Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework

Oktay Turetken1,*,†, Igor Stojanov2 and Jos J. M. Trienekens1

1Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, Eindhoven, Netherlands
2Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, Netherlands

ABSTRACT

Although the agile software development approaches have gained wide acceptance in practice, the concerns regarding the scalability and integration of agile practices in traditional large-scale system development projects are prevailing. Scaled Agile Framework (SAFe) has emerged as a solution to address some of these concerns. Despite few encouraging results, case studies indicate several challenges of SAFe adoption. Currently, there is a lack of a well-structured gradual approach for establishing SAFe. Before and during SAFe adoption, organizations can benefit greatly from a uniform model for assessing the current progress, and establishing a roadmap for the initiative. To address this need, we developed a maturity model that provides guidance for software developing organizations in defining a roadmap for adopting SAFe. The model can also be used to assess the level of SAFe adoption. We took an existing agile maturity model as a basis for agile practices and extended it with practices that are key to SAFe. The model was developed and refined with industry experts using the Delphi technique. A case study was conducted in a large organization where we evaluated the model by applying it to assess the level of SAFe adoption. © 2016 The Authors. Journal of Software: Evolution and Process Published by John Wiley & Sons Ltd.

Received 6 May 2015; Revised 11 April 2016; Accepted 18 May 2016

KEY WORDS: agile software development; agile development in large-scale; Scaled Agile Framework; maturity model; Delphi study

1. INTRODUCTION

The software development methodologies have evolved from predictive (such as waterfall) to iterative and incremental (such as Rapid Application Development and Rational Unified Process), and to agile approaches (such as Scrum, and Extreme Programming). As the popularity of agile adoption increases, the questions organizations ask themselves shifts from why to adopt agile practices to how to adopt and scale these practices. Although a large number of organizations have adopted agile practices, these approaches are often criticized of being applicable primarily to small teams and organizations rather than large enterprises with several hundreds of development teams [1]. The difficulty of adopting agile practices increases when there is a need to scale these practices. The scalability in software engineering can be defined as the property of reducing or increasing the scope of methods, processes, and management according to the problem size [2].

Despite important contributions in the academic literature, large projects, and processes, and organizational and governance aspects continue to challenge large software developing enterprises.

*Correspondence to: Oktay Turetken, Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, Eindhoven, Netherlands.
†E-mail: o.turetken@tue.nl

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2016 The Authors. Journal of Software: Evolution and Process Published by John Wiley & Sons Ltd.
The existing research suggests a need for augmenting basic agile practices to fit large settings, and providing guidance on their use and scalability.

In an attempt to scale the advantages of agile methodologies, a number of frameworks have been proposed to provide guidance for scaling agile development across the enterprise. One of the commonly known models is the Scaled Agile Framework (SAFe)\(^1\) [3, 4]. Despite some criticisms, SAFe has gained a rapid attention in the practice and has become an important choice for organizations that are in need of approaches for scaling agile development. It addresses the scalability not only by scaling up ‘some’ of the agile practices, but also by introducing new practices and concepts (such as release train, business, and architecture epics, portfolio backlog) that integrate with basic and scaled agile practices. It aims multiple benefits such as accelerated time-to-market, increased productivity and quality, and reduced risks and project costs. Although there are examples of successful SAFe adoptions that argue to have verified some of these benefits, these stories are typically narrowly focused and self-reported. The emerging growth of SAFe in industry and practice requires an academic attention.

Our review of the main sources of SAFe ([3, 4]), however, suggests a lack of a structured roadmap that can guide enterprises on the necessary preparation and adoption of SAFe. The SAFe focuses merely on describing the best practices, roles, and artifacts of agile and lean principles but makes no attempt to describe any implementation strategy or method [3]. Companies that aim to adopt SAFe can find it difficult to identify the priorities and to lead the efforts for implementing agile and SAFe practices.

A maturity model with a structured collection of agile and SAFe practices, including the dependencies between these practices, would help organizations in defining a roadmap for agile/SAFe adoption. In order to identify a prioritized roadmap for SAFe adoption, it is important to understand which practices the organization currently performs well, and which practices it does not. It is critical to periodically assess the extent by which these practices are successfully adopted and can be improved. A maturity model acts as a basis for such assessments. The objective of this study, therefore, is to introduce the SAFe Maturity Model (SAFe MM) that allows for assessing the level of SAFe adoption and helps in defining a roadmap for the implementation of agile and SAFe practices in an enterprise. As such, the specific research question that we address is

How to design a maturity model that can be used as a guideline by software developing organizations to adopt SAFe and assess the success level of SAFe adoption?

We took an existing agile maturity model as a basis in the development of the initial version of the SAFe MM. The final version is developed as a joint effort of industry experts, using the Delphi technique. The Delphi technique is a structured approach for soliciting expert opinion on a particular topic. This technique is particularly useful for this type of research as it is well aligned with exploratory theory building on complex, interdisciplinary issues, often involving a number of new or future trends [5]. To observe the validity of the SAFe MM, we applied it in a case organization to assess the achieved level of adoption of SAFe practices and reported on the steps to be taken to proceed to higher levels of achievement.

The remainder of the paper is organized as follows: Section 2 presents a brief overview of the SAFe and a discussion of maturity models for agile development. Section 3 describes the research procedure and methods that we applied in this study. In Section 4, we introduce the SAFe MM together with the elaboration of the method we followed in its construction. Section 5 presents the case study and findings regarding the application of SAFe MM in an organization. Finally, Section 6 presents our conclusions, limitations, and outlook for future research.

2. BACKGROUND AND RELATED WORK

2.1. Agile development in the large

The success of agile methods for small, co-located teams has inspired the use of agile practices in the large-scale [1]. The term ‘large-scale agile development’ has been associated with different structures

\(^1\)SAFe and Scaled Agile Framework are registered trademarks of Scaled Agile Inc.
and project settings. It typically refers to the development in large teams and/or large multi-team projects that make use of agile principles and practices in an (possibly large) organization [6].

The increasing popularity of agile methods has also brought criticism on the scalability of agile. Some prevailing views argue against the use of agile methods outside their original domain of fit [1]. Large projects and teams have different dynamics and social considerations. Coordination of work between multiple teams that are possibly distributed, architecture and planning in large projects are just few aspects that are not in the context of agile development in smaller scale. Several researchers agree that guidelines and adaptions of agile practices might be necessary when scaling along the dimensions of project size, complexity, and dispersion of team members [7]. As such, scalability of agile is arguable, as a method is scalable only if it can be applied to problems of different sizes without fundamentally changing it [2].

The arguments on this topic have continued to bring the attention of the practice and academia [1, 6, 8]. Increasing number of large and critical projects and organizations has adopted agile methods [9]. The practitioners and researchers have considered the topic as one of the top research challenges that requires further attention [1, 10].

Few works (such as [11–14]) discuss the challenges and offer guidance on the scalability of agile practices in large organizations, particularly in those that are transitioning from traditional approaches. Lindvall et al. [15] discusses on the experience of large companies in employing agile practices in large settings. They conclude that tailoring agile practices to the particular requirements of large organizations is an absolute necessity. The study showed that the greatest challenge lies not in the agile practices or the new practices put in place, but in the interface and coherence between these practices and the existing organizational processes. Another challenge is to establish support for cross team communication because large organizations often distribute teams across several physical locations, and agile practices do not address problems arising from communication and coordination between multiple teams. The works by Petersen and Wohlin [16–18] make use of case studies to conduct empirical research on the impact of incremental and agile practices in large enterprises.

The literature also reports on the metric models to help measure the impact of agile transformations in large organizations [19]. For instance, customer service request turnaround time and cycle-time for feature metrics are proposed to measure responsiveness and lead-time. Similarly, number of external trouble reports and average number of days that these reports have been unsolved are metrics suggested to address the quality of the product developed.

With the goal to address the scalability challenge, a number of frameworks has emerged that claim to offer the much needed guidance for scaling agile to the large-scale development efforts, where there are several development teams with tens or hundreds of software practitioners working in multi-year projects or programs. Most commonly known models include the Disciplined Agile Delivery [20], Large Scale Scrum (LeSS) [21, 22], and SAFe [3, 4].

Disciplined Agile Delivery, proposed by Scott Ambler, incorporates a range of lean principles and lightweight practices that are built upon a set of core agile practices. It claims to address traditional challenges of complex enterprise structures and geographically dispersed agile teams by scaling agile practices across the system lifecycle [20]. LeSS is championed by Craig Larman and Bas Vodde, and focuses on scaling Scrum over multiple teams [22]. It offers two versions: LeSS, for projects up to eight teams, and LeSS Huge, for hundreds of people working on a single product.

SAFe is developed with the aim to help organizing and managing software development in large enterprises. It comes with a package of supporting tools, books, training materials, and certification schemes supported by substantial marketing effort. SAFe has received significant attention in the agile community, including some criticisms. Albeit the sound lean principles at the core, many raised concerns about the extent by which the agile principles and values are represented, the large-scale planning, the top-down approach, and the strong emphasis on process rather than people [23, 24]. Whilst these criticisms, a number of companies that have applied SAFe reported on their experience and claimed to have achieved significant benefits [25–27]. We elaborate more into the structure, properties, and industry experiences of SAFe in Section 2.2.

These advances, however, have not alleviated the corporate concerns about the risks and challenges of scaling agile at the enterprise level. Several studies (such as [12, 13, 17, 28–30]) have outlined the risks and challenges involved in agile transformations, and others (such as [31–33]) have proposed...
some critical factors and practices, as well as maturity models for agile transformation. Project size, for instance, is considered as a top contextual factor that has the greatest risks of derailing agile projects [34, 35]. Although these studies provide important contributions in helping companies to adopt agile practices outside of the context for which they have been created, there is still a need for structural approaches and methodological guidelines for scaling agile in large settings, in particular for SAFe.

2.2. Scaled Agile Framework

Scaled Agile Framework aims to incorporate the practices of agile and lean principles at the enterprise level. Currently, SAFe is being documented in version 3 of the Big Picture Framework and is publicly available.

Figure 1 presents the SAFe Enterprise Big Picture, which is a visual representation of the framework to serve as both an organization and a process model for agile requirements practices [1]. SAFe aims to integrate existing bodies of work of Scrum, XP, Lean, and Product Development Flow.

In brief, the framework is separated into 3 levels, namely team, program, and portfolio levels. The boundaries between these three levels are arbitrary and serve as a model for abstraction of the scope and the scale between levels.

The team level of the framework consists of agile teams, which are collectively responsible for defining, building, and testing software in fixed-length iteration and releases. The SAFe framework on this level contains a blend of agile project management practices (Scrum) and agile technical practices (XP). For instance, the concept of user stories is borrowed from XP, while sprint planning, daily stand-ups are typical Scrum components. ‘Definition of done’ and retrospectives are adopted at each iteration. Teams operate on an identical cadence and iteration lengths in order to provide better integration among teams. These agile teams typically consist of 7 ± 2 team members [3].

![Figure 1. Scaled Agile Framework® Enterprise Big Picture v3](image-url)
The primary goal at the program level is to organize the agile teams at scale in order to optimize the value delivery of requirements. Furthermore, the program level also aligns the teams with a strategic vision and roadmap for each investment theme. At this level, business and architectural features are defined and prioritized in the program backlog. A major concept introduced at this level is the agile release train, which provides cadence and synchronization. The agile release train produces releases or potentially shippable increments at fixed time boundaries, typically 60–120 days [3]. The releases are planned during a 2-days release planning event, which involves all relevant stakeholders. Furthermore, a system team is formed in order to establish an initial infrastructure and to support continuous integration and end-to-end testing efforts.

The highest level in SAFe is the portfolio level, where programs are aligned to the enterprise business strategy along value stream lines. Value streams are long-lived series of system definition, development, and deployment steps used to build and deploy systems that provide continuous flow of value to the business or customer. This level is needed for enterprises, which require governance and management models. The essence lies in achieving a balance between four potentially conflicting goals [36]:

- maximizing the financial value of the portfolio by identifying value streams using Kanban systems,
- linking the portfolio to the strategy of an organization through investment themes,
- ensuring that the scope of activities is feasible by measuring appropriate metrics, and
- balancing the portfolio on relevant dimensions by defining and managing business and architectural epics, which run across value streams. Epics capture the largest initiatives in a portfolio.

Business epics describe functional or user-experience epics, while architectural epics capture the technological changes that must be made to keep the systems flowing.

Since the introduction of the SAFe in 2011, a number of companies have applied the framework and published their experiences on SAFe adoption as white papers or technical reports [25–27]. These reports state improvements in several directions such as higher ROI, 20–30% faster time to market, 40–50% decrease in post release defects, better alignment with customer needs, and increase in productivity of 20–50%, within short periods of time, that is, in the order of months. The reports also outline the challenges such as staying releasable throughout the development lifecycle due to the late discovery of defects, and defining the right level of requirement detail at the right time during the lifecycle. The studies claim that a proper preparation, orchestration, and facilitation of distributed program events are essential for successful release planning. The studies also report with confirmation that geographically distributed teams experience lower productivity because of lack of alignment and solid program execution.

2.3. Maturity models for agile development

In practice, organizations are unable to fully adopt agile development practices immediately or over a short period of time [37]. Maturity models can guide organizations in providing the directions concerning the practices and the manner that they can be introduced and established in the organization. A maturity model is a conceptual framework that comprises a collection of best practices that help organizations to improve their processes in a particular area of interest [38].

The basic purpose of maturity models is to outline the stages of maturation paths. Also based on the assumptions of predictable patterns of organizational evolution and change, maturity models typically represent theories about how an organization’s capabilities evolve in a stage-by-stage manner along an anticipated, desired, and logical path [39]. In general, maturity models are characterized by a limited and ordered number of maturity levels, and each maturity level defines characteristics or practices which have to be achieved at each level [40].

In a systematic review of literature, Schweigert et al. [41] identifies around 40 maturity models for agile development. The analysis concluded that, despite an urgent need for agile maturity models, currently none of the models proposed in the literature is widely referred in practice and academia. In addition, there is no model that addresses the practices, principles, and challenges of scaling agile in large settings.
Ozcan Top and Demirors [42] presents a case study based comparison of nine agile maturity models in terms of their fitness of purpose, completeness, definition of agile levels, objectivity, correctness, and consistency. In their study, they applied each model in a single case organization that is in the process of adopting agile practices in developing software. The assessment criteria included: fitness for purpose (whether a model’s emphasis is on assessing agile practices or not), completeness (the extent by which a model addresses major engineering and management processes within a software development life cycle), structure (whether a model defines levels which enumerate the different degrees of agility), objectivity (whether an assessment using a model is verifiable, traceable, and reproducible), correctness (the extent by which a model is aligned with agile principles), and consistency (whether a model is free of logical and temporal conflicts). Accordingly, the model proposed by Sidky (Sidky Agile Measurement Index [SAMI]) scored the highest.

The SAMI model is structured into four components (as presented in Table I): agile levels, agile principles, agile practices and concepts, and indicators [32]. Driven from the four values and 12 principles of the Agile Manifesto [43], the model defines five agile levels. Collaboration is one of the essential values and qualities of agile, and thus it is enumerated as Level 1. Developing software through an evolutionary approach is the next key principle in agile and the objective at Level 2 in SAMI. Effectiveness and efficiency in developing high quality, working software is the objective at Level 3. The key objective at Level 4 is to gain the capability to respond to change through multiple levels of feedback. Establishing a vibrant and all-encompassing environment to sustain agility is the objective at Level 5.

The SAMI model clusters 12 agile principles into five categories that group the agile practices. These principles are: (i) embracing change to deliver customer value; (ii) plan and deliver software frequently; (iii) human-centricity; (iv) technical excellence; and (v) customer collaboration.

Agile practices are a set of techniques or methods that are used for developing software in a manner that is consistent with the agile principles. Each agile level contains practices, which – when adopted – collectively leads to significant improvements in agility. These practices are also categorized under the agile principles. Table I shows the relationship between these components. The model incorporates 40 agile practices in total. Organization should adopt agile practices on lower levels first, because the agile practices on a higher level are dependent on the practices introduced at the lower levels.

Indicators are used to assess certain characteristics of an organization or project, such as its people, culture, and environment, in order to ascertain the readiness of the organization or project to adopt an agile practice.

3. RESEARCH DESIGN

In this study, we followed a design science research approach [44], as our primary objective is to develop a new software engineering artifact, that is, the SAFe MM (that can be used to assess the level of SAFe adoption in organizations that have been implementing SAFe or similar approaches). Accordingly, the research approach involves mainly (i) the definition of the problem and the objectives of the artifact, (ii) the design and development of the artifact, and (iii) the evaluation of the artifact in real life settings [45]. In Section 1, we discussed the problem and the main objective of the artifact (item i) as to assess the maturity level of SAFe adoption. The design and development of the SAFe MM (item ii) is described in Sections 4. In section 5, we present the evaluation of the SAFe MM (item iii), where we applied it in a case organization to assess its validity.

Figure 2 depicts a detailed view on the procedure that we followed in developing, applying, and evaluating the SAFe MM proposed in this paper. The first phase involved the construction of the SAFe MM. The initial version of the maturity model was constructed through a synthesis of various concepts and best practices aligned with main SAFe sources (i.e., [3, 4]). However, to increase the relevancy of the proposed model and ensure the content validity, we performed a Delphi study with a panel of domain experts to refine and finalize the model. The key benefit of this method is that it uses group decision-making techniques while involving experts from the field, which increases the validity of the research [46]. Furthermore, the anonymity of the participants and the discussions resolves the difficulties commonly associated with group interviews (such as the effects of dominant
Table I. Structure of the SAMI Model [32]. Agile levels are populated with agile practices, which are categorized within agile principles.

<table>
<thead>
<tr>
<th>Agile levels</th>
<th>Agile principles</th>
<th>Agile principles</th>
<th>Agile principles</th>
<th>Agile principles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Embrace change to deliver customer value</td>
<td>Plan and deliver software frequently</td>
<td>Human centrality</td>
<td>Technical excellence</td>
</tr>
<tr>
<td>Level 5 Encompassing</td>
<td>Low process ceremony</td>
<td>Agile project estimation</td>
<td>Ideal Agile physical setup</td>
<td>Test driven development</td>
</tr>
<tr>
<td>Level 4 Adaptive</td>
<td>Client driven iterations Continuous customer satisfaction</td>
<td>Smaller and more frequent releases Adaptive planning</td>
<td>Continuous customer satisfaction</td>
<td>Continuous integration Continuous improvement (refactoring)</td>
</tr>
<tr>
<td>Level 3 Effective</td>
<td>Risk driven iterations Plan features not tasks Backlog</td>
<td>Self-organizing teams Frequent face to face communication</td>
<td>Continuous integration Continuous improvement (refactoring)</td>
<td>Continuous integration Continuous improvement (refactoring)</td>
</tr>
<tr>
<td>Level 2 Evolutionary</td>
<td>Evolutionary requirements Planning at different levels</td>
<td>Continuous delivery</td>
<td>Unit tests 30% of level 2 and level 3 people</td>
<td>Software configuration management Tracking iteration progress No big design up front</td>
</tr>
<tr>
<td>Level 1 Collaborative</td>
<td>Reflect and tune process Collaborative planning</td>
<td>Empowered and motivated teams Collaborative teams</td>
<td>Coding standards Knowledge sharing Task volunteering</td>
<td>Customer commitment to work with development team</td>
</tr>
</tbody>
</table>
individuals [47]). The Delphi study allowed us to involve experts in the actual development of the model, which is critical and can essentially be considered as an important part of the artifact evaluation [48].

The second phase involved a case study where the model was applied to assess the adoption level of SAFe practices in an enterprise, which is in the process of adopting SAFe. Case study research is one of the most commonly used methodologies in both software engineering and information systems research [49]. Sections 4 and 5 describe the two phases that were carried out, respectively.

4. DEVELOPMENT OF THE SCALED AGILE FRAMEWORK MATURITY MODEL

We developed the SAFe MM in close cooperation with the industry experts, following a series of structured steps. First, taking an existing agile maturity model as a basis, we developed an initial version of the model. We extended the agile maturity model significantly with SAFe related practices that are published in relevant sources. Second, the initial model was evaluated and refined through a Delphi study with two rounds of feedback. A panel of industry experts was gathered for the Delphi study in order to elicit domain expertise and facilitate consensus on the domain knowledge and eventually on the maturity model.

4.1. Development of the initial model

After an extensive review of the scalability of the agile approaches, SAFe and relevant case studies, and software process oriented maturity models with emphasis on the agile approaches, we took the SAMI agile maturity model [32] as the starting point (mainly due to its comprehensive and well-organized structure, which is also confirmed by the literature [42]). We reviewed the agile practices offered by the SAMI model and evaluated their applicability as practices that SAFe implements mostly at the team level. Therefore, the SAMI model provided the agile practices that the SAFe approach requires at the team level. We adopted these agile practices as the basis for the maturity model to address the SAFe team level practices.

Next, we adapted and extended the SAMI model in accordance with the SAFe principles and practices defined in the main sources of SAFe, namely [3] and [4]. Hence, in addition to the 40 original SAMI agile practices, we defined 19 SAFe practices that were incorporated in the initial version of the SAFe MM. Both the SAMI originated agile practices and the SAFe practices that we introduced in the initial version of the model went through a review and refinement using a Delphi study as described in the following section. We review these refinements and changes also with respect to the original agile practices (of the SAMI model) in Section 4.2.3.
4.2. Model refinement using the Delphi method

The Delphi study that we organized consisted of two rounds, with a timespan between the two rounds of about 30 days. Seven SAFe and agile experts participated in both the first and second round of the study. Online questionnaires were used for collecting data during the rounds.

An important aspect of the Delphi study is the selection of the panel of experts. The panel typically consists of academic and industrial experts in order to balance the views from both theoretical and practical perspectives. However, because of the high practical relevance of the topic and to the fact that SAFe is a recent development that has not yet gained sufficient attention in the academia, the panel of experts comprised of industry professionals. All participating experts are agile coaches with multi-year experience in practical implementation of agile practices, including agile adoptions in large enterprises. Six experts (out of 7) are also SAFe Program Consultants. SAFe Program Consultants are internal change agents in an enterprise, who have domain expertise on SAFe implementation and are qualified to launch such initiatives including training management and practitioners in the organization on SAFe practices. Because of the homogeneity and internal nature of the group, only two rounds were deemed appropriate to reach a consensus. Figure 3 shows the profile of the experts in terms of the rating regarding their experience with agile and SAFe practices.

4.2.1. Delphi study round 1. The aim of the first round of the study was to elicit broad comments from the panel of experts. The initial maturity model was presented and a document was sent to the panel members, which described the model in detail including the properties of each maturity level and practice. The experts reviewed each practice to assess if it can stay as described, change (the description or position in the model in terms of the maturity level and/or principles), or has to be removed from the model. In case of change or removal, the experts were asked to provide explanations. In addition, they were subject to detailed questions regarding whether the five agile levels are sufficient and if the SAFe and agile practices are complete and aligned appropriately. In cases when experts disagreed, they were asked to elaborate their responses and rationale based on their practical experience and expertise.

4.2.2. Delphi study round 2. The primary benefit of multi-round Delphi technique is the fact that the selected Delphi participants are able to reassess their initial views after seeing the results from the previous round and to reach a consensus. Besides, because the results are presented anonymously, the effects of dominant individuals typically associated with group-based interviewing are diminished. During the second round the expert panel was confronted with results from the first round and requested to reach consensus on the proposed improvements. The improvement revisions were grouped into three categories: fundamental, additive, and corrective revisions. Fundamental revisions addressed the essential changes proposed by the experts and used for a further clarification of the existing agile and SAFe practices. Additive revisions covered the additions of SAFe practices, which were proposed during Round 1. The corrective revisions included the corrections of the existing elements, which mainly focused on aligning practices under the correct maturity levels and principles.

Figure 3. Profile of the experts participated in the Delphi study.
4.2.3. Changes in the initial model. Based on the feedback gathered in the Delphi study rounds, several alterations were done on the initial model, involving both agile practices adopted from the SAMI model and SAFe practices that were defined to address other SAFe requirements. The following are the key changes that are performed on the agile practices adopted from the SAMI model.

- Two agile practices – ‘paired programming’ and ‘agile documentation’ – were removed as they were misaligned with the SAFe principles and practices and were not applied in SAFe.
- Two agile practices were renamed to match the terminology used in SAFe. The practice ‘planning at different levels’ was renamed to ‘two level planning and tracking’ based on the SAFe naming convention. Similarly, the concept of ‘backlog’ changed into the ‘product backlog’ to provide a better representation of SAFe concepts.
- Two agile practices - ‘user stories’ and ‘product backlog’ were moved to different maturity levels because they were considered to provide the basis for other practices at higher maturity levels. The agile practice regarding user stories, which is originally at level 4 in the SAMI model, was moved to level 1. This was mainly because it was considered to provide the basis for collaboration and communication between the stakeholders in regards to requirements. Similarly, product backlog, which is originally at level 3 in the SAMI model, was moved to level 2, as it is a fundamental concept in SAFe that appears at all three SAFe levels (team, program, and portfolio). The presence of product backlogs in SAFe also fulfills the goal of maturity level 2 regarding early and continuous delivery of software by defining the epics, features, and task in the corresponding backlogs.

In addition to the changes in the agile practices adopted from the SAMI model, various changes were performed on the SAFe practices of the initial model. In brief, panel experts incorporated 5 additional SAFe practices and modified to some extent several agile and SAFe practices.

4.3. The Scaled Agile Framework Maturity Model

A set of governing rules were applied in defining agile/SAFe practices and populating them in the appropriate maturity level and principle. According to the first rule, each practice must contribute to the achievement of the maturity level objective in which it is positioned. For example, the practice ‘collaborative planning’ (L1P2) addresses directly the collaboration objective of maturity level 1. The second rule ensures the relevancy of the practice with respect to the agile principle that it is associated with. For instance, the same practice (L1P2) relates to the principle for ‘Plan and Deliver Software Frequently’. The third rule concerns the relation between the practices in such a way that practices positioned at higher levels depend on the achievements of practices at lower levels. For instance, the SAFe practice of ‘release planning’’ at level 2 depends on (demands) achieving some of the level 1 practices, such as collaborative teams and collaborative planning. Similarly, self-organizing teams at level 3 depend on having empowered and motivated teams (at level 1).

The final version of the SAFe MM obtained after consolidating the Delphi rounds results is given in Table II. The agile practices from the SAMI model that remained unchanged in the SAFe MM are displayed in black color (35 practices). The SAFe practices that have been introduced are shown in bold (24 practices). The agile practices from the SAMI model that were altered in the current model are displayed in bold-italic (3 practices). SAFe MM can be considered as a descriptive model (as opposed to prescriptive) as it describes only the essential practices that an organization should possess at a particular level of maturity.

4.4. Model evaluation by the expert panel

At the end of the second round, the experts were asked to reach a consensus on the final version of the model and evaluate it in terms of its necessity, practicality, and understandability. Figure 4 presents the results of the responses gathered from the experts.

Most of the experts agreed that the model has a practical merit and can be used in the industry. The model is also considered as easy to understand and use. However, the experts were reserved regarding the necessity of the model as five out of the seven experts stated that they neither agree nor disagree that the maturity model is beneficial and necessary for the industry. Further discussions with the experts revealed that despite their significant effort on the development of the model, there was a
<table>
<thead>
<tr>
<th>Level 5 Encompassing</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>L5P1: Low process ceremony</td>
<td>Embrace change to deliver customer value</td>
</tr>
<tr>
<td>L5P2: Continuous SAFe Capability Improvement</td>
<td>Plan and deliver software frequently</td>
</tr>
<tr>
<td>L5P3: Agile project estimation</td>
<td>Human centricity</td>
</tr>
<tr>
<td>L5P4: Ideal Agile physical setup</td>
<td>Technical excellence</td>
</tr>
<tr>
<td>L5P5: Changing organization</td>
<td>L5P6: Test driven development</td>
</tr>
<tr>
<td>L5P7: No/minimal number of level -1 or 1b people on team</td>
<td>L5P8: Concurrent testing</td>
</tr>
<tr>
<td>L5P9: Frequent face-to-face interaction between developers and users (collocated)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 4 Adaptive</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>L4P1: Client driven iterations</td>
<td>Embrace change to deliver customer value</td>
</tr>
<tr>
<td>L4P2: Continuous customer satisfaction</td>
<td>Plan and deliver software frequently</td>
</tr>
<tr>
<td>L4P3: Lean requirements at scale</td>
<td>Human centricity</td>
</tr>
<tr>
<td>L4P4: Smaller and more frequent releases</td>
<td>Technical excellence</td>
</tr>
<tr>
<td>L4P5: Adaptive planning</td>
<td>L5P9: Frequent face-to-face interaction between developers and users (collocated)</td>
</tr>
<tr>
<td>L4P6: Measuring business performance</td>
<td>Customer collaboration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 3 Effective</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3P1: Regular reflection and adaptation</td>
<td>Embrace change to deliver customer value</td>
</tr>
<tr>
<td>L3P2: Risk driven iterations</td>
<td>Plan and deliver software frequently</td>
</tr>
<tr>
<td>L3P3: Plan features not tasks</td>
<td>Human centricity</td>
</tr>
<tr>
<td>L3P4: Roadmap</td>
<td>Technical excellence</td>
</tr>
<tr>
<td>L3P5: Mastering the iteration</td>
<td>L5P9: Frequent face-to-face interaction between developers and users (collocated)</td>
</tr>
<tr>
<td>L3P6: Software Kanban Systems</td>
<td>Customer collaboration</td>
</tr>
<tr>
<td>L3P7: PSI/Release</td>
<td>L3P10: CRACK customer immediately accessible</td>
</tr>
<tr>
<td>L3P8: Agile Release Train</td>
<td>L4P9: Daily progress tracking meetings</td>
</tr>
<tr>
<td>L3P9: Self-organizing teams</td>
<td>L4P11: Customer contract revolves around commitment of collaboration</td>
</tr>
<tr>
<td>L3P10: Frequent face to face communication</td>
<td>L4P12: Continuous integration</td>
</tr>
<tr>
<td>L3P11: Scrum of Scrum</td>
<td>L3P13: Continuous improvement (refactoring)</td>
</tr>
<tr>
<td>L3P12: Continuous integration</td>
<td>L3P14: Unit tests</td>
</tr>
<tr>
<td>L3P13: Continuous improvement (refactoring)</td>
<td>L3P15: 30% of level 2 and level 3 people</td>
</tr>
<tr>
<td>L3P14: Unit tests</td>
<td>L3P16: DevOps (Integrated Development and Operations)</td>
</tr>
<tr>
<td>L3P15: 30% of level 2 and level 3 people</td>
<td>L3P17: Vision, features</td>
</tr>
<tr>
<td>L3P16: DevOps (Integrated Development and Operations)</td>
<td>L3P18: Impact on customers and operations</td>
</tr>
<tr>
<td>L3P17: Vision, features</td>
<td>L2P12: No big design up front (BDUF)</td>
</tr>
<tr>
<td>L3P18: Impact on customers and operations</td>
<td>L2P13: Product Backlog</td>
</tr>
<tr>
<td>L2P14: Customer contract reflective of evolutionary development</td>
<td>L2P15: Product Backlog</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 2 Evolutionary</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2P1: Evolutionary requirements</td>
<td>Embrace change to deliver customer value</td>
</tr>
<tr>
<td>L2P2: Smaller, more frequent releases</td>
<td>Plan and deliver software frequently</td>
</tr>
<tr>
<td>L2P3: Requirements discovery</td>
<td>Human centricity</td>
</tr>
<tr>
<td>L2P4: Continuous delivery</td>
<td>Technical excellence</td>
</tr>
<tr>
<td>L2P5: Two-level planning and tracking</td>
<td>L5P9: Frequent face-to-face interaction between developers and users (collocated)</td>
</tr>
<tr>
<td>L2P6: Agile Estimating and Velocity</td>
<td>Customer collaboration</td>
</tr>
<tr>
<td>L2P7: Release planning</td>
<td>L3P10: CRACK customer immediately accessible</td>
</tr>
<tr>
<td>L2P8: Define/Build/Test team</td>
<td>L4P9: Daily progress tracking meetings</td>
</tr>
<tr>
<td>L2P9: Software configuration management</td>
<td>L4P11: Customer contract revolves around commitment of collaboration</td>
</tr>
<tr>
<td>L2P10: Automated testing</td>
<td>L3P12: Continuous integration</td>
</tr>
<tr>
<td>L2P11: Tracking iteration progress</td>
<td>L3P13: Continuous improvement (refactoring)</td>
</tr>
<tr>
<td>L2P12: No big design up front (BDUF)</td>
<td>L3P14: Unit tests</td>
</tr>
<tr>
<td>L2P13: Product Backlog</td>
<td>L3P15: 30% of level 2 and level 3 people</td>
</tr>
<tr>
<td>L2P14: Customer contract reflective of evolutionary development</td>
<td>L3P16: DevOps (Integrated Development and Operations)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 1 Collaborative</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1P1: Reflect and tune process</td>
<td>Embrace change to deliver customer value</td>
</tr>
<tr>
<td>L1P2: Collaborative planning</td>
<td>Plan and deliver software frequently</td>
</tr>
<tr>
<td>L1P3: Empowered and motivated teams</td>
<td>Human centricity</td>
</tr>
<tr>
<td>L1P4: Collaborative teams</td>
<td>Technical excellence</td>
</tr>
<tr>
<td>L1P5: Coding standards</td>
<td>L5P9: Frequent face-to-face interaction between developers and users (collocated)</td>
</tr>
<tr>
<td>L1P6: Knowledge sharing</td>
<td>Customer collaboration</td>
</tr>
<tr>
<td>L1P7: Task volunteering</td>
<td>L3P10: CRACK customer immediately accessible</td>
</tr>
<tr>
<td>L1P8: Acceptance testing</td>
<td>L4P9: Daily progress tracking meetings</td>
</tr>
</tbody>
</table>

*Because of the extent of the material, the description of each practice listed in the Scaled Agile Framework (SAFe) Maturity Model is provided as a supplementary material available in the online version of this article at the publisher’s website.*
shared concern about bringing the term ‘maturity model’ in the agile software development arena. This is mainly because this term is typically associated with traditional plan-driven settings where there is heavy emphasis on process-orientation. They believed that there is a risk of companies giving ‘too much emphasis on chasing (maturity) levels, instead of focusing on the enhancements that improve the team performance and cooperation’. As a response to this concern raised in Round 2, we decided not to aggregate the overall assessment results to a single all-in-all maturity rating in order to help mitigate the risk, but rather keep separate maturity ratings for each practice.

Through the Delphi method we were able to explore if experts consider the pure agile practices (in the existing agile maturity model) (up)scalable as-is or if updates are necessary to make them appropriate for SAFe. They evaluated agile practices in terms their scalable adequacy [2], that is, the effectiveness of a practice when used in large settings.

5. CASE STUDY

Although our SAFe MM has been developed as a joint effort of expert practitioners through a structured method, its applicability – as a design artifact – should be evaluated in a business environment [50] where software is developed in large settings. We applied the SAFe MM in a large international company, which is in transition from a traditional plan-driven to an agile and SAFe way of developing software. The objective of the case study was to evaluate the model for validity; that is, to investigate if the artifact works in real-life settings and does what it is meant to do [51].

5.1. The case organization

The company chosen for the case study is a corporation headquartered in Europe. It employs more than 115,000 employees and operates in over 100 countries worldwide. In 2011, the company launched a change program, which aimed at redefining the way the company conducts business through making fundamental changes to its products and processes. As a part of this corporate change program, an initiative in the IT landscape was put in place to introduce new practices in the organization, which laid the foundation for the transition of the company to an agile way of working. The company introduced multi-disciplinary teams, which adopted incremental and iterative approach to software development instead of the traditional waterfall model. Furthermore, the company created output-based partnerships with several partners contributing to the development process. This led to the creation of around 150 Scrum teams, which consisted of internal employees as well as members.
from the partner companies. The teams are grouped into programs (each with varying number of teams; 5–15) with respect to the projects and the particular line of business.

A typical Scrum team consisted of company employees and external (output-based) partners. A partner typically provides the Scrum master and developers, while the company provides the product owner, business analysts, and tech leads. The product owner is a non-IT business employee very close to the company’s business problem to be solved.

Given the size of the organization and projects, scaling agile practices across the teams emerged as an immediate need. The majority of these Scrum teams were geographically dispersed because of the location of several output-based partners. In 2013, the company decided to adopt SAFe in attempt to better organize and streamline these Scrum teams. Initially, program and delivery managers took the initiative to implement varying aspects of SAFe, which was later taken over by SAFe program consultants who were formally trained on SAFe.

The transition from traditional waterfall to SAFe took place in phases at the project level. Any project initiated in the organization was labeled either as a waterfall or SAFe project and followed the practices of the corresponding approach throughout project execution. Figure 5 gives an overview of the projects performed over the years between 2010 and 2014 (July). It also marks the times around which agile practices and SAFe were introduced in the organization. Agile practices were initiated towards the end of 2010 with a few number of pilot projects. In 2011 July, the number of ongoing agile projects reached almost to the level of waterfall-based development projects. The SAFe practices were introduced in 2013. With this introduction, ongoing and starting agile projects were placed within the SAFe program, and associated SAFe practices at the program and portfolio levels were initiated. In July 2014 (i.e., in the time we applied SAFe MM in assessing SAFe maturity in the case organization), there were 156 agile/SAFe projects, and 15 waterfall-based projects that were ongoing.

Within the period between 2010 and 2014/07, 129 agile/SAFe projects were completed with an average duration of 365 days (SD: 201 days). Within the same period 63 waterfall projects were finished with an average duration of 454 days (SD: 232 days).

5.2. Application of the Scaled Agile Framework Maturity Model

For the purpose of assessing the current SAFe maturity, we randomly selected a single program within the company and conducted an assessment among the teams within this particular program. The authors performed the assessment meetings with a Scrum master and a Release Train Engineer (RTE). (RTE can be considered as a ‘chief scrum master’ who facilitates the program level processes in SAFe.) The RTE’s experience and considerations provided input for the existing SAFe practices on the program and portfolio levels of the SAFe, while at the SAFe team level, the Scrum

Figure 5. Number of ongoing and completed waterfall-based and agile/Scaled Agile Framework (SAFe)-based projects in the Case Organization over the years between 2010 and 2014.
master’s experience and reflections were consulted to evaluate the practices employed within the teams in that particular program.

The SAFe practices employed at these levels (i.e., program, portfolio, and team) are assessed using SAFe MM indicators. For each practice in the SAFe MM, we developed a set of indicators to appraise certain characteristics of that particular practice. For the SAMI practices, we adopted from the original indicators. For instance, below are two examples of indicators used for assessing the ‘Roadmap’ and ‘Mastering the iteration’ practices in the SAFe MM Level 3, respectively:

L3P4-In1: The organization has a program roadmap, which provides a view of the intended deliverables over a time horizon of 3–6 months.
L3P5-In1: The development team has effective iterations consisting of sprint planning, tracking, execution, and retrospectives.

Based on the current practices applied in the projects, all indicators were rated using an achievement scale of fully-achieved, largely-achieved, partially-achieved, or not-achieved, and confirmed either by the RTE or Scrum master. The rating scheme is adopted from the ISO/IEC 15504 assessment standard [52] where

- Not achieved (N) represents little or no evidence of achievement of the practice.
- Partially achieved (P) denotes some evidence of an approach to, and some achievement of the practice. Some aspects of achievement may be unpredictable.
- Largely achieved (L) indicates that there is evidence of a systematic approach to, and significant achievement of the practice; despite some weaknesses.
- Fully achieved (F) denotes strong evidence of a complete and systematic approach to, and full achievement of the practice without any significant weaknesses.

The assessment took around 4 h in total, with two assessments sessions of 2 h with each employee, accompanied by one of the authors of this paper. Assessment involved going through all practices and corresponding indicators to assess the entire set of practices in SAFe MM. In order to provide confirmation regarding the results of the assessment, we compiled an assessment report to present the results to relevant parties. We presented and discussed the results in a 1-hour meeting with two employees (Scrum master and RTE) participated in the assessment, and with one delivery manager and one program manager. The findings were directly confirmed by the participants, which lead to an immediate shift in the focus of the meeting to the discussion of possible solution approaches to address the weaknesses pointed out by the assessment.

5.3. Assessment results and discussions

Reiterating the rules applied in developing the model, the SAFe MM is developed in such a way that each practice contributes to the foundation required for the practices that are at higher maturity levels. For instance, the ‘Release Planning’ SAFe practice in Level 2 provides necessary basis for the ‘Agile Release Train’ practice at Level 3. Accordingly, we presume that focusing attention on Level 3 practices without satisfying Level 2 practices will be ineffective for the organization. Therefore, we expected from the case assessment to have more practices satisfied at lower levels than at higher levels.

The results of the assessment are summarized in Figure 6. As expected, the level of achievement tends to decrease towards higher maturity levels. In line with this result, the practices that are ‘fully achieved’ are at the lower maturity levels. In level 1, for instance, the majority (7 out of 10) of the practices are either fully or largely achieved. This result supported our assumptions about the structure and positioning of practices in SAFe MM.

However, the results for the practices that are ‘not achieved’ hardly confirm this inclination; that is, the practices that are ‘not achieved’ are spread over all levels. In order to have a better understanding of the underlying reasons for this situation, we analyzed all the practices that are ‘not achieved’ in more detail. The practice that was ‘not achieved’ at level 1 (L1P2) relates to the collaborative planning, where all stakeholders are expected to come together during the planning phase. Given that the teams are globally distributed, issues related to the achievement of this practice came as no surprise.

Although the lack of collaborative planning could be an enabler for scalability in the sense that it would result in a centralized planning (as in traditional plan-driven approaches), this may contradict
with some basic agile and SAFe principles and related practices, such as the decentralized, rolling wave planning [3], and empowerment of teams and members [32]. In this case, there would be a need to introduce additional practices to ensure that dependencies between features under implementation are properly addressed.

We identified this issue (of a globally dispersed structure) also as an influential factor in many of the upper level practices that failed to achieve their goals. For example, the two practices at level 2 (that the company was not able to achieve) are both related to the planning of the releases. The release planning in SAFe requires all members of the program to come together for synchronization and planning, but having teams (and sometimes team members) located in different time-zones made it challenging for organizing such critical gatherings.

At level 3, this situation manifested itself as a communication barrier between the user representatives (operations) and the development team members, which prevented them to establish a tight integration of development and operations. These themes of weaknesses in the lower maturity levels were indicated as the foremost points on which the company should direct its attention in adopting SAFe.

When the achievement in the original agile practices (that were adopted from the SAMI model as-is) and the pure SAFe practices (that were introduced through the Delphi study) are considered, we cannot distinguish a significant difference. The percentages of practices that were fully, largely, partially, and not achieved for these two groups of practices are similar.

The post-interviews with the people that were involved in the assessment and with the delivery and program managers confirmed the findings achieved from the assessments. The confirmations validated the ability of the SAFe MM to reveal and pinpoint company’s strong and weak points in achieving agile and SAFe practices, and those aspects that require immediate attention.

6. CONCLUSIONS

The challenge in adopting agile practices in developing software increases further with the need to apply these practices in large settings. The SAFe has emerged as an industry approach to address this challenge. However, the current wholesale adoption approach of the SAFe is considered risky and complex. Moreover, driven by practitioners’ efforts in the industry, there are very few studies about SAFe adoption reported in the academic literature. This is partially due to the recent development of the SAFe model.

Our research objective in this work is to design a maturity model that can be used as a guideline by organizations to adopt SAFe and assess the success level of SAFe adoption. We aimed at developing a structured approach to increase the chance of success in scaling agile practices through SAFe. We developed the SAFe MM in cooperation with industry experts. SAFe MM serves as an evolutionary path that increases organization’s SAFe maturity in stages. It prioritizes the improvement actions in adopting agile and SAFe practices.
The model was developed through a Delphi study with the participation of agile/SAFe industry experts in two rounds of feedback. The panel’s central opinion confirms the understandability and practicality of the maturity model. The Delphi study contributes significantly to the relevancy and validity of the model. Further evaluation involved an application of the SAFe MM in a large international company that is in the process of adopting SAFe. We assessed a single program of the company with respect to the SAFe MM and confirmed the points where company’s initial effort should be directed.

The SAFe MM is an important contribution to the practice and research in this body of knowledge. A uniform model for identifying the current state of maturity and providing a roadmap in adopting agile and SAFe practices would be of great help for software developing organizations which have initiated adoption of agile/SAFe practices or those which have not adopted an agile/SAFe way of working but intend to do so in the future. The SAFe MM will provide them a structured approach to assess how well they perform these practices at any point in time. By structuring these practices into maturity levels, SAFE MM provides the community with a roadmap with a set of prioritized agile/SAFe practices. Software developing enterprises will be expected to start adopting the practices in the first maturity level, and gradually move to the practices within higher maturity levels.

SAFe MM is a first attempt to guide and structure the SAFe adoption and should pave the way for future academic research in this area. For researchers exploring empirically the impact of agility and SAFe adoption in practical settings, it is important to understand how successful the organization is in adopting these practices. This is necessary to justify that any impact can be attributed to the practices that have been adopted. The SAFe MM provides the means to address that. It offers a practical mechanism for researchers and practitioners to assess the ‘level of agility’ of an organization based on the extent of successfully performed practices. So, SAFe MM provides a spectrum for the degree of SAFe adoption with levels, which is otherwise considered as a definite state of being ‘adopted’ or not. It provides a continuous and evolutionary approach to improve a software organization’s capability in adopting agile and SAFe practices. However, it is necessary to note that factors leading to an increase in business performance in the form of customer satisfaction, productivity or product quality, can go beyond the adoption of agile/SAFe practices, and can relate to diverse organizational properties and strategic actions.

6.1. Limitations and future directions

Our study has several limitations. The setup of our Delphi study in constructing the SAFe MM poses two main limitations. First limitation concerns the number of Delphi participants. Although the literature does not reach a consensus on the optimal number of subjects in a Delphi study [53], seven members can be considered as limited. We aimed at addressing this issue by bringing together experts that are highly trained and competent within this specialized area of knowledge. As a second limitation, the rounds of feedback were conducted anonymously, which allowed for the panel members to rethink their initial opinions without the influence of the other members. However, the anonymity also can lead to exclusion of group interaction, which in some cases may reduce the accuracy of group judgments [54]. To insure soundness of the gathered data we paid particular attention to panel selection and motivation, questionnaire construction, and for aggregating expert opinion.

There are also limitations pertaining to the application of the model in the case organization. Because of the organizational constraints, we were able to involve only limited number of members of the company in assessing their level of maturity in applying SAFe and for evaluating the overall findings of the assessment. These members were highly experienced and were closely and actively involved in the organizational initiative of SAFe adoption. In addition, in the case study, we were able to assess all practices in the SAFe MM. However, despite the points mentioned, involving more members to gather additional viewpoints on the application of the model and more feedback about the utility of the model would be valuable. This might have also allowed us to assess the practices in multiple programs to get a better and more accurate picture of the overall status of the initiative. On the other hand, the literature allows for some degree of flexibility in judging the degree of evaluation that is needed when a new design artifact is proposed [51], particularly with the novel artifacts when rigorous evaluation is possible mostly through longitudinal studies [55].
Although we conducted our research in a large corporation with tens of projects running concurrently, the results are derived from a single case organization. From this respect, there is a lower external validity of the empirical study, and consequently, the generalizability of these results to other enterprises can be limited. The lower generalizability effect is often associated with the case study research [56]. Furthermore, evaluation of maturity models is typically a challenging research activity that requires several applications in practice for elongated time periods to be able to observe any improvement path and to validate the usefulness of the model in providing guidance for SAFe adoption and improvement. Several assessments should be performed in different organizations and business settings to gain more credible insights into the use and benefits of SAFe MM. Henceforth, further studies are required to better assess the utility, completeness, and validity of the maturity model. Such studies should also aim at quantifying the benefits of SAFe adoption. A complete evaluation of the model’s efficacy requires longitudinal studies that relate increasing SAFe maturity with the process and product quality, as well as business performance in quantititative forms.

Finally, because of the current early state of SAFe, there needs to be further exploration of approaches and risks behind SAFe adoption possibly through case studies in different organizational settings. The feedback and lessons learned gained from these studies should provide input to improve the maturity model. In addition, the constant changes and improvements in the SAFe framework should also be reflected continuously in the SAFe MM.

REFERENCES


SUPPORTING INFORMATION

Additional Supporting Information may be found in the online version of this article at the publisher’s web-site.