Release Planning in a Hybrid Project Environment

Kristina Marner[1](✉), Sven Theobald[2], and Stefan Wagner[3]

[1] Dr. Ing. h.c. F. Porsche AG, Porschestraße 911, 71287 Weissach, Germany
kristina.marner@porsche.de
[2] Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
sven.theobald@iese.fraunhofer.de
[3] Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany
stefan.wagner@iste.uni-stuttgart.de

Abstract. *Context*: Even regulated domains like the automotive domain increasingly adopt agile software development. However, traditional sequential processes are still in use and have to coexist with the new development approaches. Collaboration between agile and hybrid projects within complex traditional product development processes is challenging, especially regarding the creation and synchronization of a qualification phase plan. *Objective*: The aim of this study is to motivate research related to the combined use of agile and traditional paradigms in release planning in the automotive domain and to report challenges from industry. *Method*: We introduce and motivate the research topic and discuss related work based on the results of a small literature study. Further, an online survey with 56 respondents from an automotive Original Equipment Manufacturer was conducted. *Results*: There is a clear research gap regarding release planning for combined agile and traditional projects. The state-of-the-practice survey identified challenges, such as a lack of transparency regarding the status quo of related projects. *Conclusions*: The research gap as well as the challenges from industry should motivate further research on this topic, in order to improve release planning processes in this specific context.

Keywords: Automotive · Agile method · Challenges · Hybrid development process · Hybrid project environment · Product development process · Qualification phase · Release planning · Traditional process

1 Introduction

Agile software development approaches promise many benefits like increased transparency, a faster response to change or a shorter time to market [1]. Nowadays, they have become the most commonly used software development approach, especially in information systems domains [1]. Companies from the strongly regulated automotive domain realize these benefits. Most companies have already started their evolutionary bottom-up transition to agility [2] by using agile development in pilot projects. In reality, these agile projects often use hybrid processes [3]. How does a hybrid environment look like? Automotive domains are situated in a hybrid project environment with two conflictive parts. On the one hand, there are processes with many milestones

A. Przybyłek and M. E. Morales-Trujillo (Eds.): LASD 2019/MIDI 2019, LNBIP 376, pp. 19–40, 2020.
https://doi.org/10.1007/978-3-030-37534-8_2

planned long time in advance, before the projects go live and serve the production and the distribution. On the other hand, there is the operational level. On this operational level, the projects try to act in a way that fits best to their project character. The majority of teams in regulated domains prefer adopting single agile practices [2] or strongly adapting existing agile methods like Scrum instead of adopting methods in their pure form [4].

Nonetheless, traditional approaches like waterfall or the V-model are still predominant in the automotive domain. Within these domains, the adoption of agile practices is hard to achieve and even not always desired [5]. This inevitably leads to a mixture of different processes ranging from completely traditional processes to agile adaptations. With these so called hybrid development approaches [3, 6], it becomes more and more complex to handle the interfaces [7] between all involved methodologies.

Especially in the automotive domain, the complexity of software and systems is constantly increasing [8]. Causes are increasing connectivity, increasing distribution of functionality to control units inside the car as well as new technologies like connected services or cloud services. Due to their complexity, automotive projects are large projects with many subprojects and suppliers. One of the many challenges in such projects is to speed up the software release cycles [5]. Creating and updating a common release plan that considers all dependencies is challenging, even more when multiple parties work with different processes and timelines. Agile approaches can cause chaos in release planning, because the two paradigms differ largely.

Thus, we want to motivate the investigation of release planning in the context of system development with coexisting agile and traditional projects and propose the following high-level goal: Identify and analyze challenges in qualification phase planning in order to identify improvements in the context of system development with coexisting agile and traditional projects from the perspective of an automotive original equipment manufacturer (OEM). The research questions below are possible ways to investigate the research field:

- What are challenges concerning qualification phase planning in a hybrid project environment?
- What are the challenges of agile projects that are embedded in a traditional development context?
- What are the challenges of traditional projects, having to synchronize with agile projects?
- What are solutions for the identified challenges?
- How should a release planning process look like to optimize synchronization in a hybrid project environment?

This work extends a previous publication [9]. The contribution of this paper is to detail and motivate the research topic by presenting an extended background and related work. We extended the results of the survey study [9] with the data from 17 additional respondents, and provide more insights into the comments of participants. By presenting these updated results, we highlight common challenges that show the need to investigate potential solutions and come up with a suitable release planning process in a hybrid project environment.

The remainder of this paper is structured as follows: The background and related work is presented in Sect. 2. Section 3 presents the research approach of the survey study, and the results are presented in Sect. 4. We conclude our work and outline future research in Sect. 5.

2 Background and Related Work

In this section, we present the background of our research and the related work. First, the hybrid project environment is presented together with our understanding of release planning. Afterwards, we provide insights into the agile initiative of an automotive OEM and present the identified related research.

2.1 Hybrid Project Environment

In the automotive domain, a hybrid project environment consists of two conflicting parts (see Fig. 1). There is the strategic framework on one side consisting of processes with many milestones planned a long time in advance before projects related to production and distribution go live. This strategic framework (shown in the upper part of Fig. 1) represents the time and content requirements, such as the product development process and thus defines a superordinate process. The Qualification Phase (QP) is the repetitive integration and testing process of an Electronic Control Unit (ECU) network, its sensors and actuators. This phase is typically defined at the beginning of a project. The maturity level is determined to release the ECU network for further testing, usage, and development. The maturity levels provide information about the development progress of functions and ECUs in relation to the target state.

The Additional Qualification Phase (AQP) is an extra qualification phase with a reduced testing scope if the level of maturity is found to be insufficient and refers to a reduced scope of ECUs. The reduced test scope refers to the inadequate target state and is defined application-specifically. An AQP is not planned in advance but established depending on the quality level of the QP. An AQP has to be executed only in case of low quality. For this reason, an AQP is not represented in Fig. 1. The selection of the test cases and the duration of the tests depend on the errors identified during the QP.

On the other hand, there is the operational level: the development of ECUs and associated software. Here projects are performed in the way that best fits the project's character (see bottom part of Fig. 1). On this level, projects are developed in an agile, hybrid or traditional way. However, all projects have one milestone in common, which triggers them to serve the next release. Represented by the grey milestone. Reality demonstrates that coordination and synchronization of these two parts no longer works successfully (symbolized by the different colors of the flashes).

A solution has to be found that synchronizes both levels and which enables coordinated release planning.

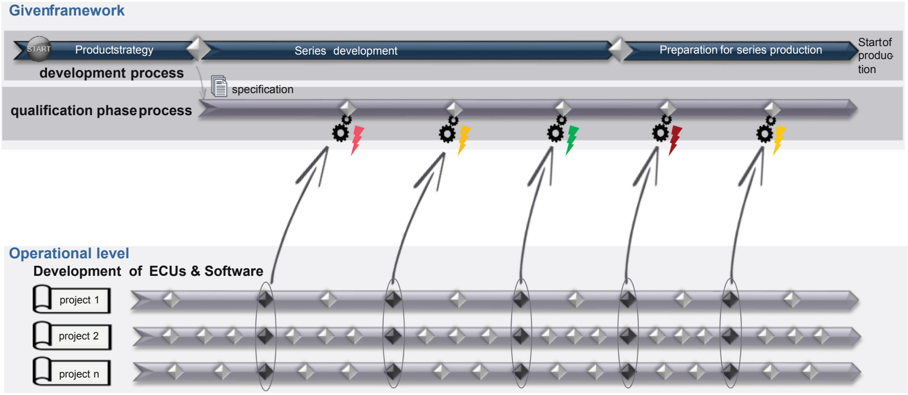

Fig. 1. Coordination between strategic and operational level

2.2 Release Planning

Release planning, in an agile way, contains all decisions to feature content of the next releases and deals with planning to develop the features [10]. Several aspects need to be considered in order to guarantee a successful release:

- Planning: Planning the content that is supposed to be realized in the next release is challenging, since the work has to be distributed on several teams. It is certainly not easy to come up with a reliable plan upfront, so that all planned content can be integrated in the end.
- Coordination: The dependencies between several teams need to be considered and managed. Transparency is necessary to know the relevant stakeholders and to synchronize work with them.
- Integration: Integrating the results from several teams into the final product. Sometimes, errors are only found when integrating, causing bug fixing and additional integration effort.
- Testing: Fully testing the results of all teams individually is already a challenge, but also the integrated product needs to be checked in order to assure all parts work together like expected.

The automotive domain is a strongly regulated domain. Therefore, this combination cannot start in a green field, as strategic frameworks define different phases of the development process.

2.3 Agile@Porsche

The benefits of agile development methods have also arrived at Dr. Ing. h.c. F. Porsche AG. For a stepwise introduction of agile practices, a set-up team called Agile@porsche was established. This team consists of representatives corresponding to the individual departments, such as development, marketing or controlling.

The Agile@Porsche team agrees upon common objectives and activities to strengthen an agile organization. Six dimensions characterize the Porsche-specific approach of agility (see Fig. 2) [11].

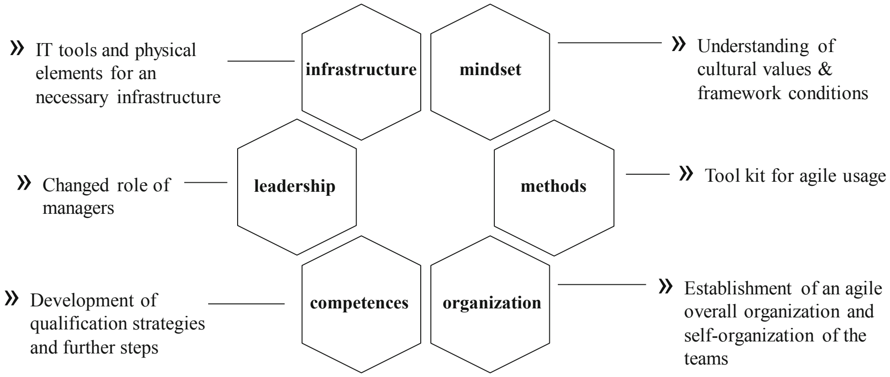
» IT tools and physical elements for an necessary infrastructure
infrastructure
mindset
» Understanding of cultural values & framework conditions
» Changed role of managers
leadership
methods
» Tool kit for agile usage
» Development of qualification strategies and further steps
competences
organization
» Establishment of an agile overall organization and self-organization of the teams

Fig. 2. Six dimensions of agility at Porsche [11]

The first dimension contains a defined understanding of cultural values and framework conditions and provides the basis. The second dimension includes a tool kit for the use of agile methods. This tool kit comprises various methods and practices that can be applied in projects. Organization is the topic of the next dimension that ensures the development of an agile overall structure and the self-organization of individual teams. The development of qualification strategies and the taking of relevant steps for an agile establishment is specified in the fourth dimension that is named competences. The fifth dimension called leadership is characterized by a changing role of leadership. The management has to learn handing responsibility over to the developers. For this purpose a definition of new roles is required and necessary. In an agile environment, the team makes the decisions and a vertical commitment is needed. The usage and implementation of agile practices is only feasible with a suitable environment. For this purpose, a dedicated infrastructure consisting of different IT-tools and physical elements is required and seen as a fundamental condition.

Guidelines for the practical implementation of these six dimensions can be found in a document [11] that offers a standardized definition of agility and a common understanding of agility at Porsche. The transition to a holistic agile organization will be implemented step by step. The state of the practice of using agility results in implementing agile methods in larger scaled projects and the development of supporting processes and organizational structures for anchoring. For the near future, the company targets at having an efficient coexistence of traditional and agile teams.

2.4 Related Work

To investigate how the proposed research topic is already covered by research, we investigated the related work. Besides considering the sources, the authors already knew, a small literature review was conducted. We defined three aspects (c.f. Fig. 3) based on our research goal and tried to identify related work about release planning that in addition covers at least one of those aspects:

1. Release Planning in Systems Engineering
2. Release Planning in the automotive domain
3. Release Planning in a hybrid project environment

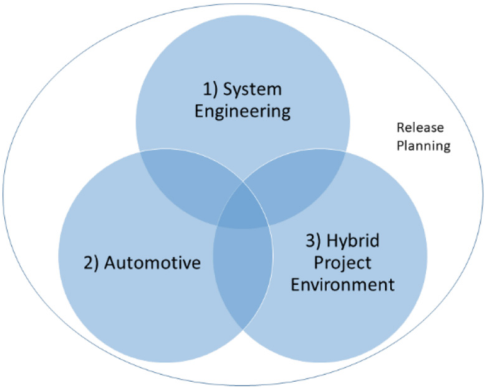

Fig. 3. Research aspects

For each category, we defined a search string (see Fig. 4) and searched on Scopus[1]. We assessed the resulting list of literature based on the title and included every literature that seemed to address any of our categories. This final list of papers was then mapped to the three categories and checked for relevancy.

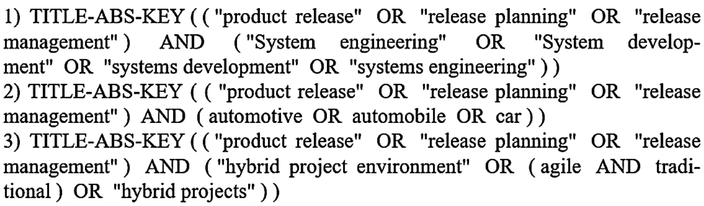
1) TITLE-ABS-KEY (("product release" OR "release planning" OR "release management") AND ("System engineering" OR "System development" OR "systems development" OR "systems engineering"))
2) TITLE-ABS-KEY (("product release" OR "release planning" OR "release management") AND (automotive OR automobile OR car))
3) TITLE-ABS-KEY (("product release" OR "release planning" OR "release management") AND ("hybrid project environment" OR (agile AND traditional) OR "hybrid projects"))

Fig. 4. Search strings

In the following, we present related work and analyze research gaps. There is a research gap considering this hybrid project environment where projects with different development paradigms meet. The HELENA study [3] investigates the combined use of agile and traditional practices in hybrid processes, but does not consider the coexistence of agile and hybrid projects and their synchronization. Theobald et al. [7]

[1] www.scopus.com.

investigates and classifies problems at the interface of agile and traditional environment. Our work in this paper can be classified in the problem field “project planning” at the interface “project-team”. We could only identify one source dealing with release planning in a hybrid project environment. [12] identified “an obvious gap in the research of release planning in large-scale agile software development organizations” in a literature review.

There is some work on release planning in the automotive domain. [13] describes software release and configuration management in the automotive domain. [14] define requirements to control and monitor dependencies with other release processes with the help of workflow support. [15] define requirements for IT-support to improve release management in the automotive domain. [16] identified key aspects of release planning in the context of software and system development projects and also captured the state of the practice for release planning in industry. There is no explicit work on release planning in systems engineering, but the contributions dealing with an automotive context mostly also cover release planning in systems engineering.

There is literature dealing with release planning in agile software development projects: single project as well as scaled projects. [17] evaluates the methods that are used by companies to plan for new software releases. [18] presents a case study where the agile release planning process in a scaled Scrum environment was evaluated. [19] describes the practice qualification phase planning and presents a case study of multi-team agile release planning with the help of this practice. [20] conducted a systematic literature study to identify agile release engineering practices. [21] conducted a literature study to report on software release planning models.

The focus of the majority of publications on release planning models aim at several kinds of mathematical models and simulations [22], which are ineffectual in complex industries [23]. Practitioners said that these approaches are either too simple to generate a benefit or so difficult that they cannot reconstruct the whole process created [24, 25].

In total, there is no direct related work that considers release planning in coexisting traditional and agile processes in the automotive domain. Some work deals with agile release planning, but none of the identified sources deals with the hybrid project environment that is targeted. In order to better understand release planning in such a hybrid project environment, we conducted a survey study at an automotive case company [9] and present the updated results in the following sections.

3 Research Approach

The research approach is presented including the research questions and design, the data collection and analysis procedure, the research site and participants, as well as the threats to validity.

3.1 Research Questions

The survey study aims to answer the following research question: What are the challenges and consequences of the qualification phase in an automotive hybrid project environment? To answer this question, three research questions were defined:

- RQ1. What are challenges concerning the qualification phase in a hybrid project environment?
- RQ2. What are the specific challenges of agile projects embedded in a traditional development context?
- RQ3. How could agility address the identified challenges?

3.2 Research Design

To answer the research questions, we selected a two-step research approach. First, we set up an exploratory, qualitative interview study within a German automotive OEM. An interview guide for identifying challenges and problems with regard to the release planning process was specified. The interview guide was tested in a pilot interview. Emerging issues, such as vague phrases, were addressed before the qualitative interview study was conducted. In the second step, an online survey questionnaire was built to validate the challenges identified from the qualitative interview study in detail.

The data collection instrument was a questionnaire that contained 31 open and closed questions structured into six categories (cf. Table 1).

Table 1. Survey questionnaire

Category	ID	Question
Context	1	What is your current role? [free text]
	2	How long have you been working in that role? [free text]
	3	What are you working on in your project? [E/E ECU, software component, function, connect service, vehicle project]
	4	Please select a sector to classify your project. [powertrain electronics, body electronics, infotainment, project is safety-critical, others]
	5	What kind of development method do you use? (agile, hybrid, or traditional) [use of adapted agile methods, hybrid methods, traditional approaches]
	6	If you are using agile or hybrid methods, please specify the method. [free text]
Qualification phase	7	What do you think about the current number of qualification phases (incl. additional qualification phase)? [too high, adequate, too low]
	8	How often are you able to generate current software versions ready to deliver? [never, seldom, often, always]
	9	Do you receive feedback about the qualification phase on time? [never, seldom, often, always]
	10	How often should a qualification phase take place in order for you to be ready to deliver? [every week, once a month, every 3 months, at larger intervals]
	11	Would additional releases in terms of partial composites with reduced test scope be helpful for safeguarding dependent ECUs? [yes, partially, no]

(*continued*)

Table 1. (*continued*)

Planning	12	Is an initial planning of content possible? [never, seldom, often, always]
	13	Does an initial planning of content make sense? [never, seldom, often, always]
	14	How often is the content of the initial planning still up-to-date at the beginning of a qualification phase? [never, seldom, often, always]
	15	How difficult is it to get planning information for the relevant counterparts? [very difficult, difficult, easy, very easy]
	16	To what extent do management decisions, external influencing factors, or externally determined decisions influence your development process? [no impact, weak impact, strong impact, very strong impact]
Integration	17	To what extent does bug fixing affect the timely implementation of planned functionalities for the next qualification phase? [no impact, weak impact, strong impact, very strong impact]
	18	It is inevitable that software versions are released that are suboptimal concerning quality or content. [yes, partially, no]
	19	What kind of activities dominate your daily routine during a qualification phase? [free text]
	20	Rate the following statement: Additional qualification phases are necessary. [yes, partially, no]
	21	Rate the following statement: Additional qualification phases are reasonable. [yes, partially, no]
Coordination	22	Is the status of development transparent to you at any time? [yes, partially, no]
	23	Is the status of development of your stakeholders transparent to you at any time? [yes, partially, no]
	24	How important is the transparency of the development status of your relevant counterparts to you? [totally unimportant, unimportant, important, very important]
	25	Rate the following statements: - Stakeholder/Interfaces are known [Disagree, rather disagree, rather agree, agree] - Quality of coordination is good. [Disagree, rather disagree, rather agree, agree]
Testing	26	Development can no longer handle the high number of bug reports. [Disagree, rather disagree, rather agree, agree]
	27	Problem resolution management can no longer handle the high number of bug reports. [Disagree, rather disagree, rather agree, agree]
	28	What are the reasons for the high number of tickets? [free text]
	29	Do all planned changes to the ECU network have to be fully tested for each qualification phase? [yes, partially, no]
	30	Do all types of tests have to be performed for every ECU for each qualification phase? [yes, partially, no]
	31	When do all ECUs have to be fully tested? [every qualification phase, depending on the changes, not mandatory]

The categories and questions were derived from the insights gained in the previous interviews and match the four aspects of release planning (c.f. Sect. 2.2). The questions were originally written in German. The questionnaire went through four review cycles by an independent researcher as well as by a specialist from the case company. The authors discussed the review comments and improved the questionnaire.

In the first category, we elicited the “Context”, such as role and experience of the participant, as well as project type, area, and the development method used (traditional vs. agile). The second category, “Qualification Phase”, aimed at evaluating how many qualification phases are feasible. The third category, “Planning”, was for evaluating the need to have an initial plan as well as external influences on such a plan. At a certain point in the development process, an initial planning of the functional scope of an ECU must be submitted for each release. In addition to general ECU information, deviations from the required functional, network and diagnostic maturity levels must also be specified. We examined the need for additional qualification phases in the fourth category “Integration”. Integration is an upstream part of the actual process and represents the integration of one or more ECUs into a whole network. Transparency of the status quo and the quality of coordination were the focus of the fifth category, “Coordination”. Finally, we covered all questions related to “Testing” in the last category, trying to evaluate which kind and intensity of tests are necessary and if and why there are so many bug reports. The test phase focuses on the execution of the qualification phase and is therefore a main activity.

3.3 Data Collection and Analysis Procedure

To identify the main challenges, the first researcher conducted 26 semi-structured interviews, which took between 30 and 60 min each. The information from each interview was incorporated into later interviews. Because these interviews did not allow for quantitative results, an online survey was conducted to confirm the challenges and to draw a more complete picture by consulting different participants. This allows for quantitative results, but gave every participant the chance to provide further qualitative results by sharing their experiences.

During the first run of the online survey (results presented in [reference]), 95 potential participants were selected based on their roles, to cover all perspectives. Then the participants were invited via an email motivating the goal of the study and outlining the contents and the time expected to answer the questionnaire. A reminder email was sent after one week. Also, one of the participants forwarded the questionnaire to an additional group of 25 people. The survey was open from November to December 2018 and resulted in 55 respondents of which 39 completely filled out the whole questionnaire.

In addition to [9], we run the online survey a second time. Reasons were to give everyone who missed the first round an opportunity to share their experiences, and to increase the number of respondents in order to increase the validity of the results. The survey was opened up at the end of February 2019 for a duration of two weeks. Afterwards, combined with the results of the first round [9], we ended up with 94 respondents, of which 56 completely filled out the questionnaire.

After extracting the data from the online survey tool into an Excel document, we analyzed the answers for completeness. There were 56 complete responses, meaning all six pages of the questionnaire had been answered and thus the survey had been officially finished. In addition, there were 38 incomplete answers where the questionnaire was not finished. Of these 38 incomplete answers, 11 respondents only finished the first category (Context), while 20 respondents did not even finish the first questions. Only three respondents finished the second category (Qualification Phase), two respondents stopped after category 3 (Planning) and 4 (Integration) each. Although we had access to the incomplete data sets, we decided to only consider the complete data sets for further analysis. Afterwards, we conducted a descriptive analysis of the individual questions and analyzed the textual answers to identify common opinions.

3.4 Research Site and Participants

This study was conducted at Dr. Ing. h. c. F. Porsche AG, a manufacturer that builds sports cars for everyday driving. The division EE within Dr. Ing. h. c. F. Porsche AG in Weissach, Germany, is responsible for the development process of electronic systems and its integration into the development process of the complete vehicle. For achieving this goal, transparent development processes and hence accurate release planning are essential.

The target population of our survey included all roles involved in the qualification phase process of automotive products where the subprojects differed in terms of the development approaches used, including agile as well as traditional methods. The sample selected consisted of stakeholders from Dr. Ing. h. c. F. Porsche AG involved in release planning activities. The participants were expected to be motivated enough to answer the comprehensive questionnaire because they anticipated improvements based on the findings that reflect their current situation.

3.5 Threats to Validity

As the results only represent one specific case, it might not be possible to generalize them. However, the fact that the case company has the same framework conditions (regulated domains, complex supplier relationships and high safety requirements) as similar OEMs, others could benefit from the findings. The issues that were identified in the earlier interviews were addressed in the questionnaire, whereas new survey participants did not have a chance to add more individual problems during the online survey. There might be a bias concerning the stakeholders who participated. Some roles are overrepresented, while other relevant roles were not represented by many participants. This might have led to results that are skewed towards the opinion of certain roles. Nonetheless, many different roles participated in the study, providing answers from many perspectives. As in all surveys, non-response bias could have led to missing the opinions of certain participants. The second round of the interview study was conducted at a later point of time, which could have led to a difference in the perception of the participants. However, a comparison of results of the first with the second round showed that the answers of all participants followed the same trend.

4 Survey Results

This section contains the demographics and context of the respondents, followed by the presentation and discussion of the results of this work structured along the research questions.

4.1 Context

The respondents' professional experience in their current role (Q1) was slightly below six years on average, with a minimum of one year and a maximum of 20 years (Q2). 18 respondents (32%) had management roles related to projects, products, functions, integration, testing, quality, data, processes, or other related disciplines. 22 participants (39%) represented the operational level. The remaining respondents had roles with responsibilities related to the environment of qualification phases (n = 16; 29%).

The respondents described their working environment using one or more categories (Q3). Most participants reported working in vehicle projects (n = 29), development of E/E components (n = 23), development of functions (n = 22), development of software components (n = 17), and connected services (n = 16). Others (n = 7) dealt with IT backend, cross-project integration, distributed functions, or quality.

27% (n = 15) of the respondents answered that their project was safety-critical. Most participants assigned their project to the area of infotainment (n = 26), followed by electronics for car bodies (n = 14) and electronics for engines (n = 8). Regarding the 25 individual answers, ten participants reported working on crosscutting topics (Q4).

Most respondents reported using traditional development or project management approaches such as the V-model or sequential approaches (n = 34). Only twelve respondents used adapted agile methods, and ten persons used hybrid approaches, which was defined as strongly adapted agile methods or use of only single agile practices (Q5). This showed that about 40% of the participants were using agile concepts at the time.

Agile implementations were based on Scrum or the Porsche-specific adaption of agile methods. One person even reported scaled agile and lean at the unit level combined with an adapted Scaled Agile Framework (SAFe). Single agile practices like daily standups, user stories, backlogs, retrospectives, or the Scrum Master role were used in traditional projects. Some respondents reported using both agile and traditional approaches at different project levels. One answer stated that agile was being used at the team level together with the V-model for whole projects, while another respondent reported using a sprint-like approach within the V-model due to highly dynamic changes in requirements. Another respondent indicated the use of different development paradigms in different life cycle phases (Q6).

4.2 RQ1: Current Challenges

In the following, the current challenges concerning the qualification phase in a hybrid project environment will be presented and discussed along the categories of the survey questionnaire.

(1) Qualification phase. The majority of all participants (n = 30; 54%) stated that the current number of releases (p.a.), including all additional qualification phases (AQPs) and special qualification, is too high (Q7). On the other hand, there are 14 participants that claim that this number is too low, while another 12 perceive this amount as the right number of releases. An analysis of the comments field of this question shows results relating to the regulated defined number of releases. It emerged from the free text that the regular number of QPs (without AQPs and special qualifications) is appropriate. Nevertheless, AQPs and special qualifications are inevitable in the project phase relating to the start of series production. The developers confirmed their opinion and asked for a higher number of qualification phases. The management group agreed with the regulated defined numbers.

Further information concerning the ordinary number of qualification phases was given by the group of developers using agile methods. For the majority of those participants, the absolute number of qualification phases is too low to use agile methods properly. They complained about the too great distance between two QP to integrate and test the new version more quickly.

The next issue concerned the delivery results (Q8). 52% of the survey participants answered that the required deliverable is seldom available in the required quality. On a closer look of these 52%, n = 12 of the management level answered that the deliverable is seldom available. In contrast, 46% of all participants replied that it is always (11%) or at least most of the time (35%) possible to create a delivery version for every requested release. Hence, there is a different perception of the definition of ‘required quality’ and what exists in reality.

45% of all participants (n = 25) answered that they mostly receive feedback about qualification phases on time (Q9), while ten participants (18%) claim to always be informed in time. A minority never receives feedback on time (n = 7; 12%), another 14 participants only do seldom (n = 14; 25%). A deeper analysis of the answers related to this question showed that it depends on the stakeholder and its required feedback. The management level wants to have early feedback in order to be able to intervene in time. However, the operational level needs feedback about the testing results in more detail, which takes time. Receiving feedback on time and in the required quality depends on different expectations and fixed targets.

The next question dealt with identifying a suitable number of qualification phases with regard to being able to generate a releasable software version (Q10). Half of the participants (n = 29; 52%) stated that qualification phases should take place at least each quarter of the year. Six participants (11%) said that the QP should take place each week, 14 participants (25%) wanted the QP to take place once a month. Only seven participants (12%) said that the QP should take place less often than each quarter of a year. In the comments, the participants emphasize that a regular QP should target testing on vehicle level. Therefore, a QP should serve the qualification of the whole vehicle network. In order to avoid big bang integration, smaller integration loops should be carried out in advance in order to achieve the greatest possible maturity for the QP.

In the last question (Q11) of this category, 50% of participants called for additional qualification phases with reduced test scopes. Only five participants (9%) said that such additional QPs would not be helpful, the remaining 23 respondents (41%) agreed in parts. On a closer look, 68% of all participants of the operational level want to have a higher number of QPs on condition that not the whole scope has to be tested. Nonetheless, this statement raises concerns, since the operational level answered in Q8 and Q10 that they are not always able to release good quality with the regular number of QPs.

(2) Planning. This category highlights the characteristics around planning. The first question (Q12) aimed at evaluating the feasibility of initial planning at the beginning of the project. 48% of the participants in our study reported that initial planning is mostly or always possible, and the others (52%) answered that such a plan is rarely or never possible. At the beginning of a project, the decisions for or against a supplier have sometimes not been made yet. That is one reason why it is difficult to generate an initial planning. Another person replied that requirements for functions are the results of testing, which is done further on in the development process.

In a further question, the participants were asked if such initial planning would be meaningful (Q13). A majority (n = 34; 61%) stated that planning at the beginning of a project is mostly (20%) or always (41%) reasonable because it is a resilient starting point for further steps. On a closer inspection, only 18% of all developers are in favor of an initial planning. The other half do not consider planning effective. Participants also mentioned the existing change management process as an argument for initial planning, which permits updates at any time.

The next question (Q14) regarding this topic dealt with the projected content before the next release in terms of timeliness. The results show that scheduled content is frequently impossible to implement in practice (80%). No participant stated that the content is still up to date at the start of a new release, only 11 participants (20%) claimed it is up to date most of the time. The majority of the participants stated that awareness still exists for high quality in planning. Planning updates have to pass a committee, which is one reason why change requests are not implemented in the current release. Also some areas, such as the area “connected car”, are very dynamic, which is another reason for the bad current state of planning, which is not up-to-date.

Receiving information about planning details from the relevant stakeholders is perceived as challenging (Q15). Most of the respondents replied that obtaining information on time is difficult (59%) or very difficult (14%), because there are no regulated tasks nor a consistent workflow for changing the relevant information. The remaining participants perceived it to be easy (23%) or very easy (3%) to receive planning information.

Another issue is the impact of management decisions during the development cycle (Q16), which implies that these cannot be implemented easily. A significant majority of the respondents rated the influence of external decisions on the course of development as strong (50%) or very strong (43%) and reported that the development of new functionality suffers from having to deal with unexpected changes demanded by management. Some respondents complained about management decisions that change the backlog priority and have severe effects on further procedures. Only four participants perceive the influence to be weak (7%), no participant stated that there is no influence.

(3) Integration. This category contains the results relating to the challenges of software and hardware integration during a development cycle. During a qualification phase, new software versions are tested at different levels of integration. The test results and even bug fixing have a great impact on the subsequent procedure (Q17). Most of the participants answered that bug fixing has a strong effect (50%) or very strong effect (29%) on their timely implementation related to the next release. Only three participants (5%) claimed that there is no influence. The remaining 16% reported a weak influence. 73% of the operational level stated that debugging has a strong effect on the upcoming tasks. Since the project plan does not consider an extra buffer, fixing bugs can even lead to delays of the next scheduled functions.

This non-existing hold is one of the main reasons for bad quality releases (Q18). Almost all interviewees admitted that delivering software versions with high quality is infeasible (46%) or partly infeasible (50%) when they also have to provide the content planned for the next release. The results considered for integration have low maturity, due to the increasing pressure of costs and deadlines.

Another question in this category dealt with the activities during a qualification phase (Q19). The main activities or tasks linked to the respective role are: Management is engaged in coordination and ensuring the scheduled scope with regard to the next release. At the operational level, tracking of test results and analysis of upcoming bug tickets are the main concerns. Both groups have to handle the subsequent deliveries.

For this reason, additional qualification phases (AQP) have been established subsequent to the original deadline. We wanted to know if such AQPs are necessary (Q20) and reasonable (Q21). 64% of all participants considered AQPs necessary, 29% partly necessary and there were only 7% of participants that claimed that AQPs are not necessary. After taking a closer look, 60% the operational level call for AQPs as a necessary action. 41% of all participants were convinced that AQPs are reasonable, 30% stated that they are partly meaningful. There were 29% of respondents that claimed that AQPs are not meaningful. However, 27% of the operational level considered AQPs to be of partly use. The main reasons given by the participants for subsequent integrations were poor software quality, lack of adherence to delivery dates on the part of the suppliers, poor scheduling without buffering, and no complete bug fixing from the previous qualification phase.

(4) Coordination. Transparency and coordination were the relevant aspects in this category (Q22). We asked whether the current development status of the respondents' own team or dependent teams is sufficiently transparent. Only 28% of all participants (n = 9) reported that their own development is transparent. The majority of respondents rated transparency as only partially existent (n = 30; 54%) or non-existent (n = 10; 18%).

The next question (Q23) dealt with the transparency of the status of projects by relevant stakeholders and relevant counterparts. Here, only 14% (n = 8) of the respondents answered that the development status of other projects is transparent for them. Most participants (n = 27; 48%) reported partial transparency, while 38% (n = 21) reported a lack of transparency. Reasons for the lack of transparency were missing time and coordination mechanisms, and the use of outdated content of the release plans. The free text fields reveal that developers agree that generating transparency is a management task.

The transparency of the status quo of a certain development project is very important and closely linked to the quality of a release (Q24). 93% of the respondents supported the statement that having a transparent software version at any time is important (45%) or very important (48%). Transparency is necessary due to the complexity, dependency, and connectivity of software engineering.

Another question aimed at getting information about the communication structures within the company and involved persons from the release planning process (Q25). The participants had to rate whether they knew their interfaces and relevant stakeholders and whether the quality of the coordination was good. This rating had to be done for several interfaces: within the team, between team and testing, within the case company, within the company group, as well as towards external suppliers. The results presented in Fig. 5 (bottom figure) demonstrate that communication quality decreases with longer communication paths: Communication within a project was perceived as good, but the quality was perceived as decreasing in communication within the company and even worse in communication with suppliers (internal means company group and external suppliers). Similarly, the relevant stakeholders and interfaces of the wider project context were reported less known than those within the team (see Fig. 5, top figure).

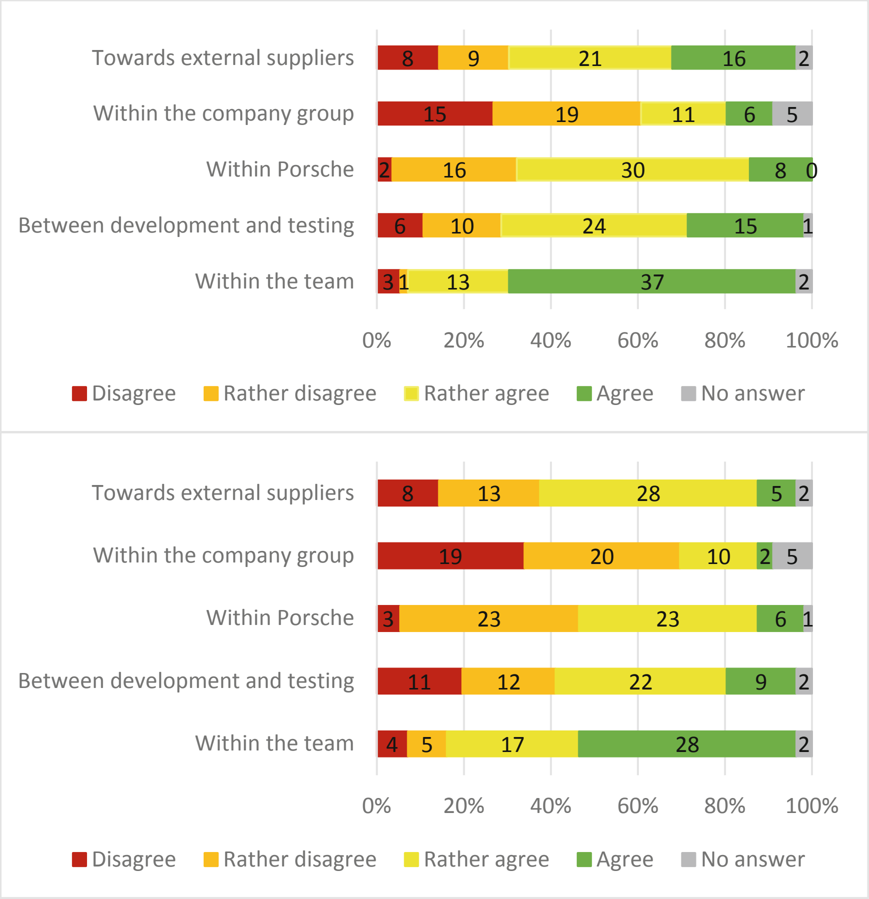

Fig. 5. Known interfaces (top) and quality of coordination (bottom)

(5) Testing. This category assesses the testing situation. The first question aimed to evaluate whether the number of bug reports is still controllable by development (Q26) or problem resolution management (Q27). Overall, the majority of the participants agreed (20%) or rather agreed (36%) that development is able to control the high number of bug reports. The remaining respondents had a tendency to disagree (25%) or disagreed (19%). Concerning problem resolution management, most participants disagreed (21%) or had a tendency to disagree (39%) that resolution management is able to cope with the number of bug reports. The minority of the participants agreed (14%) or rather agreed (25%).

Furthermore, the participants were asked about reasons for the high number of bug reports/tickets (Q28). The survey revealed that identifying errors is usually not done before the upcoming release due to insufficient development time, cost, and deadline pressure. It was reported that the intensity of testing by the supplier was not adequate. Other reasons given for the high number of error tickets were the rising complexity of the product itself, a lack of coordination within the team, and inadequate requirements engineering. Generally, it can be stated that the quality before a qualification phase is insufficient and questionable, endangering the success of the qualification phase.

Software changes may have severe effects on interfaces, which is why tests have to be done. The need for testing the software changes to the full extent for every qualification phase (Q29) was not seen by 13% of the respondents, who claimed that this is not necessary. Most respondents (55%) said that changes have to be tested to the full extent for every planned release. The remaining 32% partially agreed that testing is always necessary and specified in the comments specific situations where more testing was necessary or less testing was acceptable. Some stated that the scope of testing depends on the number of changes made or on the development phase. One respondent commented that it is not possible to test all changes; another one said that full testing is always necessary because cross-dependencies only become visible by testing within a release. Only 16% of the respondents agreed that all types of tests have to be performed in every release cycle (Q30). 36% disagreed with this statement and about half of them (48%) partially agreed. The participants further pointed out that conducting all tests is not feasible or that the necessary types of tests are predefined in the test strategy and depend on the change itself. Others reported that regression tests are often sufficient, or that full releases need to be tested more accurately than partial releases.

To save testing effort, it is important to know when comprehensive testing (including all types of tests) of all ECUs needs to be done (Q31). 79% of the respondents answered that testing needs to be done dependent on the software, hardware, or functional changes. Some respondents (16%) claimed that the ECUs have to be tested once per qualification phase, either at the beginning or at the end. 5% said that testing is not always necessary. One participant commented that due to the high product complexity and low software quality, all ECUs have to be tested as an integrated system with all possible tests, or at least with good regression tests. Another one claimed that comprehensive testing is not possible for all system parts, but major parts can be covered with a good testing strategy.

4.3 RQ2: Agile-Specific Challenges

Existing vehicle development processes emerged at a time when agility was not present yet and were formalized based on traditional development methodologies. Due to the regulations, strict production deadlines and the complexity in vehicle development, the need to have formal processes will remain. However, the potential to integrate agile processes must be evaluated in order to exploit the benefits of agility. New technologies such as cloud services implicate a stronger customer focus, to be able to respond more flexibly to customer needs, which results in conflicts with the slow and unresponsive traditional development. Innovation is happening fast in the automotive domain, and companies have to react in time to stay competitive.

Iterative cycles are already incorporated into many processes, but other concepts of agile methods initially designed for small teams are more difficult to integrate or synchronize with the existing rigid processes. The OEMs are currently performing a balancing act between fixed framework conditions and scope for flexibility. On the one hand, legal requirements, standards and production requirements must be observed and on the other hand, developers want to act more freely without being restricted by guidelines. The results of this survey indicate that this is not a simple procedure.

The survey revealed that if departments are already working with agile methods, they only use them to a certain extent. Our initial expectation was that agile methods are commonly used at least in fields such as connected car, with its digital services and shorter development cycles. The differences between our expectations and reality may be caused by the lack of a common understanding of agile methods. This is confirmed by the inconsistency of the answers by the respondents, who considered additional qualification phases necessary but at the same time did not demand more qualification phases. The reason for this may be a lack of knowledge about agile methods.

There is also a lack of suitable means of communication for short, regular exchanges aimed at establishing transparency between all participants. Such possibilities for fast feedback would also increase the overall quality of voting and benefit the flow of information. Respondents from agile projects reported that the length of release cycles is too long and does not suit agile approaches.

The fact that management decisions have such a strong influence on the further course of development illustrates that decisions are made at higher levels of hierarchy. In an optimal agile environment, the development team makes the decisions. Based on the priorities set by the Product Owner and the requirements dependencies identified by the development team, a Scrum team knows best how to achieve the best solutions. At the beginning of each iteration, they commit to a product increment that is valuable and achievable. If management forces decisions upon the team during an iteration, results can be expected to be suboptimal.

However, this is only the point of view of a single team. If each single team cannot meet their commitments, the qualification phase of an integrated product is going to raise problems. One reason is that the release plan, which considers dependencies between different projects, gets unofficially changed without being updated. That means the developers change their release plans on the operational level without having the change approved and without informing the affected interfaces.

4.4 RQ3. Improvements with Agile Methods

There are many challenges that are predestinated to be solved with agility. The survey revealed that transparency and coordination are highly important for a successful qualification phase. Some of the interviewees stated that the communication path in their department is too long, which causes loss of time and a lack of coordination. This argument is supported by the fact that some participants reported not knowing their interfaces and relevant stakeholders, resulting in bad synchronization and integration structures. By using agile development and by having small working groups with no typical hierarchy, interface management as well as short communication paths could become possible [26].

Currently, additional qualification phases are started to fix the remaining bugs or to finish some functionalities that had been planned for the previous release cycle. Due to the increased effort for these activities, the planned results for the next release cycle cannot be fully achieved, pushing a wave of additional efforts, e.g., for coordinating additional qualification phases, through the whole project. Increased transparency regarding the content that was finished in an iteration can be achieved with a definition of done and by incorporating time-boxed sprints. At the end of each sprint, the status quo is assessed, and unfinished requirements can be planned for the next sprint.

Another characteristic of sprints is that requirements are usually not changed, especially not from outside the team. This would also help to stabilize the release plan, which would help to achieve higher-quality products delivered for integration by each single team. Sprints are usually short iterations of several weeks. Respondents from agile projects reported that the length of release cycles is too long, and that they would prefer receiving feedback earlier. This issue leads to work overload and defined timelines not being achieved, which ultimately leads to lower software quality. In addition, development costs increase due to many additional qualification phases. By using agile methods and more intermediate steps, including regular assessments of the project state, discrepancies could be identified earlier.

Agile teams use face-to-face communication and daily standups to synchronize their work in order to achieve their sprint goal. In a scaled environment, so-called Scrum-of-Scrums are scaled daily standups where representatives of different teams synchronize their development status and plan their dependencies. The Scaled Agile Framework (SAFe) uses an architectural runway to coordinate architectural decisions between the single development teams to facilitate integration.

Continuous integration is commonly used in agile projects and could be of benefit in qualification phases. Integrating smaller work products incrementally can replace a larger and more complex final integration and provides early transparency about the finished content of the release as well as raising awareness of dependencies.

In general, regular retrospectives can be held at the end of each sprint, helping the team to raise issues impeding their work and improve their development process. Conducting retrospectives together with relevant stakeholders and dependent projects helps to continuously improve collaboration between teams.

5 Conclusion and Future Work

Agile development has its origins in the information systems domain. Organizations from regulated domains like the automotive domain also want to benefit from agility. Due to regulations and the complexity of systems, traditional sequential processes are still in place and have to be synchronized with the coexisting agile and hybrid development approaches. Qualification phases of traditional, hybrid as well as agile projects are difficult. Release planning in such a hybrid project environment needs to be improved. In this research, we presented this research topic by formulating research goal and questions and by detailing the research background. An investigation of related work showed that there is a research gap concerning this topic. We elicited the state of the practice from 56 respondents of a German automotive Original Equipment Manufacturer. The identified challenges, such as lack of transparency, show the problems related to release planning in this specific context. We outlined how agile concepts could improve some of the identified challenges and thus provided recommendations for practitioners.

In future work, the survey questionnaire can be adapted to collect experiences outside the case company, in order to check whether there are similar problems at other automotive companies or even companies from other regulated domains that are developing complex systems in a hybrid project environment. Finally, solutions for the identified challenges need to be identified in order to come up with a suitable release planning process in the context of co-existing traditional and agile approaches.

References

1. VersionOne: the 13th annual state of agile report (2019). www.collab.net
2. Diebold, P., Zehler, T.: The right degree of agility in rich processes. Managing Software Process Evolution, pp. 15–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31545-4_2
3. Klünder, J., et al.: HELENA Study: reasons for combining agile and traditional software development approaches in German companies. In: Felderer, M., Méndez Fernández, D., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D. (eds.) PROFES 2017. LNCS, vol. 10611, pp. 428–434. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69926-4_32
4. Diebold, P., Theobald, S.: How is agile development currently being used in regulated embedded domains? J. Softw. Evol. Process **30**(8), e1935 (2018). https://doi.org/10.1002/smr.1935
5. Hohl, P., Münch, J., Schneider, K., Stupperich, M.: Forces that prevent agile adoption in the automotive domain. In: Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., Mikkonen, T. (eds.) PROFES 2016. LNCS, vol. 10027, pp. 468–476. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49094-6_32
6. Kuhrmann, M., et al.: Hybrid software and system development in practice: waterfall, scrum, and beyond. In: Proceedings of the 2017 International Conference on Software and System Process, pp. 30–39 (2017). https://doi.org/10.1145/3084100.3084104
7. Theobald, S., Diebold, P.: Interface problems of agile in a non-agile environment. In: Garbajosa, J., Wang, X., Aguiar, A. (eds.) XP 2018. LNBIP, vol. 314, pp. 123–130. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91602-6_8

8. Broy, M.: Challenges in automotive software engineering. In: Proceedings of the 28th International Conference on Software Engineering, pp. 33–42. ACM (2006). https://doi.org/10.1145/1134285.1134292
9. Marner, K., Theobald, S., Wagner, S.: Real-Life challenges in automotive release planning. In: Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 831–839 (2019). https://doi.org/10.15439/2019f326
10. Ruhe, G., Saliu, M.O.: The art and science of software release planning. IEEE Softw. **22**(6), 47–53 (2005)
11. Dr. Ing. h.c. F. Porsche AG: Methodenleitfaden für agiles Arbeiten bei Porsche (guidelines for agility at Porsche) (2018)
12. Heikkilä, V.T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T., Järvinen, J.: Operational release planning in large-scale scrum with multiple stakeholders–a longitudinal case study at F-Secure Corporation. In: Information and Software Technology, vol. 57, pp. 116–140 (2015). https://doi.org/10.1016/j.infsof.2014.09.005
13. Sax, E., Reussner, R., Guissouma, H., Klare, H.: A Survey on the State and Future of Automotive Software Release and Configuration Management. Karlsruhe Reports in Informatics, 11 (2017). https://doi.org/10.5445/ir/1000075673
14. Bestfleisch, U., Herbst, J., Reichert, M.: Requirements for the workflow-based support of release management processes in the automotive sector. In: Proceedings of the 12th European Concurrent Engineering Conference ECEC 2005 (2005)
15. Müller, D., Herbst, J., Hammori, M., Reichert, M.: IT support for release management processes in the automotive industry. In: Dustdar, S., Fiadeiro, J.L., Sheth, Amit P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 368–377. Springer, Heidelberg (2006). https://doi.org/10.1007/11841760_26
16. Lindgren, M., Land, R., Norström, C., Wall, A.: Key aspects of software release planning in industry. In: Proceedings of the 19th Australian Conference on Software Engineering, pp. 320–329 (2008). https://doi.org/10.1109/aswec.2008.4483220
17. Danesh, A.S., Ahmad, R.B., Saybani, M.R., Tahir, A.: Companies approaches in software release planning-based on multiple case studies. JSW **7**(2), 471–478 (2012). https://doi.org/10.4304/jsw.7.2.471-478
18. Heikkilä, Ville T., Paasivaara, M., Lassenius, C., Engblom, C.: Continuous release planning in a large-scale scrum development organization at ericsson. In: Baumeister, H., Weber, B. (eds.) XP 2013. LNBIP, vol. 149, pp. 195–209. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38314-4_14
19. Heikkilä, V., Rautiainen, K., Jansen, S.: A revelatory case study on scaling agile release planning. In: Proceedings of the 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 289–296 (2010). https://doi.org/10.1109/seaa.2010.37
20. Karvonen, T., Behutiye, W., Oivo, M., Kuvaja, P.: Systematic literature review on the impacts of agile release engineering practices. Inf. Softw. Technol. **86**, 87–100 (2017). https://doi.org/10.1016/j.infsof.2017.01.009
21. Ameller, D., Farré, C., Franch, X., Rufian, G.: A survey on software release planning models. In: Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., Mikkonen, T. (eds.) PROFES 2016. LNCS, vol. 10027, pp. 48–65. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49094-6_4
22. Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S.B., Shafique, M.U.: A systematic review on strategic release planning models. Inf. Softw. Technol. **52**, 237–248 (2010). https://doi.org/10.1016/j.infsof.2009.11.006

23. Carlshamre, P.: Release planning in market-driven software product development: provoking an understanding. Requir. Eng. **7**(3), 139–151 (2002). https://doi.org/10.1007/s007660200010
24. Jantunen, S., Lehtola, L., Gause, D.C., Dumdum, U.R., Barnes, R.J.: The challenge of release planning. In: Proceedings of the Fifth International Workshop on Software Product Management, pp. 36–45 (2011). https://doi.org/10.1109/iwspm.2011.6046202
25. Benestad, H.C., Hannay, J.E: A comparison of model-based and judgment-based release planning in incremental software projects. In: Proceedings of the 33rd International Conference on Software Engineering, pp. 766–775 (2011). https://doi.org/10.1145/1985793.1985901
26. Spiegler, S.V., Heinecke, C., Wagner, S.: Leadership gap in agile teams: how teams and scrum masters mature. In: Kruchten, P., Fraser, S., Coallier, F. (eds.) XP 2019. LNBIP, vol. 355, pp. 37–52. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19034-7_3