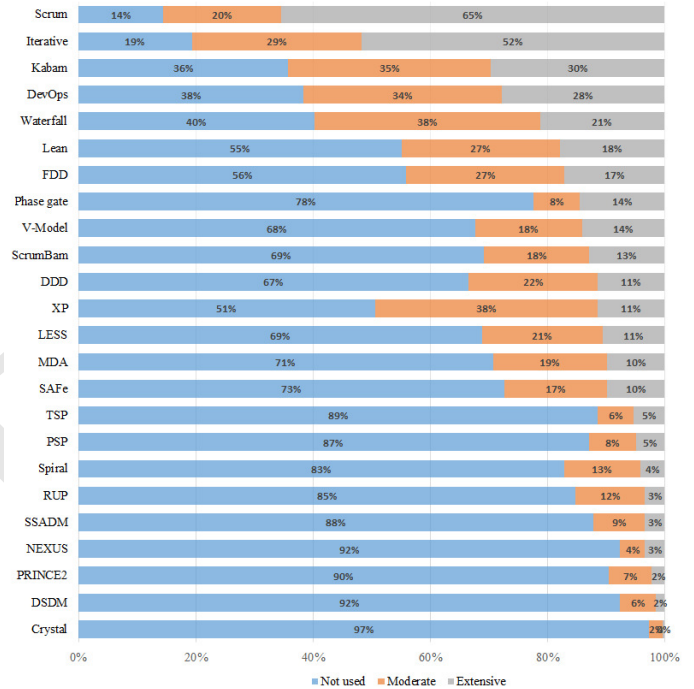

Fig. 1. Frameworks and methods used by GSD projects (*n=263*).

demographics of the participants. We then proceed to data analysis related to our research questions (I).

A. Study Population

The survey yielded 263 valid and complete responses related to GSD projects, from 33 different countries, covering 5 continents, as shown in Table IV. These projects were situated in 18 micro-sized, 29 small, 55 medium, 61 large, and 100 very large companies.

The size of the projects were characterized as small (7), medium (21), large (37), and very large (198). The respondents had a range of experience in the area of software and systems development, ranging from < 1 year (8), through 1–2 years (12), 3–5 years (35), 6–10 years (49), to >10 years (159). Twenty-three percent of respondents were Developers, 22% Project/Team Managers, 11% Product Manager/Owners, 10% Architects, 8% Quality Managers 6% Scrum Master/Agile Coaches, 6% C-level Management (e.g. CIO, CTO), 3% Testers, 3% Analyst/Requirements Engineers, 1% Trainers and 8% other role.

B. Development Approaches used in GSD

Fig. 1 presents the frameworks researched and the degree of utilization by the global projects. We collapsed the responses to simplify the representation of the framework usage:

C. Combination of Development Approaches

Respondents were asked how framework combinations and practices in the participants' organization were developed. 83% reported that they evolved from past projects over time, and 43% reported that they were planned as part of a process-improvement program (Table V).

TABLE V
HOW WERE THE COMBINATIONS OF DEVELOPMENT FRAMEWORKS AND METHODS IN YOUR COMPANY DEVELOPED? (*(Of the 189 participants, 154 answered these affirmations)*)

Option	#	%
Planned as part of a process improvement program	69	43.0
Evolved as learning from past projects over time	128	83.0

Participants were asked to define their project-specific development approach (see Table VI for results). 25% of projects reported project-specific process selection and tailoring followed defined rules, while 24% of projects selected specific practices and methods according to customer demands.

5

Option	#	%
Project specific process selection and tailoring follows defined rules	48	25
Specific practices and methods are selected according to customer demands	45	24
A project manager tailors the process in the beginning of a project	30	16
Specific practices and methods are selected in the project on demand	30	16
Project specific process selection and tailoring is supported by tools	13	7
The process is not tailored at all	10	5
Other	13	7

TABLE VII

OVERALL GOALS WHEN SELECTING AND COMBINING APPROACHES
(*n=189*).

Goal: Improved...	%
Productivity	67.0
Planning and estimation	65.0
External product quality	64.0
Frequency of delivery to customers	64.0
Adaptability and flexibility of the process to react to change	58.0
Time to market	58.0
Project monitoring and controlling	53.0
Client involvement	49.0
Internal artifact quality	48.0
Employee satisfaction	43.0
Knowledge transfer and learning	41.0
Risk management	39.0
Return-on-investment cycles	38.0
Reuse for project artifacts	35.0
Maturity of the company	31.0
Ability of the company to develop critical systems	28.0
Tool support	28.0
Staff education and development	26.0

Participants were asked to list the overall goals that drive their selection of development approaches. The main objectives reported were: improved productivity, improved planning and estimation, improved external product quality and improved frequency of delivery to customers. Table VII shows that global hybrid project delivery is merely a combination of selected approaches working together in order to increase performance beyond what any single methodology can achieve. The precise combination of techniques varies from one project to another.

The majority of participants answered that a hybrid development approach appropriately supports business projects (e.g. concerning internal or customer goals). Further, this approach helps to achieve good product quality. Participants said that their current development approach adequately addressed the external requirements (e.g. standards). The approach supported the work and helped to hasten its time to market. However, the majority of participants noted that they would change or improve their current development approach. We believe this is an example of continuing quality improvement through hybrid designs.

On the other hand, 43% of the participants stated that the combination of methods and frameworks improves the satisfaction of the team. According to [48], using agile methods and practices exclusively does not necessarily lead to a more competent development team. Rather, doing so can undermine

the quality and productivity of software, as well as social factors such as team motivation and performance. Thus, we believe that, in addition to combining methods for better product results, it is necessary to structure a philosophy for hybrid teams which can improve team satisfaction.

Fig. 2 shows a classification of the different approaches where we present data from the 263 companies who are involved in GSD. For instance, a project might adopt Scrum at all times, but use some plan-driven practices. Thus, in the sequence, global projects adopt more scrum, iterative development, and kanban-based approaches. Among the frameworks that were created to scale agility, the use of LESS and Safe was followed by Nexus.
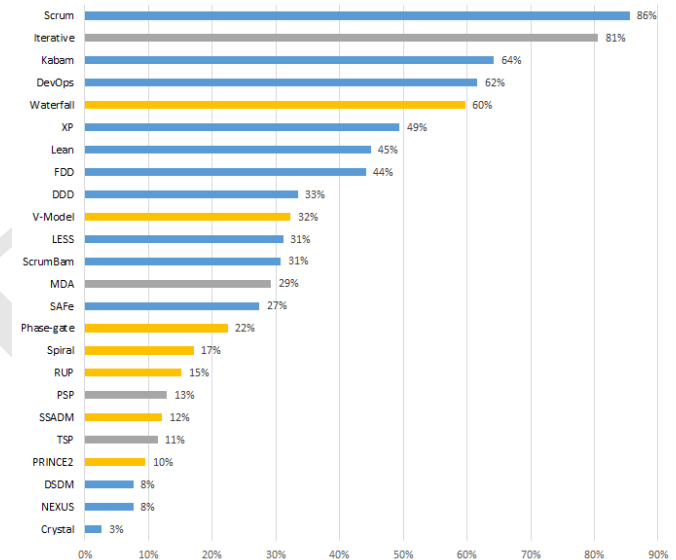

Fig. 2. Framework usage frequency (*n=263*). colors: agile: blue, traditional: orange, generic: grey.

We performed a comparison of how frequently traditional and agile methods are combined in GSD projects (Table VIII); the rows show agile-specific approaches and the columns show the frequency (extensive or moderate) of each traditional approach adopted in the GSD projects. For example, 89% of GSD projects always adopt Scrum. Combinations are adopted when Scrum is used, and the frequency of adoption of waterfall approach was 18% 'extensive' and 41% 'moderate'. This comparison indicates that GSD projects using an agile approach also use traditional approaches during execution.

We found that, in the combination of approaches in an 'extensive' way, the Waterfall approach is the most combined with agile (22%), followed by Stage-Gate model and V-model (15% each). Further information regarding GSD projects can be derived from this table: when Scrum is used, Prince2 is the least-used approach; the most combined approaches with Crystal were RUP and the V-model; when DSDM is used, it is most often combined with the Waterfall, RUP, V-model and Stage Gate traditional approaches.

Table IX shows that combining agile and traditional meth-

TABLE VIII
FREQUENCY WITH WHICH TRADITIONAL METHODS ARE PERFORMED WHEN AGILE PRACTICES ARE AT LEAST *rarely* PERFORMED BY GSD PROJECTS.

Agile practice	PRINCE2 (25²) 10%		RUP (40) 15%		Spiral (45) 17%		Waterfall (157) 60%		Stage Gate (59) 22%		SSADM (32) 12%		V-model (85) 32%		Neither trad. nor waterfall
	extensive[1]	moderate	extensive	moderate	extensive	moderate	extensive	moderate	extensive	moderate	extensive	moderate	extensive	moderate	
Scrum (225) 86%	(6) 3%	(18) 8%	(8) 4%	(29) 13%	(8) 4%	(32) 14%	(41) 18%	(93) 41%	(34) 15%	(18) 8%	(5) 2%	(23) 10%	(31) 14%	(43) 19%	(56) 25%
Kanban (169) 64%	(5) 3%	(12) 7%	(5) 3%	(22) 13%	(5) 3%	(21) 12%	(32) 19%	(67) 40%	(29) 17%	(15) 9%	(4) 2%	(13) 8%	(28) 17%	(34) 20%	(40) 24%
DevOps (162) 62%	(5) 3%	(15) 9%	(6) 4%	(18) 11%	(2) 1%	(23) 14%	(29) 18%	(62) 38%	(26) 16%	(14) 9%	(1) 1%	(16) 10%	(15) 9%	(31) 19%	(43) 27%
XP (130) 49%	(3) 2%	(11) 8%	(8) 6%	(23) 18%	(3) 2%	(27) 21%	(22) 17%	(60) 46%	(13) 10%	(13) 10%	(2) 2%	(14) 11%	(12) 9%	(32) 25%	(27) 21%
Lean (118) 45%	(5) 4%	(11) 9%	(7) 6%	(18) 15%	(7) 6%	(21) 18%	(20) 17%	(53) 45%	(24) 20%	(14) 12%	(6) 5%	(12) 10%	(24) 20%	(26) 22%	(22) 19%
FDD (116) 44%	(3) 3%	(12) 10%	(4) 3%	(16) 14%	(8) 7%	(18) 16%	(26) 22%	(50) 43%	(22) 19%	(13) 11%	(4) 3%	(15) 13%	(23) 20%	(24) 21%	(20) 17%
LESS (82) 31%	(3) 4%	(8) 10%	(6) 7%	(15) 18%	(3) 4%	(14) 17%	(15) 18%	(39) 48%	(15) 18%	(10) 12%	(4) 5%	(9) 11%	(11) 13%	(21) 26%	(12) 15%
Scrumban (81) 31%	(3) 4%	(8) 10%	(5) 6%	(14) 17%	(3) 4%	(15) 19%	(18) 22%	(33) 41%	(12) 15%	(10) 12%	(3) 4%	(13) 16%	(12) 15%	(23) 28%	(14) 17%
SAFe (72) 27%	(3) 4%	(11) 15%	(7) 10%	(16) 22%	(4) 6%	(15) 21%	(16) 22%	(37) 51%	(18) 25%	(9) 13%	(3) 4%	(13) 18%	(13) 18%	(18) 25%	(7) 10%
DSDM (20) 8%	(1) 5%	(4) 20%	(4) 20%	(6) 30%	(3) 15%	(6) 30%	(4) 20%	(11) 55%	(4) 20%	(2) 10%	(4) 20%	(5) 25%	(2) 10%	(6) 30%	(3) 15%
Nexus (20) 8%	(2) 10%	(6) 30%	(2) 10%	(8) 40%	(1) 5%	(6) 30%	(3) 15%	(8) 40%	(3) 15%	(3) 15%	(1) 5%	(8) 40%	(2) 10%	(9) 45%	(0) 0%
Crystal (7) 3%	(1) 14%	(3) 43%	(2) 29%	(3) 43%	(1) 14%	(3) 43%	(1) 14%	(3) 43%	(1) 14%	(3) 43%	(1) 14%	(2) 29%	(2) 29%	(3) 43%	(0) 0%
Agile: (254) 96%	(6) 2%	(19) 7%	(9) 4%	(31) 12%	(11) 4%	(34) 13%	(56) 22%	(101) 40%	(38) 15%	(21) 8%	(9) 4%	(23) 9%	(37) 15%	(48) 19%	(66) 26%

[1] The 'moderate' column indicates the method was used rarely or sometimes, while the 'extensive' column means a method was used often or always.
[2] The number given in parentheses indicates the number of respondents using the approach.

TABLE IX
COMPARISON OF HOW FREQUENT TRADITIONAL AND AGILE METHODS ARE COMBINED.[1]

Agile	Traditional					TOTAL
	never	rarely	sometimes	often	always	
never	(4) 44%	(0) 0%	(0) 0%	(3) 33%	(2) 22%	(9) 3%
rarely	(1) 14%	(0) 0%	(1) 14%	(2) 29%	(3) 43%	(7) 3%
sometimes	(6) 19%	(2) 6%	(8) 25%	(7) 22%	(9) 28%	(32) 12%
often	(17) 16%	(22) 21%	(23) 21%	(36) 34%	(9) 8%	(107) 41%
always	(41) 38%	(19) 18%	(19) 18%	(13) 12%	(16) 15%	(108) 41%
TOTAL	(69) 26%	(43) 16%	(51) 19%	(61) 23%	(39) 15%	(263) 100%

[1] The TOTAL column indicates the total frequency with which projects perform agile methods at the level shown; the TOTAL row indicates the total frequency with which project perform traditional methods a the level shown

ods is common. For example, looking at the columns and rows (rarely, sometimes, often and always), we see that the combination of methods occurs in 72% of the projects, as shown in Table III. Our results strongly indicate that it is easier to find hybrid GSD projects than purely agile ones. It is understood that opportunities to adapt and customize each of these approaches must meet the specific needs of the current situation to allow organizations the flexibility to develop a project and use the best methods in a particular aspect of the work.

D. Agility of Hybrid Approaches

In the survey, participants were asked, "For the following standard activities in the project or product development, please indicate to which degree you carry out activities in a more traditional or more agile manner"; participants rated their approach to performing each of the disciplines on a five point scale comprising "Fully Traditional" (1), "Mainly Traditional" (2), "Balanced between Traditional and Agile" (3), "Mainly Agile" (4) and "Fully Agile" (5). Fig. 3 summarizes responses using the average rating among small, medium, large, and very large companies.

In general, the responses indicate that activities are implemented in a balanced way between agile and traditional methods. However, management activities such as Risk Management and Quality Management are more traditionally oriented, whereas development activities, such as Implementation/Coding and Integration/Testing, tend to be conducted in a more agile manner. Also, Fig. 3 shows that small companies tend to use traditional methods more than medium, large and very large companies, that tend to be more agile.
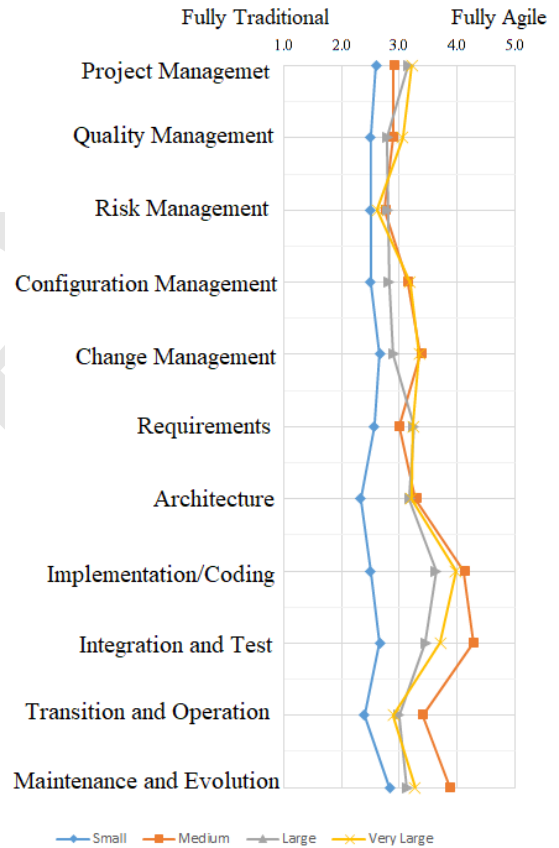

Fig. 3. Activities implementation in GSD projects (1=fully traditional to 5=fully agile; Average rating of *n=189* hybrid projects.

V. DISCUSSION

Agile adoption is commonplace, and as shown in the GSD literature, operating in a distributed setting is not a barrier to adopting agile methods [49]. In this study, however, we found that, in practice, hybrid approaches are adopted in the majority (72%) of GSD projects, while 25% only use agile methods, and 5% use only traditional methods.

Our study consisted of responses from 263 participants in GSD projects, of whom 198 participated in very large global projects and 61% had more than 10 years' experience. Regard-

ing research question RQ1 (Fig. 2), participants reported the most adopted approaches are, from highest to lowest: Scrum, Iterative Development, Kanban, DevOps and Waterfall. When combining approaches (RQ2 Table VIII) at different times in the project, the responses indicate that projects often use Scrum and Waterfall; and out of 263 GSD projects, 189 used hybrid software development approaches.

Most of the GSD projects represented in the survey results are neither predominantly planned or agile in their approach. Rather, the approaches appear to be adjusted to reflect the challenges and opportunities of each initiative (Table VI). For example, Fig. 2 shows that 86% of GSD projects adopt Scrum, and Table VIII shows that, when adopting Scrum, there is a requirement to combine it with traditional approaches. This result agrees with the conclusion of Lous et al. [50] who performed a systematic review of the Scrum literature with GSD and came to the conclusion that all Scrum adopters need to (extensively) adjust the core assets of Scrum to a GSD context. Likewise, Fitzgerald and Stol [51] affirm that GSD requires some tailoring of agile methods that were not intended for such settings. While agile methods have moved well beyond this, with frequent use of agile methods in projects within globally distributed teams, many challenges remain, such as those pertaining to cultural differences, breakdowns in communication, and optimum practices for distributing development work across sites. Individual successes rely heavily on the particular organizational context, which is increasingly recognized as an essential factor [52].

Bass [53] argued that several agile artefacts, such as user stories and sprint backlogs, are used in global projects. Further, teams adopt elements of an agile culture. However, to deal with release plans, test plans, and product architecture implementation, the methodology adopted follows a plan-driven approach. Bass [53] states that teams skilfully blend agile artefacts with traditional plan-based artefacts because of the need for coordination, and that this combination is reflected in the quality. Finally, Richter et al. [54] analysed agile GSD in an organization and identified several issues. They concluded that there is no single solution or universal software development approach. Teams and organizations use different strategies to address the manifold of challenges in software development projects. As a consequence, it appears that there is a need for agile practices, which were originally designed for collocated teams, to be modified to work in distributed environments. Jalali and Wohlin [55] also found this to be necessary.

Diebold et al. [56] showed that it is common to use a textbook method, but one that adapts to specific contexts (the project's or the company's). This was reflected in GSD projects in the results of Table VI, which shows an overview of the actual process selection and tailoring in particular projects.

An agile structure requires engagement based on the interaction of all teams, whereas traditional models determine the specific phases in which team activities occur [8]. In large and distributed projects, the main focus should be to keep the organizational structure small. The hierarchical structure should be as flat as possible, with more teams and fewer levels of management. Teams can be real or virtual, but the main difference in similar groups formed when following traditional methods of project management is that teams must have cross-links. One member of each team must attend the major meetings of the other teams. Having interconnected, self-organized teams and a flat organizational structure allows for agile teams [57].

Table VIII presents combinations of the use of traditional approaches. Of the surveyed projects that used scaling agile frameworks, such as LESS, SAFe and Nexus, the majority used traditional methods moderately or extensively. In scaled projects, the existence of an architecture team is significant, as it guides all teams on the proper implementation of the requirements, reducing development costs by proactively analysing alternatives and finding solutions that best meet the needs, adaptability, and future use [58]. As shown by [59], several case studies show that moving to SAFe feels like moving back to plan-driven methods (such as Waterfall and RUP) which include fixed increments, centralized planning, a loss of incremental and iterative development, and too much detail. Although LESS applies agile practices, we also investigated agile/plan-driven. Likewise, [57] suggests a way of combining plan-driven with agile. However, proponents of agility are unlikely to sanction such a mix.

To maintain control of large and distributed projects, the natural inclination is to introduce layers of management, improved policies, new processes, and checkpoints based on the project competencies of managers. For this reason, many of the practices and tools identified for distributed agile projects are the same as those followed in projects using traditional project management approaches. Examples include the need for an architecture team that positively influences both traditional and agile methodologies, along with quality assurance and integration teams that have similarly scaled functions for large, distributed teams [31].

A. Threats to Validity

1) Internal Validity: Each data transcript from the survey was faithfully represented in answers or comments in the study. Sampling of the participants was conducted with maximum variation, following a revised and approved questionnaire and synthesis tools, such as SPSS and Microsoft Excel, in order to increase the internal validity.

2) External Validity: In this study, independent researchers were consistently involved. Furthermore, results were compared with previous studies to find a reference for data interpretation. However, in order to generalise the results, further research in more regions is necessary. Also, the survey did not include the distribution of the teams (we do not know the geographic distance, temporal distance, cultural distance nor team size). Our results are based on responses taken from projects that are geographically distributed (across continents and countries).

VI. CONCLUSIONS

Problems related to the complexity of software development and increased focus on coordination, communication, and

collaboration are the main reasons for the interest in applying agile methods to GSD [20]. GSD uses asynchronous communication and computer-mediated activities, but the applicability of these coordination mechanisms is generally insufficient [18]. Our survey illustrates that most companies surveyed applied a combination of traditional and agile methods in their distributed projects. Although the agile approach has many advantages, our results suggest that it is not a universal solution. Consequently, we often see companies using a combination of agile and traditional methods. These hybrid approaches seem to be products of a trade-off between reaping the benefits of an iterative, flexible, and collaborative approach with maintaining some level of planning and structure.

In this study, using the publicly available HELENA Survey [14] , we aimed to highlight how GSD projects are adopting hybrid approaches. We looked at which frameworks are being used, and examine how GSD projects still use traditional plan-driven approaches. We analysed 189 responses from projects in which traditional and agile approaches were combined. Our results led us to the conclusion that most global software projects are neither purely agile or purely traditional in their approach to software development, but rather combine agile and traditional methods. The most used and combined approaches in the responses were Scrum and Waterfall. In particular, we found that projects which adopted agile scaling frameworks such as SAFe or LESS, also employed plan-driven methods. In summary, our study supports the findings of Aitken & Ilango [29], that "there is nothing really incompatible with applying all the principles and values of agile software development, along with most (if not all) of the practices, to traditional software engineering".

A. Limitations

Based on our data collection, we note four limitations. First, the data does not represent all agile, plan-driven, and GSD specific approaches. Second, there is relatively low representation of United States-based projects in the sample population, the first user community to publicise the agile movement, and the most experienced and most populous among agile user communities globally. In this study, although the United States is the leading country, with a representation of 7% of projects, it can still be considered underrepresented, since according to Agile Alliance[3] 35% of all agile user groups (29 of 83) are based in the United States. Further, their website claims that more than 58% of Agile Alliance users are from the United States. However, we do have 35 different countries represented in our dataset. Third, our data does not include information on team size, the extent of the distribution, nor the specific countries involved in the distributed project. All of these may have an impact on why a certain development approach is taken. Finally, the sample size is small, considering the large population of the agile and GSD community. More respondents could provide more

robust and accurate statistical calculations and analyses, and could also include other methods that were not observed.

Notwithstanding these observations, what is obvious to us as researchers is that, as companies are continually combining agile, traditional and global software development methods, we have identified a need for an approach by which companies can build their own bespoke software development method.

B. Future Research

Since our results show that hybrid methods are the current state of practice in GSD projects, this should be taken as a new baseline for future research. The strategies applied today are still some way from perfect when it comes to devising hybrid methods. Therefore, the first main future direction for research is to provide better strategies to devise hybrid methods.

Also, we find that agile scaling approaches still use traditional process elements for specific purposes. Notably, the finding that the waterfall model, even criticised a lot, is still frequently used. So, studying the reasons for the agile scaling approaches uses more traditional process allows us to understand potential directions better to mitigate the use of hybrid methods. Finally, we should develop strategies suitable for use by practitioners when being confronted with the need to integrate different processes in GSD projects.

REFERENCES

[1] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE software*, vol. 22, no. 5, pp. 30–39, 2005.

[2] VersionOne, Inc., "12th Annual State of Agile Development Survey," https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report, 2018, [Online; accessed 27-June-2018].

[3] R. Lockard, A. Leff, C. Johnson, J. Gifford, J. Hrcsko, T. Lightfoot, A. Cleff, M. McCalla, J. Cusack, A. Bacon, C. Murman, and P. Elia, "Agile uprising agile manifesto review," 2018, [accessed 19-November-2018]. [Online]. Available: http://podcast.agileuprising.com/manifesto-author-review/

[4] M. Marinho, J. Noll, and S. Beecham, "Uncertainty management for global software development teams," in *International Conference on the Quality of Information and Communications Technology*. IEEE, 2018, pp. 238–246.

[5] V. Vinekar, C. W. Slinkman, and S. Nerur, "Can agile and traditional systems development approaches coexist? an ambidextrous view," *Information systems management*, vol. 23, no. 3, pp. 31–42, 2006.

[6] R. Akbar and S. Safdar, "A short review of global software development (gsd) and latest software development trends," in *Computer, Communications, and Control Technology (I4CT), 2015 International Conference on*. IEEE, 2015, pp. 314–317.

[7] M. Kuhrmann, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektere, F. McCaffery, O. Linssen, E. Hanser *et al.*, "Hybrid software and system development in practice: waterfall, scrum, and beyond," in *Proceedings of the 2017 International Conference on Software and System Process*. ACM, 2017, pp. 30–39.

[8] G. Theocharis, M. Kuhrmann, J. Münch, and P. Diebold, "Is water-scrum-fall reality? on the use of agile and traditional development practices," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2015, pp. 149–166.

[9] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empirical software engineering*, vol. 15, no. 1, pp. 91–118, 2010.

[3]https://www.agilealliance.org/

[10] H.-C. Estler, M. Nordio, C. A. Furia, B. Meyer, and J. Schneider, "Agile vs. structured distributed software development: A case study," *Empirical Software Engineering*, vol. 19, no. 5, pp. 1197–1224, 2014.

[11] T. Ylikotila and P. Linna, "A collaboration model for global multicultural software development," *Information Modelling and Knowledge Bases XXII*, vol. 225, p. 321, 2011.

[12] J. Noll, A. Razzak, I. Richardson, and S. Beecham, "Agile practices for the global teaming model," in *11th International Conference on Global Software Engineering Workshops (ICGSEW 2016)*. IEEE, 2016, pp. 13–18.

[13] B. Ramesh, L. Cao, K. Mohan, and P. Xu, "Can distributed software development be agile?" *Communications of the ACM*, vol. 49, no. 10, pp. 41–46, 2006.

[14] M. Kuhrmann, P. Tell, J. Klünder, R. Hebig, S. Licorish, S. MacDonell, and Eds., "HELENA stage 2 results," *ResearchGate*, 2018, doi:10.13140/RG.2.2.14807.52649. [Online]. Available: https://www.researchgate.net/publication/329246439_HELENA_Stage_2_Results/.

[15] S. Beecham, I. Richardson, and J. Noll, "Assessing the strength of global teaming practices: A pilot study," in *10th International Conference on Global Software Engineering (ICGSE 2015)*. IEEE, 2015, pp. 110–114.

[16] S. Beecham and J. Noll, "What motivates software engineers working in global software development?" in *International Conference on Product-Focused Software Process Improvement*. Springer, 2015, pp. 193–209.

[17] M. Marinho, A. Luna, and S. Beecham, "Global software development: practices for cultural differences," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2018, pp. 299–317.

[18] M. Niazi, S. Mahmood, M. Alshayeb, M. R. Riaz, K. Faisal, N. Cerpa, S. U. Khan, and I. Richardson, "Challenges of project management in global software development: A client-vendor analysis," *Information and Software Technology*, vol. 80, pp. 1–19, 2016.

[19] I. Richardson, V. Casey, F. McCaffery, J. Burton, and S. Beecham, "A process framework for global software engineering teams," *Information and Software Technology*, vol. 54, no. 11, pp. 1175–1191, 2012.

[20] M. Paasivaara, S. Durasiewicz, and C. Lassenius, "Using scrum in distributed agile development: A multiple case study," in *Fourth International Conference on Global Software Engineering, (ICGSE 2009)*. IEEE, 2009, pp. 195–204.

[21] E. Hossain, M. A. Babar, and H.-Y. Paik, "Using scrum in global software development: a systematic literature review," in *Fourth International Conference on Global Software Engineering (ICGSE 2009)*. IEEE, 2009, pp. 175–184.

[22] S. Beecham, J. Noll, and I. Richardson, "Using agile practices to solve global software development problems–a case study," in *International Conference on Global Software Engineeering Workshops (ICGSEW 2014)*. IEEE, 2014, pp. 5–10.

[23] M. Fowler and J. Highsmith, "The agile manifesto," *Software Development*, vol. 9, no. 8, pp. 28–35, 2001.

[24] K. Conboy, "Agility from first principles: Reconstructing the concept of agility in information systems development," *Information Systems Research*, vol. 20, no. 3, pp. 329–354, 2009.

[25] C. Ebert and M. Paasivaara, "Scaling agile," *IEEE Software*, vol. 34, no. 6, pp. 98–103, 2017.

[26] M. Paasivaara, "Adopting SAFe to scale agile in a globally distributed organization," in *12th International Conference on Global Software Engineering (ICGSE 2017)*. IEEE, 2017, pp. 36–40.

[27] A. Cockburn and J. Highsmith, "Agile software development, the people factor," *Computer*, vol. 34, no. 11, pp. 131–133, 2001.

[28] T. Dyba and T. Dingsoyr, "What do we know about agile software development?" *IEEE software*, vol. 26, no. 5, pp. 6–9, 2009.

[29] A. Aitken and V. Ilango, "A comparative analysis of traditional software engineering and agile software development," in *2013 46th Hawaii International Conference on System Sciences*, Jan 2013, pp. 4751–4760.

[30] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Communications of the ACM*, vol. 48, no. 5, pp. 72–78, 2005.

[31] O. Sievi-Korte, S. Beecham, and I. Richardson, "Challenges and recommended practices for software architecting in global software development," *Information and Software Technology*, vol. 106, pp. 234 – 253, 2019.

[32] B. Boehm and R. Turner, "Using risk to balance agile and plan-driven methods," *Computer*, vol. 36, no. 6, pp. 57–66, 2003.

[33] P. Diebold and T. Zehler, "The right degree of agility in rich processes," in *Managing Software Process Evolution*. Springer, 2016, pp. 15–37.

[34] T. Dingsøyr, T. Dybå, M. Gjertsen, A. O. Jacobsen, T.-E. Mathisen, J. O. Nordfjord, K. Røe, and K. Strand, "Key lessons from tailoring agile methods for large-scale software development," *arXiv preprint arXiv:1802.05118*, 2018.

[35] B. Fitzgerald, G. Hartnett, and K. Conboy, "Customising agile methods to software practices at intel shannon," *European Journal of Information Systems*, vol. 15, no. 2, pp. 200–213, 2006.

[36] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[37] M. Kuhrmann, J. Nakatumba-Nabende, R.-H. Pfeiffer, P. Tell, J. Klünder, T. Conte, S. G. MacDonell, and R. Hebig, "Walking through the method zoo: does higher education really meet software industry demands?" in *Proceedings of the 41st International Conference on Software Engineering: Software Engineering Education and Training*. IEEE Press, 2019, pp. 1–11.

[38] J. Klünder, R. Hebig, P. Tell, M. Kuhrmann, J. Nakatumba-Nabende, R. Heldal, S. Krusche, M. Fazal-Baqaie, M. Felderer, M. F. G. Bocco *et al.*, "Catching up with method and process practice: An industry-informed baseline for researchers," in *Proceedings of International Conference on Software Engineering, ser. ICSE-SEIP. IEEE*, 2019.

[39] P. Tell, J. Klünder, S. Küpper, D. Raffo, S. G. MacDonell, J. Münch, D. Pfahl, O. Linssen, and M. Kuhrmann, "What are hybrid development methods made of? an evidence-based characterization," in *Proceedings of the International Conference on Software and System Processes*. IEEE Press, 2019, pp. 105–114.

[40] J. Klünder, P. Hohl, M. Fazal-Baqaie, S. Krusche, S. Küpper, O. Linssen, and C. R. Prause, "HELENA study: Reasons for combining agile and traditional software development approaches in german companies," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2017, pp. 428–434.

[41] M. Felderer, D. Winkler, and S. Biffl, "Hybrid software and system development in practice: Initial results from austria," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2017, pp. 435–442.

[42] N. Paez, D. Fontdevila, and A. Oliveros, "HELENA study: Initial observations of software development practices in argentina," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2017, pp. 443–449.

[43] E. Scott, D. Pfahl, R. Hebig, R. Heldal, and E. Knauss, "Initial results of the HELENA survey conducted in estonia with comparison to results from sweden and worldwide," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2017, pp. 404–412.

[44] M. Ciolkowski, O. Laitenberger, S. Vegas, and S. Biffl, "Practical experiences in the design and conduct of surveys in empirical software engineering," in *Empirical methods and studies in software engineering*. Springer, 2003, pp. 104–128.

[45] F. Shull, J. Singer, and D. I. Sjøberg, *Guide to advanced empirical software engineering*. Springer, 2007.

[46] C. Robson and K. McCartan, *Real world research*. John Wiley & Sons, 2016.

[47] C. Larman and V. R. Basili, "Iterative and incremental development: A brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003.

[48] J. Klünder, A. Schmitt, P. Hohl, and K. Schneider, "Fake news: Simply agile," *Projektmanagement und Vorgehensmodelle 2017-Die Spannung zwischen dem Prozess und den Mensch im Projekt*, 2017.

[49] R. Vallon, B. J. da Silva Estácio, R. Prikladnicki, and T. Grechenig, "Systematic literature review on agile practices in global software development," *Information and Software Technology*, vol. 96, pp. 161–180, 2018.

[50] P. Lous, M. Kuhrmann, and P. Tell, "Is scrum fit for global software engineering?" in *Proceedings of the 12th International Conference on Global Software Engineering*. IEEE Press, 2017, pp. 1–10.

[51] B. Fitzgerald and S. Klaas-Jan, "The future of software development methods," *The Routledge Companion to Management Information Systems*, pp. 125–137, 2018.

[52] T. Dybå, D. I. Sjøberg, and D. S. Cruzes, "What works for whom, where, when, and why?: on the role of context in empirical software engineering," in *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*. ACM, 2012, pp. 19–28.

[53] J. M. Bass, "Artefacts and agile method tailoring in large-scale offshore software development programmes," *Information and Software Technology*, vol. 75, pp. 1–16, 2016.

[54] I. Richter, F. Raith, and M. Weber, "Problems in agile global software

engineering projects especially within traditionally organised corporations:[an exploratory semi-structured interview study]," in *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering*. ACM, 2016, pp. 33–43.

[55] S. Jalali and C. Wohlin, "Global software engineering and agile practices: a systematic review," *Journal of software: Evolution and Process*, vol. 24, no. 6, pp. 643–659, 2012.

[56] P. Diebold, J.-P. Ostberg, S. Wagner, and U. Zendler, "What do practitioners vary in using scrum?" in *International Conference on Agile Software Development*. Springer, 2015, pp. 40–51.

[57] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, 2016.

[58] D. Leffingwell, *Scaling software agility: best practices for large enterprises*. Pearson Education, 2007.

[59] A. Putta, M. Paasivaara, and C. Lassenius, "Benefits and challenges of adopting the scaled agile framework (safe): Preliminary results from a multivocal literature review," in *International Conference on Product-Focused Software Process Improvement (PROFES)*. Springer, 2018, pp. 334–351.