Analyzing SAFe Practices with Respect to Quality Requirements: Findings from a Qualitative Study

Wasim Alsaqaf[1](✉), Maya Daneva[1](✉), Preethu Rose Anish[2](✉), and Roel Wieringa[1](✉)

[1] School of Computer Science, University of Twente, Enschede, The Netherlands
{w.h.a.alsaqaf,m.daneva,r.j.wieringa}@utwente.nl
[2] Tata Research Development and Design Center, TCS, Pune, India
preethu.rose@tcs.com

Abstract. Quality Requirements (QRs) pose challenges in many agile large-scale distributed projects. Often, project organizations counter these challenges by borrowing some heavyweight practices, e.g. adding more documentation. At the same time, agile methodologists proposed a few scaled agile frameworks to specifically serve agile organizations working on large and distributed systems. Little is known about the extent to which these proposals address QRs and the specific ways in which this happens. Moreover, evidence regarding the practical implementation of these frameworks with respect to QRs is scarce. Our paper makes a step towards narrowing this gap of knowledge. Using an exploratory research process, we analyze one well-documented framework, namely the Scaled Agile Framework (SAFe). We first analyzed the elements of SAFe as they were described in the methodological book of SAFe to identify the possible remedies to the QRs challenges reported in previous work. We then conducted a qualitative interview-based study to understand the practices that SAFe practitioners actually use to mitigate those QRs challenges. Our documentary analysis of SAFe resulted in identifying 25 SAFe elements that could (at least partially) mitigate one or more of the reported QRs challenges. Nine of those SAFe elements were reported in our interview-based study by SAFe practitioners as remedy for some of the reported QRs challenges. While practitioners attempted to use the recommended SAFe strategies for QRs, they often changed them in their own ways, or altogether resorted to heavyweight practices that the case study organizations knew from previously done non-SAFe projects.

Keywords: Agile scaled framework · SAFe · Quality requirements · Requirements engineering · Documentary analysis · Empirical research · Qualitative interview-based study

1 Introduction

The necessity of reacting quickly to the rapidly changing market, pushes large organizations to believe that the success stories of agile methods' application in the context they

L. Ardito et al. (Eds.): PROFES 2021, LNCS 13126, pp. 233–248, 2021.
https://doi.org/10.1007/978-3-030-91452-3_16

originally were designed for (e.g. small co-located teams) can be successfully repeated in large-scale distributed context. However, the transferability of experiences made in the original context to the realities of large-scale distributed contexts is far from flawless [1–4]. Although several agile scaled frameworks have been proposed by agile practitioners to guide the application of agile methods in large-scale distributed context (e.g. Scaled Agile Framework (SAFe) [5], Large-Scale Scrum (LeSS) [6], Scrum@Scale (S@S) [7]) relatively little research is published about these frameworks' effectiveness in practice, especially on an enterprise scale [2, 8]. Moreover, as per a 2018 review [3], large-scale agile enterprises adopting these frameworks report a broad range of technical and enterprise-level challenges due to resistance to change, shifts in the ways of thinking of hierarchies of requirements, lack of transparency, and lack of knowledge on proper integration of agile and non-agile ways of working. Our current paper is dedicated to one specific type of requirements challenges in large-scale agile delivery, namely those pertaining to quality requirements (QRs), such as security and usability. The paper builds upon an earlier study [9] in which the authors found that often, enterprises counter QRs challenges by borrowing some heavyweight practices, e.g. creating new artefacts (security or usability stories) or roles (e.g. security officer, UX team), and then adding these practices to their agile delivery cycle. Therein [9], is also stated that the introduction of these heavyweight practices unexpectedly brought with them new problems. But do agile scaled frameworks propose a remedy to QRs challenges in large-scale agile? If so, is the remedy effective in practice? As we found no publication answering these questions, we initiated an exploratory qualitative research process to understand and evaluate the agile methodologists' proposals for treating QRs challenges. For the purpose of our research we chose for inclusion those scaled agile frameworks deemed 'most popular' according to the 14th annual state-of-agile report issued by market observing firms [10]. As already said, the present work rests on a previously published exploratory study [9] that found 15 QRs challenges and 9 practices that agile practitioners currently use to cope with the identified challenges. We note that these findings [9] came out of an interview-based research with practitioners in enterprises committed to agile project delivery. However, these 9 practices were not collected in relation to any existing prescriptive or descriptive agile scale framework such as LeSS [6] nor agile method such as Scrum [11]. Given this background, in the present research we aim to explore those agile practices that are suggested by the most popular published agile scaled frameworks and that could help mitigate the QRs challenges which were identified in the previous work [9].

The present paper reports our results of analyzing one specific scaled framework, namely, SAFe 5.0 [5]. Our ongoing research also includes some other frameworks, however these are out of scope in this paper. In this paper, we set out to answer the following research question: *What are the agile practices suggested by SAFe that could mitigate the effect of the QRs challenges identified in* [9]*?* This question is decomposed in three sub-questions:

RQ1: Which SAFe elements described in the SAFe methodological textbook and related literature could possibly mitigate the QRs challenges defined in [9]*?*
RQ2: Do SAFe practitioners experience the QRs challenges identified in [9]*?*
RQ3: If yes, then which SAFe elements do they utilize to mitigate these challenges?

Using (i) a documentary research process [12] that takes as input the SAFe methodological textbook [5] and its related literature, and (ii) a qualitative exploratory interviews with practitioners from agile organizations [13], we analyzed the practices that the SAFe methodologists [5] proposed to use in large projects. In what follows, we first give a background information and describe related work (Sect. 2). Thereafter we describe our research process and provide definitions of the most important concepts (Sect. 3 and 4). We then present and discuss our results (Sect. 5 and 6) and the threats of validity (Sect. 7). We conclude in Sect. 8.

2 Background and Related Works

The question of how large-scale frameworks (such as SAFe [5] and LeSS [6]) treat RE problems in general, and QRs problems in particular, has become an area of active research relatively recently. A 2017 systematic literature review [14] found that while QR challenges are documented in empirical studies, relatively few solutions to them have been demonstrated to work consistently well. An empirical follow-up case study research of Alsaqaf et al. [9] indicated nine solution practices that practitioners resort to in large-scale agile projects. However (as mentioned in the Introduction), these solution practices were not traceable to any particular methodological guidelines linked to large-scale framework, be it SAFe, LeSS or S@S. Next, in 2016–2020, Kasauli et al. [15] carried out a large-scale research initiative on RE challenges and practices in large-scale agile system development. Moyon et al. [16] extends SAFe with the S^2C-SAFe framework to achieve security compliance in large-scale agile context. To the best of our knowledge, the empirical articles (e.g. [15, 16]) of these authors are among the very few who particularly focused on SAFe and LeSS in order to derive potential solutions to RE challenges. Kasauli et al. [15] also indicated that despite one might find potential solution elements in the popular large-scale frameworks and their methodological guidelines, for very few of these elements there is empirical evidence to work as assumed. Furthermore, Beecham et al. [17] examined how two scaled frameworks – SAFe and DAD [18], address software development risks in global projects, where requirements risks formed a major category. Using two longitudinal case studies implementing each framework, the authors conclude in regard to the requirements risks that these "are addressed well by both methods". However, the authors also elaborate that requirements risks are not in fact eliminated; only the impact of these risks could be reduced. We make the note however that the addressed risks refer to scope, project goals and conflicting requirements in general, and not particularly to QRs challenges.

Most recently, other researchers examined how large-scale agile project (that had adopted large-scale frameworks, e.g. SAFe) cope with specific types of QRs, such as security compliance requirements and privacy requirements [19]. The authors of [19] developed and evaluated approaches to each of these requirements that are meant to complement SAFe. The specificity of these approaches however makes it hard to generalize across solutions to other types of QRs.

3 Research Method

This study is part of an exploratory research initiative in which we analyze the practices described by well-known agile scaled frameworks from practitioner's perspective, in order to understand the extent to which these practices mitigate the impact of the QR's challenges identified in [9]. To achieve our goal we chose to develop a two-step qualitative research design [13]. This is suitable in cases in which researchers first use documentary analysis techniques for examining texts (e.g. guidelines, policies, proceedings) in order to develop sensitivity of the aspects of the phenomenon under study and to come up with assumptions about what is supposed to happen in real-world situations, and then use qualitative exploratory techniques (e.g. in-depth interviews) to understand what actually happens in the real-world context and how much this deviates from what is supposed to happen. We adopted this approach because it fits our research context and also because it has demonstrated its viability in an earlier study [17] in a context similar to ours. We designed the following research process including the following steps: **Step 1** explains our reasoning for including certain frameworks. This is described in Sect. 3.1. **Step 2** is concerned with examining the applicability of the selected agile scaled frameworks' practices – as described by the authors of each framework in their respective methodological textbook with guidelines and their repository of related documents (appendices, templates, cases) –, in mitigating the QRs challenges found in our previously published study [9]. This step is described in Sect. 3.2. It performs a documentary analysis grounded on the methodological guidelines of Appleton and Cowley [12]. The outcome of this step is a list of assumptions that agile methodologists have about how to treat QRs in large-scale projects. **Step 3** is described in Sect. 3.3 and it investigates the practices of the selected agile scaled frameworks as utilized in real life by practitioners in real-world organizations, to empirically examine their application in mitigating the identified QR's challenges (as per the perceptions and the experiences of those working in the field). As this paper is focused on one framework only (SAFe [5]), it in turn reports on Steps 2 and 3 as executed in the context of analyzing this specific framework. We describe the steps of our process in the next sub-sections.

3.1 Selecting Agile Scaled Framework

Over the years, the community of agile practitioners proposed more than 30 scaled agile frameworks [20]. Portman [20] classified those into two categories: (1) *enterprise-targeted* frameworks (e.g. SAFe [5], LeSS [6], S@S [7]) aiming to deliver complex enterprise-level products whereby the collaboration between distributed teams is essential, and (2) *web-scale-targeted* frameworks (e.g. Spotify [21], Scaled Agile Lean Development (ScALeD)[1]) aiming to support a company's IT-department in maintaining existing applications, whereby the dependencies between distributed teams are minimalized. In this paper, we focus on the first category—"enterprise-targeted frameworks" – because these frameworks match our research interest, namely the distributed and large-scale systems development context. Furthermore, for the purpose of our research initiative, we limit our selection of frameworks to those that are the most used according to the

1 http://scaledprinciples.org/.

14th annual state-of-agile report of market observers [10]. This source [10] indicates the "enterprise-targeted framework" SAFe [5] as the most popular across large organizations today. While our research initiative will include more frameworks (e.g. LeSS, and S@S), in this paper we focus solely on the agile practices of SAFe [5]. However, our choice for SAFe [5] does not imply that we prefer or recommend SAFe. The other frameworks will be investigated in our follow-up research.

3.2 Uncovering the Assumptions in SAFe About Engineering the QRs

The information about SAFe provided on its official website https://www.scaledagileframework.com/ as well as in its official textbook 'SAFe 5.0 Distilled Achieving Business Agility with the Scaled Agile Framework' [5], was taken as input into our documentary analysis. The objective of this stage of the research process was to answer RQ1. The first two researchers analyzed the SAFe practices by first reading and re-reading the textbook [5] and the text resources in the website to identify those SAFe elements that could be considered potential candidate strategies to cope with the QR challenges identified in [9]. For clarity, we list these QRs challenges in Table 1, where the first column shows the categories of the challenges as reported in [9] and the second column reports the specific challenges of each respective category in the first column.

Table 1. The QR challenges as reported in [9].

Category	Challenges
1. Teams coordination and communication challenges	Late detection of QRs infeasibility
	Hidden assumptions in inter-team collaboration
	Uneven teams maturity
	Suboptimal inter-team organization
2. Quality assurance challenges	Inadequate QRs test specification
	Lack of cost-effective real integration test
	Lengthy QRs acceptance checklist
	Sporadic adherence to quality guidelines
3. QRs elicitation challenges	Overlooking sources of QRs
	Lack of QRs visibility
	Ambiguous QRs communication process
4. Conceptual challenges of QRs	Unclear conceptual definition of QRs
	Confusion about QR's specification approaches
5. Architecture challenges	Unmanaged architecture changes
	Misunderstanding the architecture drivers

Our documentary analysis proceeded as follows: In a first round, the two researchers worked independently to come up with a list of SAFe elements for which the SAFe textbook [5] gives explicit or implicit information that the respective practice helps with QRs. In a second round the researchers got together and have discussed their identified practices of SAFe based on an argumentative discussion [22] and Conklin's dialog mapping technique for qualitative data structuring [23], in order to examine each practice's fitness in mitigating the QRs challenges reported in [9]. The goal was to reach a shared rationally-supported hypothetical mapping [23] between each SAFe element and one or more QR challenges. The result of this step was a list of assumptions indicating to us those SAFe practices that could possibly mitigate the QR challenges in Table 1. These assumptions mean theoretical mappings (we adopt the term from Beecham et al. [17]) and in Step 3, we want to complement them with real-world insights from an exploratory interview study, in order to compare and contrast our findings from theory (i.e. the SAFe textbook) and practice. We provide more details on our analytical activity and results in Sect. 5.1.

3.3 Understanding How Real-World SAFe Projects Resolve QRs Challenges

As indicated earlier, at this point of our research process, we wanted to know if practitioners in real-world organizations experiences the QRs challenges and if so, how did they mitigate their effects (RQ2 and RQ3). Toward this end, we performed a qualitative exploratory study in the context of real world large-scale distributed agile projects in two different organizations, labeled as **O1** and **O2**. Both were selected purposefully based on their size and their rich experience with SAFe implementation. The first (O1) is a large Dutch government organization with about six years of SAFe experience. O1 has about 30000 employees spread over the whole country, however the IT department is located in one big building in one city. The agile teams of the IT department are distributed within this building. The software delivered by O1 is used by both individuals and companies to manage their taxes and allowances and is subject to strict legal regulations. The SAFe variant that O1 implemented is Portfolio SAFe [5] (it is explained more in detail in the next section). The second organization (O2) is an Indian large multinational consulting company with approximately 100000 employees. O2 is one of the biggest IT companies in the world. It operates in more than hundred locations across several countries. O2 is CMMI level 5 certified and has worked with SAFe since the introduction of SAFe in 2011. The agile teams of O2 are distributed across different locations and countries. The SAFe variant they implemented is Large Solution [5]. As already stated, by doing this interview based we aimed to answer RQ2 and RQ3. To collect the data, we conducted nine semi-structured, open- ended, in-depth interviews according to the guidelines of Boyce and Neale [24]. We chose the qualitative interview-based case study approach, because our desired depth of understanding could not be achieved meaningfully through the use of survey questionnaires and closed questions. As per Benbasat et al. [25], employing such a research method is a particularly suitable to research situations in which researchers study socially constructed processes in systems development projects and seek to achieve as good as possible grasp of reality. The selection of participants was based on the following criteria: (1) they all have at least 3 years of experience in SAFe and at least 10 years of IT experience, (2) they all had exposure to tasks related to the engineering of QRs, (3) they all were willingness to participate. They were drawn purposefully from the professional circle of the first and the third authors. As we were striving for a variety of roles to cover various perspectives, we included participants employed in different jobs (See Table 2). Four of the participants are from O1 and their interviews took place in the Netherlands, with the first author as the interviewer. The other participants are from O2 and the interviews took place in India, with the third author as the interviewer. The participants' roles in SAFe and their experience are described in Table 2. All interviews were in English and took around one hour. Our interview process included the following: (i) we asked each interviewee to choose a SAFe project in which he/she was actively involved in and in which QRs played a significant role, (ii) we showed the list of the reported QRs challenges [9] and asked the participant to indicate which ones he/she observed in his/her experiences in that SAFe project, and finally (iii) we asked which SAFe elements has each participant used to cope with the experienced QRs challenges. A list with the identified QRs challenges [9] was sent to all participants before the interview, in order to give them the chance to understand the challenges correctly.

Table 2. The participants in our interview-based study.

Participant ID	Organization	Role	Years of experience in IT
P1	O2	Technical lead	12
P2	O2	Agile coach	30
P3	O2	Solution architect	14
P4	O2	Agile coach	20
P5	O2	Project manager	13
P6	O1	Agile coach	15
P7	O1	IT manager	24
P8	O1	Product manager	10
P9	O1	IT manager	18

The interviews were audio-recorded and then thereafter transcribed by a professional company to avoid biases. The first two researchers read the transcripts separately and established a mapping between the by practitioners used SAFe elements and the QRs challenges they mitigate. Using coding [26] and Conklin's dialog mapping technique for qualitative data structuring [23], these researchers sorted out the qualitative data and mapped the participants' answers against the QRs challenges in Table 1. The results of this analytical activity are reported in Sect. 5.2. However, before that, we present the SAFe framework [5] as this is important for understanding the context of our research.

4 Scaled Agile Framework (SAFe)

SAFe is described by its authors [5] as a set of principles, practices and guidance that can be used by enterprises to deliver small solutions as well as complex systems in an agile way. It is a configurable framework with four variants as described below:

Essential Safe (ES) is the smallest SAFe configuration (between 50–150 practitioners) and the fundamental building block for all other SAFe configurations. The so-called agile release train (ART) is the fundamental concept of ES. It is a team of 5–12 agile teams where each team is a cross-functional group of 5–11 practitioners. Furthermore, ES consists of two levels, namely (1) team level that contains artefacts, events and processes an agile team needs to do in order to deliver value and (2) program level which contains all artifacts, events and processes needed to coordinate the work between the agile teams at the team level. The agile teams apply several agile practices (e.g. Scrum, Extreme Programming, Kanban) to deliver their part of the solution within a sprint. ART is further responsible for planning, committing, and deploying all the work together within a Program Increment (PI) (typically between 8 and 12 weeks). Scrum of Scrums is used to coordinate the dependencies between the different agile teams.

Large Solution SAFe (LSS) is a SAFe configuration for delivering large and complex systems without the need for portfolio and strategy alignment. In essence, LSS is an ES with additional roles (e.g. Solution train, Solution train engineer), and practices (e.g. Solution engineering, Solution management). The so-called solution train (ST) is the basis of LSS which coordinates the work of multiple ARTs by using Scrum of Scrums to deliver a complex system within a shared PI. As already said, O2 uses LSS.

Portfolio SAFe (PS) is concerned with managing value streams by aligning value streams to ARTs. Values are defined as Epics and managed by an Epic owner through the so-called portfolio kanban system. Value streams are the activities needed by an enterprise to deliver end-to-end customers' value. As stated earlier, O1 uses PS.

Full Safe (FS) is the complete SAFe configuration consisting of all previous configurations, which is used in large enterprises to align portfolios to very large solutions. We note that FS, while leveraging the elements of all three configurations, does not add any new elements.

5 Results

5.1 SAFe Elements Assumed Mitigate the QRs Challenges Defined in [9]

This section summarizes our findings in regard to RQ1. We found that SAFe as presented in [5] does recognize the importance of QRs. In fact, QRs are explicitly mentioned in all SAFe configurations. SAFe treats QRs as constraints on the backlog items or as restrictions on the software design and not as independent backlog items. In Tables 3, 4, 5 and 6, we present the results of our documentary analysis. These tables show, respectively, those elements – i.e. roles, artifacts, events, practices – of all four SAFe configurations which are supposed to influence the engineering of QRs, according to the SAFe textbook [5] and the documents at the official SAFe website. In each of these tables, column 1 represents the identified SAFe element which our analysis found to be a good candidate solution to use for coping with the QRs challenges from Table 1. The second column of each table shows the SAFe configuration in which the identified SAFe element is utilized (i.e. Essential SAFe (ES), Large Solution SAFe (LSS) and Portfolio SAFe (PS)). Finally, the third column gives a short description of the respective SAFe element. A dash "-" in a cell means that SAFe doesn't mention a value for that particular element in corresponding column. We note that elements which are related to Scrum [11] (e.g. sprint, sprint review) are not taken into consideration in our analysis for two reasons: 1) they are not specific to SAFe which is the subject of this study, 2) Scrum is originally designed for a small, single, co-located teams [3] while our focus is large-scale distributed agile teams.

Table 3. The identified SAFe roles that could help cope with QRs challenges

Roles	SAFe configuration	Description
Product management	ES	A team who is responsible for the program backlog
Solution management	LSS	A team who is responsible for the Solution backlog
Lean portfolio management	PS	A team who is responsible for the portfolio backlog
Epic owner	PS	The owner of an epic from the portfolio backlog that gets implemented
System engineer	ES	An individual who is responsible for defining and communicating the technical requirements where QRs might be part of within the ART
Solution engineer	LSS	An individual who is responsible for defining and communicating the technical requirements where QRs might be part of within the ST
Agile release train (ART)	ES	A team of agile team created around values which coordinate the activity of those teams to deliver customer values
Solution train (ST)	LSS	A SAFe construction to organize the work of multiple ARTs in order of delivering complex systems

Table 4. The identified SAFe artifacts that could help cope with QRs challenges

Artifacts	SAFe configuration	Description
Program backlog	ES	A repository of all features that need to be broken down to teams user stories
Solution backlog	LSS	A repository of all (supportive) activities needed to enable the implementation of other activities
Portfolio backlog	PS	A repository of the most abstract epics
Solution intent	LSS	A repository of all knowledges and requirements of the solution to be implemented
Enabler	ES, LSS, PS	It is e.g. epic, feature of user story needed for extending the architecture to meet certain requirements

As indicated in Sect. 3, we note that Full SAFe includes all other SAFe configurations (e.g. ES, PS and LSS) and does not describe any unique specific element which only exists when implementing the Full SAFe. That explains why we did not identify specific Full SAFe elements in Tables 3, 4, 5 and 6. After identifying the SAFe elements that could mitigate the QRs challenges identified in [9], the first two researchers mapped these elements (in Tables 3, 4, 5 and 6) to the reported categories of the challenges by using Conklin's dialog mapping technique for qualitative data structuring [23]. Table 7 summarizes this mapping. The first column of Table 7 shows the reported challenges, while the second column shows those SAFe elements that could be used to mitigate the related challenge in the first column. A dash "–" in the second column means that SAFe does not explicitly specify a particular element (e.g. artifact, role, event, practice) that could mitigate the reported QR challenge in the first column. SAFe describes elements of four types – i.e. artifacts, roles, events, practices – that could (at least partially) mitigate the QRs challenges reported in [9]. E.g., the different backlogs of the different SAFe configurations (e.g. Program backlog, Solution backlog) in combination with the Solution intent artifact could be used to eliminate confusions about conceptual definitions of QRs. SAFe however treats QRs in an unambiguous way as constraints on the product backlog items (PBIs) and not as independent PBIs. Besides, SAFe introduces several roles, namely: System engineer and Solution engineer, both possibly helping mitigate architectural challenges e.g. "Unmanaged architecture changes" and "Misunderstanding the architecture drivers" (see Table 1). However, we did not find appropriate SAFe elements that could mitigate the following two QRs challenges: "Lengthy QRs acceptance checklist", "Confusion about QR's specification approaches". We discuss this further in our Discussion section.

Table 5. The identified SAFe events that could help cope with QRs challenges

Events	SAFe configuration	Description
Program Increment (PI) planning	ES	It is a two to three whole day planning session where all agile teams of an ART come together to obtain a shared understanding of the business and fill in their own backlogs by breaking down the program backlog items including QRs. It is analog to Sprint planning
Inspect and adapt (I&A)	ES, LSS	At the end of each PI, the teams of an ART come together to inspect the delivered product, quality and de development process. It is analog to sprint review and retrospective
Scrum of Scrums (SoS)	ES	Representatives of all teams of an ART gather ate least once weekly for 30–60 min to discuss teams progress and dependencies
PO sync	ES	A 30–60 min weekly meeting for all POs and product management of all agile teams to discuss PI objectives, product features and scope
Pre and post PI planning (PPPI)	LSS	Two events of two whole days occurs before and after de PI planning to coordinate and follow-up work of various ARTs in relation to the Solution train
Portfolio sync	PS	It is a monthly event for the Lean Portfolio Management to discus epics implementation, addressing dependencies and removing impediments
Innovation and planning iteration (IPI)	ES	An iteration following the last iteration of an increment. It is used to experiment, inspect test results and plan possible technical changes

Table 6. The identified SAFe practices that could help cope with QRs challenges

Practices	SAFe configuration	Description
Set-based design (SBD)	LSS	A practice of delaying a definitive design decision as long as possible until all possible options are verified and validated
Model-Based Systems Engineering (MBSE)	LSS	A model-centric approach to define, design and document complex systems
Economic framework	PS	A set of decision guidelines that should be used to evaluate the feasibility of QRs
Architectural runway	LSS	The existing code, infrastructure and architectural guidelines to help teams enabling the implementation of near-term features
Quadrant 4	–	Agile test matrix that explains which tests should be performed by which way (e.g. manual, automated) to validate QRs

Table 7. Mapping SAFe elements to QR challenges

QR challenges [9]	SAFe elements
Late detection of QRs infeasibility	SoS, Program backlog, Solution backlog. Portfolio backlog. Solution intent. PO sync, Portfolio sync, PPPI, PI planning
Hidden assumptions in inter-team collaboration	SoS, ARTs, STs, PI planning, IPI
Uneven teams maturity	Economic framework, Architectural runway, I&A, IPI
Suboptimal inter-team organization	ARTs
Inadequate QRs test specification	MBSE
Lack of cost-effective real integration test	Quadrant 4
Lengthy QRs acceptance checklist	–
Sporadic adherence to quality guidelines	Economic framework, I&A, SoS
Overlooking sources of QRs	Product management, Solution management, Lean Portfolio Management, Epic owner, System engineer, Solution engineer, Solution intent, PO sync
Lack of QRs visibility	System engineer, Solution engineer
Ambiguous QRs communication process	System engineer, Solution engineer, SoS, ARTs, STs
Unclear conceptual definition of QRs	SoS, Program backlog, Solution backlog. Portfolio backlog. Solution intent
Confusion about QR's specification approaches	–
Unmanaged architecture changes	System engineer, Solution engineer, Architectural runway, SBD, Enablers
Misunderstanding the architecture drivers	System engineer, Solution engineer, Architectural runway, SBD, MBSE, Enablers

5.2 Answers to RQ2 and RQ3

This section summarizes our answers to RQ2 (Do the SAFe practitioners experience the QRs challenges identified in [9]?) and RQ3 (If yes which SAFe practices they utilize to mitigate these challenges?) The answers came out of our qualitative analysis of the data in our interview-based study. We note that not all challenges were observable by all participating practitioners. For each of them, we analyzed the challenges this practitioner experienced and the way SAFe practices were used to confront the challenges (as per this practitioner's experience). Using coding [26] and concept-mapping practices [23], the first two researchers mapped SAFe elements identified by the practitioners to the challenges reported in [9]. Table 8 summarizes this mapping. The first column of the table shows the reported categories and their related challenges, while the second column shows the SAFe elements used by the practitioners to mitigate the related challenge in the first column. A dash "–" in the second column means that no remedy has been identified by the practitioners.

Table 8. Mapping SAFe elements used by practitioners to mitigate their QR challenges

QR challenges [9]	SAFe elements
Late detection of QRs infeasibility	–
Hidden assumptions in inter-team collaboration	SoS
Uneven teams maturity	Architectural runway
Suboptimal inter-team organization	ARTs, Component teams
Inadequate QRs test specification	Team architects, Automated test
Lack of cost-effective real integration test	–
Lengthy QRs acceptance checklist	References to external documents
Sporadic adherence to quality guidelines	Automated tools
Overlooking sources of QRs	–
Lack of QRs visibility	Business owner, PO, Product management
Ambiguous QRs communication process	PI planning, SoS, Architectural runway, ARTs
Unclear conceptual definition of QRs	–
Confusion about QR's specification approaches	DoD, Product backlog, A generic ART's DoD
Unmanaged architecture changes	Team architects, Solution architect, Enablers, Architectural runway
Misunderstanding the architecture drivers	Enterprise architect, Solution architect, Enablers, Architectural runway

In the experience of our participants, there are several remedies to the identified QRs challenges. E.g., for the challenge "Ambiguous QRs communication process", the practitioners used PI planning, SoS, and Architectural runway to cope with it. In contrast, for other challenges (e.g. "Unclear conceptual definition of QRs") the practitioners did not identify any SAFe remedy. Moreover, even when practitioners belong to same organization, they had different views on some challenges. E.g., a practitioner from O2 did not perceive "Lengthy QRs acceptance checklist" as a challenge while another practitioner of the same organization recognized this challenge: *"If they put QRs in the DoD, it will be a challenge for developers. We do not put them in DoD"*. A practitioner from O1 recognized this challenge as well: *"In the Definition of Done we reference to QRs"*. The practitioners had different views as well regarding the challenge "Unclear conceptual definition of QRs". In their experiences, some of them treated QRs as PBIs: *"Performance is important for us, we described it as user stories"*, *"We treat both functional requirements and QRs as user stories"*. Other practitioners considered QRs as constraints on the PBIs and specified them in both the acceptance criteria and in the DoD *"Performance, it comes back in the acceptance criteria"*. These practitioners in fact were aligned with the methodological source of SAFe [5] which states to treat QRs as constraints on PBIs and advices to specify them in both the acceptance criteria and in the DoD of the PBIs. For the challenge *"Late detection of QRs infeasibility"*, the practitioners did not identify any remedy. In fact, they accepted the invisibility of the QR as a consequence of complex system development *"You are building a complex software and you find out that it is in practice slightly different than you had thought. I think it doesn't matter at all what method you use, whether it is SAFe, Scrum or waterfall, you just run into that and you have to deal with that"*.

Another practitioner explained that the *"Lack of cost-effective real integration test"* is difficult to mitigate: *"It is difficult to simulate complex production environment to obtain a reasonable integration test results."* Finally, we found that in the collective experience of our practitioners, there was no remedy for the challenge "Overlooking sources of QRs" as well. As a practitioner from O1 reported: *"Within our organization we have more than six hundred systems that communicate with each other. Each of each has its own stakeholders. It is almost impossible to have an overview of all stakeholders"*.

6 Discussion

Table 7 maps the SAFe elements we (i.e. the authors) identified as remedy to the QRs challenges reported in [9], while Table 8 maps those SAFe elements actively used by practitioners to mitigate the QRs challenges. We compared both tables and noticed the contrast between the remedies assumed in the literature of SAFe and the remedies reported by the practitioners. E.g., we have identified Quadrant 4 and Economic framework as possible remedies for, respectively *"Lack of cost-effective real integration test"* and *"Sporadic adherence to quality guidelines"*. However, no participant has mentioned those SAFe elements as possible remedies for any of the reported challenges in [9], despite the use of the Portfolio SAFe configuration by their organization (O1) or the Large Solution SAFe configuration (O2). We think that this discrepancy can be explained by the fact that the implementation of SAFe in real life differs from its theory. In line with this thought, practitioners reported the use of different non- SAFe elements: *"We have a preparation team which is not a SAFe-construct. It consists of information managers who prepare the work for the agile teams. Those managers were not willing to anticipate in agile teams so we created a team for them to keep calmness"*. *"We are fan of silo's, so we have a lot of architect's flavor e.g. a business process architect, a process architect and an IT architect"*. The tendency of introducing non-SAFe elements into what was supposed to be SAFe, in the experience of our practitioners, was traceable to the *"long and consistent waterfall experience"* of our case study organizations in large-scale systems delivery projects. As one of our case study organizations was a CMM5-certified, their adoption of SAFe happened in a highly disciplined environment in which SAFe co-exists with metrics, measurement procedures, progress tracking practices and milestone-compliance practices. For this reason, it seems logical that if practitioners see a QR problem they might be open to resort to a heavyweight practice that they knew to work well from their pre-agile professional experience. Moreover, one could argue that our interviewees experienced QRs challenges because the SAFe remedies were not implemented with the due discipline and commitment, e.g. some practices might have been too much *"free-styled",* which means being used in an idiosyncratic way. While this might well be possible, we think that for a large-scale organization it might not be realistic to assume a strict SAFe implementation as per the SAFe textbook [5]. More research is needed to substantiate this.

Furthermore, we found that despite the fact that our case study organizations did employ some SAFe elements in their coping strategies for QRs, no practitioner indicated that this was done in a cost-effective way. This let us think that it might be possible that SAFe and more heavyweight methods could both be comparably expensive approaches. This seems unsurprising, knowing that dealing with QRs represents an expensive part of any large-scale project [15]. Next, we found that SAFe [5] treats QRs as constraints on the backlog items or as restrictions on the software design. This observation agrees with the treatment of QRs in heavyweight life cycle models (e.g. [27, 28]) where QRs are operationalized into functional requirements and architecture design choices. Next, it's worthwhile noting that our practitioners experienced SAFe as a complex framework. A practitioner from O1 explains *"SAFe is very precisely defined, almost a work instruction. It is a complex framework, have you seen how big is the SAFe book in comparison to other agile scaled frameworks' books?."* This reflection is in line with observations of [29, 30] regarding the SAFe's complexity.

Finally, we found that SAFe integrated different elements that could be recognized as heavyweight non-agile elements such as Model-Based Systems Engineering. This finding is in line with [9]. The authors of [9] reported that heavyweight practices often get integrated in organizations' agile way of working to encounter QR's challenges.

7 Threats of Validity

This section discusses the possible validity threats [13] to our empirical research design and our findings. First, we employed documentation research method [12] which is a reflexive approach to the analysis of documents that contain information about the phenomenon we wish to study. There is a possible validity threat related to the types of documents and the ability to use them as reliable sources of evidence on the social world. We think this threat is minimal because we used the most important repository of documents of the SAFe community of practitioners. The documents we reviewed form the basis for this community's education and certification programs. We were also conscious of researchers' bias. However, we mitigated this by including two academic researchers (the second and the fourth co-authors) next to the two industry researchers (the first and the third authors). This diversity of backgrounds was used to remove any bias and checking tacit assumptions about the reviewed documents. Furthermore, our interview study included 9 practitioners in two organizations. Their perceptions and experiences can not be generalized to all possible large-scale enterprises using SAFe. However, because these practitioners were with various roles and operated in very different cultural contexts (India and the Netherlands), we think that they cover a broad range of professional perspectives to QRs in SAFe. Acknowledging this diversity, we think it might well be possible to have similar findings if we included more participants from organizations similar to O1 and O2. As stated in [31], contextual similarity in terms of process-oriented thinking, experience with both agile and heavyweight development practices in large and very large projects, might lead to similar organizational mechanisms that might produce similar effects in other similar but different organizations (e.g. government agencies with 30000+ employees and large companies).

8 Conclusion

This paper investigated the possible remedies suggested by SAFe [5] to mitigate the effects of QRs challenges identified in a previous study [9]. We have examined SAFe with respect to those QRs challenges by (i) performing documentary analysis where the official SAFe literature has been investigated and (ii) conducting an interview-based study to understand the SAFe elements used by practitioners to cope with the QRs challenges. Our results show that SAFe contains 25 elements (e.g. roles, artifacts, events, practices; see Tables 3, 4, 5, 6) that are assumed to be used to mitigate the impact of the identified QRs. Nine of those elements were also mentioned by the practitioners involved in the interview-based study (see Table 8). We think that SAFe practitioners, due to the complexity of the SAFe framework, implement SAFe differently from the SAFe textbook [5]. In our opinion this difference in implementation could be the reason for the discrepancy between the SAFe elements identified through our documentary analysis and those identified in our exploratory interview-based study. Our follow-up research step will be to extend this work by investigating the actual reasons behind the discrepancy between the practical implementation of SAFe in real-world projects and the suggested implementation described by the SAFe authors [5]. The follow-up study will help us define a practical set of remedies to the QRs challenges identified in [9].

References

1. Smart, J.: To transform to have agility, don't do a capital A, capital T agile transformation. IEEE Softw. **35**, 56–60 (2018)
2. Conboy, K., Carroll, N.: Implementing large-scale agile frameworks: challenges and recommendations. IEEE Softw. **36**, 1–9 (2019)
3. Kalenda, M., et al.: Scaling agile in large organizations: practices, challenges, and success factors. J. Softw. Evol. Process **30**, e1954 (2018)
4. Bick, S., et al.: Coordination challenges in large-scale software development: a case study of planning misalignment in hybrid settings. IEEE Trans. Softw. Eng. **44**, 932–950 (2018)
5. Richard, K., Leffingwell, D.: SAFe 5.0 Distilled Achieving Business Agility with the Scaled Agile Framework, 1st edn. Pearson Education, London (2020)
6. Larman, C., Vodde, B.: Large-Scale Scrum more with Less. Pearson Education, London (2016)
7. Sutherland, J.: The Scrum@Scale guide - the definitive guide to Scrum@Scale: scaling that works. In: Scrum@Scale, pp. 1–19 (2019). https://www.scrumatscale.com/scrum-at-scale-guide/
8. Paasivaara, M., et al.: Adopting SAFe to scale agile in a globally distributed organization. In: ICGSE 2017, pp. 36–40 (2017)
9. Alsaqaf, W., et al.: Quality requirements challenges in the context of large-scale distributed agile: an empirical study. Inf. Softw. Technol. **110**, 39–55 (2019)

10. COLLAB.NET and VERSIONONE.COM: 14th Annual State of Agile Report. VersionOne (2020). https://stateofagile.com/?_ga=2.145189495.276092471.1591726593-1008038165.1591726593#ufh-i-615706098-14th-annual-state-of-agile-report/7027494
11. Schwaber, K., Sutherland, J.: The scrum guide. Scrum.Org and ScrumInc, p. 19 (2017)
12. Appleton, J.V., Cowley, S.: Analysing clinical practice guidelines. A method of documentary analysis. J. Adv. Nurs. **25**, 1008–1017 (1997)
13. Yin, R.K.: Case Study Research Design and Methods. 5th Revise. Sage Publications Inc. (2013)
14. Alsaqaf, W., Daneva, M., Wieringa, R.: Quality requirements in large-scale distributed agile projects – a systematic literature review. In: Grünbacher, P., Perini, A. (eds.) REFSQ 2017. LNCS, vol. 10153, pp. 219–234. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54045-0_17
15. Kasauli, R., et al.: Requirements engineering challenges and practices in large-scale agile system development. J. Syst. Softw. **172**, 110851 (2021)
16. Moyon, F., et al.: How to integrate security compliance requirements with agile software engineering at scale? In: Morisio, M., Torchiano, M., Jedlitschka, A. (eds.) PROFES 2020. LNCS, vol. 12562, pp. 69–87. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64148-1_5
17. Beecham, S., et al.: Do scaling agile frameworks address global software development risks? An empirical study. J. Syst. Softw. **173**, 110823 (2021)
18. Ambler, S.W., Lines, M.: Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. IBM Press (2012)
19. Wagner, T.J., Ford, T.C.: Metrics to meet security & privacy requirements with agile software development methods in a regulated environment (2021)
20. Portman, H.: Scaling Agile in Organisaties. Van Haren Publ. (2017)
21. Kniberg, H., Ivarsson, A.: Scaling agile @ spotify - with tribes, squads, chapters & guilds (2012)
22. Hitchcock, D.: The practice of argumentative discussion. Argumentation **16**, 287–298 (2002)
23. Conklin, J.: Dialog mapping: reflections on an industrial strength case study. In: Kirschner, P.A. et al. (eds.) Visualizing Argumentation. CSCW, pp. 117–136. Springer, London (2003). https://doi.org/10.1007/978-1-4471-0037-9_6
24. Boyce, C., Neale, P.: Conducting in-depth interviews: a guide for designing and conducting in-depth interviews. Evaluation **2**,1–16 (2006)
25. Benbasat, I., et al.: The case research strategy in studies of information systems. MIS Q. 369–386 (1987). https://www.jstor.org/stable/248684?seq=1#page_scan_tab_contents
26. Charmaz, K.: Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis. Sage (2006)
27. Kassab, M., et al.: An ontology based approach to non-functional requirements conceptualization. In: 4th International Conference on Software Engineering Advances, ICSEA 2009, pp. 299–308 (2009)
28. Mart, S., et al.: Dealing with non-functional requirements in model-driven development : a survey. IEEE Trans. Softw. Eng. **45**(4), 818–835 (2021)
29. Putta, A., Paasivaara, M., Lassenius, C.: Benefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review. In: Kuhrmann, M., et al. (eds.) PROFES 2018. LNCS, vol. 11271, pp. 334–351. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03673-7_24
30. Ebert, C., Paasivaara, M.: Scaling agile. IEEE Softw. **34**, 98–103 (2017)
31. Seddon, P., Scheepers, R.: Towards the improved treatment of generalization of knowledge claims in IS research: drawing general conclusions from samples. Eur. J. Inf. Syst. **21**, 6–21 (2011)