Check for updates

Received: 12 February 2021 | Revised: 20 August 2021 | Accepted: 28 August 2021

DOI: 10.1002/smr.2382

RESEARCH ARTICLE - METHODOLOGY

Journal of Software: Evolution and Process WILEY

Goals and challenges in hybrid software development approaches

Nils Prenner[1] | Carolin Unger-Windeler[2] | Kurt Schneider[1]

[1]Software Engineering Group, Leibniz University Hannover, Hannover, Germany

[2]Research and Development, Baker Hughes, Celle, Germany

Correspondence
Nils Prenner, Software Engineering Group, Leibniz University Hannover, Hannover, Germany.
Email: nils.prenner@inf.uni-hannover.de

Abstract

The number of companies that use agile methods increases steadily. However, these companies often do not implement a pure agile approach but combine agile and plan-based methods to so-called hybrid development approaches. However, the development of these approaches is rather difficult for the companies, since agile and plan-based approaches often follow opposite concepts. To benefit from agile and plan-based approaches at the same time, the companies have to identify and address the conflicts between agile and plan-based methods. The conflicts depend on the goals that are pursued with the implementation of agile and plan-based methods. However, there is no overview of the exact goals that are pursued in hybrid approaches and which challenges and conflicts arise with them. Therefore, we conducted a systematic mapping study to gather and analyze the goals and challenges in hybrid development approaches. The mapping study is focused on literature that presents the actual needs and goals of companies and projects. Based on our results, we present the influence factors that cause conflicts in hybrid approaches and discuss how these conflicts can be addressed.

KEYWORDS
agile development, hybrid development approaches, plan-based development, systematic mapping study

1 | INTRODUCTION

Software companies are under constant market pressure to produce new features fast, cost effectively, and with high quality.[1] Therefore, they continuously strive to find the best way to produce software that is attuned to their special needs. Since the introduction of agile development, the number of companies that introduce agile methods and replace plan-based processes has increased steadily.[2] Plan-based approaches have the problem that they are resistant to changing requirements and that developers get late feedback from the customer and stakeholders.[3] By their iterative way of working, agile methods address these problems. However, also agile methods have their deficiencies, especially at managing a large-scale or distributed development and the compliance to security and safety standards.[2,4] Exactly for these issues, plan-based development methods were created.[5]

Recent studies show that companies often do not use only agile methods but rather a mix of agile and plan-based methods in order to combine the benefits of both approaches.[6-9] Approaches where different methods are combined are called *hybrid development approaches* (short: hybrid approaches) and have become state-of-the-practice.[10] Kuhrmann et al.[8, p. 2] define hybrid approaches as

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
© 2021 The Authors. Journal of Software: Evolution and Process published by John Wiley & Sons Ltd.

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

This definition considers every development approach where a combination of different methods and practices from agile or plan-based approaches is used. This includes also approaches where only different agile methods are combined. In this work, we concentrate on approaches where agile and plan-based methods are used, that is, a subset of hybrid approaches. We call these approaches agile-plan-based hybrid approaches (short: APH approaches).

1.1 | Importance and benefits of the topic

The variety of business areas and contexts in the field of software development makes it impossible to create a development approach that fits all companies.[11] For example, a company that only has a small development team has to follow a different process than one with many teams that work simultaneously on the same product. Therefore, software companies have to create an approach that fits their special needs. Klünder et al[10] found out that most development approaches are created by an evolutionary process where experience is the driver of process change. However, experience is only useful if a process that already exists is improved whereas the creation of a new development approach is difficult.[12] This applies particularly to the creation of APH approaches where companies can choose from a variety of methods and practices[13] and agile and plan-based approaches with their often opposite principles clash.[3] The opposite principles of agile and plan-based methods create conflicts and challenges for companies when they use APH approaches. For the successful implementation of an APH approach, where both approaches are pursuit at the same time, the companies have to address the challenges and conflicts that arise by finding a compromise between both. The occurrence of conflicts and challenges depends strongly on the goals that are pursued because they influence which principles and thus methods are used. Longtime, APH approaches were discussed based on the combination of methods without the goals that companies try to reach. There is no structured approach that supports companies in finding and addressing the conflicts. Also existing models for the support of the creation of hybrid approaches, for example, by Bohem and Turner[3] or Tell et al,[9] do not consider these goals and arising challenges in hybrid approaches. Therefore, it is important to examine the goals and challenges in APH approaches.[13] We want to lead the way toward a better understanding of the goals that are pursued by APH approaches and which goals create a need for agile or plan-based methods. By a better understanding of the goals and challenges in APH approaches and what APH approaches have to accomplish, new research directions and needs are identified for the research community. The practitioner community also benefits from the results because they are more aware for the challenges in APH approaches and can mitigate their impact.

1.2 | Contribution

Several experience reports, case studies, and other investigations of APH approaches were already published. These publications name goals and describe challenges that companies have and face when they use APH approaches but are mostly focused on a single case. Therefore, research lacks the overview of these goals and challenges.[13] For the first step of getting a better understanding of the goals behind APH approaches and the related challenges, we want to form a baseline of the scientific knowledge. For that, we conducted a *systematic mapping study (SMS)*. Based on our results, we present in this publication three contributions:

1. We identify 13 goals for the use of agile methods and 12 goals for the use of plan-based methods in APH approaches and analyze how these goals are addressed. Among the most important goals for agile methods are the adaptability of change and the dealing with uncertain requirements. Plan-based methods are mostly used to meet safety, security, and regulatory requirements and to manage large scale development.
2. We identify eight challenges and conflicts that arise in APH approaches. These challenges come mainly from the need to implement agile and plan-based methods simultaneously for the same activity.
3. Based on our results, we present the influencing factors that cause conflicts in APH approaches. In order to guide future research and support the creation of APH approaches, we discuss how these conflicts can be addressed.

1.3 | Outline

The remainder of the paper is structured as followed: In Section 2, we present the principles of agile and plan-based development and present the body of knowledge in the field of APH approaches, followed by Section 3, where we explain our research questions and research design. In

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Section 4, we present the results of our SMS, and in Section 5, we discuss our findings. After presenting and discussing the results, we present the factors that cause conflicts in APH approaches and discuss how these conflicts can be addressed in Section 6. In the end, we conclude and present future work in Section 7.

2 | BACKGROUND AND RELATED WORK

In this section, we present background information about plan-based and agile development. Further, we give an overview of the body of knowledge of hybrid approaches and the related work in this field.

2.1 | Plan-based development

Plan-based development approaches demand that the customers' requirements are clear and complete at the beginning of a project.[14] This enables the conduct of software projects in phases with a detailed defined process. The most prominent model for plan-based approaches is the waterfall model.[9] The model consists of mainly five phases, namely, requirements analysis, design, implementation, testing, and operations.[5] Each of the phases is accompanied by an extended documentation and validation process before the next phase can start to ensure quality.[15] Because all requirements are gathered at the beginning, the whole project can be planned in advance and detailed plans and deadlines can be created.[3] This structure is especially suited for large-scale projects or safety-critical applications.[3]

The problem of plan-based approaches is that the customers often do not fully know the requirements and that projects are often subject to unforeseen changes.[16] Because of the detailed plans in plan-based approaches, changes are hard to integrate into the process.[3] Additionally, plan-based approaches generate late results and feedback because the product is deployed a whole at the end of the project.[5]

2.2 | Agile development

In order to address the difficulties of plan-based approaches, in the early and mid-1990s, the movement of agile development formed. In this movement, a lot of different methods and frameworks emerged. All these practices and methods are gathered under the *Agile Manifesto* that defines the values and principles of agile development.[17] They all adopt a form of iterative development where the product is not deployed only at the end of the project but at regular intervals. In this way, feedback is constantly gathered and changes are approved, even late in the development.[15] In contrast to plan-based development, agile methods are people centered because they are seen as the success for a development project.[3] Agile methods propagate self-organization as the best way to manage development teams and that managers should change from a command and control towards a supportive attitude.[16] In agile development the most often used agile framework is the Scrum framework that is often combined with practices from Extreme Programming.[9]

2.3 | Hybrid development

Soon after the publication of the *Agile Manifesto*, Boehm and Turner[3] identified that agile development is not suited for all types of projects and that companies have to combine agile and plan-based methods. They investigated the differences between agile and plan-based development and discussed which approach is suitable for which kind of project. From this, they derived five factors that define the home ground for agile and plan-based approaches. These factors are as follows:

- Dynamism: Requirements change rate
- Culture: Chaos vs. order
- Size: Number of personnel
- Criticality: Loss due to the impacts of defects
- Personnel: Skill level

A low change rate for requirements is rather suited for plan-based approaches and a high one rather for agile approaches. If the staff rather thrives under order, a plan-based approach is appropriate. Otherwise, if the staff is striving on chaos, an agile approach is suitable.[3] A project with a small number of personnel is rather suitable for an agile approach and a large project for a plan-based approach. If errors inside the software can threaten several lives, a plan-based approach should be used. If only the comfort of users is affected, an agile approach is sufficient enough.

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Experienced developers with a high skill level can master agile development. Employees with less experience and skill level are more suitable for a plan-based approach.[3]

If the examination of a project shows that it neither lies completely in the agile nor in the plan-based home ground an APH approach is needed. For that, Boehm and Turner[3] present a risk-based approach, but they do not describe how these risks influence generally the development approach and which principles have to be followed to meet the demands of a project. Their approach is based on the assumption that different parts of the software can be encapsulated by architecture and can be developed separately from each other.[3] Based on that, the different parts can be addressed by the suitable development approach. Also, they do not discuss the simultaneous presence of agile and plan-based practices for the same activity but rather describe them as interchangeable bricks.

Barlow et al[18] also propose a model to guide the decision regarding the use of an either plan-based, agile, or APH development approach. They put also into account the possibility of requirement changes and the size of the project. Further, they consider the type of existing dependencies in a development project. If there are primarily sequential dependencies a rather plan-based approach should be used. Otherwise, if the dependencies are primarily reciprocal a rather agile approach should be used. However, the nature of the dependencies in a development approach strongly depends on how the development approach is structured and which principles are followed. Therefore, the type of dependencies is a result of the development approach and not the other way around. As long the requirements cannot be completely defined at the beginning of the project and changes occur, a project will always have reciprocal dependencies. Today's software development operates in a fast-evolving environment where requirements are often uncertain and therefore reciprocal dependencies are necessary. However, following the recommendations of Barlow et al[18] for the creation of APH approaches, projects should be analyzed regarding where sequential dependencies can be used and where reciprocal dependencies are necessary. Similar to Boehm and Turner,[3] they lack the description of how their model can be used to build a concrete development approach.

Thesing et al[14] conducted an interview study to investigate the advantages and disadvantages of plan-based or agile approaches. Based on their results, they derive a model to determine whether a project should follow a plan-based or agile approach. First, they define exclusion criteria for agile development. Projects that cannot be separated into increments or where changes are not possible are excluded from agile development. In a second step, they define further criteria based on project constraints and the people and culture. The project constraints consider the clarity of the scope, the duration of the project, and the need for rapid results, and need for fixed resources. The people and cultural aspects consider the organizational type and culture and the size and characteristics of the development team.[14] Thesing et al.[14] mention the existence of APH approaches only in passing and do not describe how their model can be used to create APH approaches.

Heeager[19] investigated if agile and disciplined approaches are combinable or only compatible. She defines that both approaches are combinable if a project follows a quality assurance standard and uses at the same time an agile approach without compromises. On the other hand, both approaches are only compatible if the project follows a quality assurance standard and can at the same time integrate some agile principles and practices. She concludes that agile and disciplined approaches are only compatible and need to be adapted. Further discussions of this topic were made by McMichael and Lombardi[20] and Fritzsche and Keil.[21] Discussions regarding the applicability of agile in a distributed context were made by Ramesh et al.[22] or in a large scale context by Cao et al.[23] They all conclude that agile approaches have to be adapted to fit in these contexts and partly plan-based methods have to be used. To address these contexts, some case studies and proposals of APH approaches were published.[24] In 2011, West[2] stated that most companies combine a waterfall-like process with the Scrum framework. He named this combination "Water-Scrum-Fall". He also predicted that the future of software development is less about a specific method but the right set of practices for a certain situation or problem.

Although hybrid approaches were discussed early on in the research community, there was no research about how far hybrid approaches are spread in the industry and which methods were used. To fill this gap, Theocharis et al.[6] performed a systematic literature review to examine which development approaches were used and combined in practice. The research questions of this publication are shown in Table 1. Their results show

TABLE 1 Overview of related systematic literature reviews in the field of hybrid development approaches

Title	Author	Year	Research questions	# of examined papers
Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices	Theocharis et al.	2015	Which development approaches are used in practice? Which development approaches are combined in practical use? Are there patterns and trends observable?	22 papers
When Agile Meets Waterfall	Kusters et al.	2017	What are the issues (risks and problems) at the interface of agile and traditional development which impact coordination and cooperation?	12 papers
How are Hybrid Development Approaches Organized? - A Systematic Literature Review	Prenner et al.	2020	What are the activities in hybrid development approaches and how are they arranged? How are the activities conducted in hybrid development approaches?	24 papers

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

that the waterfall model and Scrum are among the most used methods and that they are often combined. To further investigate the use of agile and plan-based methods and based on their results, the *HELENA study** was created. The study was addressed to practitioners, and questions about their methods and practices use were asked. The study was published in 2017 and resulted in 1467 data points with 691 complete answers.[25] Noll and Beecham[26] analyzed the data and found out that in 66% of the cases an APH approach was used. They state that the use of more agile or plan-based methods depends rather on the mindset than technical factors.

Klünder et al[10] found out that the most often methods to form combinations are Scrum, Iterative Development, Kanban, Waterfall, and DevOps. They also investigated the reasons for companies to use different method combinations. The goals most often addressed by companies are improved productivity, improved external product quality, and improved planning and estimation. However, this investigation does not take into account what the companies want to achieve by choosing a concrete method or practice.

Tell et al[9] also analyzed the data and found no dependencies between the application of different method combination and the company's size or business field. Instead, they state that hybrid approaches are state-of-the-practice. They could confirm the assumptions of West[2] that a lot of companies follow the "Water-Scrum-Fall" like process. They also confirm West's[2] statement that the future of software development lies in finding the right set of practices instead of a concrete development method. In their analysis, they found a set of core practices that are used consistently among the analyzed data. The set contains of automated unit testing, code reviews, coding standards, prototyping, and release planning. They show how often these practices are combined with the base methods also found by Klünder et al.[10] In a deeper factor analysis, Klünder et al.[12] investigated the influence of factors, for example, company size or business area, on the choice of methods. They found few context factors that influence the choice of development methods and conclude that there must be further factors. We state that these missing influence factors are the goals that are pursued with the use of agile and plan-based methods. However, this aspect is not considered by the HELENA study and has to be further investigated. The HELENA study also considers only the used methods and practices and considers them as building bricks. Klünder et al[27] investigated which methods and practices are used for which project disciplines. Their results show that methods and practices are often not used as intended and practitioners often deviate from existing methods. They discuss that practitioners have to be aware whether they deviate from a method, how they do it and, why. Before applying a method, practitioners have to analyze the reasons for which the method was created. Otherwise, methods may are used in a way or a context they were not designed for.

Similar to the other presented research, Klünder et al[27] also only investigate whether certain methods and practices are applied or not without considering the inner structure of hybrid approaches. The inner structure considers how the phases, activities, roles, and artifacts from plan-based and agile approaches are arranged and connected in hybrid approaches.[13] However, the investigation of the inner structure of hybrid approaches is important in order to understand them.[9,24]

Overall, there is little research about the inner structure of hybrid approaches. Boehm and Turner[28] investigated the challenges of implementing agile processes in traditional companies. They identified conflicts in the area of the development process, the business process, and the staff. However, only the parallel existence of agile and traditional processes are considered. Theobald and Diebold[29] and Kusters et al[30] investigate the problems at the interfaces between agile and traditional processes, yet only here a parallel existence of both processes is considered. Theobald et al[29] identified in the context of a workshop problems on the interfaces between the team and the project, the project and the organization, to the customer, to subcontractors, and to regulatory authorities. Kusters et al[30] conducted a systematic literature review, shown in Table 1. They discovered problems in the organization and structure, business process and control, and documentation and communication.

In order to strengthen the knowledge about the inner structure of APH approaches, we conducted in Prenner et al[24] a systematic literature review to collect and analyze descriptions of APH approaches. The research questions for this publication are shown in Table 1. The problem with proposed APH approaches is that the building of them depends on the respective goals. Therefore, if a proposal works in one particular case, it does not have to work necessarily in another one. Since APH approaches have to be individualized, we analyzed the found approaches regarding their general structure to support the building of APH approaches. In Figure 1, we present the found methods to structure APH approaches.[24] The first method we found was the Waterfall-Agile-Method (WAM) where a waterfall process is used with an integrated agile part during development. The second is the Waterfall-Iterations-Method (WIM) where an iterative process is used and inside of the iterations, a waterfall phase model is applied. The third method is the Pipeline-Method (PM) where the phases of the waterfall model are conducted in parallel for different increments. N is the increment that is currently under development in iteration I. At the same time, the requirements are gathered for increment $N + 1$ and the tests are conducted for increment $N - 1$. In the iteration $I + 1$, all increments are shifted to the next phase. We also discovered that the WIM or the PM are often integrated into the WAM during the development phase.

Our investigation of the structure of APH approaches showed that agile and plan-based processes are not separated in APH approaches but are intertwined into each other. Therefore, we concentrate our research on APH approaches where both processes are used at the same time and compromises between both have to be made. To find these compromises is very difficult, and there is a lack of research that is concerned with this area. To create a compromise, first, the goals behind APH approaches and the challenges that arise with them have to be understood.

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

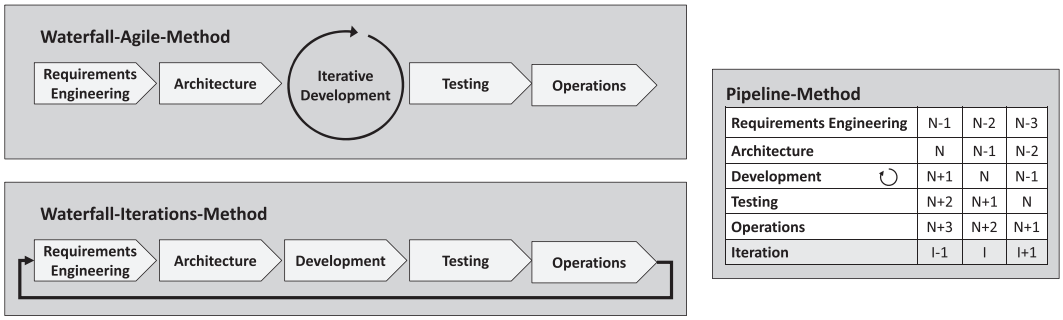

FIGURE 1 Overview of the structures of APH approaches presented in Prenner et al.[24]

3 | RESEARCH DESIGN

This work aims at getting a better understanding of the goals and challenges in APH approaches. There are publications about APH approaches in form of case studies and experience reports but their results are often limited to one case. In order to combine the insights from these case studies and create an overview of the existing scientific knowledge, we decided to perform a systematic mapping study. In contrast to systematic literature reviews, where the aim is to look for evidence, systematic mapping studies are designed to get an overview of a research area and structure it.[31] In this section, we describe our research questions and research method.

3.1 | Research questions

For our systematic mapping study, we formulated the following three research questions:

RQ1: What are the mentioned goals that lead to a combination of agile and plan-based methods in APH development approaches?
Common knowledge is that companies use APH development approaches to combine the benefits of agile and plan-based approaches. An overview of what the companies want to accomplish by the use of agile and plan-based methods inside of APH approaches is missing and these aspects are therefore rather unclear. It is important to get an overview of the goals in APH approaches to identify and understand the conflicts that arise in APH approaches. Therefore, we want to get a scientific overview of the knowledge about the goals in APH approaches.

RQ2: How are the goals inside of APH development approaches addressed?
There is also no overview of which agile or plan-based methods or practices are used to reach the goals in APH approaches. This is important to lead further research, understand the conflicts and support companies in the creation of APH approaches. Therefore, we want to investigate how the goals are addressed with this research question.

RQ3: What are the challenges in APH development approaches?
In order to support the creation of APH approaches, we have to understand which challenges arise with them. Because case studies and experience reports are limited to their case, an overview of the challenges is missing. Therefore, we want to combine the insights of the different publications with this research question.

3.2 | Research method

For planning and execution of our systematic mapping study, we follow the proposed procedure for systematic mapping studies by Petersen et al.[32] We performed a database search and extended the procedure of Petersen et al[32] by conducting a snowball process to reach more relevant publications.

3.2.1 | Definition of the search string

For conducting the database search, the research area of the systematic mapping study has to be defined by using keywords. In our systematic mapping study, we investigate publications that are concerned with the combination of agile and plan-based approaches. The combination of agile

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

and plan-based approaches is called *hybrid development approaches* in literature.[8,26] Because we concentrate our work on software development and to keep the phrase as generic as possible, we chose *hybrid software development* as one key-phrase. However, a lot of authors just reference to the combination of agile and plan-based methods, when they describe hybrid approaches. Therefore, in order to reach further publications, we added the terms *agile*, *plan-based* and *combine* to our search string. In many publications, the term *traditional* is used as a synonym for *plan-based*; hence, we added also this term.[8,33,34] To introduce agile methods and to form APH approaches, companies often *integrate* agile methods into a preexistent plan-based approach; hence, we added this term to our search string.[35,36] In their work on APH approaches, Boehm and Turner[3] stated that companies need to *balance* agile and plan-based approaches to be successful and minted this term in this context. Therefore, we also added this term as a synonym to our search string.

Boehm and Turner[3] also discuss that the size of the project team and the criticality (loss due to the impact of defects) are important influencing factors for creating APH approaches. A large project team and a high level of criticality tend to lead to the use of APH approaches. To reach further publications that describe the reasons behind APH approaches, we added the terms *large-scale* and *safety critical* combined with *agile* to our search string. The keywords were aggragated to form the following search string:

> *"hybrid software development" OR (agile AND (("traditional software development" OR plan-based) AND (integrate OR combine OR balance)) OR large-scale OR "safety critical"))*

3.2.2 | Definition of inclusion and exclusion criteria

For increased objectivity during the filtering and selection process, we defined inclusion and exclusion criteria. The specific criteria are shown in Table 2. In our SMS, we want to examine the goals and reasons behind the use of APH approaches and how these goals are reached. Therefore, we only included those papers that describe the intentional use of an APH approach. Often APH approaches are used by companies during the transformation from a plan-based process to an agile one to better deal with the introduction of agile practices and to not lose productivity.[37] However, these approaches are not intentional and are only there for a short period. Therefore, they do not describe the normal reasons that lead to the use of an APH approach. Additionally, agile practices are sometimes used during the development phase of a project in an otherwise strict waterfall process.[38] In this way, the principles of plan-based approaches are completely implemented without a compromise with the agile ones and the project does not benefit from both agile and plan-based approaches. Our research about the structure of APH approaches shows that their success comes from the interweaving of agile and plan-based approaches.[24] According to Heeager,[19] we consider these approaches as *compatible approaches* where agile and plan-based methods are only made compatible but are not combined. As a result, we excluded these approaches from our SMS.

Since we want to form a baseline of the scientific knowledge of the goals and challenges of APH approaches, we focus our SMS on publications that bear relations to the industry, companies, or projects by a case study, experience report, interview study, problem identification, or similar. We also include publications that performed literature reviews to gather knowledge about these aspects. To answer RQ1 and RQ2, we include papers that mention pursued goals in APH approaches and describe concrete means to implement these goals. In order to get papers that only answer RQ1 but not RQ2, we also include papers that only describe the goals in APH approaches without providing means. To answer RQ3, we include papers that describe challenges and success factors for APH approaches.

TABLE 2 Inclusion and exclusion criteria

	Criteria	Description
Inclusion	IC_1	The paper describes goals that are pursued with an APH development approach and means to reach those goals.
	IC_2	The paper describes goals that are pursued with an APH approach.
	IC_3	The paper describes challenges when using an APH approach.
	IC_4	The paper describes success factors or practices for balancing agile and plan-based aspects in APH approaches.
Exclusion	EC_1	The paper describes a transition or transformation towards agile software development, respectively a not intentional APH approach.
	EC_2	The paper treats agility as an island in an otherwise strict waterfall approach.
	EC_3	The paper has no reference to the industry, companies or projects.
	EC_4	The paper is neither written in German nor in English.
	EC_5	The paper is not a peer-reviewed contribution to a conference or journal.

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3.2.3 | Selection of databases

We selected five databases for our systematic mapping study: Google Scholar, IEEE, ACM, Science Direct, and SpringerLink. In all these databases, we conducted a thorough search. The databases have partially different search options. In order to fit the options, we adapted our search string accordingly.

3.2.4 | Execution

Our selection process is shown in Figure 2. The main part of our SMS was conducted in May and June 2020. In April 2021, we checked the databases for additionally published papers. We formed a start set followed by two iterations of snowballing. In the following, we describe the selection process in detail.

For the formation of our start set, we conducted an automated search with our search string in the selected databases. In the first step, we applied our inclusion and exclusion criteria to the titles of the publications and identified 3378 papers. After excluding all duplicates, 275 papers remained. In the next step, we read the abstracts and excluded 104 papers based on their abstracts and keywords. Accordingly, 171 papers remained for the content analysis. In the last step, we read the content of the papers thoroughly and excluded 139 papers. After this step, 32 papers remained for our start set.

In the first iteration of our SMS, we performed for each paper of our start set a forward and backward search. According to Wohlin et al,[39] we considered for the backward search the references of each paper and for the forward search all papers in which the respective paper is at least once cited. In this process, we identified 200 papers. After excluding all duplicates, 68 papers remained. First, we also analyzed here the abstracts

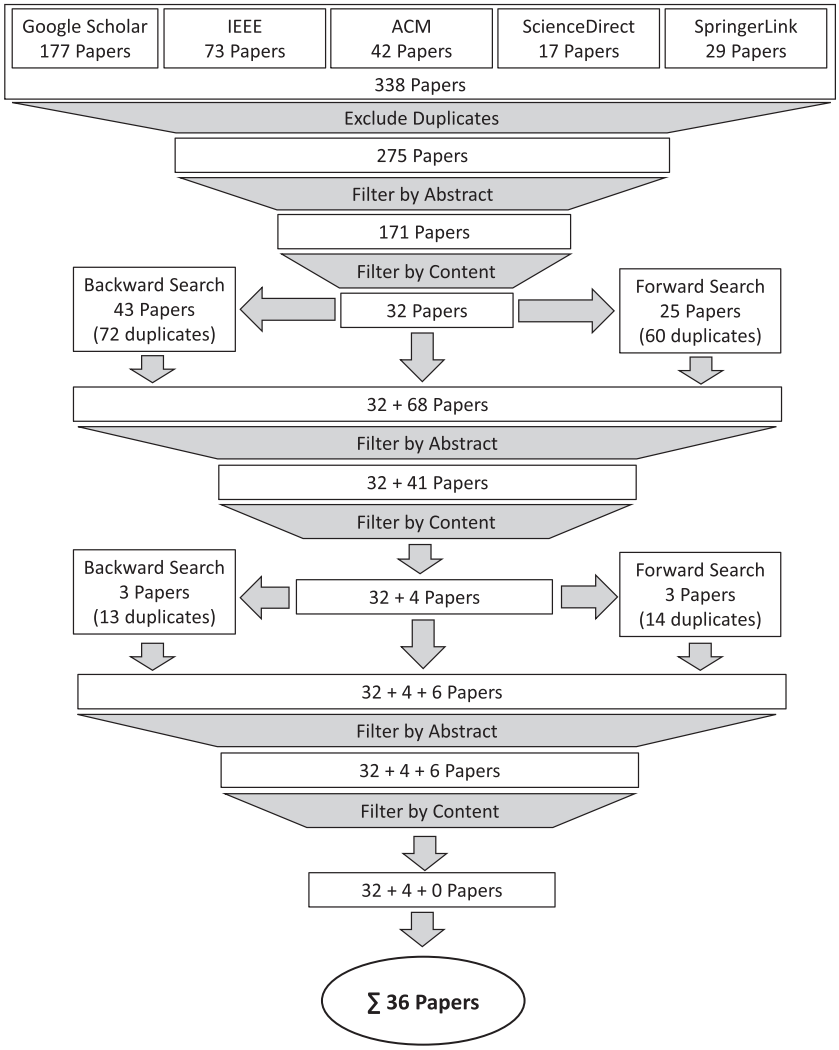

FIGURE 2 Representation of the search and filtering process after Klünder et al.[7]

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

and keywords of the papers. During this process, we excluded 27 papers. After this, we read again thoroughly the content of the remaining 41 papers. Here, we excluded 37 papers and four papers remained from our first snowballing process.

On the remaining four papers, we performed a second iteration of snowballing. During the second iteration, we identified 33 papers. After we excluded all duplicates, six papers remained. Then we read again the abstracts and keywords of all six papers but did not exclude any papers in this process. In the last step, we again read thoroughly the content of the remaining papers and excluded all six papers. After the whole selection process of our SMS, we identified a total of 36 papers.

3.3 | Data extraction and analysis

For the data extraction of the relevant informations from the publications, we created an extraction sheet shown in Table 3. Each field in the extraction sheet has a name and value. Additionally, as far as possible, we mapped the fields of the extraction sheet to the corresponding research questions. After the data extraction from the publications, we used thematic analysis to analyze the data.[40] For that, we first assigned initial codes to the extracted text passages that answer our research questions. In a second step, we formed from the codes themes by constant comparison of the initial codes.

3.3.1 | Threats to validity

Our work is subject to some threats to validity. In this section, we discuss, according to Maxwell[41] and Petersen et al,[31] descriptive, theoretical, and interpretive validity and generalizability,

Descriptive validity

Descriptive validity considers the correctness of the made observations. In order to mitigate this threat, we formulated a concrete extraction sheet to facilitate the data collection. The extraction sheet enables to trace the data back to its source. In order to mitigate possible bias in the extraction of the data, the first and the second authors participated both in this activity. To ensure a consensus between both, the extraction of information from single papers were discussed.

Theoretical validity

Theoretical validity considers the extent to which we were able to collect the relevant aspects. Our goal is to understand the goals and challenges in APH approaches better. For that, we wanted first to get an overview of the scientific knowledge about these aspects. Scientific publications consider APH approaches often from a theoretical point of view. However, these publications do not report about the actual goals behind APH approaches. In order to mitigate this threat, we excluded papers from our SMS that have no connection to the industry, companies, or projects.

In order to mitigate the threat of missed publications, we formulated a search string that spans broadly the research area of APH approaches. To adapt to the research area, we further added synonyms for the different aspects of our search string, for example, *traditional* or *balance*, that are used in the research about APH approaches and discussed the search string with our colleagues. We also added two environments, namely, *large-scale* and *safety-critical*, in the area of software engineering that are known for the use of APH approaches in order to reach further

TABLE 3 Data extraction sheet

	Name	Value	RQ
General	Study ID	Integer	
	Title	Name of publication	
	Author Name	List of Names	
	Publication Year	Calendar Year	
Content	Agile Goals	List of agile goals that are pursued	RQ1
	Agile Means	List of tuples of agile goals and implemented agile means	RQ2
	Plan-based Goals	List of plan-based goals that are pursued	RQ1
	Plan-based Means	List of tuples of plan-based goals and implemented plan-based means	RQ2
	Challenges	List of mentioned challenges and problems	RQ3
	Success Factors	List of mentioned success factors	RQ3
	Research Method	What research method is applied	

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

publications. We also searched multiple databases and performed a snowball process with two iterations until no new papers emerged. This process led to four additional papers.

The SMS was executed by the first and second authors to reduce the mono researcher bias. However, the inclusion and exclusion of papers is a subjective decision, and there is a threat to wrongly include or exclude a paper. To mitigate this threat, we formulated concrete inclusion and exclusion criteria to increase the objectivity of the selection process. Uncertainties regarding the inclusion and exclusion of certain papers were discussed among the authors. The subsequent data extraction was mainly performed by the first author, but the second author reviewed the process.

Interpretative validity

The interpretative validity considers the conclusions drawn from the data. Most found results were mentioned multiple times in different papers. We conclude that the papers support each other and that the results of our study are valid. Therefore, we consider this as a minor threat and consider the found papers sufficient to answer our research questions.

Generalizability

We cannot guaranty that the goals we discovered represent all goals that are pursued in APH approaches by companies. There may exist companies that have other or undiscovered goals. The same applies to the found means and challenges. Further research is needed to confirm the generalizability of the results.

4 | RESULTS

Table 4 lists all publications we selected during the SMS. Most of the publications are case studies or design science approaches where demands from industries are addressed. This supports our statement that research about the challenges and goals in APH approaches is mainly limited to specific cases and there is no general overview of these aspects. There are also no attempts to combine the insight of these case studies. Further, there is little research that compares different APH approaches and tries to generate insight across different cases. Therefore, we are among the first that combine and analyze the insights of different cases studies in the field of APH approaches.

> Finding 1: Research about the goals and challenges in APH approaches is limited to case studies but a combination and overview of the insights is missing.

In the following sections, we present the results from our systematic mapping study according to our research questions.

4.1 | Research questions 1 and 2

Because RQ1 and RQ2 depend on each other, we answer both research questions combined. The overall goal behind the use of APH approaches is to combine the benefits of plan-based and agile approaches and eliminate their weaknesses at the same time. However, this is more or less common sense among practitioners and researchers. In order to show the detailed reasons, we present in the following why companies need certain plan-based and agile concepts at the same time and what they do to achieve this.

4.1.1 | Agile goals and means in APH approaches

In this section, we present the findings for the reasons behind the use of agility and how the goals are reached by the use of agile practices and methods. Table 5 summarizes all agile goals in the first column. The goals are sorted in descending order based on the numbers of papers which mention that specific goal. The papers that mention this goal are documented in the second column. In the third column, as far as described by the papers, we document how these goals are addressed (and the references to the respective papers in the fourth column). Some paper only described a goal without a corresponding means. In these cases, further research is needed to examine how companies address these goals.

The *adaptivity to changes* is the most often mentioned reason for agile methods in APH approaches. This means to be adaptable to requirement changes or changes in the technology choices. One means to reach this goal is to have an *iterative development*.[48,52] Iterations enable

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 List of selected publications

Title	Author	Year	Research Design	Reference
Adaptive Software Development for Developing Safety Critical Software	Abdelaziz et al.	2015	Design science	42
Hybrid Project Management: Agile with Discipline	Adelakun et a.	2017	Case study	43
Managing Requirements Volatility while 'Scrumming' within the V-Model	Anitha et al.	2013	Case study	44
Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry	Batra and Xia	2010	Case study	45
Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings	Bick et al.	2018	Case study	46
The project management cocktail model: An approach for balancing agile and ISO 21500	Binder et al.	2014	Design science	47
How Extreme does Extreme Programming Have to be? Adapting XP Practices to Large-scale Projects	Cao et al.	2004	Experience report	23
An Iterative Approach for Development of Safety-Critical Software and Safety Arguments	Ge et al.	2010	Case study	48
A Hybrid Model for IT Project with Scrum	Hayata and Han	2011	Case study	49
Agile Software Development and its Compatibility with a Document-Driven Approach? A Case Study	Heeager and Nielsen	2009	Case study	50
Introducing Agile Practices in a Documentation-Driven Software Development Practice: A Case Study	Heeager	2014	Case study	51
Meshing Agile and Plan-Driven Development in Safety-Critical Software: A Case Study	Heeager and Nielsen	2020	Case study	52
Systematic Piloting of Agile Methods in the Large: Two Cases in Embedded Systems Development	Heidenberg et al.	2010	Case study	53
Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson	Heikkilä et al.	2017	Case study	54
Does a Hybrid Approach of Agile and Plan-Driven Methods Work Better for IT System Development Projects?	Imani et al.	2017	Case study	55
Understanding Agile Software Development in Practice	Kautz and Madsen	2010	Case study	56
Coexisting Plan-driven and Agile Methods: How Tensions Emerge and Are Resolved	Laux and Kranz	2019	Case study	57
Adopting Agile Practices when Developing Medical Device Software	McHugh et al.	2015	Case study	4
Traditionalisation of Agile Processes: Architectural Aspects	Matkovic et al.	2018	SLR	58
Challenges of Aligning Requirements Engineering and System Testing in Large-Scale Agile: A Multiple Case Study	Gomes De Oliveira Neto et al.	2017	Case study	59
"Rafting the Agile Waterfall" – Value based Conflicts of agile Software Development	Pechau	2011	Case study	60
Scrumconix: Agile and documented method to AGSD	Portela and Borrego	2016	Design science	61
Agile software development methodology for medium and large projects	Qureshi	2012	Case study	62
Ambidexterity in Agile Distributed Development: An Empirical Investigation	Ramesh et al.	2012	Case study	63
The Many Lives of an Agile Story: Design Processes, Design Products, and Understandings in a Large-Scale Agile Development Project	Read and Briggs	2008	Design science	64
Towards Agile Engineering of High-Integrity Systems	Paige et al.	2008	Design science	65
The role of the IT-Project Manager in Organizations that Balance Agile and Traditional Software Development	Tjørnehøj	2018	SLR	66
Investigating the Role of Architects in Scaling Agile Frameworks	Uludag et al.	2017	SLR	67
When agile meets the enterprise	Waardenburg and Vliet	2013	Case study	68
How Much Up-Front? A Grounded Theory of Agile Architecture	Waterman et al.	2015	Case study	69
Coordination In Large Agile Projects	Xu	2009	Case study	70
The Effect of Complexity and Value on Architecture Planning in Agile Software Development	Waterman et al.	2013	Interview study	71

(Continues)

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 (Continued)

Title	Author	Year	Research Design	Reference
Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation	Dingsøyr et al.	2018	Case study	72
Exploring Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective	Scheerer and Kude	2014	Case study	73
Agile Development in a Medical Device Company	Rottier and Rodrigues	2008	Experience report	74
A Case Study of Agile Software Development for Safety-Critical Systems Projects	Islam and Storer	2020	Case study	75

TABLE 5 Agile goals and means in APH approaches

Goals/reasons	Papers	Means	Papers
Adaptability to change (requirements, technology etc.)	4,23,43,44,48,50-54,61-65,67,70,75	Iterative Development (Scrum Framework)	48,52,75
		Integrated and collocated team	63
		Decentralized decisions	70
Dealing with uncertain requirements	45,47,51,52,55,62,68,71	N/A	N/A
Fast development and short time-to-market	42,43,49,51,57,63,75	Iterative Development (Scrum Framework or XP)	49,51,75
		Autonomous teams	57
Improvement of communication	23,53,75	Scrum Framework	75
Improvement of the visibility of the project's status	53,61	Scrum Framework	61
Increasing of productivity	47,61	Scrum Framework	61
Decreasing of costs	42,51	N/A	N/A
Business Alignment / Increasing of customer satisfaction	43,44	Customer collaboration	44
Increasing developers' motivation	54,75	Scrum Framework	75
Adaptability of plans	45	Fast decision making	45
Improving predictability of resources	53	Continuous planning	53
Creating early value	68	N/A	N/A
Early discovery of problems	75	Scrum Framework	75

companies to get constant feedback from the customer and stakeholders and plan for new or changing requirements. Especially the Scrum framework supports this way of working with a dedicated product owner, who constantly manages the backlog and the review and planning meetings in each Sprint. The use of an *integrated and collocated team* can support and fasten the integration of changes during the development process since all team members can collaborate closely together.[63] The fast integration of changes is a critical factor for the success of agile development. Therefore, *decentralized decision making* is a way to accelerate this process, since change requests do not need to first climb up and down the hierarchical ladder.[70] The second most-often mentioned goal is the *dealing with uncertain requirements*. At the beginning of a project, the customer and stakeholders often do not know what they really need and how the end product should look like.[45] Here, we did not identify a specific means to address this goal other than a general use of agile methods.

The next goal that is mentioned is *fast development and short time-to-market*. To reach this goal, again the use of *iterative development*, especially the Scrum framework or XP are mentioned.[49,51] These methods enable companies to produce early value and provide a lean process that helps to fasten the development process. *Autonomous teams* can further fasten the development process because decisions and agreements can be made without the slowing consultation of upper management.[57] The next three goals are the *improvement of communication*,[23,53,75] *improvement of the visibility of the project*,[53,61] and *increasing of productivity*.[47,61] All three goals are addressed by the implementation of the *Scrum framework*. The next goals is the *decreasing of cost*.[42,51]

An improved *customers' satisfaction and business alignment* are also goals that are mentioned.[43,44] Therefore, the agile methods emphasize *close collaboration with the customer* and stakeholder over the whole project duration.[44] The next mentioned goal is *increasing of developers' motivation*.[54,75] The goal is also addressed by the *Scrum framework*.[75] A further goals is the *adaptability of plans*.[45] In order to reach this goal, *fast decisions* are mentioned as a means.[45] Companies want to *improve the predictability of resources* by the use of agile methods.[53] More concrete, the use of *continuous planning* was mentioned to reach this goal. The next goal is that companies want to *create early value* for their customers.[68] The last mentioned goal is to *discover problems early*.[75] Here again, the *Scrum framework* is mentioned to reach this goal.[75]

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Finding 2: The most important goals for the use of agile methods in APH approaches are the adaptability to change, dealing with uncertain requirements, and fast development and short time-to-market.

4.1.2 | Plan-based goals and means in APH approaches

In this section, we present the goals behind the use of plan-based concepts and how companies reach these goals by the use of plan-based methods. In Table 6, the goals and the means are represented. The table is structured in the same way as for the agile goals.

The most often goal that is mentioned is the *meeting of safety, security, and regulatory requirements*. The means that is most often described to meet this goal is an *up-front requirements analysis*.[44,50-52] Also, an *up-front architecture design* is used as a means.[47,48] Further, the use of the WIM, we described in Section 2, is mentioned to meet these requirements.[42,50] Similar to the description of more effort up-front is also the use of a

TABLE 6 Plan-based goals and means in APH approaches

Goals/reasons	Papers	Means	Papers
Meeting of safety, security and regulatory requirements	42,44,47,48,504,51,52,65,75	Up-front requirements analysis	44,50-52,75
		Up-front architecture design	47,48
		Using Waterfall-Iterations-Approach	42,50
		Testing phase at the end of the project	42
		Up-front hazard analysis	52
		Using V-Model	4
		Using Pipeline-Approach	65
Managing and coordination of large scale development	43,45,53-55,62,70,72,73	Central team for coordination and planning	45,70,73
		Up-front architecture design (high level)	55,72
		Long-term planning	45,62
		Pipeline-Approach	53,54
		Up-front requirements analysis	62
		Contracts	62
		Up-front scope definition	62
		Risk analysis	62
		Formal communication and control	62
Need for planning reliability and fixed resources	42,43,46,54,60,75	Emphasis on planing and control	46
		Up-front defining of constraints	43
		Up-front requirements analysis	75
Maintaining of big picture and dependencies	47,58,61,67,75	Up-front architecture design	47,58,67
		Using Waterfall-Iterations-Approach for architecture	61
		Up-front requirements analysis	75
Dealing with distributed development	61,63	Up-front requirement analysis (visibility of project)	61,63
		Up-front architecture design	63
		Formal communication (gate keeper)	63
Clarification of project goal and deliverables	45,48	Up-front requirements analysis	45,48
		Up-front planning	45,48
Controlling of changes	50,62	Formal change control	50,62
Decrease of costs for development	69,75	Up-front architecture design	69,75
Dealing with complex development	43,46	Using Pipeline-Approach	46
Traceability	42	N/A	N/A
Improving of risk management	61	Waterfall-Iterations-Approach for risk management	61
Testing of embedded / integrated software	65	Testing phase at the end of the project	65

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

testing phase at the end of the project described.[42] Also, a *up-front hazard analysis* is described to support the meeting of safety requirements.[52] Further, the use of the V-Mode[4] and the PM[65] is mentioned.

The second most often mentioned goal for the use of plan-based methods is the *management and coordination of large scale development*. The term large-scale describes here that more than one or two teams are involved in one project. To coordinate and manage these multiple teams, different plan-based means are mentioned. The most often mentioned means in this context is the implementation of a *central team for coordination and planning*.[45,70,73] Further means are an *up-front architecture design*[55,72] and *long-term planning*.[45,62] Also, the PM that we presented in Section 2 is used to better handle development at scale.[53,54] Further, the use of an *up-front requirements analysis* and of *contracts*[62] is mentioned. Also, the *definition of the scope of the project up-front* is used to manage multiple teams.[62] Further, *risk management* and *formal communication and control* are mentioned.[62]

Another reason for plan-based concepts in APH approaches is the *need for planning reliability and fixed resources*.[42,43,46,54,60] To reach this goal, companies emphasize more on *planning and control*,[46] *define constraints for the project up-front*,[43] and use more *up-front requirements analysis*.[75] The next mentioned goal is *maintaining of the big picture and dependencies in projects*.[47,58,61,67] Two mentioned means to reach this goal are based on a higher architectural effort. It is described that an *up-front architecture design*[47,58,67] can support this goal or the use of the WIM with an emphasis on the design phase.[61] Also more *up-front requirements analysis* can support this goal.[75]

Not only the management of large scale projects lead the companies to the use of plan-based methods but also the *dealing with distributed development*.[61,63] The use of an *up-front requirements analysis*[61,63] and an *up-front architecture design*[63] are mentioned to increase the visibility of the project. Also in this context, more *formal communication* methods are used.[63] More concrete, the role of a *gate keeper* is described that coordinates the communication between two different development sites.[63] *Clarification of the project goal and deliverables* are a further reason to use plan-based methods.[45,48] Here, an *up-front requirements analysis* and an *up-front planning* are used to reach this goal.[45,48] The *controlling of changes* is a further goal that is met by a *formal change control*.[50,62] The constant investment into architecture costs time and therefore money. In order to reduce this effort, an *up-front architecture design* is used to *decrease the development costs*.[69,75] Also the *dealing with a complex development* is met by more plan-based way.[43,46] Here, the use of the PM is mentioned to cope with complexity.[46]

The need for *traceability* is also among the reasons for the use of plan-based methods.[42] In contrast to the other, no concrete means are described to reach this goal. Also to have a *improved risk management*, the use of the WIM is described for a constant risk analysis.[61] The last goal is the *testing of embedded or integrated software*.[65] For this goal, a testing phase at the end of the project is mentioned.[65]

> Finding 3: The most important goals for the use of plan-based methods in APH approaches is the meeting of safety, security, and regulatory requirements, the management and coordination of large scale development, and the need for planning reliability and fixed resources.

4.1.3 | Combinations of goals

In order to investigate which goals are often pursued in combination, we analyzed the data presented in Tables 5 and 6 regarding frequent item sets by using the Apriori algorithm. The results of this analysis are shown in Figure 3. We chose for the generation of the frequent item sets a minimum support of three. The figure reads from the right to the left and represents the found tuples of goals. The numbers show how often we discovered each tuple. For example, we found the combination of adaptability to change, meeting safety, security and regulatory requirements and dealing with uncertain requirements two times, whereas only the combination of adaptability to change and meeting of safety, security and regulatory requirements eight times. The goal of adaptability to change was in total mentioned 18 times. The presented combinations of three goals do not have the minimum support of three, although all contained subtuples of two goals satisfy the minimum support of three. Therefore, we decided to include the combinations of three goals in the figure.

Regarding the results, agile goals are often mentioned in combination, whereas we found no frequent item set where plan-based goals are mentioned in combination. That indicates that companies have often one central goal that they pursue with a plan-based approach introduce agility to pursue further goals. The adaptability to change is the goal that is the most combined with other goals. Also companies that have to follow safety, security, and regulatory requirements have the need to react to changes during development or to deal with uncertain requirements. The adaptability to change is also often mentioned in the combination with other agile goals, like a fast development or the improvement of communication.

After analyzing the data regarding frequent item sets, we analyzed how the found combinations were addressed by the publications. For this analysis, we only include the publications that address all goals of the analyzed goal combination. The results from this analysis are presented in

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

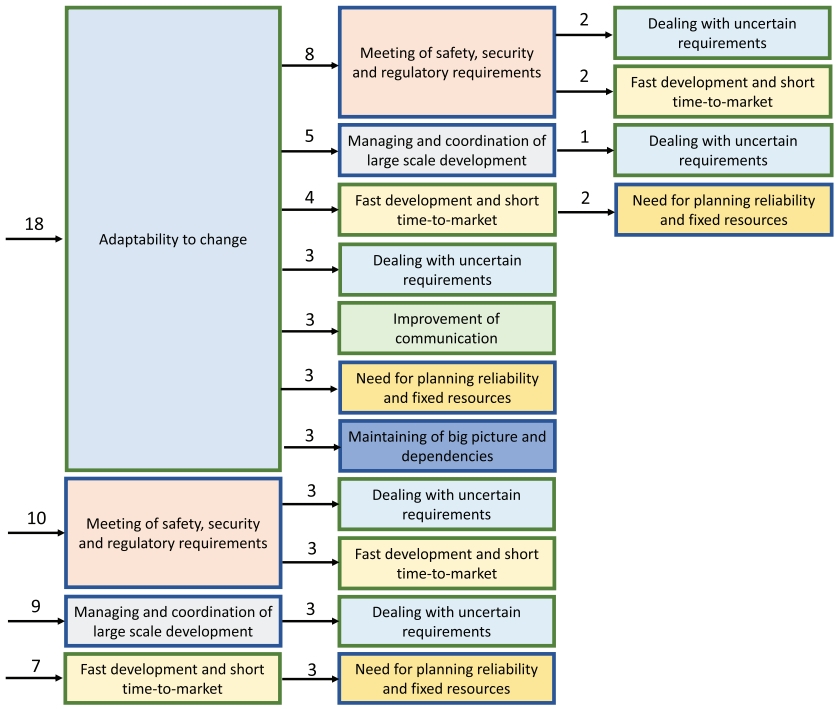

FIGURE 3 Results of the analysis for frequent item sets among the mentioned goals (minimum support = 3). The green framed rectangles represent agile goals whereas the blue framed rectangles represent plan-based goals

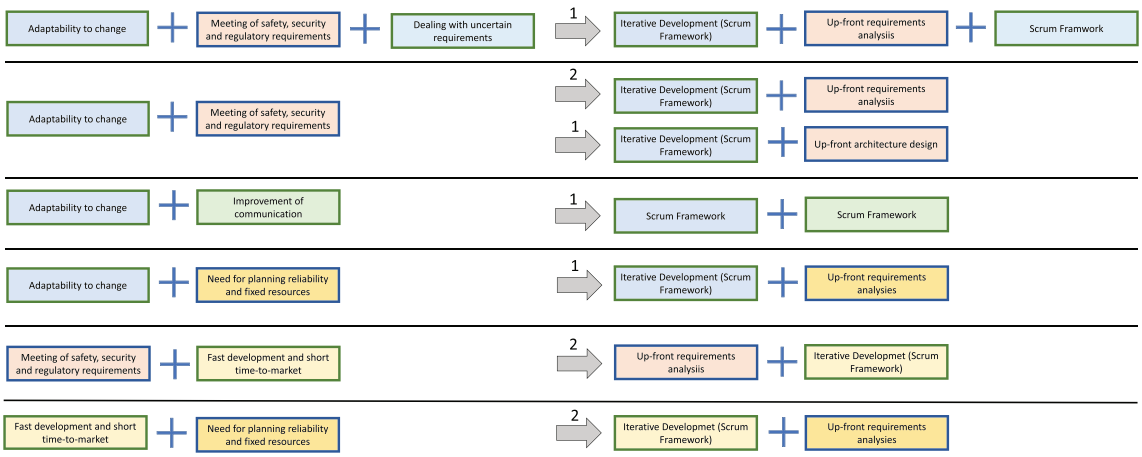

FIGURE 4 Analysis regarding how the frequent item sets of method combination are addressed. The combined goals are shown in the left and how the goals are addressed is shown in the right. The goal and the respective means have the same color for easier identification. The number over the gray arrow shows how many publications we found that describe the respective goal combination and how all these goals were addressed

Figure 4. The results show that the Scrum framework is used as a base method because the adaptability to change is a dominant goal. Other practices like an up-front requirements analysis or an up-front architecture design are added to the development process if needed. This finding is also supported by the results of Tell et al[9] that are based on the data of the HELENA study.

Finding 4: The Scrum framework is used as a base method for APH approaches and other agile or plan-based methods and practices are added if needed.

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 7 Challenges in APH development approaches

Challenge	Papers
Management of requirement changes	44,50,52,59,68,75
Conflict between up-front and continuous requirements engineering	51,52,54,63,68
Conflict between up-front and continuous architecture and design	42,48,67,71,72
Conflict between central decision making and self-organized teams	46,57,63,66,68
Difficulties by a separated testing phase	49,52,73-75
Tension between the business and the development unit	51,57,68
Conflict between long- and short-term planning	52,57,75
Conflict between explicit documentation and tacit knowledge	50,68,75

4.2 | Research question 3

In this section, we describe the challenges found in the context of APH development approaches. In Table 7, we listed the found challenges and sorted them by the number of mentions.

4.2.1 | Management of requirement changes

This is the most often mentioned challenge. Requirement changes have to be incorporated and often the scope of the project has to be adapted.[44] These aspects have to be properly managed. It is important to react fast to requirement changes in order to not lose momentum.[59] The difficulty is that often a high effort is needed to change high-level requirements or specifications.[50,52,59] Also, the analysis of the impact of changes requires time and effort.[59,75] In APH approaches, different requirement documents often exist in parallel for the different levels of requirements. In cases of requirement changes, these different documents have to be changed across all requirements level in order to keep them synchronized.[59]

4.2.2 | Conflict between up-front and continuous requirements engineering

Projects need often both up-front and continuous requirements engineering. The challenge here is to decide how much effort should be invested for the up-front requirements analysis and the level of detail in which requirements are elicited.[52] Too little effort, in the beginning, can lead to requirements on a too high level that are not really understood and not sufficient enough.[52] Also, standards and regulations demand often a sufficient requirements analysis phase.[4] On the other hand, requirements are often uncertain at the beginning of a project and emerge only during the runtime of the project.[51] Therefore, too much effort up-front, when the requirements are uncertain, hinders a successful project because requirements may be elicited which are wrong. In these cases, the project should put more emphasis on continuous requirements engineering.

At the beginning of a new project, the different aspects that influence the decision about a more up-front or emergent requirements engineering have to be weighed against each other. The most often mentioned aspect that demands continuous requirements engineering is when a lot of requirements changes are anticipated.[51,52,54] Also when the requirements are not certain and unclear, a more continuous requirements engineering is recommended.[51,68] A more continuous requirements engineering is used to reach a better business alignment and closer customer collaboration.[44] The use of up-front requirements engineering delays the development and therefore the first deployment of the product. In cases where early value and rapid results are needed, a more emergent requirements engineering is recommended.[68]

However, there are also a lot of aspects that demand more up-front requirements engineering. One is the coordination of different teams or the coordination of different locations in a distributed development.[62,63] The up-front analysis of requirements increases the visibility of the requirements and therefore for the whole project and improves the planning and managing of a project.[61] Next to that, a lot of regulations and standards demand a properly up-front analysis of the requirements.[51,52] Therefore, a more up-front requirements engineering should be used. Also to understand the wishes of the customer and stakeholders better and to clarify requirements, more up-front requirement analysis can help.[45,48] This also supports the risk analysis and management for the project.[42,50]

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.2.3 | Conflict between up-front and continuous architecture and design

The risk of an incremental or emergent architecture is that it may grow in an accidental way that may do not represent the best solution.[71] However, too much emphasis on architecture up-front increases the risk of costly rework due to uncertain requirements.[55,62,71] Therefore, often a simultaneous use of up-front and continuous architecture is needed. The challenge is to decide how much emphasis is needed on either of them.

The biggest influence on the balance between the up-front and emergent architecture is the complexity of the project.[71] The use of existing frameworks significantly decreases the complexity of an application because they can be used as prepared modules. Hence, less up-front effort is needed to define the architecture and a more emergent architecture and agile way of working can be used.[58,69,71] However, if no prepared frameworks can be used and the application has to be built from scratch, the complexity is increased and a more up-front architecture design is needed.[71] Also, legacy systems can be a source of complexity and require more up-front investment in architecture.[71] The use of more up-front architecture design is not just caused by the complexity of the build application but also by coordination reasons. An up-front architecture design can help to coordinate different teams and different development locations.[55,63,72] The up-front architecture design helps to keep the big picture and overview of the project and therefore to plan and manage dependencies.[47,58,67] Also, a lot of standards and regulations require the usage of an up-front architecture design.[47,48] An up-front architecture design also supports the risk analysis for the project.[43,50] However, it also delays the first deployment of the product. If rapid results are needed, a more emergent architecture is required.[68,71]

4.2.4 | Conflict between central decision making and self-organized teams

Central decision making is accompanied by a vertical communication where requests are transported upwards the hierarchical ladder and decisions are transported downwards back to the team.[70] On the other hand, mutual adjustment is the practice to enable self-organized teams because here problems and decisions are discussed among the team members.[70] The tension between central decision making and self-organized teams comes from the need to give the teams decision power on the one hand but properly coordinate inter-team relations at the same time on the other hand.[57] When decisions have to first climb up and down the hierarchical ladder, much time has passed before they can be implemented. In self-organized teams, decisions are made on the team level and because of this shorter way, decisions and the whole development process are faster.[57] However, mutual adjustment has its limitations when too many people are involved and mutual adjustment becomes chaotic.[63] This is especially important when it comes to the management of change requests.[50,62]

The challenge here is to balance both concepts of mutual adjustment and central decision making and using them at the same time. Teams should solve problems by mutual adjustment as far as possible. However, for that, the single team often lacks the complete overview of the project in order to exactly know who to address with certain problems.[46] Often also too many teams are involved for a successful mutual adjustment.[46] In these cases, coordination should be handed to the central decision instance so they can resolve issues in a nonchaotic way.[70,73] In order to properly implement this management system, a two-way feedback channel between the teams and central decision instance has to be established.[46]

4.2.5 | Difficulties by a separated testing phase

In APH approaches, testing is often separated from the development either by a testing phase at the end of the project or for example by the use of the PM.[24] A challenge in APH approaches is to handle the errors that are found during the separated testing phase.[49,52,73] Errors that are found, for example, during the testing phase of the PM interrupt the current development iteration.[73] A separated testing phase often also means that a separated teams is responsible for formulating and conducting the tests. This brings an additional challenge because in this way the testers have less insight in the development of the product and it is harder for them to understand the product and formulate tests for it.[75] Therefore, it is beneficial to find and fix errors directly during the development.[50]

4.2.6 | Tension between the business and the development unit

In APH approaches, next to the development department, there is often an additional business unit that is responsible for maintaining the requirements and handling of the stakeholders.[24] The challenge occurs by the segregation of these two units. The business unit wants to have fixed dates where certain features are ready in order to advertise these features and plan the creation of the product. However, with the emphasis of a negotiable scope, advertised by agile methods, the business unit feels a loss of control over the development unit.[57] The development unit on the

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

other hand grips over missing feedback for finished features and prioritization from the business unit since the business unit considers their work as done after handing over the requirements.[68] But in an agile setting, business unit and the development department have to work constantly together over the whole runtime of the project, but the necessary shared responsibility is often missing.[57] The additional business unit also leads to less contact between the developers and the customer respectively the users.[51] However this direct contact is considered as a key element of agile development.

4.2.7 | Conflict between long- and short-term planning

Companies require planning reliability and therefore fixed resources, for example, time or funding.[50,75] Therefore, they often use long-term planning. However, projects still operate under uncertainty, for example, availability of staff, and therefore can only use short-term planning.[57] Additionally, the use of milestones hinders the agility of iterations.[52] These aspects lead to tension between short- and long-term planning.

Companies need short-term planning to be adaptable to changes.[45] Because the allocation of resources is much more correct in the short-term, continuous short-term planning can help to improve the planning for a project.[53] Also, continuous planning can increase the visibility of the project because the status of the project is reviewed continuously.[53,61]

4.2.8 | Conflict between explicit documentation and tacit knowledge

Agile methods rely on tacit knowledge to speed up the development process. However, a lot of standards and regulations require concrete documentation.[50] Heeager and Nielsen[50] describe the challenge of combining a document-driven approach with agile development. The challenge here is to decide which information have to be documented and which can be held as tacit knowledge.[50,75]

> Finding 5: The challenges in APH approaches are mostly based on the need to simultaneous use of agile and plan-based methods and practices for the same activity, e.g. requirements engineering or coordination, and arise from the decision about how much emphasis on the agile or plan-based practices is needed.

5 | DISCUSSION

With the increasing detection of the use of APH approaches in research, the question about the motives behind APH approaches raised.[6,8] The question is if companies have legitimate reasons to combine agile and plan-based methods or if they are stuck in a transition toward agility and if APH approaches are even feasible.[19,23,50,71] By the results from our first research question, we have a better understanding of what companies want to reach with APH approaches. Companies are not stuck in an agile transition but use APH approaches intentionally and found that a combination of agile and plan-based methods can help them to reach their goals.

To benefit at the same time from agile and plan-based methods, the respective goals have to be met simultaneously. Companies cannot benefit from agile and plan-based methods at the same time if agile development is only an island in an otherwise strict waterfall approach. In these cases, the waterfall approach is too dominant and the benefits of agile development vanish. The analysis of the goals shows that agile and plan-based methods often have to be applied for the same activity and it is difficult to decide how much emphasis is needed on the agile or plan-based implementation of an activity. For example, to support the handling of uncertain requirements and safety and security requirements, an up-front and iterative requirements engineering have to be used simultaneously. Although this is the solution how companies can benefit from agile and plan-based practices at the same time, the results from our third research question show that this also causes the main challenge in APH approaches. If both ways are applied at the same time, neither one of them can be applied to its full extent and a conflict arises between both. To solve the conflict, a compromise between both has to be found. That means that the creation of APH approaches is more about the analysis of how the activities have to be performed and which principles have to be applied based on the goals than the choice of concrete development methods. Principles describe the rules and assumptions on which a development approach is based. An example of an agile principle is that changing requirements are welcomed, even late in the development. On the other hand, an example of a plan-based principle is that development is predictable as long all requirements are known up-front.

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Finding 6: The creation of APH approaches depends first and foremost on the analysis of the goals, how the activities have to be performed, and which principles have to be applied.

Only in the second step after the analysis of the activities and principles, the proper methods have to be chosen to implement the principles. However, the results of research question two show that the goals are often addressed by individual practices and not by the implementation of whole methods. For example, if there is a need for an up-front requirements engineering, it does not implicate that all the other phases of the waterfall model are needed as well. The methods are there to give structure to the development approach but to address the goals the right practices have to be applied.

Finding 7: Methods are used in APH approaches to provide structure for the development approaches but their practice sets have to be adapted to the goals.

To be successful, the conflicts that arise by the simultaneous use of agile and plan-based practices for the same activity have to be solved. Among the analyzed publications, we discovered a series of successful APH approaches.[23,44,45,49] In these cases, for the activities where agile and plan-based practices were used, compromises were found. Further, the plan-based and agile practices were properly integrated into each other by the implementation of unifying practices that the agile and plan-based practices connect. For example, Scheerer et al[73] describe how agile and plan-based practices can be connected for the coordination activity. To connect autonomous teams and central decision making, they describe a compromise between both. The teams are self-organized unless their decisions interfere with the overall goals or concerns multiple teams. In these cases, central decision making is used. To implement both practices at the same time, a central team was created that was responsible for creating and communicate the direction of development. The product owners were the connection between the development teams and the central team. Therefore, to create a compromise, the practices have to be adapted. Tell et al[9] state that the development of hybrid approaches depends more on how the practices are arranged and connected. Based on our results, we support this statement. Because the practices have to be adapted to solve the conflicts in APH approaches, we further state that it additionally depends on how the practices are performed. Klünder et al[12] tried to find coherences between different context factors and the choice of practices but found only a few dependencies. We state that this result is because they do not take into account how the practices are performed but only if they are applied or not.

Based on our discussion, we state that to form a successful APH approach, the activities where agile and plan-based practices have to be used simultaneously have to be identified. To implement the chosen principles, the suitable methods have to be used, adapted, and combined. As we described, the practices in APH approaches can not be applied to their full extent but a compromise between both has to be found. Therefore, the practices have to be adapted to solve the conflicts. Companies have to be aware of the conflicts in order to create successful APH approaches.

Finding 8: If agile and plan-based practices have to be implemented simultaneously for the same activities, a compromise has to be found between the practices and additionally unifying practices have to be implemented to address the conflicts.

6 | ADDRESSING THE CONFLICTS IN APH APPROACHES

In the publications, we discovered only the existence of the conflicts were mentioned. However, there is not much research regarding how these conflicts can be addressed. As we discussed, companies have to be aware of the conflicts in order to address them. Therefore, we list for each main activity, based on our results, the identified goals and contextual factors that lead to the conflicts in the main activities. Because the aspects are only based on our results, there are additional influence factors that are missing. We plan to discover these factors in future research. The considered activities are requirements engineering, architecture, planning, coordination, documentation, and testing and operations. Further, we discuss, based on our knowledge about APH approaches and information found in literature, how the conflicts can be address if the goals require

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

both an agile and plan-based implementation for the respective activity. Many of our recommendations are based on our previous work about the organization of APH approaches in Prenner et al.[24]

6.1 | Addressing the requirements engineering conflict

Figure 5 shows the influencing factors for the requirements engineering activity. The main question in the requirements engineering activity is when certain requirements are elicited.[76] According to this, in order to create a compromise, a decision has to be made in regards to the number of requirements that should be elicited up-front. Because most goals behind an up-front requirements engineering are concerned with the coordination of the development process, enough requirements should be gathered to create a big picture of the needed product. Since the detailed requirements rather change than the high-level ones, the requirements should not be gathered in too much detail. Here, the analysis of the *possibility of changes* of requirements can help to determine if the gathering of requirements should be further pursued or project should change to an iterative requirements engineering. Requirements that are likely to change should be marked and considered as a risk to keep track of them. If requirements become too uncertain, it is another indicator to switch to an iterative requirements engineering. The areas where requirements are uncertain should also be marked to keep track that these areas have to be further explored during the iterative requirements engineering. If the creation of early value is critical, the amount of up-front requirements engineering has to be carefully examined. However, especially in the area of large-scale development and complex requirements, we recommend getting at least a big picture of the project. Here, it is important to analyze what is considered as the greater demand and the greater risk. This determines whether the development should be delayed or the consequence of a not so clear big picture of the product under development can be taken.

If the project has to follow certain regulations, it has to be analyzed if the regulations really require an up-front requirements engineering or if the regulations also can be met by an iterative requirements engineering approach that uses the waterfall principle, for example, the PM.

After switching to an iterative requirements engineering, the business unit must know that their work is not done and that they have to derive detailed requirements from the high-level requirements and that they have to work constantly together with the developers.[68] Here, it has to be decided how to integrate requirements engineering into the development. This depends on the number of requirements already gathered, the stakeholder group, the requirements' complexity, and the number of development teams. The presence of each of these factors aggravates the software engineering process further. To deal with these aspects, we recommend the often used PM.[61] It enables the coordination of changes and the development team and is scalable from a rather noncomplex requirements engineering to a complex one with many stakeholders. However, the danger of the PM is the formation of silos since the developers and the formed business unit for the PM are often separated. To mitigate this risk, the business unit and development unit should work together as closely as possible and should be co-located. If the developers have to be more integrated into the requirements analysis, for example, for feasibility analyses, the WIM can be used or even combined with the PM. There, an additional requirements engineering phase can be integrated into each iteration.

6.2 | Addressing the architecture conflict

Figure 6 shows the influencing factors for the architecture activity. If the reasons behind an up-front architecture are met, the up-front architecture phase should be stopped and the project should change to an iterative way of working. If the architecture is created in too many details, the integration of changes is difficult.[16]

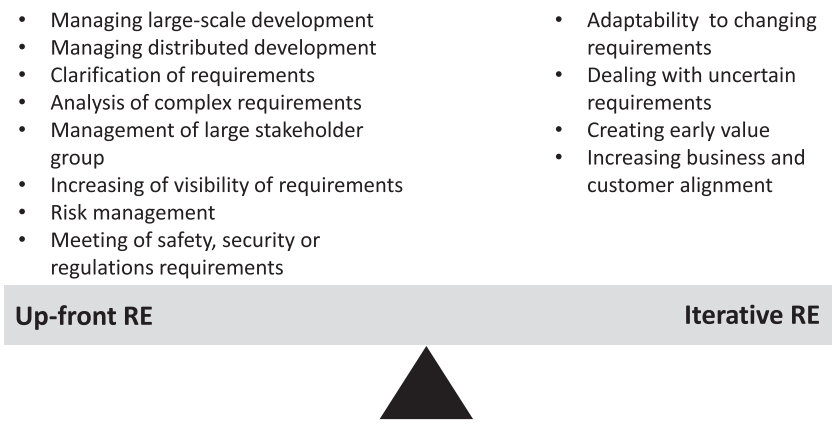

FIGURE 5 Influencing goals on the requirements engineering activity

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

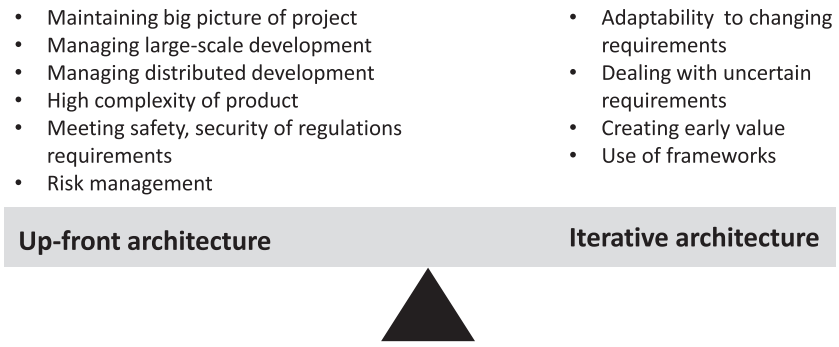

FIGURE 6 Influencing goals on the architecture activity

FIGURE 7 Influencing goals on the planning activity

Here, it has to be analyzed if the architecture is sufficient enough to create a big picture, manage the complexity and address the important security and safety requirements. If an early value is critical, this analysis is important. If standards or regulations have to be followed, it has to be examined if a full up-front architecture is necessary or if also a structured iterative approach like the WIM is sufficient. Following the requirements, the areas of the architecture that are very likely to be afflicted by changes should be marked or the detail level should be decreased. Also, the areas of uncertain requirements should be marked to be aware of potential extensions of the requirements. These areas should be treated as risks.[3]

When the project switches to a continuous architecture, its implementation has to be discussed. If the architecture requires much effort on the inter-team level or is done by an extra team of architects, the PM should be used. Otherwise, if the developers have to be more integrated into building the architecture, the WIM is recommended. If the complexity of the product is not high anymore or mostly frameworks are used, the extra architecture phase in the PM or the WIM can be omitted. The architecture activities can be performed during the development phase.

6.3 | Addressing the planning conflict

Figure 7 shows the influencing factors for the planning activity. The question here is how detailed an up-front plan should be. The plan should facilitate the coordination of different teams. It is legitimate to estimate completion dates with a scope. However, this strongly depends on the requirements and resources. Completion dates can only be estimated with certain requirements and a low changing possibility. The more these aspects do not apply, the more the estimation of completion dates must be treated as uncertain. Also, the needed resources can be estimated up-front but also only as far as they are certain otherwise they also have to be treated as uncertain. Most importantly is to not have the combination of a fixed scope with a fixed deadline because it does not consider the possibility of changes and instability of resources.[16] Instead, the plans have to be constantly examined if the dates and estimations are still valid or have to be changed.

It is important to understand that the work is not done after the initial up-front planning, but planning needs constant attention. Because the overall plan is only on a high level and mostly only as far as reasonable certain estimations can be made, it is important to start early enough the further planning when uncertain estimations become more certain over time. This is especially important in a large-scale setting to align teams in time.

To implement continuous planning, we identified the use of the PM with a central team as the best way. The PM is designed to plan iterations in advance and the length of the iterations can be adapted to how far estimations can be made. The central team is there to supervise the overall plan and communicate adjustments to the teams.

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

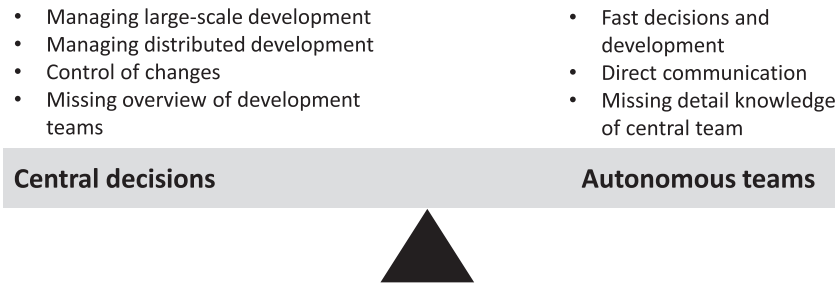

FIGURE 8 Influencing goals on the coordination activity

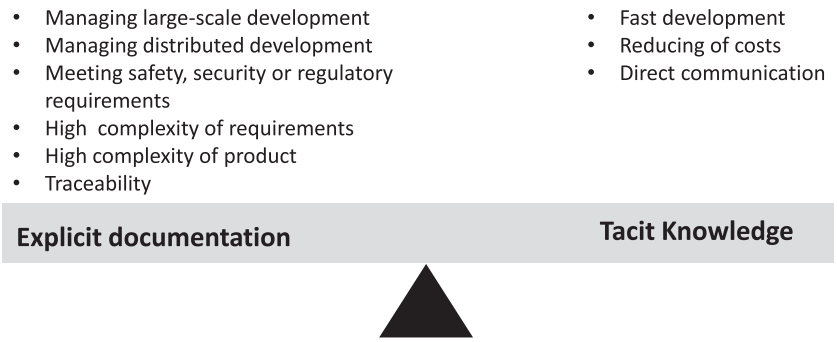

FIGURE 9 Influencing goals on the documentation activity

6.4 | Addressing the coordination conflict

Figure 8 shows the influencing factors for the coordination activity. To find a compromise, it should be determined which decisions can be made at the team level and which have to be made on the interteam level. Xu[70] proposes that the central team should provide directions and support to the teams. The teams should coordinate and make their own decisions at the team level as far as they can. If the coordination cannot be handled anymore at the team level or decisions influence the overall directions or concerns multiple teams, the coordination and decisions should be taken over by the central team. Control is especially important in cases of requirement changes.[77] The danger of only using a single person is that this person can get fast overloaded or delays the process when not available.[78]The danger of a group is that the members first have come together to make a decision. A middle way is to have a small group of people who can also partly make decisions on their own.

6.5 | Addressing the documentation conflict

Figure 9 shows the influencing factors for the documentation activity. Especially in the area of a fast reaction to changing requirements documentation plays a crucial role. To provide enough documentation without losing velocity, the amount of documentation should be minimized to the necessary. The key for a good documentation process is to store each information at a single location, with the result, that information only has to be updated in one place in case of changes. For further support, a minimization of the number of different mediums for saving information is important. This facilitates the search for information and the connection between them.

Because of the nature of hybrid approaches, often a hierarchy of requirements is used to support the combination of up-front and iterative requirements engineering, for example, epics and stories. Unfortunately, this hierarchy aggravates the before mentioned means, because information is often saved twice or even more. A way to generate also here support is to create links between the requirements of higher and lower detail level.[79] Important is to choose tools for the requirements process that support these kinds of links.

6.6 | Addressing testing and operations

Figure 10 shows the influencing factors for the testing and operations activity. In general, we distinguish between separated and integrated testing and operations. Separated testing and operations mean that these activities are performed detached from development in their own phase.

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

- Meeting safety, security or regulatory requirements
- High complexity of product
- Product can only be tested as whole (e.g. embedded development)

- Creating early value
- Minimize impact of errors
- Early error detection

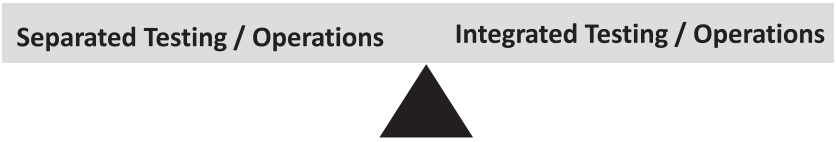

FIGURE 10 Influencing goals on the testing and operations activity

This is implemented by the testing and operations phases in the PM or WIM or with phases at the end of the project. On the other hand, integrated testing and operations mean that these activities are performed in connection with the development and no extra phase is used. To address the conflict in the testing and operations activity, the companies have to analyze which testing activities can be performed during development to minimize the separated testing effort to avoid interruptions and detect errors early.

7 | CONCLUSION AND FUTURE WORK

A lot of companies use APH approaches, but building these approaches from scratch is difficult for the companies. For a long time, it was difficult to properly support these companies in this process because there was a lack of understanding of the companies' goals behind APH approaches. Only with this knowledge, the research community can concentrate its effort to meet the goals of these companies and support them to build successful APH approaches. We addressed this lack of understanding in the context of an industry-focused systematic mapping study. We found out that APH development approaches are mostly used to meet security and safety requirements and manage large-scale or distributed development while remaining adaptable to changes and deal with uncertain requirements at the same time. We also investigated the challenges that come with the use of APH approaches. We identified eight main challenges in the use of APH approaches. These challenges come from the conflicts that emerge when for the same activity agile and plan-based principles are simultaneously used in one project. The difficulty is to pursue the practices that address, respectively, the agile and plan-based goals at the same time.

To create a successful APH approach, a compromise between both approaches has to be found. For that, the companies have to be aware of the conflicts in APH approaches and have to address them. Based on their goals, they have to decide how much emphasis they have to put on each of the agile and plan-based practices and create a balance between both. Additionally, they have to implement practices that connect both agile and plan-based implementation of one activity. We identified in our SMS a general lack of research on how these conflicts in APH approaches can be addressed. To fill this gap, we presented the factors that lead to conflicts in APH approaches and discussed how the conflicts can be addressed. However, more research in this area is needed. The research community has to stop to analyze only the use of methods and practices and consider them only as building bricks but has to investigate how the practices and methods have to be adapted in order to solve the conflicts in APH approaches. Therefore, our next research step is an interview study in different companies to first validate our findings and examine whether there are more goals and challenges to be considered in APH approaches. However, foremost the interview study is designed to investigate what the companies already do to balance the agile and plan-based approaches in their APH approach and identify future research directions.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in the mentioned databases used for the literature review.

ORCID

Nils Prenner https://orcid.org/0000-0002-2955-0394
Carolin Unger-Windeler https://orcid.org/0000-0002-1392-8882

ENDNOTE

* Hybrid development approaches in software system development: https://www.helenastudy.wordpress.com

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

REFERENCES

1. Theobald S, Prenner N, Krieg A, Schneider K. Agile leadership and agile management on organizational level—a systematic literature review. In: *PROFES'20*. Cham: Springer International Publishing; 2020:20-36.
2. West D, Gilpin M, Grant T, Anderson A. Water-scrum-fall is the reality of agile for most organizations today. *For Res*. 2011;26:1-17.
3. Boehm B, Turner R. Using risk to balance agile and plan-driven methods. *Computer*. 2003;36(6):57-66.
4. McHugh M, McCaffery F, Coady G. Adopting agile practices when developing medical device software. *J Comput Eng Inf Technol*. 2015;4, 9307:2.
5. Royce WW. Managing the development of large software systems: concepts and techniques. In: *ICSE'87*. Los Alamitos, CA, USA: IEEE Computer Society Press; 1987:328-338.
6. Theocharis G, Kuhrmann M, Münch J, Diebold P. Is water-scrum-fall reality? On the use of agile and traditional development practices. In: *PROFES'15*. Springer; 2015:149-166.
7. Klünder J, Hohl P, Schneider K. Becoming agile while preserving software product lines: an agile transformation model for large companies. In: *ICSSP'18*. New York, NY, USA: ACM; 2018:1-10.
8. Kuhrmann M, Diebold P, Münch J, et al. Hybrid software and system development in practice: waterfall, scrum, and beyond. In: *ICSSP 2017*. New York, NY, USA: Association for Computing Machinery; 2017:30-39.
9. Tell P, Klünder J, Küpper S, et al. Towards the statistical construction of hybrid development methods. *J Softw Evol Process*. 2021;33(1):e2315.
10. Klünder J, Hebig R, Tell P, et al. Catching up with method and process practice: an industry-informed baseline for researchers. In: *ICSE-SEIP'19*. IEEE Press, Piscataway, NJ, USA; 2019: 255–264.
11. Brooks FP. No silver bullet essence and accidents of software engineering. *Computer*. 1986;20(4):10-19.
12. Klünder J, Karajic D, Tell P, et al. Determining context factors for hybrid development methods with trained models. In: *ICSSP'20*. New York, NY, USA: Association for Computing Machinery; 2020:61-70.
13. Prenner N. Towards improving the organization of hybrid development approaches. In: *ICSSP'20*. New York, NY, USA: Association for Computing Machinery; 2020:185-188.
14. Thesing T, Feldmann C, Burchardt M. Agile versus waterfall project management: decision model for selecting the appropriate approach to a project. *Procedia Comput Sci*. 2021;181:746-756.
15. Keshta N, Morgan Y. Comparison between traditional plan-based and agile software processes according to team size project domain (A systematic literature review). 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) 2017.
16. Beck K. Embracing change with extreme programming. *Computer*. 1999;32(10):70-77.
17. Beck K, Beedle M, Bennekum VA, et al. Manifesto for Agile Software Development 2001. http://www.agilemanifesto.org/
18. Barlow JB, Giboney J, Keith MJ, et al. Overview and guidance on agile development in large organizations. *Commun Assoc Inf Syst*. 2011;29(2):25-44.
19. Heeager LT. The agile and the disciplined software approaches: combinable or just compatible? *Inf Syst Dev*. 2013;35-49.
20. McMichael B, Lombardi M. ISO 9001 and agile development. In: *AGILE'07*. IEEE; 2007:262-265.
21. Fritzsche M, Keil P. Agile methods and CMMI: compatibility or conflict? *e-Informatica*. 2007;1:9-26.
22. Ramesh B, Cao L, Mohan K, Xu P. Can distributed software development be agile? *Commun ACM*. 2006;49(10):41-46.
23. Cao L, Mohan K, Xu P, Ramesh B. How extreme does extreme programming have to be? Adapting XP practices to large-scale projects. *37th Annual Hawaii International Conference on System Sciences* 2004: 10 pp.
24. Prenner N, Unger-Windeler C, Schneider K. How are hybrid development approaches organized? A systematic literature review. In: *ICSSP'20*. New York, NY, USA: Association for Computing Machinery; 2020:145-154.
25. Kuhrmann M, Tell P, Klünder J et al. HELENA Stage 2 Results. 2018
26. Noll J, Beecham S. How agile is hybrid agile? An analysis of the HELENA data. In: *PROFES'19*. Cham: Springer International Publishing; 2019:341-349.
27. Klünder J, Busch M, Dehn N, Karras O. Towards Shaping the Software Lifecycle with Methods and Practices. 2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE) 2021: 1-11.
28. Boehm B, Turner R. Management challenges to implementing agile processes in traditional development organizations. *IEEE Softw*. 2005;22(5):30-39.
29. Theobald S, Diebold P. Interface problems of agile in a non-agile environment. *Agile Process Softw Eng Extreme Prog*. 2018;123-130.
30. Kusters R, van de Leur Y, Rutten W, Trienekens J. When agile meets waterfall: investigating risks and problems on the interface between agile and traditional software development in a hybrid development organization. In: *ICEIS'17*. INSTICC Press; 2017:271-278.
31. Petersen K, Vakkalanka S, Kuzniarz L. Guidelines for conducting systematic mapping studies in software engineering: an update. *Inf Softw Technol*. 2015;64:1-18.
32. Petersen K, Feldt R, Mujtaba S, Mattsson M. Systematic mapping studies in software engineering. In: *EASE'08*. Swindon: GBR; 2008:68-77.
33. Mirza MS, Datta S. Strengths and weakness of traditional and agile processes—a systematic review. *J Softw*. 2019;14(5):209-219.
34. Špundak M. Mixed agile/traditional project management methodology—reality or illusion? *Procedia - Social and Behavioral Sciences* 2014; 119: Selected papers from the 27th IPMA (International Project Management Association), World Congress, Dubrovnik, Croatia, 2013, 939, 948.
35. Karlström D, Runeson P. Integrating agile software development into stage-gate managed product development. *Empir Softw Eng*. 2006;11(2): 203-225.
36. Chang HF, Lu S. Toward the integration of traditional and agile approaches. *Int J Adv Comput Sci Appl*. 2013;4(2).
37. Hoda R, Noble J. Becoming agile: a grounded theory of agile transitions in practice. In: *ICSE'17*. Los Alamitos, CA, USA: IEEE Computer Society; 2017: 141-151.
38. Kasauli R, Knauss E, Nakatumba-Nabende J, Kanagwa B. Agile islands in a waterfall environment: challenges and strategies in automotive. In: *EASE'20*. Association for Computing Machinery. New York, NY, USA; 2020: 31–40.
39. Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A. *Experimentation in Software Engineering: An Introduction*. Norwell, MA, USA: Kluwer Academic Publishers; 2000.
40. Braun V, Clarke V. Using thematic analysis in psychology. *Qual Res Psychol*. 2006;3(2):77-101.
41. Maxwell J. Understanding and validity in qualitative research. *Harvard Educational Review*. 1992;62(3):279-300.
42. Abdelaziz AA, El-Tahir Y, Osman R. Adaptive software development for developing safety critical software. International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE) 2015:41-46.

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

43. Adelakun O, Garcia R, Tabaka T, Ismail R. Hybrid project management: agile with discipline. In: *CONF-IRM*. Association for Information Systems; 2017.
44. Anitha PC, Savio D, Mani VS. Managing requirements volatility while "Scrumming" within the V-Model. *3rd International Workshop on Empirical Requirements Engineering (EmpiRE)* 2013:17–23.
45. Batra D, Xia W, Meer D, Dutta K. Balancing agile and structured development approaches to successfully manage large distributed software projects: a case study from the cruise line industry. *Commun Assoc Inf Syst*. 2010;27:379-394.
46. Bick S, Spohrer K, Hoda R, Scheerer A, Heinzl A. Coordination challenges in large-scale software development: a case study of planning misalignment in hybrid settings. *IEEE Trans Softw Eng*. 2018;44(10):932-950.
47. Binder J, Aillaud LI, Schilli L. The Project Management Cocktail Model: An Approach for Balancing Agile and ISO 21500. *Procedia - Social and Behavioral Sciences* 2014; 119: 182-191. Selected papers from the 27th IPMA (International Project Management Association), World Congress, Dubrovnik, Croatia, 2013.
48. Ge X, Paige RF, McDermid JA. An iterative approach for development of safety-critical software and safety arguments. *Agile Conference*. 2010;35-43.
49. Hayata T, Han J. A hybrid model for IT project with Scrum. *Pxroceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics* 2011: 285-290.
50. Tordrup L, Nielsen P. Agile software development and its compatibility with a document-driven approach? A case study. *ACIS Proc*. 2009;205-214.
51. Tordrup L. Introducing agile practices in a documentation-driven software development practice: a case study. *J Inf Technol Case Appl Res*. 2014;14:3-24.
52. Heeager LT, Nielsen PA. Meshing agile and plan-driven development in safety-critical software: a case study. *Empir Softw Eng*. 2020;25(2):1035-1062.
53. Heidenberg J, Matinlassi M, Pikkarainen M, Hirkman P, Partanen J. Systematic piloting of agile methods in the large: two cases in embedded systems development. *Product-Focused Softw Process Imp*. 2010;47-61.
54. Heikkilä VT, Paasivaara M, Lasssenius C, Damian D, Engblom C. Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson. *Emp Softw Eng*. 2017;22(6):2892-2936.
55. Imani T. Does a hybrid approach of agile and plan-driven methods work better for IT system development projects? *Int J Eng Res Appl*. 2017; 07(03):39-46.
56. Kautz K, Madsen S. Understanding Agile Software Development in Practice. International Conference on Information Resources Management 2010.
57. Laux I, Kranz J. Coexisting plan-driven and agile methods: how tensions emerge and are resolved. International Conference on Information Systems (ICIS) 2019.
58. Matkovic P, Maric M, Tumbas P, Sakal M. Traditionalisation of agile processes: architectural aspects. *Comput Sci Inf Syst*. 2018;15(1):79-109.
59. Gomes De Oliveira Neto F, Horkoff J, Knauss E, Kasauli R, Liebel G. Challenges of aligning requirements engineering and system testing in large-scale agile: a multiple case study. *IEEE 25th International Requirements Engineering Conference Workshops (REW)* 2017:315–322.
60. Pechau J. Rafting the agile waterfall: value based conflicts of agile software development. *Proceedings of the 16th European Conference on Pattern Languages of Programs* 2011:1-15.
61. Portela LT, Borrego G. Scrumconix: agile and documented method to AGSD. *IEEE 11th International Conference on Global Software Engineering (ICGSE)* 2016:195-196.
62. Qureshi MR. Agile software development methodology for medium and large projects. *Software, IET*. 2012;6(4):358-363.
63. Ramesh B, Mohan K, Cao L. Ambidexterity in agile distributed development: an empirical investigation. *Info Sys Res*. 2012;23(2):323-339.
64. Read A, Briggs RO. The many lives of an agile story: design processes, design products, and understandings in a large-scale agile development project. *45th Hawaii International Conference on System Sciences* 2012: 5319-5328.
65. Paige RF, Charalambous R, Ge X, Brooke PJ. Towards agile engineering of high-integrity systems. *Computer Saf Reliab Secur*. 2008;30-43.
66. Tjørnehøj A. The role of the IT-Project Manager in Organizations that Balance Agile and Traditional Software Development. IRIS: Selected Papers of the Information Systems Research Seminar in Scandinavia 2018(9).
67. Uludağ, KM, Xu X, Matthes F. Investigating the role of architects in scaling agile frameworks. *IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)* 2017:123-132.
68. van Waardenburg G, van Vliet H. When agile meets the enterprise. *Inf Softw Technol*. 2013;55(12):2154-2171.
69. Waterman M, Noble J, Allan G. How much up-front? A grounded theory of agile architecture. *IEEE/ACM 37th IEEE Int Conf Softw Eng* 2015; 1:347-357.
70. Xu P. Coordination in Large Agile Projects. Review of Business Information Systems (RBIS) 2011;13, 4.
71. Waterman M, Noble J, Allan G. The effect of complexity and value on architecture planning in agile software development. *Int Conf Agile Softw Dev*. 2013;238-252.
72. Dingsøyr T, Moe NB, Fægri TE, Seim EA. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Softw Eng*. 2018;23(1):490-520.
73. Scheerer A, Schimmer T, Kude T. Coordination in large-scale agile software development: a multiteam systems perspective. *47th Hawaii International Conference on System Sciences (HICSS)* 2014:4780-4788.
74. Rottier PA, Rodrigues V. Agile development in a medical device company. *Agile Conf*. 2008;218-223.
75. Islam G, Storer T. A case study of agile software development for safety-critical systems projects. *Reliab Eng Syst Saf*. 2020;200:106954.
76. Lucia AD, Qusef A. Requirements engineering in agile software development. *J Emerg Technol Web Intell*. 2010;2:212-220.
77. Kamal T, Zhang Q, Akbar MA. Toward successful agile requirements change management process in global software development: a client–vendor analysis. *IET Softw*. 2020;14(3):265-274.
78. Cataldo M, Herbsleb JD. Communication networks in geographically distributed software development. In: *CSCW'08*. New York, NY, USA: Association for Computing Machinery; 2008:579-588.
79. Liskin O. How artifacts support and impede requirements communication. In: *REFSQ'15*. Cham: Springer International Publishing; 2015:132-147.

How to cite this article: Prenner N, Unger-Windeler C, Schneider K. Goals and challenges in hybrid software development approaches. *J Softw Evol Proc*. 2021;33(11):e2382. doi:10.1002/smr.2382

20477481, 2021, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2382 by Cochrane Germany, Wiley Online Library on [04/06/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License