IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2024, accepted 4 June 2024, date of publication 14 June 2024, date of current version 26 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3414614

RESEARCH ARTICLE

Evidence-Based Quality-Aware Agile Software Development Process: Design and Evaluation

PERTTI KARHAPÄÄ [1], WOUBSHET BEHUTIYE [1], PERTTI SEPPÄNEN [1], PILAR RODRÍGUEZ [1,2], MARKKU OIVO [1], XAVIER FRANCH [3], SILVERIO MARTÍNEZ-FERNÁNDEZ [3,4], (Member, IEEE), LIDIA LÓPEZ [3], MICHAŁ CHORAŚ [5,6], ALESSANDRA BAGNATO [7], SANJA AARAMAA [8], AND JARI PARTANEN [9]

[1]M3S Research Unit, Faculty of Information Technology and Electrical Engineering (ITEE), University of Oulu, 90014 Oulu, Finland
[2]Department of Languages, Computer Systems and Software Engineering, Faculty of Computer Science, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[3]Department of Service and Information System Engineering, UPC—BarcelonaTech, 08034 Barcelona, Spain
[4]Fraunhofer Institute for Experimental Software Engineering (IESE), 67663 Kaiserslautern, Germany
[5]ITTI Sp. z o.o., 61-612 Poznań, Poland
[6]Institute of Computer Science and Telecommunications, Bydgoszcz University of Science and Technology, 85-796 Bydgoszcz, Poland
[7]Softeam, 75016 Paris, France
[8]Nokia Oy, 90620 Oulu, Finland
[9]Bittium Wireless Ltd., 90590 Oulu, Finland

Corresponding author: Pertti Karhapää (pertti.karhapaa@oulu.fi)

This work was supported by the Q-Rapids project though the European Union's Horizon 2020 Research and Innovation Program under Grant 732253.

ABSTRACT Agile software development (ASD) aims to deliver high-quality software. However, software quality often receives less attention in ASD due to focus on functionality. Elicit requirements for, design, validate, implement, and evaluate an evidence-based quality-aware ASD process that organizations can use for better integration of quality management. Using design science research, we collaborated for over 1.5 years with four organizations applying ASD. Involving a total of 35 practitioners, we designed new evidence-based quality-aware ASD processes, which were validated through simulated scenarios, implemented, and evaluated by the practitioners. A general ASD process model, derived from the company-specific models, was created; it shows activities, artifacts, roles, and development phases where evidence can be used to improve software quality. The new process model contributed to higher quality awareness, improved bottleneck identification, release readiness, and quality issues prioritization in the studied companies. We also learned that companies must be mindful of the rules that determine the use of evidence. Integrating the use of quality related evidence into the ASD process can make it quality-aware, support software quality management, and enable timely reaction through fast feedback loops. The evidence-based quality-aware ASD process is applicable in any Scrum-like ASD process. Applicability in other forms of ASD, however, requires further research.

INDEX TERMS Agile software development process, evidence-based, software process improvement, quality-awareness, quality requirements.

I. INTRODUCTION

Managing software quality within agile software development (ASD) is an important industry concern, where the trade-offs between speed and quality can dictate the success or failure of software projects [1], [2]. Despite enhanced quality being a key reason for adopting ASD methods, only 45% of adopters report a positive impact on software quality [3]. Because ASD focuses on rapid delivery of functionality [4], quality often receives less

The associate editor coordinating the review of this manuscript and approving it for publication was Giuseppe Destefanis.

attention [5]. There is a need to better integrate quality requirements (QRs, also referred to as non-functional requirements that characterize software quality [6]) better into ASD.

One possible way to achieve this integration is to utilize data. A characteristic of ASD is that it produces a lot of data. Because of the iterative nature of ASD and focus on automatizing through tools (e.g., Gerrit, Jenkins, Jira) which produced data constantly [7], [8], the data can be used for decision-making [9] as companies are turning toward data analytics to become more data driven [10]. In fact, considerable research has been done during the past two decades regarding the utilization of data and measurement for quality management in ASD [11]. However, utilizing data and becoming a data-driven organization is not straightforward. A lack of understanding on how to integrate data into the software development process has been identified as one problem [12].

The goal of our study is to design an ASD process that integrates the management of QRs and software quality into the ASD process alongside with management of functionality where decisions regarding quality are based on evidence. By "decisions regarding quality," we mean decisions about both QRs and software quality in general (attributes like, reliability or efficiency). In our earlier study [13], [14], our focus was initially on management of QRs; however, we found out that practitioners preferred not to separate management of QRs and management of software quality. Therefore, in the remaining text, we use the term "quality" to mean both "QRs and software quality."

To achieve integration, we utilized design science research (DSR) [15] to design an evidence-based quality-aware ASD process represented as a general ASD model. The practices of the model are adaptable in various ASD processes, either in full or in part, contributing to improved software quality. To evaluate the utility of the model, it must be implemented in the real world. However, software development organizations in the real world follow their own specific ways of working, not a general description of ASD. Therefore, we studied the phenomenon of quality management in the real-world contexts of the software development organizations of our study and designed, validated, implemented, and evaluated the new processes. In doing so, we generalized these experiences to create the evidence-based quality-aware ASD process.

The present study is a continuation of our earlier work [13], [14] in the Q-Rapids project,[1] and we have worked together with the same four software development organizations as in [14]. For the current study, altogether, 35 practitioners participated in 17 initial interviews, seven design workshops, four validation sessions, five evaluation workshops, and 15 final interviews.

[1] http://q-rapids.eu

Our contributions are the following:

- The evidence-based quality-aware ASD process model that can be used by practitioners for improved quality management in any current ASD process.
- Requirements to consider when transforming an ASD process to an evidence-based quality-aware ASD process.

The background and related work can be found in the next section. Section III gives an overview of DSR and describes how it was applied in our study. Section IV introduces the problem context, the requirements elicited for the processes, and explains the design of the solution. Section V presents the solution. In section VI, we describe the implementation and evaluation of the new process. Section VII discusses the findings with implications for practitioners and researchers together with the validity considerations of the study. Finally, section VIII presents the conclusions.

II. BACKGROUND AND RELATED WORK

We divide this section into several parts in which we present findings from earlier research on the related areas and present the problem context and Q-Rapids project, in which the current study was conducted.

A. AGILE SOFTWARE DEVELOPMENT AND CHALLENGES WITH QUALITY

One of the core principles of ASD is to focus on "working software over comprehensive documentation" [4], that is, deliver working software to the customer soon and frequently. The iterative nature of ASD and incremental discovery of requirements enable this [16]. However, as a result focus tends to shift toward functionality rather than quality. Even though the practitioners of ASD are aware of the importance of quality, studies about QRs in ASD show that there is not enough time for properly considering QRs [17]. Other identified challenges include: the limited ability of ASD to handle QRs [18], limitations in testing QRs [19], neglecting QRs [20], lack of overall picture of QRs [19], customers overlook QRs [20], customers not being aware of important QRs [21], and limited experience of QR management in ASD teams [22], to name a few (full list in [13]).

B. DATA-DRIVEN SOFTWARE DEVELOPMENT AND SOFTWARE ANALYTICS

Because of the short iterations of delivering working software, practitioners need to make frequent and fast decisions [23]. These decisions tend to be based on experience, intuition, opinions, and various criteria [24]. Several other shortcomings in decision-making include the following: decisions are often based on a limited understanding of functionality; the quality measures of decisions are invisible to teams; decisions are made via poor communication and documentation methods; support for the automatic presentation of data is lacking; and decisions made during iterations may not be tracked and documented [25]. Because ASD produces large amounts of data, it has been proposed that there is great potential for data-driven decision-making in ASD context [26].

The emergence of artificial intelligence and machine learning has increased interest in the better utilization of evidence in decision-making. This is done through software analytics, the aim of which is to "empower software development individuals and teams to gain and share insight from their data to make better decisions" [10]. This insight includes learning from previous decisions to provide predicting capabilities [27]. Software analytics could automate the analysis of the vast amount of data and enable evidence-based support for the decision makers [28]. This approach ensures decisions are traceable back to their underlying data and reasons. Decision-making is becoming more data driven, and evidence based.

C. EVIDENCE-BASED DECISION-MAKING

Many studies have shown the potential of evidence-based decision-making [29]. Many of these studies tend to focus on specific aspects or stakeholders [30]. For example, Lou et al. [31] develop a system to aid developers in incident management, Cito [32] works on how to support developers in anticipating runtime problems with runtime information, and Stroulia et al. [33] provide managers with insights about team performance. Augustine et al. [34] focus on multiple stakeholders highlighting the importance of drill-down capabilities and visualization of a support system for data-driven decision-making. The work by Figalist et al. [35] presents a generic model for software analytics to provide insights for a broader range of stakeholders to support decision-making on operational, tactical, and strategic levels. Our focus, while considering multiple stakeholders, is on managing quality with the help of data in the ASD process. One of the challenges is the lack of understanding how data can be integrated into the software development process [12]. Our process model shows how the data can be utilized as evidence in the ASD process, that is, in what activities, by whom, for what decisions, and in what frequency.

D. THE PROBLEM CONTEXT

Already in an earlier study [14], We studied the problem context and validated the need for an improved, more inclusive process from the point of view of quality. The work carried out in the previous study is continued in these same organizations in the present study. In the earlier study, we found that, despite having notable differences in details of their development processes, on a higher level of abstraction, the studied organizations worked in a very similar manner. One organization managed development according to Scrum [36] "almost by the book." Another used a mix of Scrum and Kanban. An organization that developed software on a large scale applied Scrum practices in development on team level. One organization did not follow any predefined agile method, but Scrum like practices were still identified. The organizations are described in detail in Section IV.

In our previous study [14], we identified a set of challenges in the management of quality, many of which had been identified in the literature. These include prioritizing the functional requirements over QRs; difficulties in including QRs in user stories; unclear QR specifications; delayed feedback on QRs; uncertainty in measurement parameters; and the generally fuzzier nature of QRs compared with their functional counterparts. Although not all organizations faced every challenge and the extent varied among them, practitioners were aware of these issues. Recognizing the significance of QRs and software quality, they sought to improve proactive quality management rather than reacting to issues. As the organizations were already using development data to their advantage, these data were also used to aid in the management of quality. However, QRs still seemed somewhat detached, and we identified opportunities to utilize evidence even further to integrate management of quality better into their ASD processes.

E. Q-RAPIDS

The work of the current study has been done in the Q-Rapids project [37]. As the goal of Q-Rapids was to develop a data-driven solution for managing QRs and software quality in ASD, it provided us with an opportunity to study the phenomenon. The solution runs on a quality model based on the quality needs of the company and helps the development organization to focus on the quality perspective [8]. ISO 8042 and ISO/IEC 25010 standards [38], [39] were used as a starting point for defining a common understanding of "quality" and to find the quality aspects important for the organizations. Later, goal question metric approach (GQM) [40] was used to find the factors, metrics, and data to make quality measurable in a quality model that enabled continuous and reliable assessment of software quality.

Fig. 1 presents the idea of how Q-Rapids helps manage quality. Data can be collected when the software is developed, when it is running, and when users are using it. The data are analyzed according to the quality model and presented on a dashboard as evidence. Decision makers and developers alike may utilize the evidence in management of QRs and software quality. The outcome of the present study shows in detail how evidence provided by a solution such as Q-Rapids can be integrated into the ASD process for better management of quality, thus making the process more quality aware and evidence based.

III. RESEARCH METHOD OVERVIEW

As our aim was to design a process, we chose to follow design science research (DSR) [15]. DSR is the design and study of artifacts in context with the goal of improving something in the context when interacting with the artifact. The design is informed by the knowledge context and social context. The social context provides design problems, constraints,

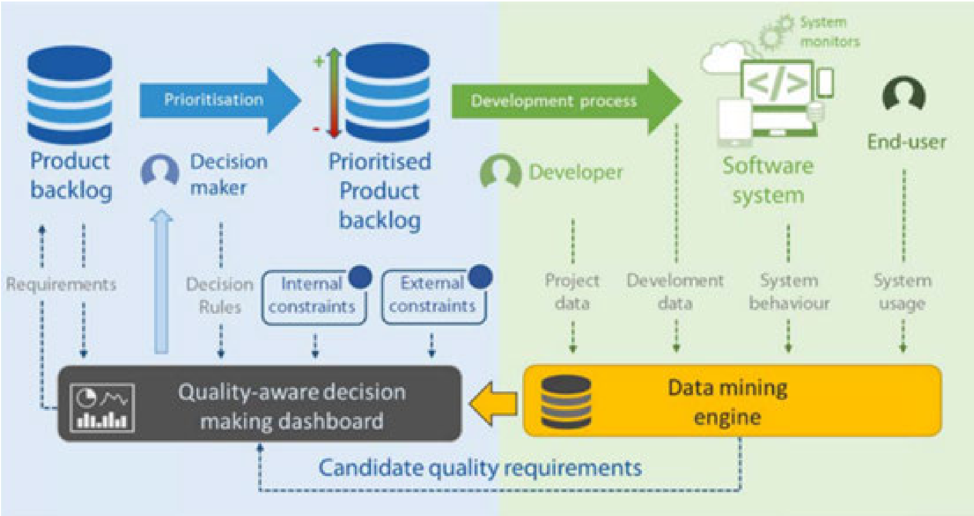


FIGURE 1. The Q-Rapids solution integrated into ASD to help manage QRs and software quality (from [37]).

and requirements for the solution and the knowledge context related previous knowledge. The design science project iterates between the design and investigation of the artifact. Wieringa [15] refers to this as the design cycle. In this cycle, the problem is investigated, the treatment is designed, and it is validated. Here, validation means predicting how the artifact would interact with its context if implemented. Knowledge questions about the artifact can provide answers to further improve the design in a new cycle. The evaluation of the artifact is part of a larger engineering cycle after it has been implemented in the real world. In the implementation evaluation, we can empirically investigate whether the requirements for the artifact have been fulfilled. This can feed back into a second engineering cycle. In the end, the design provides solutions for the social context and new knowledge for the knowledge context.

The artifact in the present study is the new process represented as a model that has the goal of improving management of software quality in ASD by increasing use of evidence in quality-related decisions and making ASD more quality aware. The knowledge context consists of the scientific literature about ASD processes, management of QRs and software quality in ASD, evidence-based decisions-making in ASD, and related existing problem-solving knowledge. The social context consists of the organizations of our study applying ASD methods.

In our study, we executed one iteration of the design cycle, as described above, in each organization. However, we applied two iterations within the design activity before validation. The design activity resulted in organization-specific process models. In validation, we used simulated scenarios of decision-making based on the designed process models. Expert opinion was used to justify that the artifacts would contribute toward filling the requirements for the processes and goals of the project. Implementation and evaluation were conducted as one integrated and iterative activity over a longer period. Because implementation would affect several areas of the ASD process, it was done gradually by implementing smaller changes at a time to avoid disrupting software production. The results of the iterations were used to further improve and update the process models. The general model was designed along these activities by abstracting from the organization-specific models. The DSR process is presented in Fig. 2, and Table 1 gives details about timing and how many participants of which organization participated in each activity.

The list below gives a brief description of the activities. The following section describes them in detail.

- *Problem identification* involved studying the state of the art in the form of a literature review [13] and study of practice in a multiple case study [14]. The collected data of this activity consisted of existing quality-related management practices, challenges, solution proposals, and general requirements for the new process. The relevant findings of these studies are reported in Sections I and II, and the companies are presented in Section IV-A.
- *Definition of objectives* were partly drawn from the previous two studies and partly from the present study. We collected specific process requirements by interviewing practitioners; that is, the data consisted of interview data that then resulted in the specific requirements. In this activity, we also formed an initial idea about what should be included in the final artifact, that is, the general evidence-based quality-aware ASD process. The requirements are presented in Section IV-B.
- In *design and development*, we studied how evidence can be utilized for quality management together with the practitioners in the context of their own ASD processes. As our focus was on the process, we examined the aspects of who was doing what, where in the process, and for what reason. We worked together with the practitioners in two focus group sessions [41] separately in each company. The data consisted of the activities

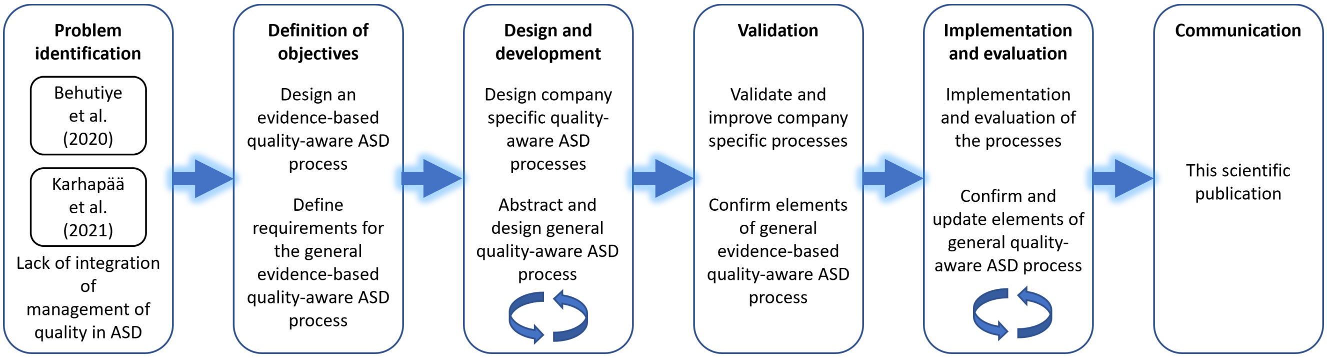


FIGURE 2. Design science research process followed in the study.

TABLE 1. Cycles of design, validation, and evaluation and number of participating practitioners.

Phase	Activity	Time	# of participants and focus group session length in minutes			
			Case A	Case B	Case C	Case D
Objectives	Interviews	March–May 2018	4	4	9	1
Design	Design focus group I	April	4 (124)	3 (123)	3 (117)	2 (162)
	Design focus group II	June	3 (143)	4 (198)	3 (189)	-
Validation	Validation	October	4 (90)	8 (90)	11 (90)	2 (90)
Implementation and evaluation	Eval focus group I	February 2019	5 (95)	5 (78)	-	-
	Eval focus group II	April	4 (90)	5 (83)	-	-
	Eval focus group III	May	3 (92)	3 (67)	-	-
	Interviews	June–October	4	4	4	3

of the processes, the flow of activities and artifacts in the processes, and proposals of improved quality management. The goal was to design the new quality-aware processes for each organization. The design started with the software development organizations' current way of working (later referred to as "as-is processes") and resulted in process models specific to each organization. The details of the models were used to add components to the general process model. This is described in Section IV-C1.

- *Validation* of the process models was done in validation sessions separately in all organizations. The collected data consisted of practitioners' perceptions regarding the feasibility of the new processes and data to further improve the processes. The goal was for the practitioners to inspect the designed specific process models and assess whether they were implementable before attempting actual implementation. Through these validation sessions, we confirmed what should be included in the general model as well. A detailed description is presented in Section IV-D, while the solution is presented in Section V.
- *Implementation and evaluation* of the quality-aware processes were done after the validation sessions. A big-bang approach was avoided in implementation because the development organizations could not risk disrupting the development of real products. Instead, the organizations took the new process into use in an explorative manner gradually, focusing on the activities that made sense at the time. Evaluation was done during this implementation. Evaluation was conducted in three different workshops and additional follow-up interviews. The data consisted of usages reported by the practitioners through an online form and interview data. Data included who did what in what activity for what reason and what was the outcome. Details were confirmed and adjusted in the evaluation sessions. These data and those from the final interviews answered whether the requirements for the new processes had been met. The goal was to concretely evaluate whether the requirements of the processes had been fulfilled and whether they contributed to the goals of increased quality awareness and utilization of evidence in quality-related decisions. The details are described in Section VI.
- Finally, *Communication* refers to this publication.

IV. PROBLEM CONTEXT AND OBJECTIVES

In this section, we describe the companies of our study and identify requirements for the new evidence-based quality-aware process.

A. FOUR ORGANIZATIONS INTEGRATING QRS INTO ASD

The problem context consists of four software development organizations, that is, the industrial partners of Q-Rapids project. The organizations are summarized in Table 2. We have used Petersen and Wohlin [42] to characterize the organizations according to different context facets. The organizations differ in the domain of the product and product type, size and structure of the development organizations, and target markets. They all develop software utilizing agile methods, while also wanting to improve the management of QRs and software quality of the product in general.

TABLE 2. Companies and context adapted from Petersen and Wohlin [42].

Context facet	Context element	Company A	Company B	Company C	Company D
Product	**Name**	Product A	Product B	Several products	Product D
	Domain	Software development	Telecommunications	Telecommunications	Healthcare
	Product type	Commercial software modeling tool	Commercial embedded secure communication device	Common platform software for telecommunications	Contract-based custom software solution
	Maturity of product	Long-lived mature product	New product	Long-lived mature product	New product
	Customization	General product, same for all customers	Product for specific niche market, same for all customers	General products	Product tailored to different customers
	Criticality of product for organization	Critical (flagship)	Critical (flagship)	Important (internal customers)	Important (external customers)
Process	**Software development process (Agile, plan-driven, ...)**	Agile based, continuous, and incremental development of features	Scrum-based, continuous, and incremental development of features	Plan-driven and Scrum-based development of features	Mixture of Scrum and Kanban for incremental development of custom software solutions
	Start of agile adoption	2006	2007	2009 to 2011	2012
	Distributed development	No	Yes (nationally)	Yes (globally)	No
	Developers in development of the product or component	22	50	500	Up to 9
	Team size	Up to 10	6 to 8	9	Up to 9
Organization	**Size (number of employees)**	>1000	~1000	>100 000	~100
	Hierarchical organizational model	No	No	Yes	No
	Organizational unit involved in the study	Development team from one location developing one component of the product	Development team in company headquarters developing the product	One local development team from one business line developing a component for that business line	Development team from one location
Market	**Type of customer**	Software developers	National organizations	On component level—internal customers using the same platform for different communication technology services On system level—nations, large organizations	Institutions, business organizations
	Setting	Business-to-user	Niche market	Business-to-business (B2B)	Contractual development, B2B
	Constraints	Competitive market of similar tools	High security and safety, confidentiality, standards and regulations	Servicing multiple networking standards, standards and regulations	Competitive market
	Access to customers or users	Direct access to customers and users	Limited access to end users	Access to internal customers, limited access to end users	Direct access to customers and end users. Very frequent contacts with customers

Company A is a middle-sized software house. The organization develops a modeling tool for global markets targeted for professional use. Development is based on agile practices following the principles of Scrum in a company-specific adaptation and on bi-annual releases of new functionality. The practitioners in this organization found QRs and software quality elusive many times, challenging to know what to measure, and did not have any prescribed method for managing them. They felt the need to manage them better to plan releases better and avoid surprises in the late stage of development.

Company B is a middle-sized organization developing product families of complex embedded devices utilizing Scrum almost by the book. The products are offered to institutional customers, who set both regulatory and customer-specific functional and quality requirements on the products, including the embedded software. Some of the products are safety and security critical and, so Company B needs to have well-documented processes, which include the management of QRs. With these products, there is a need for full traceability of every decision from customer request to final product.

Company C is a very large, international organization developing large systems of complex embedded devices. The products are targeted to institutional customers, who set both regulatory and customer-specific functional and quality requirements for the products. The organization has a long tradition of continuous process development. The actual process status differs, however, from the other organizations because there are several legacy processes in use in different parts of the organization, which is divided by both geographical locations and technical disciplines. The organization was in the process of transforming development activities according to SAFe (scaled agile framework) [43]. The practitioners felt that they could improve the process with respect to some decision points regarding QRs because, sometimes, problems may arise very late in development.

Company D is a smaller software house (compared with previous organizations) developing customer-specific software solutions on a contract basis. The development projects are typically short and separate from each other. The functional and quality requirements of the solutions are in principle set directly by the customers. The organization has a well-established Scrum-type software development process. The activities for functionality planning and setting of functional and quality requirements are conducted together with the customers, as well as the final validation of the systems to be delivered. The organization wanted to find a way for more precise control of the development process and improve root cause analysis.

All the organizations develop several products or work on several projects. Different products dictate different kinds of requirements and might have different types of quality concerns. Even though the study focused on the development of a single product per organization or the specifics of the individual organizations, we sought commonalities across the organizations. During the study, some common quality concerns were found, including maintainability, reliability, functional suitability, and productivity [44]. By generalizing and focusing on commonalities we aimed to demonstrate the applicability of the findings in varying ASD contexts.

From these organizations, we involved 35 practitioners. The participants, their roles, and their experience are included in Table 3 together with data on who participated in which activity. Our aim was to get different roles involved to collect all relevant details of the activities relevant from a quality perspective in software development. The practitioners were selected by the companies based on their own evaluation of who had the best understanding and knowledge about their way of working. A contact person, acting as a champion for the study within each company, mediated our requirements regarding practitioners.

B. REQUIREMENTS FOR THE NEW PROCESS

The goals of this activity were twofold. First, we aimed to verify our understanding of the development organizations' as-is processes from our previous study [14] and, second, collect requirements for the new process. The collected requirements served as a starting point for the design of the new process.

Based on our earlier study of the literature [13] and in line with the objectives of the Q-Rapids project [37], we formulated three main objectives for the process: **Quality awareness** - Because ASD is prone to focus more on functionality, the new process should enable reaction to quality issues; **Evidence-based decision-making** - Because evidence is abundant in ASD, it could be utilized better in the quality-related decision-making process; and **Agility** - As one of the key drivers of agile adoption is the adaptability to change, the new process should not be too rigid.

To fulfill these objectives, we needed more detailed requirements from the practitioners. The two first authors conducted 17 individual semi-structured interviews with the practitioners (Interview I column in Table 3). First, we confirmed our current understanding of their as-is software development processes and quality management practices. Then, we focused on utilization of evidence in the management of quality.

All interviews were recorded, and the recordings were transcribed by professionals. Process-related data were extracted into an Excel file and categorized under the three objectives. The first author analyzed and summarized data from three companies, while the second author did the same for one company. We compared and reviewed each other's results since both authors were present in all interviews. Later, the practitioners also reviewed the results when designing of the new processes started, setting the foundation for organization-specific models and the goals for the evidence-based quality-aware ASD processes.

To avoid biasing interviewees, the interviews focused on the specific elements of their existing ASD processes rather than the three objectives. This allowed the practitioners to focus on the details of their own ASD processes. For instance, when organizations were considering the adoption of the Q-Rapids solution, our questions centered on its usage rather than evidence utilization in general. We asked about how they were using it or were planning to use it, challenges preventing usage, and any possible observed or expected benefits or challenges. In our analysis, we generalized from usage of Q-Rapids solution to usage of evidence in general. For example, if they preferred using Q-Rapids outputs in a specific development activity, we inferred that similar evidence from other tools could be used for the same purpose in that activity.

The elicited requirements are listed at the end of this section. Here, we will show a few examples of the data collected through the interviews. Especially in the larger organizations, several interviewees expressed the importance of being able to react to quality issues in a timely fashion. They explained that, sometimes, quality issues arise very late in development. Experience showed that addressing quality issues after-the-fact can be costly. From this, we elicited the requirement that the process should highlight and enable fast

TABLE 3. Practitioners participating in the study (Exp = Experience, D = Design focus group, E = Evaluation focus group, SW = Software, dev = Developer).

Company	Participant	Role	Years of exp in company	Interview I (minutes)	D1	D2	Validation	E1	E2	E3	Interview II (minutes)
A	P1	SW architect, SW dev	13	52	X	X		X	X	X	39
	P2	Project manager	8	66	X	X	X	X	X	X	60
	P3	R&D leader	11		X						
	P4	Project manager	12				X	X	X	X	47
	P5	Executive manager	32	46	X	X	X	X	X		40
	P6	R&D engineer, SW dev	1					X	X		
B	P7	DevOps specialist	5	65	X	X	X	X	X	X	55
	P8	Tester	8	68	X	X	X	X	X		
	P9	Process manager	17	62	X	X	X	X			38
	P10	Project manager	19	51	X	X	X	X	X		
	P11	Project manager	0.5				X	X	X	X	30
	P12	Team leader	2.5				X				
	P13	Business owner	1				X				
	P14	Build manager	1								42
C	P15	Project manager	3				X				40
	P16	Quality manager	34				X				43
	P17	Process expert	2.5	101	X	X	X				83
	P18	Quality manager	25	45	X	X					
	P19	SW engineer, SW dev	8	54	X	X					38
	P20	Quality Specialist	24	61							
	P21	Fault manager	33	37							
	P22	SW engineer	19				X				
	P23	Tester, integrator	9				X				
	P24	Technical lead in testing	20+				X				
	P25	Project manager	3				X				
	P26	Test architect	25				X				
	P27	Project manager	2				X				
	P28	SW dev	18	50			X				
	P29	Technical leader	10	48							
	P30	Fault manager	20	65							
	P31	Quality specialist	19	58							
D	P32	Product owner	12	85	X		X				43
	P33	Product owner	0.5				X				
	P34	Product owner	2.5								43
	P35	Project manager	N/A								43

reaction on quality issues. This means that quality issues should be available in activities where they are relevant and where the practitioners can react to them as soon as possible.

Some interviewees anticipated that access to more accurate and timely quality information would enable more precise estimation, planning, and prioritization. The practitioners expressed their need for supportive evidence in decision-making. Although not explicitly an activity in process models, this requirement—should support experts in decision-making with timely evidence—influences the flow of artifacts and should be detailed in activity descriptions. Key activities include planning and reviewing development results. In larger organizations, access to all development data is not always granted to everyone, implying a consideration of different access to different roles. However, to avoid overemphasis on high-level indicators without considering the underlying details, it is beneficial to provide all roles access to comprehensive evidence on quality issues.

The interviewees in all organizations expressed concerns related to the introduction of new processes. There is a risk that the new process would interfere with their current agile way of working and may slow down development. In addition, after the introduction of the example support system, Q-Rapids, the practitioners were concerned that the system and the new process would be too "rigid." That is, a new system should not force them to work in a different way. From this, we arrived at the requirements of not introducing major changes to their current ways of working and that the new process should be lightweight and easy to adapt. In addition, as the largest organization was in the process of organizing development according to SAFe, they expressed that the new process needs to be compatible with SAFe. For the process model, this meant that levels of SAFe should be considered.

Based on the analysis of the interviews, we formulated the following requirements for the new process according to the three objectives defined earlier:

- **Quality awareness**.
  - No compromising on quality aspects.
  - Should highlight and enable fast reaction on quality issues.
- **Evidence-based decision-making**.
  - Should support experts in decision-making with timely evidence.

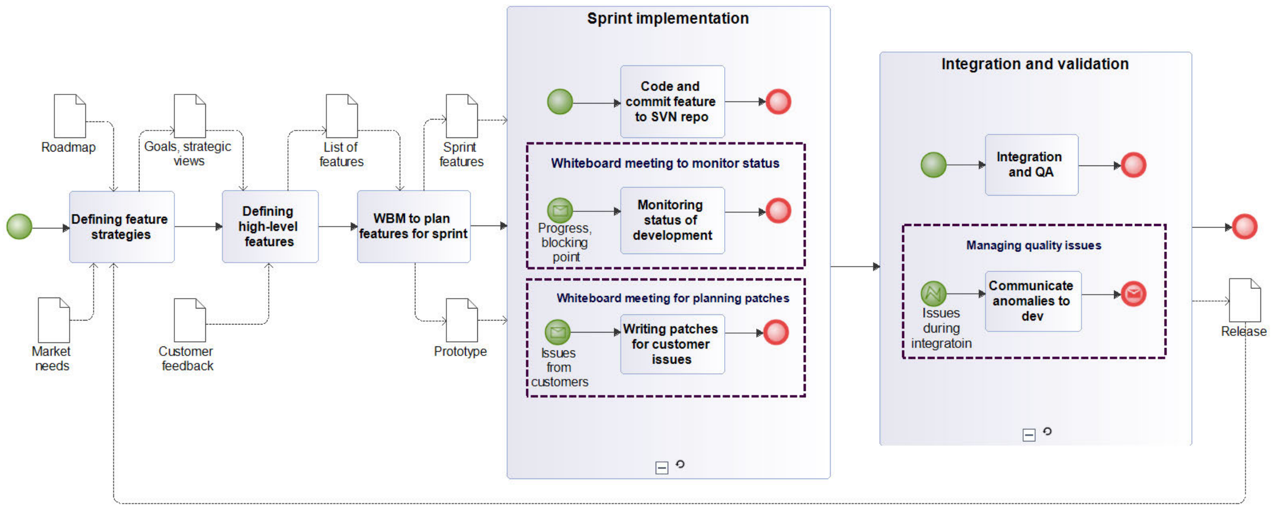


FIGURE 3. As-is software development model of Company A. (WBM - whiteboard meeting).

- Should support different roles on different levels in the organization with relevant evidence for that level and role.
- Should grant access to all relevant evidence regarding a quality issue, despite the level of role.

- **Agility**.
  - Should not impede agility.
  - Should not slow down development.
  - Need to be lightweight and easy to adapt.
  - Should not require major changes to current ways of working.
  - Need to be compatible with SAFe.

C. DESIGN

The goal of this activity was to design and validate the new organization-specific, evidence-based, quality-aware ASD processes. We designed the processes together with the practitioners as models specific to each development organization. These models were to be used in validation of the new processes before implementing changes in the real processes. At the same time, we, the researchers, designed the general evidence-based quality-aware ASD process with the help of the details from the organization-specific models.

The new process design was conducted in two workshops in all organizations except for Company D, in which we conducted only one workshop. The practitioners in Company D already had a clear understanding of how and for what they wanted to utilize evidence. The practitioners participating in the workshops are presented in Table 3 in columns D1 and D2. The population sample included developers, a tester, a DevOps specialist, a process expert, and managers with various areas of expertise. A criterion was that the participants have knowledge about the software development process and management of QRs or software quality. The two first authors participated and moderated all of the workshops.

In the first part of the first workshops, we confirmed and corrected details of the as-is process models of the organizations' software development, here with a special focus on the management of quality. These models were based on the previous case study [14] and findings of Interview I.

Based on relevant literature, our knowledge about the Q-Rapids project, and our understanding of organizations' development processes, we proposed where in the process QRs and evidence regarding software quality could be utilized and how. This was done in the second workshop. The practitioners, with their expertise about their own processes, provided their input by confirming, proposing changes, declining proposed changes, or proposing new tasks in the activities or even new activities. Finally, the practitioners voted on those activities that they saw as bringing the most benefit for improved quality management. The details of those activities were collected in a data collection form and later integrated into the process descriptions and utilized for the design of the general model. The second workshop was conducted in the same way, with the results of the first workshop serving as the starting point.

1) INTEGRATING USE OF QUALITY-RELATED EVIDENCE

To keep the text shorter, we present here only the as-is model and final outcome before evaluation. Fig. 3 shows a simplified process model of Company A, which we use as an example. The model focuses on those activities where decisions regarding quality take place together with the relevant artifacts from a quality perspective.

The development process starts with the executive board of the company *defining feature strategies* based on a company *roadmap* and analysis of the *market needs*. These and *customer feedback* are used as input to define features on a higher level for the next six-month release plan. Quality aspects of the product are an important consideration when deciding on the future direction of the development of

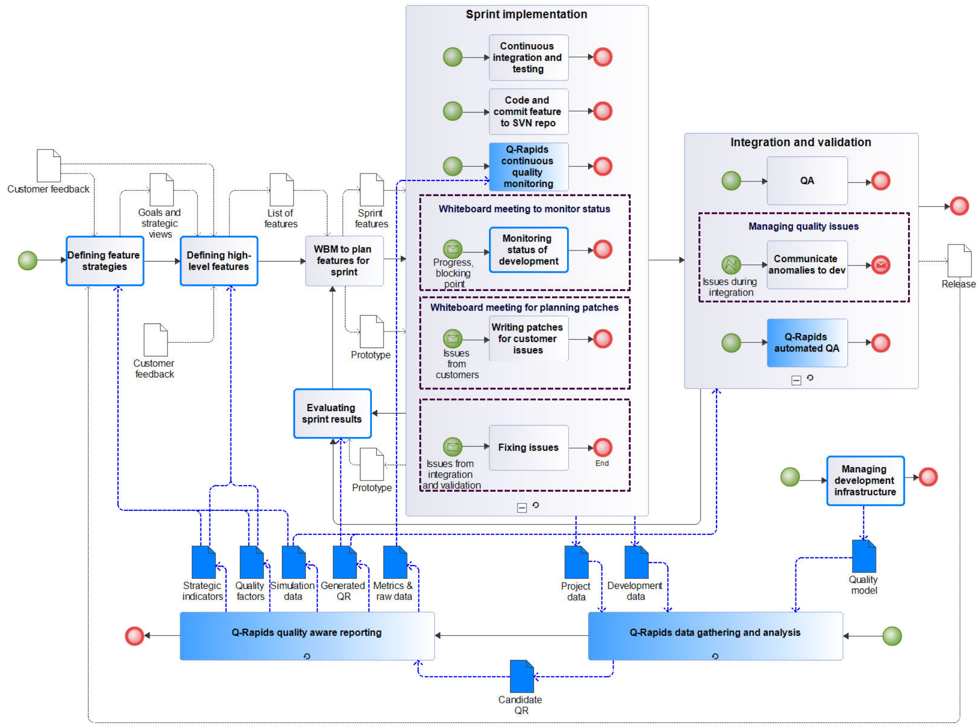


FIGURE 4. New evidence-based quality-aware ASD process in Company A.

the product. In particular, customer feedback may contain many quality-related items. The *list of features* is inspected during a whiteboard meeting (WBM), where it is decided what features should be implemented in the next sprint. Quality-related concerns are included as separate items in the backlog if it is clear how the quality can be measured. After *integration and validation*, which includes quality targets for the product, the output is a new *release* that in turn, is used as input in the planning of the next release.

Fig. 4 shows the result of the work conducted together with the practitioners in Company A. The activities that were deemed to be the best candidates for making the process quality aware by utilizing evidence are listed below. In the figure, new activities have blue gradient filling, new artifacts have blue filling, and previously existing activities that can benefit from utilizing the new evidence have blue borders.

- *Defining feature strategies* - It was deemed important that the integration of quality starts as early as possible when planning new features. The new evidence makes sure that quality aspects are not overlooked in this early phase.
- *Defining high-level features* - Following previous activity, the evidence related to quality needs to be taken into consideration when deciding about features that will be implemented during the next six-month release. Both POs and managers are involved in this task.
- *Evaluating sprint results* - An activity that was missing from the simplified process model but was deemed important to allow for faster feedback on quality issues provided by the evidence. POs, testers, and developers may reflect on the work done from a quality perspective.
- *Monitoring status of development* - An activity that is triggered when something is blocking development and requires a meeting to inspect the status more closely. The new evidence would indicate if any aspect of quality is decreasing to such an extent that immediate reaction is necessary. The severity of the triggering event is usually of such a degree that it may require the attention of managers, POs, and developers alike.
- *Managing development infrastructure* - An activity that was existing, although not explicit, in the example organization, was added to the model. It includes the management of the complete development infrastructure, and in this case, the development organization had clear roles for this. As with any other tool, the Q-Rapids solution needs management as well.
- *Q-Rapids continuous quality monitoring* - A new activity that practitioners deemed a task belonging inside existing activities, in which the roles would depend on those activities. In this task, the relevant role would monitor the status of software quality by monitoring the indicators, factors, and metrics relevant for that role. It was decided that this activity would be included as an "extra" activity because the exact contents were not perfectly clear yet at this stage.
- *Q-Rapids automated QA* - A quality assurance activity. Although the name implies that it does not include a human role, it was decided to be included in the process

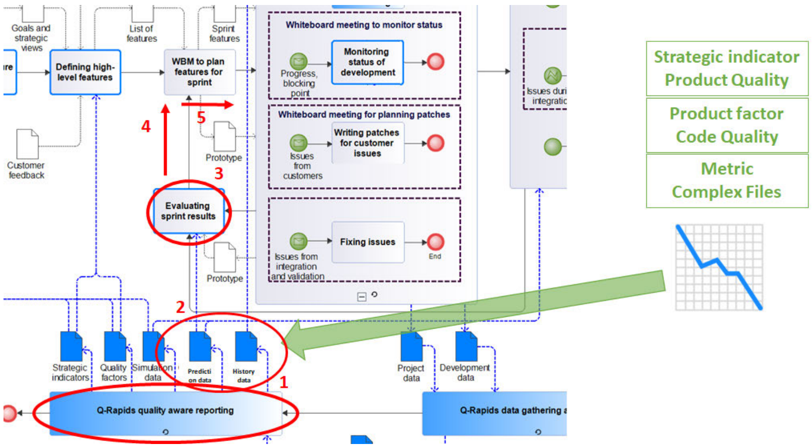


FIGURE 5. Simulation scenario 1 in the validation session in the example organization, Company A.

model to highlight the role of the support system, especially in quality assurance.

The quality management support system was added as two activities at the bottom of the figure: *Q-Rapids data gathering and analysis* takes input from development and processes it to produce the evidence, and *Q-Rapids quality aware reporting* provides access to the evidence and different features of the support system. The inputs *Product data* and *Development data* encapsulate the different raw data produced during development that are required by the *Quality model*. These are included as outputs of the general activity *Sprint implementation*. The *Quality model* is also an input, from *Managing development infrastructure*, because the support system needs it to know how to process data. Moreover, the process model was updated with the artifact *Prototype*, a working prototype that will eventually be released as input from *Sprint implementation* to *Evaluating sprint results*. This was missing in the initial process model.

Other changes compared with the as-is model include *Continuous integration and testing* and *Fixing issues*, an activity triggered by faults in, for example, testing. Due to time limitations, we did not work on the details of potential use of evidence in these activities; they were included as possible activities relevant for managing quality and to be worked on in the future. "Integration," which came from *Integration and QA*, was dropped because integration happens in development. The artifacts of *Company roadmap* and *Market needs* were removed because their scope was of too high a level and they did not provide any meaningful input to be utilized currently.

The results in the case of the other development organizations were similar. Quality should be taken into consideration early on, and evidence would enable taking quality properly into account when prioritizing the content of a release. Likewise, evidence would be critical for including quality properly in the prioritization of backlog items for the sprints. Monitoring the status of quality on a daily basis was deemed a necessary practice to enable fast reaction on "slipping" quality. These results served as takeaways for the general quality-aware process. These results also meant minimal changes to the current development processes. The activities or flow of activities would not need to be changed but only changes of the tasks within the activities.

D. VALIDATION

The goal of the validation was to have practitioners assess whether the designed company-specific processes would be implementable before evaluation in the real world. That is, the practitioners assessed whether the proposed changes in the process would fit into their ASD processes without compromising agility or speed of development while being more inclusive from a quality perspective.

Two scenarios were created that would be run as simulations with the help of the process models in the validation sessions. The scenarios were kept the same for all organizations except for changes in details according to their respective individual processes. The scenarios are presented below. We devised a script to follow in the validation sessions to ensure equal treatment among different validation sessions and reduced researcher bias. Henceforth, we refer to the simulated scenarios as simulations.

The validation sessions were conducted separately in the four organizations and were moderated by different researchers. The three first authors moderated the validation sessions in Company B and Company C, the seventh author moderated the validation session in Company D, and the eighth author moderated the validation session in Company A. The population sample consisted of managers, product owners (POs), and developers. The participants can be found in Table 3 in the column Validation.

The simulations focused on the diverse workflows of the processes with the goal of analyzing whether the new processes satisfy the agility requirement. The focus in the workflows was on activities identified when designing the new processes. This would indicate whether the requirement of quality awareness has been fulfilled. The simulations

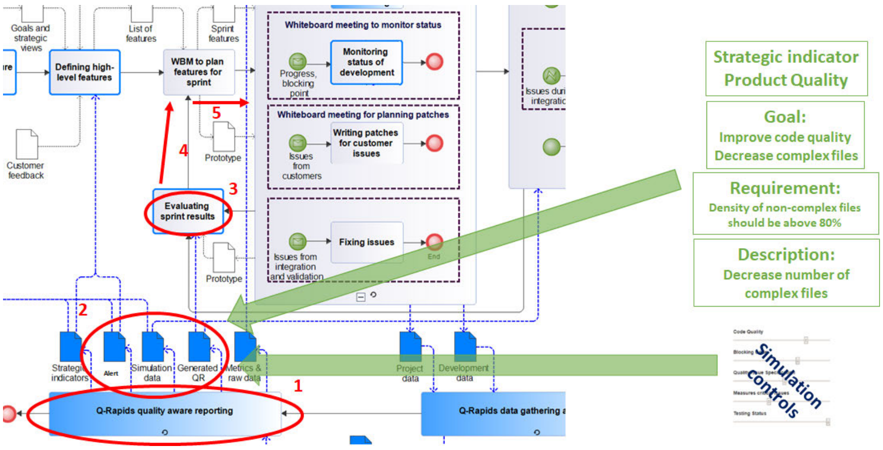


FIGURE 6. Simulation scenario 2 in the validation session in the example organization, Company A.

focused also on the usage of the different evidence provided by the example support system in decision-making. This would indicate whether the new processes could satisfy the requirement of being evidence based. If all these requirements are fulfilled, the new processes would contribute toward better integration of quality.

The workflows consisted of the activities in simulations *"Follow the process model to monitor software quality"* and *"Follow the process model to inspect a generated QR and simulation of quality."* We provide the simulations here as presented in the example organization, Company A.

In the first simulation depicted in Fig. 5, a PO monitors quality during a sprint using a dashboard (1). The PO notes a declining trend in the *Product quality* strategic indicator (2) and, upon further analysis, identifies a concerning decrease in the *Code quality* product factor and *Complex files* metric (2). Utilizing the prediction feature indicates these trends will continue if unaddressed (2). Consequently, the PO decides to address the declining product quality in the next *Evaluating sprint results* activity (3). A decision is made to inform developers about the situation, urging them to reduce the number of complex files and consider this in their implementation (4 and 5).

In the second simulation, in Fig. 6, a PO gets an alert saying that the *Product quality* strategic indicator is out of the accepted range and requires attention (1 in Fig. 6). The support system proposes a QR to be included into the backlog to mitigate the situation (2). The QR follows a predefined template and states the following:

- Goal: Improve code quality, increase the density of non-complex files
- Requirement: The density of non-complex files should be above 80%
- Description: This pattern expresses the need of decreasing the number of complex files

The PO knows that there are other factors contributing to product quality than code quality and wants to explore the option of adjusting another factor (2). The PO utilizes the simulation feature before deciding whether to include or not the proposed QR. If included or addressed in any other way, it might require the PO to update certain acceptance criteria or create a new task. Hence, in the process, it would mean that the PO evaluates the situation in the task of *Evaluating sprint results* (3). Following this, the content of the sprint is planned before resuming implementation (4 and 5).

The QR in the simulation is made up and may not be a realistic one. The purpose was not to focus on the details of the QR but rather on the process model to find whether the workflow was realistic.

In both simulations, the participants inspected the process model and its description in detail by answering the following questions:

- Is the flow of activities appropriate and complete?
  - If not, why?
- In your opinion, is the workflow agile enough?
  - If not, why?
- In your opinion, is the workflow rapid enough?
  - If not, why?
- In your opinion, does the workflow include QRs better?
  - If not, why?

All answers were collected with a data collection form and were used to further modify and improve the process models. Comments were also allowed for "Yes" answers. The recorded number of yes and no answers gave a general overview of the practitioners' perceptions regarding whether the proposed process would bring any improvement. Elaborations for no answers were important for refining the processes before implementation and evaluation, ensuring they did not impede software development. The validation results, which are summarized in Table 4, show the number of participants with positive versus negative perceptions. This includes assessments of the process's suitability for the organization, its impact on the agility of QR management, its effect on the speed of QR management, and the integration of QRs into the process.

Appropriateness: Most practitioners found the process suitable for their existing workflows. Only in Company D

TABLE 4. Summary of validation replies (Sim = Simulation, Company B had five participants, but one of them took part in simulation 1 only and another participant in simulation 2 only).

	Company A		Company B		Company C		Company D	
	Sim 1 Yes/No	Sim 2 Yes/No	Sim 1 Yes/No	Sim 2 Yes/No	Sim 1 Yes/No	Sim 2 Yes/No	Sim 1 Yes/No	Sim 2 Yes/No
Appropriateness	3/0	2/1	2/2	4/0	6/2	4/4	1/1	0/2
Agility	3/0	3/0	4/0	4/0	3/5	6/2	2/0	1/1
Speed	3/0	2/1	2/2	3/1	0/8	2/6	2/0	1/1
Inclusion of QR	1/2	2/1	2/2	3/1	5/3	4/4	2/0	2/0

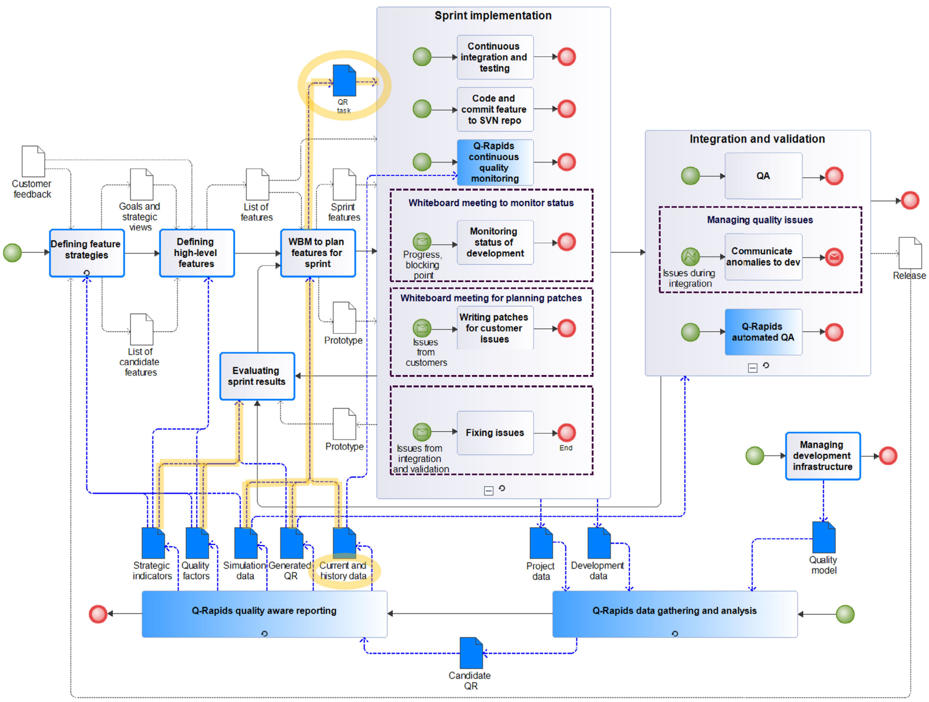


FIGURE 7. Modified model of the evidence-based quality-aware ASD process adopted in Company A.

did both participants feel that the workflow of simulation 2 did not fit well. The written feedback revealed that it was due to the details of the example QR rather than the process. Similar feedback came from Company A; while agreeing on the process's appropriateness, they noted that a new QR cannot be an input to implementation directly. Instead, the inclusion of the QR should be decided in the *Whiteboard meeting (WBM) to plan features for sprint*, where a task is derived from the QR. Thus, the *QR* artifact was linked to the WBM activity, and a *QR* task artifact was added as the output of that meeting and input for *Sprint implementation*. We updated the process model with the required changes, which are presented in Fig. 7 and are highlighted in yellow.

Agility: The proposed workflow was considered agile by most organizations, except Company C, where a majority in simulation 1 viewed it as less agile. In Company C, certain quality-related decisions were made in activities equivalent to those of program or large solution levels in SAFe, in less frequent activities. The practitioners suggested delegating some decisions to development teams to enhance agility. This was particularly evident in the feedback about the speed of development, where making quality decisions in infrequent meetings was seen as inefficient compared with handling them in daily activities. Consequently, the process for Company C and the general process were refined based on this feedback.

Speed: As noted above, the score for speed was low in the case of Company C. Other practitioners deemed that the process could increase speed, but it could slow down development initially if the number of quality issues is high.

Inclusion of QRs: Most of the practitioners found that QRs and quality in general would be better integrated in the proposed workflow.

The details from the process in the example organization as well as details from the other organizations were added into the general model, which is presented in the next section.

V. THE SOLUTION—EVIDENCE-BASED QUALITY-AWARE ASD PROCESS

The solution is an enhanced general ASD process model that integrates software quality artifacts and decisions, including QRs, metrics, factors, and indicators, into ASD. As the

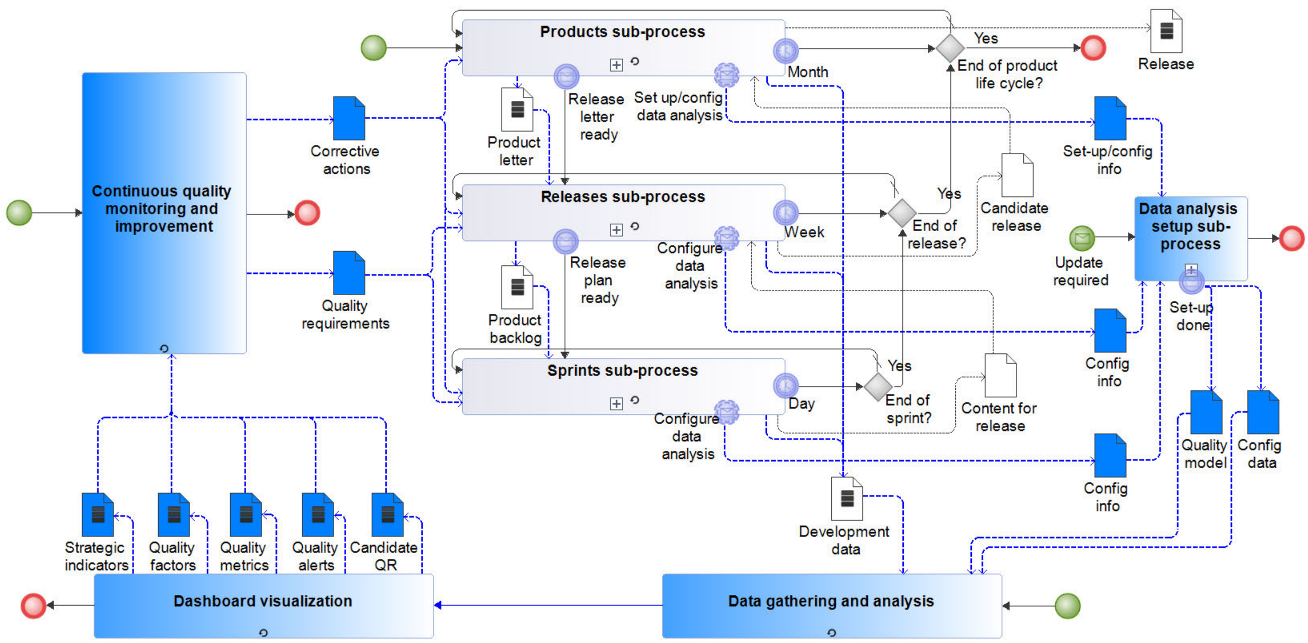


FIGURE 8. **General evidence-based quality-aware ASD process overview.**

largest company required the process to be compatible with SAFe [43], the model uses three levels of abstraction beneath the general overview. This did not conflict with the other organizations. However, SAFe is not a requirement for adopting any or all of the details presented in the model. The details are merely represented at different levels of abstraction, in which team-level details can be found in the lowest level and project management details in the highest level. The following sections present the general overview and each level of the model.

A. PROCESS OVERVIEW

Daily development activities produce results that are reviewed at progressively higher levels in the hierarchy, with less frequent reviews as the level increases. This hierarchy does not necessarily equate to the hierarchy of decision-making. Regardless of company or product size, the organizations of our study developed small product increments daily, with integration results reviewed weekly or bi-weekly. Ultimately, the results were also reviewed before release in, for example, a six-month interval. Thus, we chose to use a layered architecture.

The evidence-based quality-aware ASD process model, which is shown at the highest level in Fig. 8, consists of three sub layers represented as subprocesses in the process model and referred to as such in the text. These are *Products* subprocess, *Releases* subprocess, and *Sprints* subprocess. All companies implemented a two-level structure—sprints and releases. Moreover, Companies B and C, which were developing large solutions, utilized a third level to plan and execute activities for a full product. Company A deployed a similar kind of higher level for customer cooperation.

The management practices in the *Products* and *Releases* subprocesses are highly specific to each case, being influenced by factors such as company size, release complexity, concurrent technologies being developed in case of embedded products, business and customer cases, and various constraints. Therefore, the quality-aware ASD process summarizes the management structures through the principle of periodic status meetings, leaving the actual implementation up to the organization implementing the process. The continuous and agile nature of these subprocesses is realized by assuming that management activities are conducted continuously.

The key principles of each subprocess are as follows:

1) subprocesses are organized in a continuous, incremental, and agile manner.
2) Planning and execution items' abstraction levels vary with the subprocess, from product ideas to code.
3) Each subprocess splits into *planning*, *execution*, and *validation* activities, where *Releases* and *Sprints* serve as the execution for *Products* and *Releases* respectively.
4) Each subprocess has its own validation activities, with lower level subprocesses delivering tested software packages and higher levels performing final integration and validation.
5) The pace of subprocesses is adjustable based on product and company needs.
6) *Sprints* subprocess follows Scrum principles, while *Releases* and *Products* are less formal, being based on organizations of our study.
7) Quality management in each subprocess focuses on the quality issues relevant for the abstraction levels of the planning and execution items.

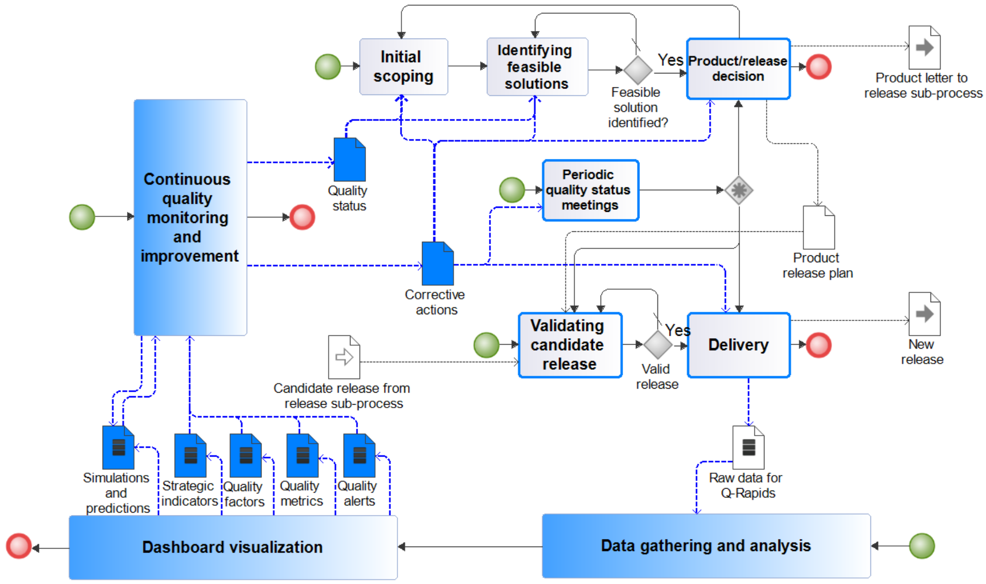


FIGURE 9. Products subprocess of the evidence-based quality-aware ASD process.

8) Roles in *Releases* and *Products* subprocesses are generally defined, leaving the companies to decide the actual roles and responsibilities, whereas the *Sprints* subprocess follows Scrum-defined roles.

The data relevant for quality management support are proposed to be used in each subprocess by all personnel relevant for decision-making, quality monitoring, and corrective action planning and execution. Although stakeholders in different subprocesses may have different areas of interest, as we saw in the example organization, the process model proposes full utilization of all evidence in all subprocesses. In the following sections, we present each of the subprocesses starting from the higher level going down to lower levels.

B. PRODUCTS SUBPROCESS

The *Products* subprocess outlines the development phases for high-level planning and validation of new products or releases, representing top management's contribution to development. It is assumed to be continuous, even though it can be one-shot in cases of customer-specific products requiring distinct planning and customer approval testing phases. This subprocess is divided into planning and validation activities, with its structure being depicted in Fig. 9.

The planning phase in the *Products* subprocess is triggered by a customer request or business opportunity being discovered internally, hence leading to the *Initial scoping*. Due to such early planning phases being extremely case specific, it is summarized as *Identifying feasible solutions*. It includes finding solutions to identified functionality and possible problems of the new product or release. Quality-related evidence is valuable in these early phases. Upon successful identification of the most feasible solutions in all relevant areas, more detailed planning commences, including documenting the business case, product or release targets, technical solutions and constraints, time and personnel constraints, and other information required for the *Releases* subprocess to start detailed planning.

Utilization of the quality status of the products or releases under development, when planning new products or releases, depends on the similarity of the earlier products or releases. The bigger the similarity, the more of the evidence supports the quality-related planning of the new product or release.

QRs or software quality-related evidence provides the *Products* subprocess with several scenarios for the new product or release planning:

1) While searching for feasible solutions for the new product or release, top managers of relevant business and technology areas monitor the quality status of the products or releases under development. The identified quality issues are taken into account and may trigger such corrective actions for the new product or release that need initiatives from the top management.
2) While searching for feasible solutions, top managers of the relevant technology areas monitor the quality of the products or releases under development to evaluate the feasibility of the potential solutions from a quality perspective.
3) While searching for feasible solutions, the top business and technology managers monitor the quality of the products or releases under development to set feasible targets for the new product or release by considering the quality perspective.
4) While deciding for a new product or release, the company-level quality managers set the high-level quality targets to the new product or release.
5) The final quality status of the product or release to be delivered is monitored by the managers in charge of the

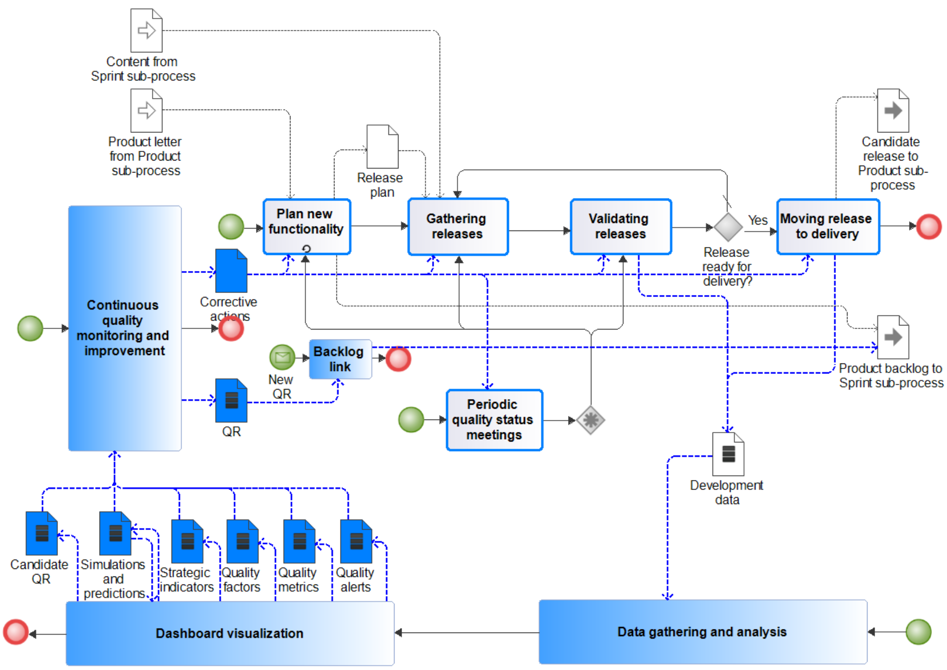


FIGURE 10. Releases subprocess of the evidence-based quality-aware ASD process.

product or release and used as a criterion for the final approval and delivery decision.

6) A company can decide to implement management-level *Periodic quality status meetings* to stay informed about the quality of the developed product and, if needed, steer resources toward work on quality issues.

The *Products* subprocess performs formal validation of new products or releases provided by the *Releases* subprocess, making final delivery decisions. Quality-related information assists in assessing whether the desired quality level is achieved. Due to the highly case-specific nature of validation activities, the process model outlines them broadly, leaving detailed implementation to the organizations using the process.

C. RELEASES SUBPROCESS

The *Releases* subprocess defines the development phases and activities for delivering software in the form of preplanned functionality bundles, that is, releases. Typically, a release contains functionality increments, error corrections, and value-adding modifications. It details how the releasing of new functionality is planned and how the releases developed in *Sprints* subprocess are validated. Serving as the implementation engine for the *Products* subprocess, it leverages the *Sprints* subprocess as the implementation engine.

The *Releases* subprocess consists of continuous planning and validation activities, which are generally faster paced than those in the *Products* subprocess. The quality-aware ASD process model allows organizations to determine the timing of the *Releases* subprocess based on case-specific constraints. Its overall structure is shown in Fig. 10.

The *Releases* subprocess, which is guided by the high-level product definition from the *Products* subprocess, plans the functionality for new products or releases. Planning details, like prioritization and breaking down epics into implementable parts, vary among companies and are typically left to their discretion. Corrective actions from previous quality issues should be considered in the planning, and any potential QRs may be added to the product backlog through a *Backlog link*. The quality-aware ASD process advises maintaining separate logical repositories for planning items like epics, features, and user stories, which is in line with the practices used in the organizations of our study.

In the *Releases* subprocess, new functionality accumulates at the pace of sprints, with specific integration and validation tests conducted. The extent of variation in validation environments and tests from the *Sprints* subprocess is flexible, being tailored to case-specific constraints.

The QR and software-related evidence provides the *Releases* subprocess with several scenarios for the planning of new functionality:

1) While planning new functionality, the key stakeholders (product or release managers, quality managers, POs) monitor the strategic indicators and QRs of the release under development to plan corrective actions for the new release.
2) While planning new functionality, the product release managers and POs monitor the QRs generated for the release under development to set quality-related requirements for the new release.

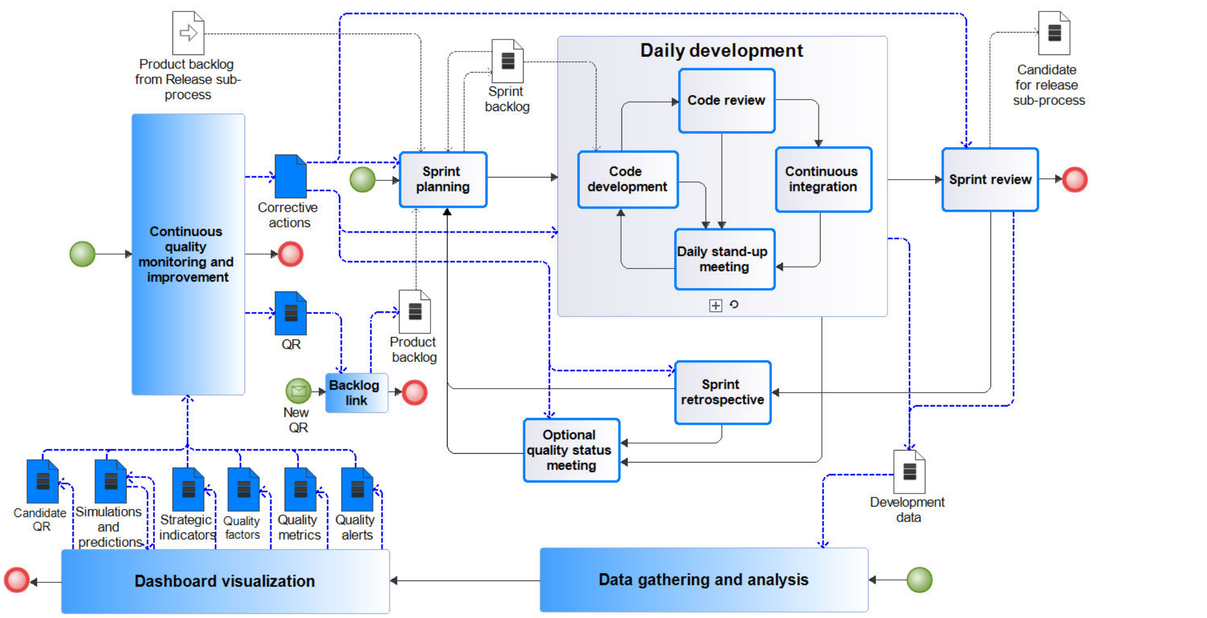


FIGURE 11. Sprints subprocess of the evidence-based quality-aware ASD process.

3) While prioritizing the epics, features, and user stories of the new release, the release managers and POs monitor the quality status of the release under development to understand how quality-related matters impact the priorities.
4) While planning the new release, the release managers and the POs monitor the quality status of the release under development to balance the plan between new functionality, refactoring, and quality-corrective actions.
5) During the sprint-based development of a release, the release managers and release quality managers monitor the quality status continuously to keep track of the maturity of the release and decide on potential corrective actions in the *Periodic quality status meeting*.
6) While planning and conducting validation testing, the release quality managers and test managers monitor the quality status of the release to focus testing on the areas that need special consideration.
7) While deciding for an approval of a release in *Moving release to delivery*, the release managers and release quality managers utilize the total quality status and open quality issues, such as quality alerts and open QRs, as acceptance criteria.

D. SPRINTS SUBPROCESS

The lowest level of the process model includes task planning, software implementation, and unit testing activities. The *Sprints* subprocess implements the plans defined in the *Releases* subprocess. It utilizes a straightforward implementation of Scrum, as presented in Fig. 11, because this process represents the way of working of the studied organizations the closest and because it is the most adopted agile method [3].

In addition to standard Scrum elements, we propose an *Optional quality status meeting*, as implemented by several organizations in our study. Integrating these meetings into the layered process model, specifically within the *Releases* or *Sprints* subprocesses, proved challenging. Therefore, it is suggested as an optional component in the *Sprints* subprocess, giving organizations the discretion to implement it. This meeting can be omitted if quality decisions are effectively addressed in standard meetings. However, an aperiodic meeting in the *Sprints* subprocess, like in Company A, allows for quicker feedback, as called for in Company C.

The *Sprints* subprocess involves developing software components and sprint-based continuous integration and testing. The quality-aware ASD process does not fix any specific sprint length but refers to the well-known recommendations of the scrumming-type agile methods. Work items for sprints, such as user stories and tasks, originate from the *Releases* subprocess planning activities and are stored in a product backlog managed by a PO, as per Scrum. These items also include corrective actions and QRs generated by the quality management support system.

Sprints incrementally deliver functional software, with continuous testing within the *Sprints* subprocess. However, final validation and approval occur in the *Releases* subprocess, adhering to the principle that a higher organizational level, rather than the implementing teams, oversees and approves the work results.

The QR and software quality-related data provide the *Sprints* subprocess with several scenarios:

1) Developers and testers can monitor the software's quality, focusing on product and process quality metrics and factors. *Quality monitoring* may be triggered by quality alerts or based on individuals' own initiatives.

TABLE 5. Usage until first evaluation workshop in Company A.

	Usage #	Reported activity	Corresponding activity in general model	User	Purpose
Workshop I	1	Code and commit features to SVN repo	Sprint level / Code development	Developer	Inspecting *Bug correction performance* metric while developing a new feature
	2	Evaluating sprint results	Sprint level / Sprint retrospective	Product manager	Inspecting strategic indicators when checking the progress of development
	3	WBM to plan features for sprint	Sprint level / Sprint planning	Product manager	Using simulation feature when planning and defining tasks to improve the state of *Product readiness* indicator
	4	Evaluating sprint results	Sprint level / Sprint retrospective	Project manager	Inspecting strategic indicators when checking the status of development
	5	Evaluating sprint results	Sprint level / Sprint review	Developer	Inspecting raw data for metrics when trying to find if the previous delivery was the source for new errors
	6	Evaluating sprint results	Sprint level / Sprint retrospective	Project manager	Inspecting details of *Testing performance* quality factor when trying to identify weaknesses in the testing process

2) QR and software quality data monitoring can be part of *Sprint review* and *Sprint retrospective* meetings, where the team may reflect on how successful the way of working in the sprint was from quality perspective and use the evidence as support for planning of actions for solving possible quality issues in the next sprints.
3) The PO may use the evidence in *Sprint planning* to balance refactoring and development of new functionality and for setting priorities of the backlog items in the next sprint from the perspective of product and process quality.
4) The POs, test engineers, and quality engineers overseeing quality in *Sprints* subprocess may conduct *Optional quality status meetings* together with the key stakeholders from *Releases* subprocess to monitor and manage the total product and process quality utilizing all the QR and software quality-related data.

VI. THE EMPIRICAL EVALUATION OF THE SOLUTION

The goal of this activity was to both implement the new processes and evaluate whether they fulfill the requirements for the process (Section IV-B). To avoid disrupting software development, implementation was gradual, integrating new practices exploratively. The evaluation exercise, which was conducted in three stages over a five-month period with follow-up interviews after three to four months, was qualitative and focused on the new process adoption from the practitioners' perspective. Because implementing the entire general process was not feasible, organizations adopted parts of it based on their specific ASD models, allowing for an evaluation of the general model through its implemented parts.

During implementation and evaluation, the practitioners from Companies A and B reported their use of the new support system through an online form, detailing what new evidence was utilized, by whom, for what purpose, when, in what activity, and outcomes. In Company C, due to unforeseen changes, the support system was reallocated to another project and was not used for decision-making within the time frame of the study. The practitioners in Company D had a clear view of how they would utilize the example support system already early on in the project. Evidence was mainly utilized for identifying weaknesses in the development process, and they did not see the need to explore further utilization at the time. After all, the intention was that a company may adopt the whole process or only parts of it according to their needs.

In Companies A and B, three workshops led by the first three authors were conducted to verify data from the online form and gather additional information for evaluation. The workshops took place in February, April, and May (columns E1, E2, and E3 of Table 3). Follow-up interviews (Table 3, last column) were conducted by the two first authors during September–October time frame in companies A, B, and C. These one-hour individual semi-structured interviews focused on continued evidence utilization, adherence to the process, additional activities, impact observation, and fulfillment of new process requirements. In Company D, a group interview that was led by the seventh author took place in June.

For Company C, where quality management support system integration was unsuccessful, interviews aimed at gathering insights into research cooperation with a large software development organization, but they are not included in this report. The following sections describe the outcomes of the evaluation workshops and final interviews, briefly covering Company B's reported usages and focusing in more detail on Company A, the example organization.

A. EVALUATION I

By the first evaluation workshop in February, six usage instances were recorded in Company A and three in Company B. In Company B, the developers mainly used the system during *Sprint implementation* to verify that changes in the metrics were reflected correctly in quality factors and strategic indicators. A project manager monitored metric updates during a daily Scrum meeting, seeking software quality improvements. Company B's focus was on building trust in the evidence provided by the quality management support system. The six instances of usage recorded in Company A are summarized in Table 5, showing what

TABLE 6. Usage after first evaluation workshop until second evaluation workshop.

	Usage #	Reported activity	Corresponding activity in general model	User	Purpose
Workshop II	7	Evaluating sprint results	Sprint level / Sprint review or retrospective	Developer	Inspecting history trend of *Bug correction throughput* metric at the end of sprint
	8	Evaluating sprint results	Sprint level / Sprint review or retrospective	Project manager	Inspecting history trend of *Timely feature delivery* metric at the end of sprint
	9	Evaluating sprint results	Sprint level / Sprint retrospective	Project manager	Inspecting details of strategic indicators to prepare for WBM

activities of the general model the reported usages correspond to.

Quality Awareness: The reported usages reflect increased quality awareness, with stakeholders receiving appropriate quality-related evidence during relevant activities. For instance, a developer reviewed the *Bug correction performance* metric (Usage #1) during development to track task progress. A product manager utilized Q-Rapids' simulation in *WBM to plan features for sprint* activity (Usage #3) to explore ways to enhance the *Product readiness* strategic indicator. Additionally, the *Evaluating sprint results* activity involved four instances of usage, focusing on monitoring development status and identifying improvement opportunities. These instances demonstrate that quality aspects are both highlighted and readily accessible to practitioners, even though they do not conclusively show if the process enables a rapid response to quality issues.

Evidence-Based Decision-Making: The reported data suggests a shift toward more evidence-based decision-making. The practitioners noted the strength of using evidence to plan and prioritize tasks for upcoming sprints, here by focusing development efforts to enhance product readiness. This indicates support for evidence-based decision-making by supplying the necessary evidence to relevant practitioners during appropriate activities.

Agility: The practitioners did not yet see any impact on the requirements related to agility. The use of evidence aligns well with their current practices without affecting the development speed. In Company B, usage focused on building trust in the system, making it premature to assess its fulfillment of the process requirements.

B. EVALUATION II

Between the first and second evaluation workshops, three usages were collected in both Companies A and B. In Company B, usage focused on building trust: a developer inspected trends of quality factors in sprint implementation, a DevOps engineer utilized evidence to assess software quality while defining tasks in sprint planning, and a project manager looked for ways to improve software quality. The main outcome was an increased emphasis on software quality. In Company A, usage was more goal driven (Table 6) and confirmed during the workshop to occur in the activity *Evaluating sprint results*, corresponding to the general model's *Sprint review* or *Retrospective*. However, Company A did not separate these activities, so usage could not be distinctly attributed to either.

Quality Awareness: The data again highlight the impact on quality awareness and usefulness of evidence. A developer reviewed the historical trends of *Bug correction* metrics at sprint's end to verify planned bug corrections. Additionally, the developer found monitoring these metrics beneficial during implementation for achieving a bug-free release, even though this was not formally reported as a usage.

Evidence-Based Decision-Making: A project manager monitored *Timely feature delivery*[2] strategic indicator and its details at the end of a sprint, verifying that new feature implementation was on track for the next release. The same manager also reported the third usage (usage #9) aimed at gaining an overview for the upcoming *WBM to plan features for sprint*. This quick, informative overview was noted as a significant benefit, facilitating discussions on quality in whiteboard meetings more effectively than before.

Agility: Regarding agility requirements, the process appeared not to hinder agility or slow down operations. One requirement is minimal change to existing processes. The usages indicate that the practitioners now review new evidence in or while preparing for meetings, a minor change that has not been perceived negatively.

C. EVALUATION III

Between the second and third evaluation workshops, nine usages were recorded: three from Company B and six from Company A. In Company B, usage was limited because the underlying support system's integration was immature, making the data unreliable for decision-making and only indicative of software quality. A developer evaluated quality during sprint implementation, a project manager checked actual development progress in a daily Scrum meeting against system outputs, and another developer's usage was related to management of the system itself to check that trends were reflected correctly.

In Company A, usage started to be more widespread, and the evidence was used for real decision-making. This provided stronger proof for whether the new process meets its requirements. Workshop discussions indicated that reported usages led to increased evidence utilization. The usages in Company A are summarized in Table 7.

[2]A strategic indicator measuring project management performance [45].

TABLE 7. Usage after second evaluation workshop until third evaluation workshop.

	Usage #	Reported activity	Corresponding activity in general model	User	Purpose
Workshop III	**11**	Managing development infrastructure	Overview / Data analysis setup subprocess	Integrator	Updating Q-Rapids with new quality policy related to QR alerts
	12	Quality assurance	Sprint level / Continuous integration	Quality Engineer	Alerts – Inspecting an alert related to three of the monitored quality metrics: *Critical issues ratio*, *Non-blocking files*, *Comment ratio*.
	10	WBM to monitor status of development	Sprint level / Optional quality status meeting	Project manager	Monitoring status of development due to suddenly changing quality metric values.
	13	Quality assurance	Sprint level / Continuous integration	Quality Engineer	QR generation – Sent to project manager information related to quality issues in the project and the generated QR (usage #12)
	14	Evaluating sprint results	Sprint level / Sprint review	Project manager	Alert and QR generation – Received notification related to potential quality issue and of new QRs (*Ratio of open issues*, and *Ratio of properly commented files*) -> Led to new task in *WBM to plan features for sprint*
	15	Q-Rapids continuous quality monitoring	Any level / Continuous quality monitoring and improvement	Project manager, Quality manager	Monitor status of project in preparation of WBM to plan features for sprint -> Led to *Managing development infrastructure* due to fault in certain metrics data

Quality Awareness: When inquired about usage #10 (Table 7), the practitioners said they noted unexpected changes in metrics during daily development (*Continuous integration* within *Daily development* of the *Sprints* subprocess). Although not blocking normal operations, which would trigger Company A's *Whiteboard meeting to monitor status of development*, they opted for an unscheduled meeting during sprint implementation to address the issue. This corresponds with the *Optional quality status meeting* in the general model, illustrating practitioners' ability and decision to respond to quality issues immediately.

In the meeting, they discovered a malfunction in the connector linking a development tool to the new support system. Restarting the connector normalized the metric values. Therefore, in fact, this led to an unreported instance of managing the development infrastructure (*Data analysis subprocess* in the general model), corresponding to *Managing development infrastructure*. Usage #11 involved updating the threshold values of the metrics to align with the company's revised quality policy. These usages highlight the focus on quality and the process's capacity for immediate quality issue response.

Usages #12 and #13, conducted by a quality engineer during the *QA* activity in *Integration and validation*, involved responding to alerts about three quality metrics—*Critical issues ratio*, *Non-blocking files*, and *Comment ratio*—exceeding the threshold values (usage #12). The system suggested creating a QR, which the engineer then generated and forwarded to the project manager with additional information (usage #13), demonstrating heightened quality awareness.

Evidence-Based Decision-Making: In usage #14, a project manager reviewed two generated QRs during *Evaluation of sprint results*, concerning the *Ratio of open issues* and *Ratio of properly commented files*. A task was created to address the first QR, while the second was dismissed, with the manager opting to remind developers about coding rules instead. This instance illustrates the use of evidence in decision-making.

Usage #15 marked the first reported instance within the new *Q-Rapids continuous quality monitoring* activity, which was part of the preparations for the next *WBM to plan features for sprint*. Although similar activities were previously reported under *Evaluating sprint results*, this instance was distinct. The project manager and a quality manager jointly investigated a sudden drop in metric values to zero. Drawing from prior experiences, they opted for a quick review rather than a full meeting, leading to notifying the integrator who restarted a connector. Restarting the connector was not reported in the data collection form. Once again, evidence helped make informed decisions.

Agility: The above usages demonstrate that the support system and process collectively enabled the practitioners to respond to quality issues in an agile way, allowing for immediate decision-making activities upon issue identification.

D. FINAL INTERVIEWS

We briefly summarize here the status of each organization at the time of the final interviews but focus mostly on Company A.

Company D: In Company D, the integration of the system stabilized early but was not as extensive as in Companies A and B. The organization had gone through the same steps of identifying needs; defining indicators, factors, and metrics for the quality model, and setting up the data collection and processing. However, their primary focus was on monitoring the development process, with a PO being the main user of the system for tracking process-related indicators and metrics. One PO identified a development bottleneck, initially perceived as a testing issue, but later traced this to resource constraints in the code review and merge phase before testing. This insight enabled targeted improvement actions. The POs appreciated the system's contribution to more accurate planning and estimation at both individual and sprint levels, process performance enhancement, and root cause analysis for process delays.

Company C: In Company C, the system was not actively used. Company C's strict rules on development data access hindered the acquisition of data initially deemed necessary, leading to the decision to pilot Q-Rapids in another project. During the interviews, they had just begun piloting the system in this project.

Company B: In Company B, evidence was mainly used in sprint planning, implementation, and review, but not for decision-making to the extent seen in Company A. The support system's maturity did not allow for full integration. Strict safety and security requirements necessitated verification of all development tool data; a goal not achieved during the study. Efforts were ongoing to build trust in the data. Consequently, no impact on agility or development speed was observed. However, the interviewed practitioners noted a shift toward a more quality-focused and data-driven development culture.

Company A: In Company A, usage continued as earlier, with an additional activity involving planning for the next release. Quality evidence was helpful in prioritizing unresolved blocking issues and agreeing on features for the upcoming release. Evidence utilization remained prominent in planning, development, review, and monitoring activities. The practitioners reported daily use of evidence for quality monitoring and decision-making, both in meeting preparations and during meetings. This led to more informed and quicker task prioritization as development status became clearer, simplifying the identification of next steps for quality improvement.

Company A saw significant improvement in its rapid response to quality issues. A project manager noted a 20% increase in the *Process performance* indicator when including data from before the project. The system also proved effective in more accurately predicting the readiness of the next release.

Meeting the Requirements: The practitioners from Company A reported that integrating quality evidence positively influenced agility, development speed, and quality management. Despite changes in decision-making and meeting preparations, they felt the requirement to maintain the existing process was met, with the product manager noting the process was enriched rather than changed.

A common challenge across all companies was setting up the support system, particularly connecting the various tools that lack existing connectors, hence requiring in-house development. This and any modifications to the quality model require manual intervention. Although Company A viewed this as a standard development task, it may be seen as an additional burden because it does not directly advance the product.

VII. DISCUSSION

This section explores the impact of the integrated quality-aware processes on the present study's organizations and examines the usefulness of the general quality-aware process model.

A. EFFECTS OF THE EVIDENCE-BASED QUALITY-AWARE PROCESS

The evaluation findings highlight improvements in both quality management and the software development process itself, which can be summarized in the following five points:

1) Agility: Company A found that the new process and support system data enabled quicker, more informed meeting preparation and decision-making.
2) QR management: Generating QRs and deriving tasks for development was started in Company A. This also raised awareness of quality.
3) Process performance: Company D saw improved understanding of several aspects of the Scrum process. Company A reported an improvement in process performance of 20%, signifying contribution to the stakeholder goals.
4) Planning: Company D experienced enhanced planning accuracy and issue management, while Company A observed more precise planning of release launch based on better testing and build stability insights.
5) Bottleneck identification and resolution: Company D identified root causes of development bottlenecks, and Company A prioritized unresolved issues for agreeing on the features for the next release.

Additionally, some organizations began addressing the challenges identified in the initial case studies [14]. Company A improved QR specification with greater detail and on several levels of granularity, and the solution aided in identifying key QRs. Company B, despite full integration not being achieved, saw that the value of QRs was easier to see, due proper attention early on. Company A also reported increased visibility of QRs during development and received real-time feedback on QRs, aligning with their areas of interest for improvement.

The evaluation results allow us to conclude the following about meeting the initial requirements:

Quality awareness. The quality-aware ASD process supports quality awareness from three perspectives:

- Continuous quality assessment and monitoring.
- Incremental and semi-automatic elicitation of QRs based on a continuous analysis of quality-related data.
- Offers a real-time, both holistic and detailed, overview of software quality status.

The tool and its integration into the ASD process alone are not solely responsible for the observed benefits. Although it increases awareness, effective use of the provided evidence still depends on the decision-makers.

Evidence-based decision-making. Quality-related decisions rely on insights from a data analysis engine, visualized on a dashboard for diverse roles. Information about the relevance of evidence for specific roles, activities, and purposes can be detailed in the process model descriptions. Although the model does not dictate what evidence to use, it may guide how to identify relevant evidence.

Agility. The quality-aware ASD process offers lightweight quality management support. Initially, usage might be more intensive, but once integrated, it seamlessly complements agile practices, facilitates quick responses to quality issues, and allows organizations to choose relevant quality aspects rather than relying on predefined indicators.

Agile prioritizes ''individuals and interactions over processes and tools'' [4], and our approach might initially seem contradictory. However, agile methods often emphasize functionality at the expense of the focus on quality. Implementing this system, alongside its usage practices, not only heightens quality awareness, but also assigns quality as a responsibility to the relevant stakeholders, like POs. The evidence provided by the system can help justify a higher priority of quality items in the backlog.

B. IMPLICATIONS FOR PRACTITIONERS

Based on our collaboration with the industry partners of Q-Rapids, we propose the following recommendations:

- Efforts to better include QRs and software quality into the ASD process should be made. An evidence-based support system can be integrated into ASD, particularly in Scrum. Few extra activities or roles are needed beyond those already existing in agile frameworks.
- Because ASD already utilizes tools for collection and analysis of data, these should be used for monitoring development progress from a quality perspective. It can guide decisions on adding tasks and QRs to backlogs, guide testing activities, and in process improvement by identifying weaknesses.
- Consider appointing a role focused on ensuring that connectors gather and store relevant data. The responsibility is similar to managing development infrastructure.
- Update descriptions of work tasks to guarantee the production of all relevant data.
- Integrate the support system into weekly monitoring, daily stand-ups, and planning activities for continuous quality monitoring, quick quality issue responses, and prioritizing quality aspects.
- Incorporate process-related metrics into sprint retrospectives to track progress toward quality goals and identify development bottlenecks.
- POs, test engineers, and quality engineers should use evidence from sprints as lessons learned, such as balancing refactoring with new development and prioritizing backlog items, to infuse quality insights from previous sprints into the planning and execution of future sprints.

Integrating a system like Q-Rapids is a nontrivial task, requiring commitment from managers, team leaders, and developers. Identifying the relevant quality indicators, selecting correct quality factors and metrics, and collecting appropriate raw data require cooperation. Connector development for various tools can be prone to error, as seen in Company A's initial connector issues. Building trust in the evidence may also be time-consuming.

More recent publications present similar findings [46], [47]. Becoming data driven is not merely about plugging in new tools and hiring experts. Meaningful indicators need to be defined, availability and completeness of the required data need to be guaranteed, and there is a need to build trust in the data analysis. It requires a change in the mindset of all involved stakeholders and in the organization's culture. This change needs to be managed properly, which means commitment from stakeholders of all levels of the organization. As we did in our study, and as Figalist et al. [47] proposes, starting small helps with building trust and with cultural change.

Eventually, in our study, the efforts led to a system that integrates smoothly with the ASD process, providing valuable quality insights and positively affecting agility, development speed, and quality management. The system's adaptability allowed for updates aligning with Company A's internal quality policy, without necessitating major changes to their existing ASD process.

C. IMPLICATIONS FOR RESEARCHERS

Despite the present study's duration of over one and a half years, the use of the Q-Rapids solution for managing QRs and software quality remains at an exploratory stage. Not all activity details were fully developed in the participating organizations, suggesting that many potential evidence applications remain unexplored. Future studies in these organizations, particularly after extended use of the support system, could uncover additional practices.

A key lesson is the importance of trust in data. In Company B, while recognizing the benefit of data-driven decisions over subjective judgments, the practitioners emphasized verifying data accuracy themselves. Their production of security and safety-critical products, which require traceable decisions, further underscored this need for trust. Exploring how practitioners can build trust in data emerges as a crucial research area.

Recognizing the limitations mentioned in the validity threats section, the utility of the evidence-based quality-aware ASD process should be examined in organizations falling beyond the present study's scope. Initially, Q-Rapids was intended to use both system and user data during software operation. However, due to project constraints and confidentiality issues around user data, the current study concentrated only on project and development data. Given that system and user data can offer crucial insights into software quality, future research is recommended to explore this aspect.

In addition, further research should also incorporate findings from more recent studies that were not available to us during our study. For example, Sdiri et al. [46] have complied a set of impediments for transitioning to a data-driven organization. A subset of these are organizational impediments, which were not fully explored in our study and could provide further insights into the transition. Figalist et al. [47] explore key drivers that prevent successful integration of analytics tools, and they also propose solutions. The integration of Q-Rapids into the ASD process could benefit greatly from these findings.

D. THREATS TO VALIDITY

As DSR is still maturing as research methodology, there is yet no common agreement on what aspects of validity should be considered [48]. Therefore, we choose to use Runeson and Höst's [49] guidelines, originally for case studies, to reflect on validity threats.

To ensure rigor, meticulous planning was applied in data collection and analysis. Interviews and focus groups were designed by one set of researchers, another set reviewed and enhanced them. Data collection was consistently conducted in pairs to minimize researcher bias and enhance reliability. However, a more significant threat to validity lies in the selection of organizations, participating practitioners, and use of the Q-Rapids support system.

Basing the requirements for the evidence-based quality-aware process on the organizations' current way of working presents a threat to construct validity. A wider set of organizations could provide a wider set of requirements, leading to a wider range of practices to utilize evidence in the management of quality. Additionally, focusing interviews on the Q-Rapids solution may have biased interviewees toward its capabilities, potentially overlooking alternative outcomes that different tools might offer.

External validity is also a concern. The design of the quality-aware processes was tailored to each organization's specific methods, focusing on their existing workflows. Although we generalized these into a final process model, different organizations could result in different workflows. Additionally, exploring agile methods beyond Scrum is recommended, because all studied organizations used ASD approaches closely resembling Scrum.

Even though we aimed to include different practitioners at different times, some of the participants were the same throughout the study. It is possible that some degree of maturation took place, which might affect the findings of the final evaluation. Therefore, this is a threat to the internal validity. Also, because the aim was to raise quality awareness, we cannot rule out that the project itself may have raised awareness of quality, not just the process.

The case companies did not implement the entire evidence-based quality-aware ASD process from the general model, instead adapting relevant parts to fit their specific processes. Although this approach allowed for evaluation of the general model through its implemented parts, further studies in diverse companies are needed to assess the model's utility across a broader spectrum of development organizations.

An additional limitation was the unsuccessful integration in two cases. In Company B, the maturity of the system hindered trust in the data. In Company C, internal restrictions on data access complicated implementation of the system. Although these are notable findings, the potential benefits of successful integration in situations like Companies B and C remain uncertain. Future research should include organizations with characteristics similar to B and C to further explore this.

VIII. CONCLUSION

Using design science research (DSR), we developed an evidence-based quality-aware ASD process to enhance the management of QRs and software quality. The phenomenon was studied over 20 months across four different software development organizations, involving 35 practitioners from various development roles who engaged in two rounds of interviews and six focus group sessions.

The major takeaways and results of our work are as follows:

- the designed general quality-aware ASD process with its subprocesses, that is, products subprocess, releases subprocess, and sprints subprocess,
- the elicited requirements for the quality-aware process from the scientific literature and the practitioners,
- guidance on how to implement the quality-aware ASD process,
- the evaluation of the new quality-aware processes showing a positive impact on ASD, and
- the designed new quality-aware ASD processes individually for the organizations.

Using one development organization as an example, we have demonstrated the design, validation, implementation, and evaluation of its specific quality-aware ASD process. Alongside this, we created a generic evidence-based quality-aware ASD process model for other ASD organizations to adopt. This model features three abstraction levels: products, releases, and sprints, encompassing standard ASD practices and activities. An optional quality status meeting can be included to address quality issues, but it is not mandatory.

The process fulfills quality awareness through continuous and automatic assessment and monitoring of quality aspects, enabling evidence-based decision-making through data collected from development. The quality aspects are defined based on the needs of the development organization.

We observed increased quality awareness, enhanced agility, and better process performance. The practitioners noted quicker responses to quality issues, improved backlog prioritization, and more precise planning for upcoming releases. One organization successfully identified and resolved development bottlenecks. However, infrastructure installation can be complex in large organizations that have many different tools, and trust in the evidence is important, particularly for secure and reliable systems.

Additional research is needed to assess the evidence-based quality-aware ASD process in various contexts, potentially enriching our model with new practices. Larger organizations and those using ASD methods different from Scrum would be particularly valuable for these studies.

Furthermore, we believe that our approach of creating and evaluating the processes can be reproduced and reused by other organizations willing to add quality aspects into ASD.

REFERENCES

[1] N. Abbas, A. M. Gravell, and G. B. Wills, "The impact of organization, project and governance variables on software quality and project success," in *Proc. Agile Conf.*, Aug. 2010, pp. 77–86.

[2] (2015). *Capgemini Finland, World Quality Report 2015-16*. [Online]. Available: https://www.capgemini.com/fi-en/resources/world-quality-report-2015-16/

[3] *Digital.ai. 15th Annual State of Agile Report*. Accessed: Nov. 2, 2019. [Online]. Available: https://digital.ai/resource-center/analyst-reports/state-of-agile-report

[4] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, and D. Thomas. (2001). *Agile Manifesto*. [Online]. Available: https://agilemanifesto.org/

[5] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: An empirical study," *Inf. Syst. J.*, vol. 20, no. 5, pp. 449–480, Sep. 2010.

[6] K. Pohl, *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam-Foundation Level-IREB Compliant*. Rocky Nook, 2016.

[7] A. E. Hassan, "The road ahead for mining software repositories," in *Proc. Frontiers Softw. Maintenance*, Oct. 2008, pp. 48–57.

[8] S. Martínez-Fernández, A. M. Vollmer, A. Jedlitschka, X. Franch, L. López, P. Ram, P. Rodríguez, S. Aaramaa, A. Bagnato, M. Choras, and J. Partanen, "Continuously assessing and improving software quality with software analytics tools: A case study," *IEEE Access*, vol. 7, pp. 68219–68239, 2019.

[9] C. Matthies and G. Hesse, "Towards using data to inform decisions in agile software development: Views of available data," in *Proc. 14th Int. Conf. Softw. Technol.*, 2019, pp. 552–559.

[10] R. P. L. Buse and T. Zimmermann, "Information needs for software development analytics," in *Proc. 34th Int. Conf. Softw. Eng. (ICSE)*, Jun. 2012, pp. 987–996.

[11] L. López, X. Burgués, S. Martínez-Fernández, A. M. Vollmer, W. Behutiye, P. Karhapää, X. Franch, P. Rodríguez, and M. Oivo, "Quality measurement in agile and rapid software development: A systematic mapping," *J. Syst. Softw.*, vol. 186, Apr. 2022, Art. no. 111187.

[12] R. B. Svensson and M. Taghavianfar, "Toward becoming a data-driven organization: Challenges and benefits," in *Research Challenges in Information Science*. Limassol, Cyprus: Springer, 2020, pp. 3–19.

[13] W. Behutiye, P. Karhapää, L. López, X. Burgués, S. Martínez-Fernández, A. M. Vollmer, P. Rodríguez, X. Franch, and M. Oivo, "Management of quality requirements in agile and rapid software development: A systematic mapping study," *Inf. Softw. Technol.*, vol. 123, Jul. 2020, Art. no. 106225.

[14] P. Karhapää, W. Behutiye, P. Rodríguez, M. Oivo, D. Costal, X. Franch, S. Aaramaa, M. Choras, J. Partanen, and A. Abherve, "Strategies to manage quality requirements in agile software development: A multiple case study," *Empirical Softw. Eng.*, vol. 26, no. 2, p. 28, Mar. 2021.

[15] R. J. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*. Berlin, Germany: Springer, 2014.

[16] A. Sillitti and G. Succi, "Requirements engineering for agile methods," in *Engineering and Managing Software Requirements*, 2005, pp. 309–326.

[17] M. G. Jaatun, "Software security activities that support incident management in secure DevOps," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, Aug. 2018, pp. 1–6.

[18] F. Gilson, M. Galster, and F. Georis, "Extracting quality attributes from user stories for early architecture decision making," in *Proc. IEEE Int. Conf. Softw. Archit. Companion (ICSA-C)*, Mar. 2019, pp. 129–136.

[19] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements challenges in the context of large-scale distributed agile: An empirical study," *Inf. Softw. Technol.*, vol. 110, pp. 39–55, Jun. 2019.

[20] W. Alsaqaf, M. Daneva, and R. Wieringa, "Understanding challenging situations in agile quality requirements engineering and their solution strategies: Insights from a case study," in *Proc. IEEE 26th Int. Requirements Eng. Conf. (RE)*, Aug. 2018, pp. 274–285.

[21] B. M. Aljallabi and A. Mansour, "Enhancement approach for non-functional requirements analysis in agile environment," in *Proc. Int. Conf. Comput., Control, Netw., Electron. Embedded Syst. Eng. (ICCNEEE)*, Sep. 2015, pp. 428–433.

[22] P. Mohagheghi and M. E. Aparicio, "An industry experience report on managing product quality requirements in a large organization," *Inf. Softw. Technol.*, vol. 88, pp. 96–109, Aug. 2017.

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

[24] H. H. Olsson and J. Bosch, "From opinions to data-driven software R&D: A multi-case study on how to close the 'open loop' problem," in *Proc. 40th EUROMICRO Conf. Softw. Eng. Adv. Appl.*, Aug. 2014, pp. 9–16.

[25] M. L. Drury-Grogan, K. Conboy, and T. Acton, "Examining decision characteristics & challenges for agile software development," *J. Syst. Softw.*, vol. 131, pp. 248–265, Sep. 2017.

[26] R. B. Svensson, R. Feldt, and R. Torkar, "The unfulfilled potential of data-driven decision making in agile software development," in *Agile Processes in Software Engineering and Extreme Programming*. Montréal, QC, Canada: Springer, 2019, pp. 69–85.

[27] M. Nayebi, G. Ruhe, R. C. Mota, and M. Mufti, "Analytics for software project management - where are we and where do we go?" in *Proc. 30th IEEE/ACM Int. Conf. Automated Softw. Eng. Workshop (ASEW)*, Nov. 2015, pp. 18–21.

[28] T. Menzies and T. Zimmermann, "Software analytics: So what?" *IEEE Softw.*, vol. 30, no. 4, pp. 31–37, Jul. 2013.

[29] E. Brynjolfsson, L. M. Hitt, and H. Kim, "Strength in numbers: How does data-driven decisionmaking affect firm performance?" Apr. 2011. [Online]. Available: https://ssrn.com/abstract=1819486

[30] T. M. Abdellatif, L. F. Capretz, and D. Ho, "Software analytics to software practice: A systematic literature review," in *Proc. IEEE/ACM 1st Int. Workshop Big Data Softw. Eng.*, May 2015, pp. 30–36.

[31] J.-G. Lou, Q. Lin, R. Ding, Q. Fu, D. Zhang, and T. Xie, "Software analytics for incident management of online services: An experience report," in *Proc. 28th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Nov. 2013, pp. 475–485.

[32] J. Cito, "Developer targeted analytics: Supporting software development decisions with runtime information," in *Proc. 31st IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Sep. 2016, pp. 892–895.

[33] E. Stroulia, I. Matichuk, F. Rocha, and K. Bauer, "Interactive exploration of collaborative software-development data," in *Proc. IEEE Int. Conf. Softw. Maintenance*, Sep. 2013, pp. 504–507.

[34] V. Augustine, J. Hudepohl, P. Marcinczak, and W. Snipes, "Deploying software team analytics in a multinational organization," *IEEE Softw.*, vol. 35, no. 1, pp. 72–76, Jan. 2018.

[35] I. Figalist, C. Elsner, J. Bosch, and H. H. Olsson, "Fast and curious: A model for building efficient monitoring- and decision-making frameworks based on quantitative data," *Inf. Softw. Technol.*, vol. 132, Apr. 2021, Art. no. 106458.

[36] K. Schwaber, *Agile Project Management With Scrum* (Developer Best Practices). Pearson Education, 2004. [Online]. Available: https://books.google.fi/books?id=6pZCAwAAQBAJ

[37] L. Guzman, M. Oriol, P. Rodriguez, X. Franch, A. Jedlitschka, and M. Oivo, "How can quality awareness support rapid software development—A research preview," in *Requirements Engineering: Foundation for Software Quality*. Essen, Germany: Springer, 2017, pp. 167–173.

[38] *Quality Management and Quality Assurance-vocabulary*, Standard ISO 8402, 1994. [Online]. Available: https://www.iso.org/standard/20115.html

[39] *Software Process: Improvement and Practice*, Standard ISO/IEC 25010, 2011. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35733

[40] V. Basili, A. Trendowicz, M. Kowalczyk, J. Heidrich, C. Seaman, J. Münch, and D. Rombach, *Aligning Organizations Through Measurement: The GQM+ Strategies Approach*. Berlin, Germany: Springer, 2014.

[41] M. C. Tremblay, A. R. Hevner, and D. J. Berndt, "Focus groups for artifact refinement and evaluation in design research," *Commun. Assoc. for Inf. Syst.*, vol. 26, p. 27, 2010.

[42] K. Petersen and C. Wohlin, "Context in industrial software engineering research," in *Proc. 3rd Int. Symp. Empirical Softw. Eng. Meas.*, Oct. 2009, pp. 401–404.

[43] D. Leffingwell, *SAFe® 4.0 Reference Guide: Scaled Agile Framework® for Lean Software and Systems Engineering*. Reading, MA, USA: Addison-Wesley, 2016.

[44] S. Martinez-Fernández, A. Jedlitschka, L. Guzmán, and A. Maria Vollmer, "A quality model for actionable analytics in rapid software development," in *Proc. 44th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2018, pp. 370–377.

[45] P. Ram, P. Rodriguez, M. Oivo, and S. Martínez-Fernández, "Success factors for effective process metrics operationalization in agile software development: A multiple case study," in *Proc. IEEE/ACM Int. Conf. Softw. Syst. Processes (ICSSP)*, May 2019, pp. 14–23.

[46] B. Sdiri, L. Rigaud, R. Jemmali, and F. Abdelhedi, "The difficult path to become data-driven," *Social Netw. Comput. Sci.*, vol. 4, no. 4, p. 385, May 2023.

[47] I. Figalist, C. Elsner, J. Bosch, and H. H. Olsson, "Breaking the vicious circle: A case study on why AI for software analytics and business intelligence does not take off in practice," *J. Syst. Softw.*, vol. 184, Feb. 2022, Art. no. 111135.

[48] K. R. Larsen, R. Lukyanenko, R. M. Mueller, V. C. Storey, D. VanderMeer, J. Parsons, and D. S. Hovorka, "Validity in design science research," in *Designing for Digital Transformation. Co-Creating Services With Citizens and Industry*. Kristiansand, Norway: Springer, 2020, pp. 272–282.

[49] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Softw. Eng.*, vol. 14, no. 2, pp. 131–164, Apr. 2009.

PERTTI KARHAPÄÄ received the M.Sc. degree in information processing science from the University of Oulu, Finland, in 2016, where he is currently pursuing the Ph.D. degree with the M3S Research Unit, Faculty of Information Technology and Electrical Engineering.

He is also a Researcher and a University Teacher with the M3S Research Unit, Faculty of Information Technology and Electrical Engineering, University of Oulu. His research interests include agile software development, software development processes, requirements engineering, alignment of requirements engineering and testing, quality requirements, and data structures and algorithms. He has been actively involved in the European Union's H2020 Q-Rapids project.

WOUBSHET BEHUTIYE received the Ph.D. degree in software engineering from the University of Oulu, Finland.

He is currently a Postdoctoral Researcher with the M3S Empirical Software Engineering Research Unit, University of Oulu. He has been actively participating in and contributing to national and international research projects. He teaches software engineering courses at the University of Oulu and Nanjing Institute of Technology, Oulu School. His research interests include agile software development, software processes, continuous software engineering, requirements engineering, software quality, technical debt management, and evidence-based software engineering.

Dr. Behutiye has served as a Reviewer and a PC Committee Member in software engineering journals and conferences, such as *IET Software*, the *Journal of Software Evolution and Processes*, LASD, and ICSOB.

PERTTI SEPPÄNEN received the M.Sc. degree in electrical engineering and the Ph.D. degree in information processing science from the University of Oulu, Finland, in 1983 and 2018, respectively.

After receiving the M.Sc. degree, he has worked in different research and development management positions in the global ICT industry. Since 2014, he has been with the University of Oulu. In the H2020 Q-Rapids Project, his research focus was on quality-aware agile software processes. His research interests include software startups and vehicle software.

Dr. Seppänen has contributed as a PC member to several international conferences on software research.

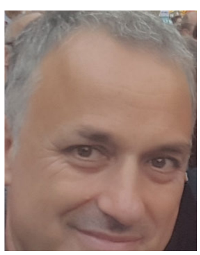
PILAR RODRÍGUEZ received the Ph.D. degree in computer science from the University of Oulu, Finland, in 2013.

She is currently an Associate Professor with Universidad Politécnica de Madrid, Spain, and a Docent with the University of Oulu. Her research interests include empirical software engineering, agile and lean software development, software quality, value-based software engineering, and human factors in software engineering.

Dr. Rodríguez served as a PC Member for conferences, such as ESEM, EASE, and XP. She serves habitually as a Reviewer for leading SE journals, such as IEEE TRANSACTIONS ON SOFTWARE ENGINEERING and *Empirical Software Engineering*. Recently, she has been the Leader of Work Package 2 in the H2020 Q-Rapids Project.

MARKKU OIVO held several positions at VTT, from 1986 to 2000. He had visiting positions the University of Maryland, from 1990 to 1991; Schlumberger Ltd., Paris, from 1994 to 1995; Fraunhofer IESE, from 1999 to 2000; the University of Bolzano, from 2014 to 2015; and Universidad Politécnica de Madrid, in 2015. He worked at Kone Company, from 1982 to 1986. He was the Vice President and the Director of Research and Development at Solid Company, from 2000 to 2002. He has been a Professor and the Head of the M3S Research Unit, University of Oulu, Finland, since 2002. He has initiated and managed more than 100 national and international research projects and programs with tens of millions of euros in national and international funding.

He is a Founding Member and the Chair of the Steering Committee of ISERN. He has served as the chair and a committee member in organizing numerous international conferences and has been a reviewer for top journals.



XAVIER FRANCH received the Ph.D. degree in informatics from Universitat Politécnica de Catalunya (UPC—BarcelonaTech), in 1996.

He is currently a Professor of software engineering with UPC—BarcelonaTech. His research interests include software engineering, including requirements engineering, empirical software engineering, green AI, and agile software development.

Dr. Franch is a member of the IST, REJ, JSS, and Computing editorial boards. He served as the PC Chair at CAiSE'22, RE'16, ICSOC'14, CAiSE'12, and REFSQ'11, and a GC for ESEM'24, RCIS'22, PROFES'19, REFSQ'11, RE'08, and PROFES'19.

SILVERIO MARTÍNEZ-FERNÁNDEZ (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computing from UPC—BarcelonaTech.

He was a Postdoctoral Fellow with the European Research Consortium for Informatics and Mathematics, from 2016 to 2018, and the Operative Project Manager with Fraunhofer IESE, Germany, from 2018 to 2019. He is currently a Researcher with more than 60 peer-reviewed publications and H-factor of 19 (according to Google Scholar). He is a Co-Principal Investigator of the Spanish project "Towards Green AI-Based Software Systems: An Architecture-Centric Approach (GAISSA)" (https://gaissa.upc.edu/en). In EU framework programs, he acted as the Evaluation WP Leader in Q-Rapids (H2020, RIA). His research interests include empirical software engineering, green AI, AI-based systems, software analytics, and reference architectures.

Dr. Martínez-Fernández has been the PC Co-Chair in several international conferences and workshops, such as ESEM 2024, ESELAW@CIbSE 2021, PROFES 2019, CESI@ICSE 2018, and QuASD@PROFES 2017–2018. He has also been a reviewer of multiple journals and a PC member of international conferences.

LIDIA LÓPEZ received the Ph.D. degree in computing from the Technical University of Catalonia (UPC—BarcelonaTech), Spain, in 2013.

She is currently a tenure-eligible Lecturer with UPC—BarcelonaTech. Previously, from 2021 to 2023, she was a Research Engineer with Barcelona Supercomputing Center (BSC). Since 2007, she has been involved at UPC—BarcelonaTech as an Assistant and Associate Teacher and a Researcher. She has participated in several international research projects, such as HealthyCloud (H2020, 2021–2023; coordination team and researcher), Q-Rapids (H2020; 2015–2019; work-package leader), and RISCOSS (FP7; 2015–2017; work package leader, and partner representative). Her research interests include software engineering, focused on requirements engineering, open-source software, empirical software engineering, and data-driven decision-making processes in agile software development.

Dr. López has been a PC Member at several international conferences, such as RCIS, ICSOFT, SAC, and CIbSE. She has also been the PC Co-Chair in international conferences and workshops, such as CIbSE and iStar. She reviewed journal articles, such as IST, *Journal of Social Sciences*, and IEEE Software.

MICHAŁ CHORAŚ is currently a Full Professor with Bydgoszcz University of Science and Technology, Poland, where he is also the Head of the Teleinformatics Systems Division and the PATRAS Research Group. He was granted a Full Professor title, in December 2021. He was a Project Coordinator for the successful H2020 SIMARGL project (secure intelligent methods for advanced recognition of malware and stegomalware). He is also the Project Coordinator/Manager and a Security Consultant for national/international bodies and private organizations. He is the author of more than 330 reviewed scientific publications. He has been involved in more than 20 EU projects, such as APPRAISE, AI4CYBER, SPARTA, STARLIGHT, SocialTruth, CIPRNet, Q-Rapids, and InfraStress). His research interests include AI, machine learning, data science, and pattern recognition in several domains, such as cyber-security, fake news detection, anomaly detection, data correlation, biometrics, and critical infrastructure protection.

ALESSANDRA BAGNATO received the Ph.D. degree in computer science from TELECOM Sud-Paris and the University of Évry Val d'Essonne, France, and the M.Sc. degree in computer science from the University of Genoa, Italy.

Since May 2018, she has been at Softeam Software (Docaposte Group), where she is currently a Research Scientist and a Research Responsible. She coordinates the Softeam Research and Development Team's research activities covering fields, such as model-based system engineering (MBSE), the design of cyber physical systems of systems (CPSoS), cloud and big data in application areas, such as eGovernment, eHealth, Fppd, and agriculture. She was with the TXT Corporate Research Division headquartered in Milan, from 1999 to September 2012, where she was involved as a Project Coordinator, the Project Manager, and/or the Technical Leader in several European research projects. She has led the Modelio team's research activities around innovative projects on model-driven engineering methods in the Modelio workbench in the area of cyber-physical systems, cloud, and big data (Horizon Europe Foodity, H2020 MORPHEMIC, ECSEL AIDOaRt, H2020 Databio, H2020 CPSwarm, H2020 QRapids, H2020 CROSSMINER), GDPR and privacy (H2020 PoSeID-on and ANR UPCARE), and on measuring software engineering (ITEA 3 MEASURE).

SANJA AARAMAA received the Ph.D. degree in information processing science from the University of Oulu, Finland, in 2017.

She is currently the Area Product Owner, Architecture and Specification (A&S) at RF Platforms DU, Nokia Mobile Networks (MN). In this position, she is responsible for planning A&S deliveries and managing area-specific backlogs within RF Platforms. She is also the H2020 Q-Rapids Project Manager for Nokia. Previously, she was the RF Platforms DU Nokia Operations Owner at Nokia Mobile Networks (MN). She has conducted empirical research, since 2008, in several publicly funded (ITEA, Business Finland, H2020) research projects in close collaboration with industry.

JARI PARTANEN received the M.Sc. degree in industrial management from the University of Oulu, Finland, in 1990.

He is currently the Director of Quality and Research at Bittium. Bittium www.bittium.com has undergone a transition toward a leaner and more agile way of working in the last few years and has taken up the use of approaches, such as Continuous Integration Toward Continuous Deployment, Continuous and Transparent Planning, as well as approaches like embedded DevOps practices toward RegOPs. He is an Active Researcher with more than 20 peer-reviewed publications.

• • •