

How Agile Is Hybrid Agile? An Analysis of the HELENA data

John Noll¹ and Sarah Beecham²

¹ University of Hertfordshire, j.noll@herts.ac.uk

² Lero, the Irish Software Research Centre, University of Limerick, Ireland,
sarah.beecham@lero.ie

Abstract. **Context:** Many researchers advocate “tailoring” agile methods to suit a project’s or company’s specific environment and needs. This includes combining agile methods with more traditional “plan driven” practices.

Objective: This study aims to assess to what extent projects actually combine agile and traditional practices.

Method: Data from the HELENA survey of nearly 700 projects were examined to assess how many projects combine agile methods and traditional *methods*, and also to what extent they used different software development *practices*.

Results: The data show that, overall, two-thirds of the projects in the survey combine agile and traditional methods to some extent. However, projects that combine agile and traditional *methods* are significantly less likely to use agile *practices* than projects that solely use agile methods.

Conclusions: We hypothesize that the mindset of an organization, rather than technical necessity, determines whether a project will adopt a hybrid vs. purely agile approach.

Keywords: Agile Development Methods; Empirical Software Engineering; Hybrid Agile Development

1 Introduction

Proponents of agile software development methods assert that to be “agile” a project must follow the methodology. For example, Kent Beck claimed that, for Extreme Programming to work, a project must adopt all twelve XP practices, because they support and rely on each other [3].

But researchers and practitioners advocate tailoring agile methods to suit a project’s or company’s specific environment and needs [4], and empirical evidence has confirmed that projects do tailor methods in practice [7, 11, 12, 16, 20, 22]. This may include combining agile methods with more traditional “plan driven” approaches, as well as combining different agile methods.

But how frequently are agile and traditional methods combined into so-called “hybrid” approaches? And, given that agile methods employ known, proven practices, but in an “agile” way [3, 6, 15], is there any difference in practice adoption between hybrid approaches and purely agile projects? This study aims to assess to what extent projects actually combine agile and traditional methods and practices.

To find out, we analyzed data from the HELENA survey [14] of nearly 700 projects, to assess how many projects using agile methods also used traditional methods, and also to what extent projects that might be classified as “purely agile³” or “hybrid” use different software development practices.

The analysis shows that nearly two-thirds of projects reported in the survey combine agile and traditional methods to some degree, but 30% of projects *never* use traditional methods (Table 1, column 1). Further, projects that often use agile methods, but never use traditional methods, are significantly more likely to use agile software development *practices* than projects that combine agile and traditional methods. This suggests that hybrid projects *replace* agile practices with traditional practices, rather than *augment* agile methods with traditional practices. We hypothesize that this difference is due as much to an organization’s “mindset” as to technical requirements.

The remainder of this paper is organized as follows: in Section 2 we introduce the background to the problem, and define our research questions. Section 3 describes the method used. In Section 4 we present the results; Section 5 discusses their implications and presents our conclusions.

2 Background

2.1 Agile and Traditional Development Approaches

Agile adoption in industry is growing year by year. According to the State of Agile Report published in December 2018 [23], the top reasons for this are to accelerate software delivery, enhance ability to manage changing priorities, increase productivity, improve business/IT alignment, enhance software quality and enhance delivery predictability. The agile philosophy, as described in the Agile Manifesto, centers around *a set of iterative and incremental software engineering practices* [10]. However, the actual practices contained within this *set* have evolved over time, and vary from project to project. They are even said to be revolutionary [7]. In a recent survey, researchers identified 36 such agile practices [14]. As coined by Bertrand Meyer [18], “every project I’ve seen embraces a subset of the chosen method’s ideas, rejecting those that don’t fit its culture or needs ...”. Also, according to Meyer [18], agile methods are not a negation of what came before, and can be considered as another brick in the software development wall. Conboy concurs [6], stating that although agile methods are not entirely new or that different to development methods used before, the movement can be branded as an alternative to traditional development methods [6].

Kuhrmann et al [12] argue that “hybrid approaches emerge naturally because of the challenges accompanying a migration to agile,” suggesting that hybrid approaches are driven by need rather than mindset. Marinho et al [17] also suggest that global development projects use a combination of agile and traditional methods because of the need to maintain a level of forward planning and structure while benefitting from a flexible and collaborative approach offered by agile practices. These views support

³ “Purely agile” means a project often or always uses at least one agile method, but never uses any traditional methods; see Section 4 for details.

Tell and colleagues’ recommendation that research should focus on the practices used, rather than debate about what is the ‘right’ method [22].

There are hints in the empirical research to suggest that this reliance on traditional methods might be due to project size since large projects may require more coordination and heavier methodologies than offered by agile methods [1, 9]. Or possibly there is a perceived need for upfront precise and comprehensive planning, and defining the architecture prior to testing and implementation, that push project managers to hold onto traditional plan driven methods [19]. It appears that in practice, adapting the development approach to the context demands a balance of methods and practices [2, 4, 24].

2.2 Research Questions

In light of these arguments, we seek to answer the following two research questions:

1. To what extent is a hybrid approach combining agile and traditional *methods* used in practice? In this context, we define a *method* as an approach to managing a software project that uses a set of software development *practices* in a certain way; examples of methods are Scrum, Waterfall, and Feature Driven Development (see Section 3 for a complete list of methods considered).
2. What *practices* are used by projects that employ such a hybrid approach? A *practice* is a particular technique for developing software, such as pair programming or model-driven development, that may be part of many different methods (see Section 4 for the list of practices considered).

3 Method

In attempt to answer our research questions, we used the HELENA (‘Hybrid dEveLopmENt Approaches in software systems development’) data set that is the result of a multi-year effort to investigate how real software projects combine agile and traditional software development methods [14]. More details can be found on the official website⁴.

HELENA used an online survey [5, 21] to collect data from practitioners about the development methods they use in their projects, and what practices they use [13].

The survey comprises 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) [14]. The survey was distributed to personal contacts of the participating researchers, through posters at conferences, and to mailing lists and social media; in total, the survey yielded 1,467 responses, of which 690 were complete [14]; the results presented in this paper are based on the 690 complete responses.

3.1 Data Analysis

The analysis in this paper focuses on two aspects of the survey: what *methods* are used by projects, and what *practices* are used. Following Kuhrmann and colleagues [13],

⁴ HELENA Survey: <https://helenastudy.wordpress.com/>

methods were categorized as *traditional*, *agile*, or *generic*. Scrum, Safe, Lean, LESS, Nexus, XP, Kanban, DevOps, ScrumBan, Crystal, DSDM and Feature-driven development (FDD) were categorized as agile, while Waterfall, Spiral Model, V-Model, RUP, PRINCE2 and SSADM were categorized as traditional. Iterative development⁵, Domain-Drive Design (DDD), Model Driven Architecture (MDA), Team Software Process (TSP), and Personal Software Process (PSP), were classified as generic, since the approach does not fit into either the agile or traditional category. Respondents were asked to rate the frequency with which their project applied each of these methods on a five point scale ranging from *never* to *rarely*, *sometimes*, *often*, and *always* used. The HELENA survey also asked respondents to rate the frequency with which a project uses 36 development practices, on the same five point scale. When analyzing practice use, we followed Tell and colleagues [22] approach and classified a practice as “used” when respondents stated their projects used the practice *rarely*, *sometimes*, *often*, or *always*.

4 Results

Table 1 shows the frequency that projects use and combine agile and traditional methods. The majority (458 of 690, or 66%) of projects combine agile and traditional development methods; these are shown in **boldface** in Table 1. Conversely, 171 projects (25%) *often* or *always* use at least one agile method, but *never* use any traditional methods; these are shown in *italics* in Table 1. Finally, only 43 projects (6%) *never* use agile methods.

Table 1. Frequency of traditional and agile method combination.

agile	traditional					TOTAL
	never	rarely	sometimes	often	always	
never	(21) 3%	(1) 0%	(3) 0%	(10) 1%	(8) 1%	(43) 6%
rarely	(2) 0%	(1) 0%	(3) 0%	(13) 2%	(5) 1%	(24) 3%
sometimes	(16) 2%	(9) 1%	(24) 3%	(24) 3%	(17) 2%	(90) 13%
often	(55) 8% †	(53) 8%	(56) 8%	(82) 12%	(20) 3%	(266) 39%
always	(116) 17% †	(44) 6%	(42) 6%	(33) 5%	(32) 5%	(267) 39%
TOTAL	(210) 30%	(108) 16%	(128) 19%	(162) 23%	(82) 12%	(690) 100%

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 projects perform traditional methods at the level shown.

Entries in **bold** are classified “hybrid” while entries marked with a † are classified “pure agile.”

⁵ While Iterative development is a key feature of agile methods some traditional approaches (RUP and the Spiral Model, for example) also incorporate iterations [15]; consequently, we followed Kuhrmann et al. [13] and classified iterative development in the generic category.

Table 2. Comparison of practices use by 171 Agile and 458 Hybrid projects.

practice (n=36)	Agile (n=171)	Hybrid (n=458)	diff p value	practice (n=36)	Agile (n=171)	Hybrid (n=458)	diff p value
retrospectives*	68%	48%	20 ***	onsite customer*	20%	19%	1 0.8
refactoring*	63%	43%	20 ***	iteration planning*	58%	57%	1 0.8
cont integration*	75%	56%	19 ***	scrum-of-scrums	19%	19%	0 1
cont deployment*	56%	38%	18 ***	coding stds	70%	71%	-1 0.9
auto unit test	67%	50%	17 ***	auto theorem prv	2%	4%	-2 0.3
pair programming*	30%	15%	15 ***	security testing	25%	29%	-4 0.4
expert/team estimates	56%	41%	15 ***	burn down charts	37%	41%	-4 0.5
backlog mgt	75%	60%	15 ***	end-to-end testing	50%	55%	-5 0.3
limit WIP	44%	30%	14 **	auto code gen	16%	21%	-5 0.3
collective code own*	51%	37%	14 **	model checking	4%	10%	-6 *
code review	73%	59%	14 **	destructive test	6%	12%	-6 *
velocity-based plan	36%	23%	13 **	design reviews	36%	43%	-7 0.1
TDD*	33%	20%	13 ***	prototyping	33%	41%	-8 0.1
def of done*	62%	49%	13 **	formal estimates+	2%	13%	-11 ***
daily standup*	69%	56%	13 **	use case modeling	14%	30%	-16 ***
iter/sprint reviews	69%	58%	11 *	arch spec	32%	49%	-17 ***
user stories	63%	53%	10 *	formal spec+	6%	24%	-18 ***
release planning	69%	65%	4 0.4	detail design/spec	16%	34%	-18 ***

Values marked '%' the fraction of projects in the column that use the practice *rarely*, *sometimes*, *often*, or *always*. The practices are sorted by the difference in fractions, where $diff = Agile\% - Hybrid\%$. The statistical significance of the difference is indicated in the p value column, where '*' is p value $\leq .05$, '**' is p value $\leq .01$, and '***' is p value $\leq .001$; p values $> .05$ are shown as their actual values.

Practices marked with '*' in column 1 are classified as "agile"; marked with '+' are classified as traditional [12].

We compared the frequency that purely agile projects use certain practices, to hybrid projects that combine agile and traditional methods. We define a *purely agile* project as one that respondents stated *often* or *always* uses at least one agile method, but *never* uses any traditional method. We define a *hybrid* project as one that respondents stated uses at least one agile method, and also uses at least one traditional method, where "uses" means at least *rarely*. In Table 1, purely agile projects are marked with a †, and hybrid projects are shown in **boldface** type.

Comparing purely agile projects to hybrid projects, Table 2 shows the difference in the fraction (as %) of projects in each category that use a given practice, and whether that difference (*diff* column) is significant according to the Chi-square comparison of proportions. A practice was considered to be "used" if respondents reported it was used more often than *never* (in other words, at least *rarely*).

This table shows that there are several practices associated with agile methods [12] – refactoring, continuous integration, continuous deployment, collective code ownership, definition of done and daily standups – that are performed by a majority of Agile

projects, but significantly fewer Hybrid projects. Also, pair programming and test driven development (TDD) are performed significantly more often by purely agile projects, although not by a majority.

Conversely, formal estimates and formal specification – that are classified as traditional practices [12] – are performed by a minority of Hybrid projects, but significantly less by Agile projects.

Only four practices – release planning, iteration planning, coding standards, and end-to-end testing – are performed by a majority of projects at the same frequency whether purely agile or hybrid.

5 Discussion and Conclusions

Our results show that a majority of projects in the HELENA survey combine agile and traditional methods. This is consistent with other findings that show companies tailor methods to suit their context [7, 11, 12, 16, 20, 22].

However, our results also show that a substantial number of projects are able to use only agile methods, without employing any traditional methods; this is consistent with the views of agile method advocates such as Kent Beck [3].

Further, Table 2 shows that purely agile projects – those that use agile methods often or always, but *never* use traditional methods – use agile *practices* significantly more often than hybrid projects. In addition, purely agile projects use traditional practices significantly less often than hybrid projects. This suggests that hybrid projects *replace* agile practices with traditional practices, rather than *augment* agile methods with traditional practices.

Tell and colleagues argue that agile methods have changed the mindset, but not the practices, that projects adopt, stating that their results “reveal a small core of practices used by practitioners regardless of the (hybrid) development method selected [22].” While our analysis also shows there is a set of practices – release planning, iteration planning, coding standards, and end-to-end testing – that are likely to be used by a majority of projects in both categories, our analysis also suggests that being “agile” is more than just adopting an agile mindset and method; rather, those projects that state that they solely use agile *methods* also report that they actually use agile *practices* as well.

Returning to our research questions:

RQ1: To what extent is a hybrid approach combining agile and traditional methods used in practice? The answer to this question is, the majority (66%) of projects in the HELENA survey combine agile and traditional methods (see Section 4).

RQ2: What practices are used by projects that employ such a hybrid approach? The answer to this question is somewhat more nuanced. All practices queried in the HELENA survey are used by at least some hybrid projects; however, some practices are used more often than others. As shown in Table 2, agile practices are used significantly less often by hybrid projects than by purely agile projects, and certain traditional practices are used significantly more often by hybrid projects than agile projects.

The HELENA survey did not provide extensive opportunities for respondents to explain why they adopt methods and practices, so we cannot say definitively why the

adoption of practices is different between hybrid and purely agile projects. However, Michal Doležel suggested a possible reason, which he called “institutional logic” [8]: organizations with an agile “logic” will view agile methods, and practices, more favorably than organizations with a traditional “logic.” As such, we propose the following hypotheses based on this notion:

H1: Organizations with an agile mindset are more likely to be purely agile, and therefore adopt agile practices, regardless of technical drivers.

H2: Organizations with a traditional software development mindset are more likely to adopt a hybrid approach, and therefore adopt traditional practices, regardless of technical drivers.

Acknowledgments This work was supported, in part, by Science Foundation Ireland grants 10/CE/I1855 and 13/RC/2094 to Lero - the Irish Software Research Centre (www.lero.ie).

Bibliography

- [1] Aitken, A., Ilango, V.: A comparative analysis of traditional software engineering and agile software development. In: 2013 46th Hawaii International Conference on System Sciences. pp. 4751–4760 (Jan 2013)
- [2] Akbar, R., Safdar, S.: A short review of global software development (gsd) and latest software development trends. In: Computer, Communications, and Control Technology (I4CT), 2015 International Conference on. pp. 314–317. IEEE (2015)
- [3] Beck, K., Gamma, E.: Extreme Programming Explained: Embrace Change. Addison Wesley, Boston, USA (Nov 2000)
- [4] Boehm, B., Turner, R.: Using risk to balance agile and plan-driven methods. *Computer* **36**(6), 57–66 (2003)
- [5] Ciolkowski, M., Laitenberger, O., Vegas, S., Biffl, S.: Practical experiences in the design and conduct of surveys in empirical software engineering. In: Empirical methods and studies in software engineering, pp. 104–128. Springer (2003)
- [6] Conboy, K.: Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research* **20**(3), 329–354 (2009)
- [7] Diebold, P., Zehler, T.: The right degree of agility in rich processes. In: Managing Software Process Evolution, pp. 15–37. Springer (2016)
- [8] Doležel, M.: Possibilities of applying institutional theory in the study of hybrid software development concepts and practices. In: Kuhrmann, M., Schneider, K., Pfahl, D., Amasaki, S., Ciolkowski, M., Hebig, R., Tell, P., Klünder, J., Küpper, S. (eds.) Proceedings of the International Conference on Product-Focused Software Process Improvement (PROFES 2018). Lecture Notes in Computer Science, vol. 11271, pp. 441–448. Springer (2018)
- [9] Dyba, T., Dingsoyr, T.: What do we know about agile software development? *IEEE software* **26**(5), 6–9 (2009)
- [10] Fowler, M., Highsmith, J.: The agile manifesto. *Software Development* **9**(8), 28–35 (2001)

- [11] Klünder, J., Hohl, P., Fazal-Baqaie, M., Krusche, S., Küpper, S., Linssen, O., Prause, C.R.: HELENA study: Reasons for combining agile and traditional software development approaches in german companies. In: International Conference on Product-Focused Software Process Improvement. pp. 428–434. Springer (2017)
- [12] Kuhrmann, M., Diebold, P., Munch, J., Tell, P., Trektore, K., McCaffery, F., Garousi, V., Felderer, M., Linssen, O., Hanser, E., Prause, C.R.: Hybrid software development approaches in practice: A european perspective. *IEEE Software* **36**(4), 20–31 (July 2019)
- [13] Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektore, K., McCaffery, F., Linssen, O., Hanser, E., 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. pp. 30–39. ACM (2017)
- [14] Kuhrmann, M., Tell, P., Klünder, J., Hebig, R., Licorish, S., MacDonell, S.: Helena stage 2 results. Tech. rep., HELENA consortium (11 2018)
- [15] Larman, C., Basili, V.R.: Iterative and incremental development: A brief history. *Computer* **36**(6), 47–56 (2003)
- [16] Marinho, M., Luna, A., Beecham, S.: Global software development: practices for cultural differences. In: International Conference on Product-Focused Software Process Improvement. pp. 299–317. Springer (2018)
- [17] Marinho, M., Noll, J., Richardson, I., Beecham, S.: Plan-driven approaches are alive and kicking in agile global software development. In: International Symposium on Empirical Software Engineering and Measurement (ESEM). ACM/IEEE (2019)
- [18] Meyer, B.: Making sense of agile methods. *IEEE Software* **35**(2), 91–94 (March 2018)
- [19] Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies. *Communications of the ACM* **48**(5), 72–78 (2005)
- [20] Paez, N., Fontdevila, D., Oliveros, A.: HELENA study: Initial observations of software development practices in argentina. In: International Conference on Product-Focused Software Process Improvement. pp. 443–449. Springer (2017)
- [21] Shull, F., Singer, J., Sjøberg, D.I.: Guide to advanced empirical software engineering. Springer (2007)
- [22] Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S.G., Münch, J., Pfahl, D., Linssen, O., Kuhrmann, M.: What are hybrid development methods made of?: an evidence-based characterization. In: Proceedings of the International Conference on Software and System Processes. pp. 105–114. IEEE Press (2019)
- [23] VersionOne, Inc.: 13th Annual State of Agile Development Survey. <https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508> (2018), web page, accessed 5 August 2019.
- [24] Vinekar, V., Slinkman, C.W., Nerur, S.: Can agile and traditional systems development approaches coexist? an ambidextrous view. *Information systems management* **23**(3), 31–42 (2006)