

Hybrid Software Development Approaches in Practice: A European Perspective

Marco Kuhrmann¹, Philipp Diebold², Jürgen Münch³, Paolo Tell⁴, Kitija Trektere⁵, Fergal McCaffery⁵, Vahid Garousi⁶, Michael Felderer⁷, Oliver Linssen⁸, Eckhart Hanser⁹, Christian R. Prause¹⁰

¹Clausthal University of Technology, Germany, E-mail: kuhrmann@acm.org

²Fraunhofer IESE, Germany, E-mail: philipp.diebold@iese.fhg.de

³Reutlingen University, Germany, E-mail: j.muench@computer.org

⁴IT University Copenhagen, Denmark, E-mail: pate@itu.dk

⁵Dundalk Institute of Technology, Ireland, E-mail: kitija.trektere@dkit.ie , fergal.mccaffery@dkit.ie

⁶Wageningen University, Netherlands, E-mail: yahid@vgarousi.com

⁷University of Innsbruck, Austria, E-mail: michael.felderer@uibk.ac.at

⁸FOM University of Applied Sciences for Economics and Management, Germany, E-mail: oliver.linssen@fom.de

⁹Baden-Württemberg Cooperative State University, Germany, E-mail: hanser@dhbw-loerrach.de

¹⁰DLR Space Administration, Germany, E-mail: christian.prause@dlr.de

Correspondence to: Marco Kuhrmann, E-mail: kuhrmann@acm.org

© IEEE 2017. **Preprint.** This is the author's version of the work. The definite version was accepted in IEEE Software, Issue assignment pending,

The final version will be made available in the IEEE Digital Library

Hybrid Software Development Approaches in Practice: A European Perspective

Marco Kuhrmann¹, Philipp Diebold², Jürgen Münch³, Paolo Tell⁴, Kitija Trektere⁵, Fergal McCaffery⁵, Vahid Garousi⁶, Michael Felderer⁷, Oliver Linssen⁸, Eckhart Hanser⁹, Christian R. Prause¹⁰

¹Clausthal University of Technology, Germany, E-mail: kuhrmann@acm.org

²Fraunhofer IESE, Germany, E-mail: philipp.diebold@iese.fhg.de

³Reutlingen University, Germany, E-mail: j.muench@computer.org

⁴IT University Copenhagen, Denmark, E-mail: pate@itu.dk

⁵Dundalk Institute of Technology, Ireland, E-mail: kitija.trektere@dkit.ie, fergal.mccaffery@dkit.ie

⁶Wageningen University, Netherlands, E-mail: vahid@vgarousi.com

⁷University of Innsbruck, Austria, E-mail: michael.felderer@uibk.ac.at

⁸FOM University of Applied Sciences for Economics and Management, Germany, E-mail: oliver.linssen@fom.de

⁹Baden-Württemberg Cooperative State University, Germany, E-mail: hanser@dhbw-loerrach.de

¹⁰DLR Space Administration, Germany, E-mail: christian.prause@dlr.de

Correspondence to: Marco Kuhrmann, E-mail: kuhrmann@acm.org

Abstract

Agile and traditional development approaches are used in combination in today's software development. To improve the understanding and to provide better guidance for selecting appropriate development approaches, it is important to analyze such combinations in practice. Results obtained from an online survey strongly confirm that hybrid development approaches are widely used in industry. Our results show that hybrid development approaches: (i) have become reality for nearly all companies; (ii) are applied to specific projects even in the presence of company-wide policies for process usage; (iii) are neither planned nor designed but emerge from the evolution of different work practices; and, (iv) are consistently used regardless of company size or industry sector.

Keywords

Software process; software development method; software development practice; agile practices; traditional development approach; hybrid development approach

1 Introduction

As there is no one-size-fits-all software development approach, teams and organizations use different approaches to address the manifold challenges of software development projects. They rarely follow the pure approach by implementing a process by the book. For instance, in 2011, West claimed that the "Water-Scrum-Fall" [1] had become reality for software systems development, and previous studies (box "Related Studies") as well as experience suggest that this claim is true. Yet, many questions concerning so-called *hybrid development approaches* (box "A Definition") are still open. In this article, we are particularly interested in answering the following key questions:

- (1) What different development approaches are used and in which combinations?
- (2) Why are combinations developed and used?
- (3) Do industry sector and company size matter?

Answers to these questions would enable us to better characterize how and why particular development approaches are used to address improvement goals and problems in respective areas.

2 Objectives and Method

To shed light on the characteristics of hybrid development approaches and to begin answering the above questions, we initiated an exploratory multistage international research project named HELENA (*Hybrid DEveLopmENt Approaches in software systems development*), and collected data from 69 practitioners across Europe. In total, the questionnaire comprises 25 questions aimed at collecting data on general process use, process use in the context of norms and standards, process improvement, and experiences. Due to the open and explorative nature of the presented study, we intentionally sacrificed full control over the population and accepted issues coming along with an explorative study. For instance, we did not provide definitions to the participants regarding the terminology used (e.g., method, process, approach) and categorized data only during data analysis according to the definitions from [9]. The heart of the questionnaire is a list of 40 different development approaches, which we crafted from several previously conducted studies (e.g., [11], [13], see also box “Related Studies”). For these approaches, we asked the participants to select the ones used in their project. The survey accepted answers from May to June 2016, and the data collection strategy applied was ‘convenience sampling’ using a number of mailing lists of IT clusters/networks and social media (e.g., Twitter, LinkedIn, Facebook, and ResearchGate) within the relevant communities. The full study material (questionnaire, raw data, etc.) can be obtained online¹.

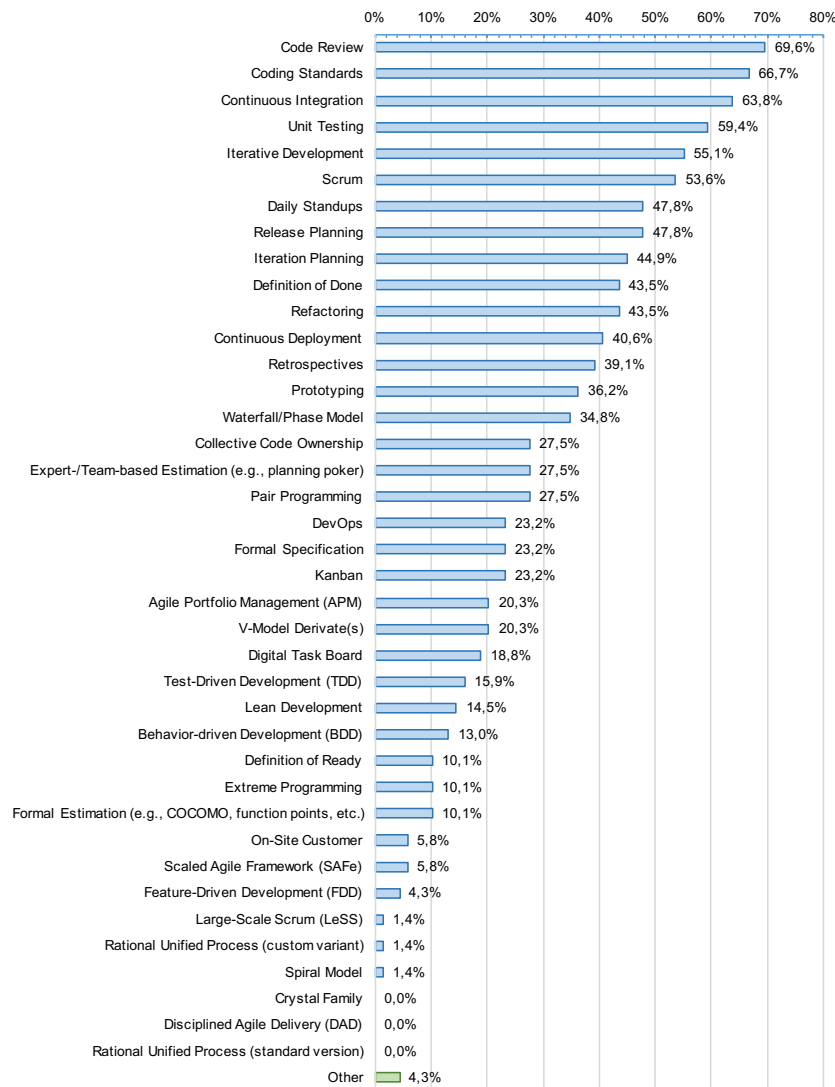
3 HELENA Findings

We present the findings of our survey along the key questions presented above.

3.1 What different development approaches are used and in which combinations?

Since the terms “process”, “method”, “approach”, and “practice” are difficult to define precisely, we decided to not provide participants with a pre-categorized list. Instead, the provided list comprises 40 different development approaches in alphabetical order. The participants were asked to select all the approaches they use and, if necessary, add further ones. An overview of the basic demographic information can be taken from the box “Demographics”. Figure 1 shows a large variety of approaches and highlights the ones most frequently selected, e.g., code review 69.6%, continuous integration 63.8%, unit testing 59.4%, Scrum 53.6%, and the Waterfall/phase model 34.8%.

¹ HELENA online material, available from ResearchGate: <https://www.researchgate.net/project/HELENA-SURVEY-Hybrid-dEveLopmENt-Approaches-in-software-systems-development>



1
2 **Figure 1 Overview of the development approaches applied in practice—the chart is based on 729 selections in**
3 **the provided list of 40 different development approaches.**

4
5 To analyze the combination of development approaches in detail, in a first step, we categorized the different
6 approaches according to the definitions from [9]. Each approach was categorized as either “method” or
7 “practice”, and as “traditional”, “agile” or “both”². In a second step, we created a matrix to analyze the
8 pairwise combinations. Figure 2 shows this matrix, and shows our classification of the different approaches
9 as well as the frequency of the combinations. The darker the color of a particular cell, the more often the
10 combination of two approaches occurred in the answers, e.g., code review and coding standards (38), the
11 Waterfall model and Scrum (10), and Scrum and Kanban (13) are often combined. Taking the different
12 categories into account, Figure 2 clearly shows that traditional and agile approaches are combined with each
13 other, and that combinations occur among all kinds of development approaches. In summary, with our data,
14 we can confirm a trend previously observed in [7] that:

- 15 (1) different (i.e., traditional and agile) approaches are combined, and
16 (2) agile practices are combined with each other.

² The terms “agile” and “traditional” are used in the largely accepted understanding that separates elements pre-2001 (i.e., the writing of the agile manifesto) from the ones that emerged afterwards. Elements classified as “both” indicate practices and methods that have characteristics of both categories.

	Method	Method																																								
		Traditional					Agile					Both					Practice					Both																				
		Rational Unified Process (custom)	Rational Unified Process (standard)	Spiral Model	V-Model Derivate(s)	Waterfall/Phase Model	Agile Portfolio Management (APM)	Behavior-driven Development (BDD)	DevOps	Disciplined Agile Delivery (DAD)	Extreme Programming	Feature-Driven Development (FDD)	Kanban	Large-Scale Scrum (LeSS)	Lean Development	Scaled Agile Framework (SAFe)	Scrum	Crystal Family	Iterative Development	Prototyping	Formal Estimation (e.g., COCOMO)	Formal Specification	Collective Code Ownership	Continuous Deployment	Continuous Integration	Daily Standups	Definition of Done	Definition of Ready	Digital Task Board	Iteration Planning	On-Site Customer	Pair Programming	Refactoring	Retrospectives	Test-Driven Development (TDD)	Code Review	Coding Standards	Expert-/Team-based Estimation	Release Planning	Unit Testing		
Method	Traditional	1																																								
	Agile				7	2	1	3		2				2		6		1		8	7	3	5	2	5	8	7	5		2	8	1	2	7	6	3	13	10	3	8	9	
	Both																																									
	Practice																																									
	Both																																									
	Traditional																																									
	Agile																																									
	Both																																									
	Practice																																									
	Both																																									
	Other												2					1		2	2		1	1		2	2	2	2	1		1	2	2		2	2	1		2	2	

Figure 2 Overview of the pair-wise combination of the different development approaches. The chart is based on 729 approach selections, reported by the 69 participants.

Figure 1 and Figure 2 only show which approaches are used and how they are pairwise combined. In [10], we focused on statistically analyzing different combination patterns. This analysis reveals that there are few “base methods”, which serve as umbrella for integrating the different approaches. These base methods are the *V-Model*, *Waterfall/Phase Model*, *Scrum*, *SAFe*, and the generic *Iterative Development*. The clusters statistically constructed around these “base methods” are in line with the two confirmed trends above.

In summary, practitioners use a multitude of different development approaches, and the development approaches are usually combinations of traditional and agile ones.

3.2 Why are combinations developed and used?

Figure 1 and Figure 2 show that different approaches are used and combined. Therefore, we investigate the reasons behind this situation by asking several detailed questions:

Defined policies: First, we asked the participants if a company-wide policy concerning process use exists. 52.2% of the participants stated that their company has a defined process or a set of processes, which are applied to all projects. Contrarily, 20.3% stated that each business unit has its own standards, rules, and policies and, finally, 27.5% stated that each project has the freedom to select the most suitable development approach individually. That is, about half of the participants adopt company-level policies, while the rest selects the development approach more freely.

Planned development: As we, grounded in earlier studies, expected a combined use, we asked the participants how a particular combination of approaches was developed. Only 19.6% of the participants stated that the used approach is an outcome of a planned process improvement program. While for the remaining 83.9%, the majority of the approaches emerges from experiences collected over time, i.e., the actual approach evolves and is usually not subject to a controlled development – a practice in line with the core agile principles from the manifesto for agile software development often referred to as “inspect and adapt”. In this context, 23.2% of the participants also state that they adapt the approach in response to a specific situation, which is in line with the 27.5% above, who select their development approach individu-

1 **Motivation for a hybrid approach:** To understand triggers that motivate practitioners developing a hybrid
2 approach, we asked the participants for the main motivation for developing their approach.

3 Referring to the different industry sectors (see box “Demographics”), external norms, standards, and rules
4 can be considered drivers behind the development of hybrid approaches. From the 69 participants, 41 partic-
5 ipants (59.4%) stated that standards, norms, and regulations are relevant for their companies and/or pro-
6 jects. From these 41 participants, 23 stated that the companies are challenged by implementing the required
7 standards and agility, whereas the remaining 18 state that implementing agility is not a problem – but ful-
8 filling the requirements of the standards is. In a nutshell, almost 2/3 of the respondents face requirements
9 brought in by standards, and more than half of these consider the combination of standard-driven and agile
10 development challenging.

11 Beyond the standards, we also asked the participants to report further motivating drivers. In total, 56 partic-
12 ipants provided information, which we coded and categorized. Table 1 shows the summary of the described
13 motivations. The top-3 categories concern project/product management and commitment, evolution and
14 pragmatism, and project operation and improved flexibility. For instance, hybrid approaches are seen as a
15 route towards more stable yet flexible project teams. By including well-known “classic” approaches, hybrid
16 approaches help improving management commitment while developers get flexibility by using agile prac-
17 tices (in line with [12], see also Sect. 3.1). Remarkably, evolution and pragmatism was mentioned in 18
18 statements. That is, the current approach was neither planned nor designed (see above), but evolved from
19 the different work practices applied, which was commented by one participant as follows: “It is what
20 works”.

21 **Table 1 Summary of the major motivating drivers for implementing hybrid development approaches. The table**
22 **shows the 9 major drivers crafted from 56 stated motivating drivers (89 items in total).**

Motivation trigger	#	Examples
Project/Product Management and Commitment	27	Improve stability of the team, project type, product lifecycle management, integration of high-level Waterfall-like and low-level Agile approaches, lacking readiness of customers (e.g., contracting, pricing), management does not buy in, business people are scared of new things
Evolution and Pragmatism	18	Constant evolution, stepwise company transition towards agile, increasing amount of software parts in complex systems, requirement of the day-to-day life – “it came up naturally” and “we do what works”
Project Operation and Improved Flexibility	15	Flexibility of teams and resources, select the best-fitting approach, fast product feature evolution, meeting deadlines, better handling volatile requirements
Client Constraints	13	Lacking readiness and understanding of agile, customer satisfaction, client-domain requirements
Company Constraints	8	Company size, keeping existing business running, company history, company philosophy
Business Constraints	8	Business context and target domain requirements, e.g., management, standards, different target domains to be addressed

23
24 **3.3 Do industry sector and company size matter?**
25 Due to the variety of industry sectors to which software has become key, we were also interested in whether
26 the use of hybrid approaches differs according to either sector or company size, and whether a general trend
27 could be identified. An overview of the different industry sectors and company sizes involved in our study
28 can be found in the box “Demographics”. Table 2 shows a breakdown of our data based on (a) company
29 size and (b) industry sector. The relative distribution of the different approaches is based on the participants’
30 selection. For example, for the industry sector “Web Applications and Services”, Table 2 reads as follows:
31 this industry sector was selected by 20 participants, and these 20 participants selected 279 (=100%) different

development approaches of which 3.9% are classified as traditional method, 17.2% as agile method, 45.2% as agile practice, and so forth.

Table 2 Overview of the relative use of the different approaches (a) per company size and (b) per industry sector. For each item, the quantity is given, e.g., the number of companies of a particular size, and the number of approaches selected per item. Note: while participants had to select exactly one company size, one company can be engaged in different industry sectors (multiple selection was enabled). The percentages given in this table always refer to the number of selected approaches for the respective category.

	QTY	Method			Practice			Other	Approaches Selected
		Traditional	Agile	Both	Traditional	Agile	Both		
Micro (<10)	12	6,6%	15,4%	13,2%	5,5%	33,0%	25,3%	1,1%	91
Small (11-50)	14	3,2%	16,7%	7,7%	0,6%	47,4%	24,4%		156
Medium (51-250)	12	4,9%	16,2%	9,2%	1,4%	40,8%	26,8%	0,7%	142
Large (>250)	14	4,3%	16,0%	7,4%	3,7%	40,5%	27,6%	0,6%	163
Very Large (>2500)	17	8,5%	15,8%	7,9%	5,1%	38,4%	24,3%		177
Average		5,5%	16,0%	9,1%	3,3%	40,0%	25,7%	0,8%	
Deviation		1,6%	0,3%	1,7%	1,8%	3,5%	1,2%	0,2%	
Web Applications and Services	20	3,9%	17,2%	7,2%	3,2%	45,2%	23,3%		279
Medical Devices and Health Care	17	7,1%	16,8%	9,2%	4,3%	38,0%	23,9%	0,5%	184
Public Sector/Public Contracting	16	2,6%	16,2%	8,8%	3,9%	44,3%	24,1%		228
Financial Services (e.g., Banking, Insurance, Trading)	15	4,2%	16,8%	7,9%	4,2%	42,1%	24,7%		190
Cloud Applications and Services	14	4,5%	21,1%	8,0%	3,0%	41,2%	22,1%		199
Other Information Systems (e.g., ERP, SAP, etc.)	14	5,7%	19,4%	8,6%	4,6%	39,4%	21,7%	0,6%	175
Telecommunication	13	6,0%	20,5%	8,6%	4,6%	37,7%	22,5%		151
Other	13	7,4%	15,7%	10,2%	0,9%	37,0%	28,7%		108
Automotive Software and Systems	11	4,4%	17,5%	8,0%	4,4%	40,1%	25,5%		137
Logistics and Transportation	9	7,4%	14,8%	8,2%	3,3%	44,3%	22,1%		122
Mobile Applications	7	2,3%	17,4%	6,8%	2,3%	48,5%	22,7%		132
Other Embedded Systems and Services	6	5,4%	18,3%	7,5%	7,5%	36,6%	23,7%	1,1%	93
Defense Systems	5	6,0%	19,4%	9,0%	4,5%	41,8%	19,4%		67
Aviation	2	5,3%	5,3%	15,8%		36,8%	36,8%		19
Home Automation	2	3,8%	19,2%	11,5%	3,8%	42,3%	19,2%		26
Games	1		15,0%	10,0%		50,0%	25,0%		20
Robotics	1	4,3%	17,4%	8,7%	4,3%	43,5%	21,7%		23
Space Systems	1	8,3%	16,7%	8,3%	8,3%	37,5%	20,8%		24
Average		5,2%	16,9%	9,0%	4,2%	41,5%	23,8%	0,7%	
Deviation		1,6%	3,3%	2,0%	1,7%	3,9%	3,8%	0,2%	

Based on the data from Table 2, Figure 3 visualizes the averaged data. The figure shows clearly that neither company size nor industry sector make a difference. Figure 3 shows that traditional and agile approaches are combined with each other regardless of company size and industry sector. Moreover, the shares of the different approaches are comparable (for numbers and deviations, see Table 2).

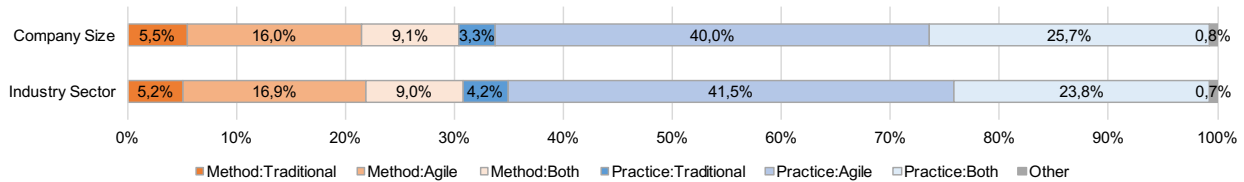


Figure 3 Overview of the process use categorized following the approach classification (a) per company size and (b) per industry sector.

4 Conclusion

Traditional software development processes, complemented by norms, standards, and rules, should support development and product quality. However, these have a negative connotation since they limit flexibility to allow for plannability, predictability, compliance, or simpler pricing/contracting models. Compared to agile methods, these approaches are considered heavy-weight and inflexible, to the point that companies are forced to look for alternatives to increase agility [8]. Hybrid development approaches represent a solution that, regardless of company type and industry sector, enables companies to benefit from both worlds by providing clients and management a safe environment and developers with the demanded flexibility. Our study shows that hybrid development approaches emerge gradually: a majority of 83.9% states that the development approach emerges from experience. Moreover, even though 52.2% of the participants state

1 that the company defines a standard approach, more than a quarter states that development approaches are
2 selected more individually and are adapted during projects in response to given situations.

3 Our report presents the first step into a deeper investigation on the use of hybrid development approaches.
4 Yet, more research is necessary to improve the understanding of the benefits of these approaches. HELENA
5 will continue, and we cordially invite practitioners to participate in future stages of HELENA by sharing
6 their experience.

8 References

- 9 [1] West, D., Gilpin, M., Grant, T., Anderson, A., “Water-Scrum-Fall Is The Reality Of Agile For
10 Most Organizations Today”, available online from: [http://www.storycology.com/up-](http://www.storycology.com/uploads/1/1/4/9/11495720/water-scrum-fall.pdf)
11 [loads/1/1/4/9/11495720/water-scrum-fall.pdf](http://www.storycology.com/uploads/1/1/4/9/11495720/water-scrum-fall.pdf), Forrester Research, 2011, Last accessed: Dec. 14,
12 2016
- 13 [2] Garousi, V., Coşkunçay, A., Betin-Can, A., Demirörs, O., “A survey of software engineering
14 practices in Turkey,” *Journal of Systems and Software*, vol. 108, pp. 148-177, Oct. 2015, DOI:
15 10.1016/j.jss.2015.06.036
- 16 [3] Vijayarathy, L., Butler, C., “Choice of Software Development Methodologies: Do Organiza-
17 tional, Project and Team Characteristics Matter?” *IEEE Software*, vol. 33, no. 5, 2015, pp. 86-94.
- 18 [4] Cusumano, M. et al., “Software Development Worldwide: The State of the Practice,” *IEEE Soft-*
19 *ware*, vol. 20, no. 6, 2003, pp. 28-34.
- 20 [5] Neill, C., Laplante, P. A., “Requirements Engineering: The State of the Practice,” *IEEE Software*,
21 vol. 20, no. 6, 2003, pp. 40-45.
- 22 [6] Jones, C., “Variations in Software Development Practices,” *IEEE Software*, vol. 20, no. 6, 2003,
23 pp. 22–27.
- 24 [7] Theocharis, G., Kuhrmann, M., Münch, J., Diebold, P., “Is Water-Scrum-Fall Reality? On the Use
25 of Agile and Traditional Development Practices,” *Proc. of Intl. Conf. on Product-Focused Soft-*
26 *ware Process Improvement*, Springer, ser. *Lecture Notes in Computer Science* 9459:149-166,
27 2015.
- 28 [8] Nerur, S., Mahapatra, R., Mangalaraj, G., “Challenges of Migrating to Agile Methodologies,”
29 *Communications of the ACM*, vol. 48, no. 5, 2005, pp. 73–78.
- 30 [9] Diebold, P., Zehler, T., “The Right Degree of Agility in Rich Processes,” in Kuhrmann et al.
31 (Eds.) *Managing Software Process Evolution*, Springer, pp. 15-37, 2016
- 32 [10] Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektere, K.,
33 McCaffery, F., Linsen, O., Hanser, E., Prause, C. R., “Hybrid Software and System Development
34 in Practice: Waterfall, Scrum, and Beyond”, *Proc. of Intl. Conf. on Software and Systems Process*,
35 ACM, 2017, (in press).
- 36 [11] Kuhrmann, M., Méndez Fernández, D., „Systematic Software Development: A State of the Prac-
37 tice Report from Germany”, *Proc. of Intl. Conf. on Global Software Engineering*, IEEE, pp. 51–
38 60, 2015.
- 39 [12] Murphy, B., Bird, C., Zimmermann, T., Williams, L., Nagappan, N., Begel, A., “Have agile tech-
40 niques been the silver bullet for software development at Microsoft?” In *Intl. Symp. on Empirical*
41 *Software Engineering and Measurement*, IEEE, pp 75-84, 2013
- 42 [13] VersionOne, “State of Agile Survey”, Available from: [http://www. versionone.com/agile-re-](http://www.versionone.com/agile-re-)
43 [resources/more-resources/blogs/](http://www.versionone.com/agile-re-resources/more-resources/blogs/), 2006-2014.
- 44 [14] Dikert, K., Paasivaara, M., Lassenius, C., “Challenges and Success Factors for Large-scale Agile
45 Transformation: A Systematic Literature Review,” *Journal of Systems and Software*, vol. 119,
46 2016, pp. 87–108.

1 Boxes

2 Hybrid Software Development Approach: A Definition

3 We use the following definition of hybrid software development approaches:

4 *A hybrid software development approach is any combination of agile and/or traditional (plan-driven or*
5 *rich) approaches that an organizational unit adopts and customizes to its own context needs (e.g., applica-*
6 *tion domain, culture, processes, project, organizational structure, techniques, technologies, etc.).*

7

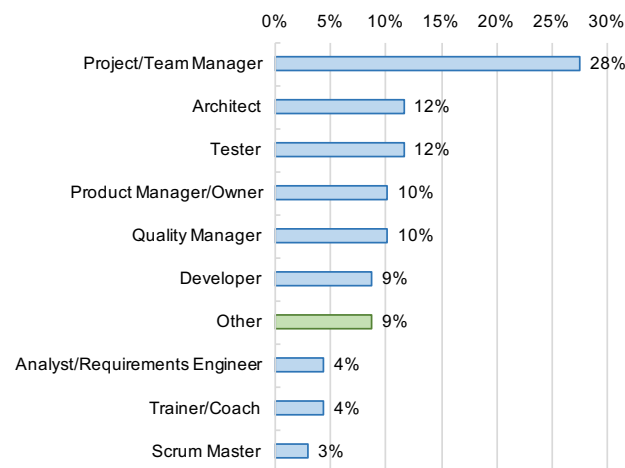
8 Related Studies

9 Recently Vijayasarathy and Butler [3] investigated the question of whether organizational, project and team
10 characteristics influence the choice of development methods. They found a huge variety of development
11 methods applied and strong associations between the characteristics and development approaches used.
12 Compared to older studies, such as [4], [5], they found an increased use of agile methodologies. However,
13 even in 2003, Jones [6] found a substantial diversity in development methods. In another survey from Tur-
14 key with 202 participants, Garousi et al. [2] found Waterfall (53%), agile/lean (45%), and incremental mod-
15 els (38%) the most frequently used. Similar results were found in the studies conducted in Germany [7]. We
16 argue that hybrid approaches emerge naturally due to the challenges coming along with a migration to agile
17 as listed in [8] and [14].

18

19 Demographics

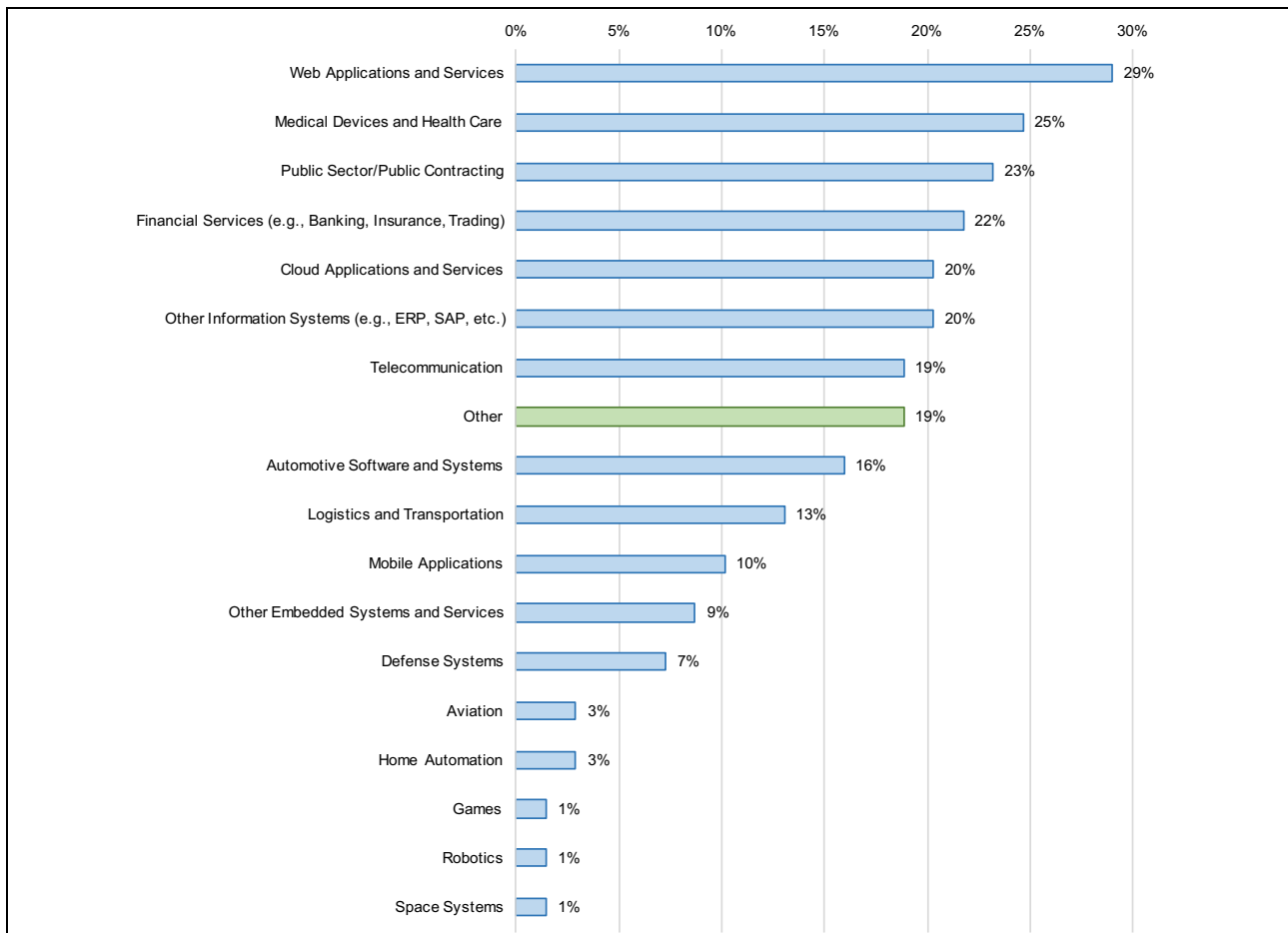
20 The HELENA survey comprised 25 questions¹, and received 69 responses from 16 countries of which five
21 countries had at least five data points: Germany (n=21), Austria (n=10), Turkey (n=9), Ireland (n=6), and
22 Denmark (n=5). Figure 4 shows the roles the respondents have, of which 28% are project or team managers
23 and represent the largest group. Another 20% was identified as product managers/owners and quality man-
24 agers (10% each). The more technical roles comprise architects and testers (12% each) and developers (9%).
25 Another 9% was categorized as “others”, which includes safety managers, compliance managers, and C-
26 level managers. Hence, the profiles cover the whole software and system development lifecycle.



27

28 **Figure 4 Overview of the roles respondents hold.**

29 The respondents' companies cover all sizes and a variety of industry sectors: 17.4% micro-sized (less than
30 10 employees), 20.3% small (11-50 employees), 17.4% medium-sized (51-250 employees), 20.3% large
31 (more than 250 employees), and 24.6% very large (more than 2,500 employees). Figure 5 gives an overview
32 of the participants' industry sectors.



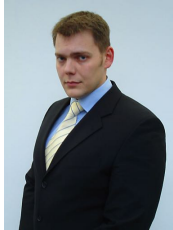
1
2 **Figure 5 Overview of the industry sectors covered by the respondents (the questionnaire allowed for multiple**
3 **selections).**

4 The data shows that companies, even small and medium-sized ones, are active in several industry sectors.
5 Finally, our data shows that approx. 2/3 of the companies use distributed/virtual teams in software/system
6 development of which 26.1% work in a globally distributed way, 20.3% use regionally and 18.8% use na-
7 tionally distributed teams. The remaining 34.8% stated to not implement distributed development.

8

1 Authors (CVs)

2



1) Marco Kuhrmann (Germany)

Marco Kuhrmann is a senior research associate and division manager at the Institute of Applied Software Systems Engineering (IPSSSE) at the Clausthal University of Technology. Contact him at

kuhrmann@acm.org.



2) Philipp Diebold (Germany)

Philipp Diebold is a researcher at the Fraunhofer Institute for Experimental Software Engineering (IESE) leading the topic of agile development in the Process Engineering department. Contact him at philipp.diebold@iese.fraunhofer.de.



3) Jürgen Münch (Germany)

Jürgen Münch is a Professor of Software Engineering and Innovation at Reutlingen University, Germany. His research centers on lean product management, startup and innovation methods, digital business models, business model validation, and agile engineering. Contact him at juergen.muench@reutlingen-university.de.



4) Paolo Tell (Denmark)

Paolo Tell is an assistant professor at the computer science department at the IT University of Copenhagen (ITU). Contact him at pate@itu.dk.



5) Kitija Trektore (Ireland)

Kitija Trektore is a researcher at Regulated Software Research Center (RSRC) in Dundalk Institute of Technology (DKIT). Contact her at kitija.trektore@dkit.ie.



6) Fergal McCaffery (Ireland)

Fergal McCaffery is Director of the Regulated Software Research Centre in Dundalk Institute of Technology (DKIT). He is also the competence leader within Lero for medical device software engineering. Contact him at fergal.mccaffery@dkit.ie.



7) Vahid Garousi (The Netherlands)

Vahid Garousi is an Associate Professor of Software Engineering in Wageningen University, The Netherlands. Contact him at: vahid@vgarousi.com



8) Michael Felderer (Austria)

Michael Felderer is a professor at the University of Innsbruck's Institute of Computer Science and a senior consultant for QE LaB Business Services. Contact him at michael.felderer@uibk.ac.at.



9) Oliver Linssen (Germany)

Oliver Linssen is Professor of Business Informatics at FOM University of Applied Sciences for Economics and Management. He is also a Honorary Professor at Bergische Universität Wuppertal (BUW). Contact him at oliver.linssen@fom.de.



10) Eckhart Hanser (Germany)

Eckhart Hanser is a Professor of Software Engineering at the Baden-Wuerttemberg Cooperative State University (DHBW) Loerrach, head of the Center for Agile process models of the DHBW and speaker of the process models group of the German Informatics Society (GI). Contact him at hanser@dhbw-loerrach.de.



11) Christian R. Prause (Germany)

Christian Prause heads the field Software Product Assurance in the Product Assurance of the DLR Space Administration. Contact him at christian.prause@dlr.de.