

Scaling Agile across the Global Organization: An Early Stage Industrial SAFe Self-Assessment

Mohammad Abdur Razzak, Ita Richardson
Lero, the Irish Software Research Centre,
University of Limerick
abdur.razzak,ita.richardson@lero.ie

Clodagh Nic Canna
Ocuco Ltd
clodagh.niccanna@ocuco.com

John Noll
University of East London
j.noll@uel.ac.uk

Sarah Beecham
Lero, University of Limerick
sarah.beecham@lero.ie

ABSTRACT

The adoption, scaling and tailoring of agile methods depends on several factors, such as the size of the software development organization, business goals, and operative model. The Scaled Agile Framework (SAFe) was developed to support organizations in scaling agile practices across the enterprise.

Large multi-national enterprises report that adoption of SAFe led to significant productivity and quality gains. However, whether these benefits translate to small to medium sized enterprises (SMEs) is unknown.

As such, in this study we ask: *To what extent can SAFe practices be implemented in a global SME?* We administered three surveys to members of the development organization of an Irish SME, to identify and evaluate the adoption rate of SAFe practices in the Global environment.

We found teams and program level personnel are transitioning well towards SAFe. But the Portfolio level personnel appear not to have, as yet, adopted many SAFe practices.

In an SME context it might not be necessary for Portfolio level members to fully adopt SAFe providing they are supportive of their teams. The SAFe self-assessment highlighted where training is required, to improve the vertical communication between teams, program level and upper management.

CCS CONCEPTS

• **Software and its engineering** → **Software creation and management**; *Software development process management*; Collaboration in software development;

KEYWORDS

Global Software Development; Scaling agile; Scaled Agile Framework; Team level; Program level; Portfolio level;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICGSE '18, May 27–29, 2018, Gothenburg, Sweden

© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5717-3/18/05...\$15.00
<https://doi.org/10.1145/3196369.3196373>

ACM Reference Format:

Mohammad Abdur Razzak, Ita Richardson, John Noll, Clodagh Nic Canna, and Sarah Beecham. 2018. Scaling Agile across the Global Organization: An Early Stage Industrial SAFe Self-Assessment. In *ICGSE '18: 13th IEEE/ACM International Conference on Global Software Engineering, May 27–29, 2018, Gothenburg, Sweden*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3196369.3196373>

1 INTRODUCTION

Nowadays, software development organizations look for support in three areas: firstly, how to implement agile methods, secondly, how to implement agile methods within a global environment, and finally, how to ensure this can be achieved while continuing to grow. These lead to several challenges. For example, scaling agile continues to be a challenge in software development where the associated growth calls for strong co-ordination among teams as well as within the project [1, 27]. Scaling agile in globally distributed projects adds to the complexity [16], since *global distance* creates new challenges for successful scaling of agile practices [24].

A number of frameworks have been proposed for scaling agile across the enterprise; the Scaled Agile Framework (SAFe) is one of the most adopted of these models [29]. SAFe has become popular with practitioners and is an important choice for organizations who are scaling their agile development processes. The literature indicates that SAFe is aimed at large-scale organizations. However, small-to-medium-size enterprises (SMEs) are also interested in SAFe as it provides an enterprise roadmap for adopting agile [24]. Although the adoption of SAFe is increasing, little research exists to identify how SAFe, and other scaling agile frameworks, are adopted in global SMEs [17].

SAFe has a self-assessment suite of surveys to measure the degree to which organizations implement SAFe practices at the team, program, and portfolio levels of the organization. In order to test how the metrics proposed in SAFe apply to all levels, we applied this suite of surveys to an SME engaged in global software development, who are transitioning to SAFe; we sought to answer the following question: *To what extent can SAFe practices be implemented in a global SME?*

This paper is organized as follows: Section 2 provides a background to scaling agile frameworks, Section 3 presents

the method we used in our empirical study, while Section 4 summarizes our key findings. Section 5 discusses the implications of these results. Section 6 concludes the study.

2 BACKGROUND

2.1 Global Software Development

Improved communication technologies, access to global talent, cheaper labor, proximity to new markets and legal requirements have all contributed to the growth in Global Software Development (GSD) [30]. Traditionally, GSD has followed a plan driven, structured, waterfall approach, where tasks are allocated according to where they appear in the software development lifecycle [4, 13]. It was considered that agile methods, originally envisaged for *small projects* and *co-located teams* [1, 9] would be a poor fit for GSD because the approaches differ significantly [21]. In contrast to GSD, agile methods tend to rely on informal processes and regular face-to-face communication to facilitate coordination, whereas distributed software development relies on formal mechanisms. Yet there is a growing trend for globally distributed companies to adopt agile methods [7]. Adopting agile practices such as short iterations, frequent builds, and continuous delivery all pose challenges to configuration management and version management [19]. But, practices such as short iterations increase transparency of Work-in-Progress (WIP) and provide a big picture of project progress to stakeholders [18]. However, to implement agile practices in a global software organization, developers need to have more autonomy as well as decision-making power [6, 14].

2.2 Agile Scaling Frameworks

Scott Ambler pointed out several factors that need to be considered when scaling agile development, such as team size, geographical distribution, entrenched culture, system complexity, legacy systems, regulatory compliance, organizational distribution, governance and enterprise focus. In general, productivity and quality are the two main concerns of any organization when adopting a scaling agile paradigm [2].

The Agile Scaling Knowledgebase (ASK)¹ developed a matrix of different agile frameworks, namely *Scrum-of-Scrums* (SoS)², *Large Scale Scrum* (LeSS)³, *Scaled Agile Framework* (SAFe)⁴, *Disciplined Agile Delivery* (DAD)⁵, *Spotify Model*⁶, *Nexus*⁷, and *Scrum at Scale*⁸. This matrix shows that SAFe, launched in 2012 by Dean Leffingwell [12] focuses on large enterprises and takes a scaled approach to agile adoption.

The 11th Annual State of agile report [29] reported that, with 28% of respondents using it, SAFe is the most used scaling method. In contrast, LeSS, DAD, and Nexus are reported

to have a significantly lower take-up rate. However, many practitioners consider SAFe to be too heavy and complex due to highly prescriptive role-and-process scheme and evolving to “the new waterfall” [3]. In our recent study, we also identified that, in smaller organizations, adopting the many different ceremonies as well as dedicated roles within a team may not be possible, or necessary, to meet business goals [23]. For this reason, we are interested in the implementation of SAFe across the entire enterprise, as adopted in our case study company, since there is a lack of scientific studies on the actual usage of these prescribed practices [17].

2.3 Scaled Agile Framework (SAFe)

SAFe can be viewed as a container for several existing agile approaches. It is scalable and modular, and was primarily developed for organizing and managing agile practices in large enterprises. Early adopters of SAFe report that the application of the practices contained in this framework has led to significant productivity and quality improvements [11]. The literature shows that SAFe adoption is widespread in sectors including manufacturing and financial services [11, 16, 20, 26]. SAFe 4.0 is organized into four layers: Team, Program, Value Stream, and Portfolio.

2.3.1 Team Level. In SAFe, all teams are part of the Agile Release Train (ART); ARTs are the central construct of the program level. Teams are collectively responsible for defining, building, and testing software in fixed-length iterations and releases [12]. The team events (Backlog Refinement, Sprint Planning, Sprint Review) are an integral part of SAFe, and help to reduce coordination overhead between teams. These teams typically consist of 7-9 members and teams operate on identical cadence and iteration lengths in order to provide better integration among teams [26]. But, adoption of only Scrum at the team level could lead to additional problems in task synchronization. To resolve this issue, SAFe introduces the *Release Planning* meeting after every five iterations to synchronize team tasks [12]. All teams on an ART are synchronized and integrated via common iterations that provide a valuable increment of new functionality. At the end of each iteration, the teams perform a system demo for ART integration.

2.3.2 Program Level. At the program level, the development teams and other resources are applied to the ongoing development mission. At this level, teams, roles, and activities are organized around the ART which delivers a continuous flow of incremental releases of value [12]. Program level also discovers, defines and develops features and enablers that are required by the business to realize the vision and roadmap. To manage the flow and make it visible to all stakeholders, a program kanban is used at this level to ensure that features are prioritized prior to Program Increment (PI) planning. There are three primary functions namely Release Train Engineer (RTE), who is the chief Scrum Master, who optimizes the flow of the value through program kanban, PI planning, and Inspect & Adapt workshop (I&A); Product Manager, who

¹<http://www.agilescaling.org/home.html>

²<https://www.agilealliance.org/glossary/scrum-of-scrums/>

³<https://less.works>

⁴<http://www.scaledagileframework.com>

⁵<http://www.disciplinedagiledelivery.com>

⁶<http://blog.crisp.se/2012/11/14/henrikkniberg/scaling-agile-at-spotify>

⁷<https://www.scrum.org/resources/nexus-guide>

⁸<https://www.scruminc.com/scrum-scale-case-modularity/>

becomes the internal voice of the customer, acting as a liaison between the customer and Product Owner, and System Architect who defines the overall architecture and nonfunctional requirements [12].

2.3.3 Value Stream Level. The Value Stream level is intended for builders of large and complex solutions that require multiple ARTs. The primary purpose of this level is to describe Lean-Agile approaches to system development [12]. Since our SME is not yet involved in multiple ARTs we do not study this level in this study.

2.3.4 Portfolio Level. In SAFe, the portfolio level is the level with highest level of concern that encapsulates the people and processes to provide funding and governance mechanisms to meet the strategic objectives [12]. This level has a bidirectional connection to the business: in the first direction, it provides the strategic themes to guide the portfolio towards the larger, specific and itemized business objectives. The second direction provides a constant flow of portfolio context back to the enterprise [12].

2.4 Measurement of Software Process Improvement

The success of software development organizations depends on *predictions* and *commitments* they make to their products. These two elements help a software organization to understand their capabilities and to develop an achievable plan for the delivery of products and services [5]. By applying process assessment, a software development group can understand its current process maturity against a target reference model, identify practices to improve, and identify those that are missing, but should be implemented. Assessments can range from a full formal organizational appraisal, to informal facilitated sessions [31] that define ‘as is’ processes [22] or process artifacts [10]. While assessments such as the Capability Maturity Model Integration (CMMI) [25] and ISO/IEC 15504 [28] exist, these do not explicitly consider agile methods; there is a need for an assessment tool for continuous improvement of agile processes [8].

SAFe provides 18 types of metrics across all levels. These evaluate how well the organization is progressing toward the Portfolio, Program, and Team’s business and technical objectives. That helps organizations to adjust their behavior based on the metrics being used to measure their systems and evaluate their performance. In this study, we adopted the SAFe self-assessment metrics to assesses and evaluate the frequency of SAFe adoption across an organization. This self-assessment is quick, flexible and cost efficient, also useful to measure long-term organizational performance.

3 METHOD

3.1 The Case Organization

The company we studied, Ocuco Ltd., is a medium-sized Irish-based software company that develops practice and lab management software for the optical industry. Ocuco Ltd.

Table 1: List of participants.

Role	Team (n=19)	Program (n=11)	Portfolio (n=6)	Country
Project Manager (Scrum Master)	7	6	–	Canada, France, Ireland, Norway, Poland, UK
Developer	6	–	–	Canada, Ireland
Quality Assurance	3	–	–	France, Ireland, USA
Product Owner	3	–	–	Canada, Ireland, USA
Quality Assurance Lead	–	1	–	Ireland
Database Administrators	–	1	–	Ireland
Technical Support	–	1	–	Ireland
Development Manager	–	1	1	Ireland
Product Manager	–	1	1	Spain
Chief Executive Officer	–	–	1	Ireland
Chief Technology Officer	–	–	1	Ireland
Regional Management/Sales	–	–	2	France, Italy

employs approximately 300+ staff members in its software development organization, including support and management personnel. Ocuco Ltd.’s annual sales approach €20 million from customers internationally. The development takes place across nine countries, all of which are represented in our survey results (see Table 1).

3.2 Data Collection and Analysis

We administered three SAFe self-assessment surveys (Team⁹, Program¹⁰, Portfolio¹¹) to 70 team, 12 program, and 10 portfolio members in July, 2017.

The SAFe self-assessment survey comprises 25 questions at the Team level, 43 questions at the Program level, and 42 questions at the Portfolio level. Each question has both a quantitative element (ordinal scale) and an optional qualitative element (comment) that allowed participants to explain their ranking if needed. The ordinal scale has six possible response options (ranging from ‘never’ to ‘always’) to measure the frequency of practice use according to each area. We also included definitions of all SAFe terms along with each survey, to help participants understand SAFe terminology.

To analyze the collected survey data, we extracted all qualitative and quantitative data, aggregated all data points to get an overall view of all team members and to measure the frequency of practices used by members according to each area in different levels within the organization, then looked at comments to understand the reasons behind the qualitative levels. We used content analysis to analyse the qualitative data collected. In total, we received 19 responses out of 70 at the team level, 11 responses out of 12 at the program level, and 6 responses out of 10 at the portfolio level. Table 1 shows a breakdown of the roles of all respondents.

4 RESULTS

4.1 SAFe at Ocuco Ltd.

Figure 1 represents the early stage SAFe adoption of the three layers of SAFe at Ocuco Ltd.. Company management divide their software development personnel into three levels: the portfolio level is responsible for strategic decision making, the program level is responsible for all project level activities, and

⁹<http://www.scaledagileframework.com/metrics/#T4>

¹⁰<http://www.scaledagileframework.com/metrics/#PF7>

¹¹<http://www.scaledagileframework.com/metrics/#P6>























































Portfolio	Program	Team									
CFO	Product Manager	PO	T1	T2	T3	T4	T5	T6	T7	T8	T9
CSO											
CEO	Architect Lead	Tech Lead									
CTO	QA Lead	Dev									
Director of Development		QA									
		SM									
		PM									

Figure 1: SAFe at Ocuco Ltd. (CFO=Chief Financial Officer, CSO=Chief Sales Officer, CEO=Chief Executive Officer, CTO=Chief Technical Officer, QA=Quality Assurance, PM=Project Manager, PO=Product Owner, Tech Lead=Technical Lead, Dev=Developer, SM=Scrum Master)

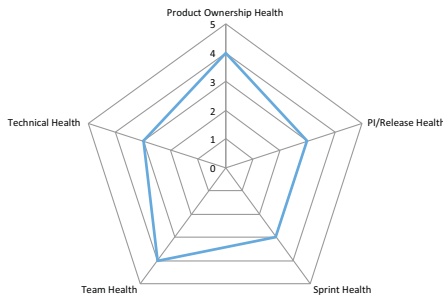


Figure 2: SAFe Team Self-Assessment (values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always).

finally the team level is responsible for actual development activities. As Ocuco Ltd. is still in the early stage of their SAFe transformation, all existing roles were distributed across these levels as shown in Figure 1.

In this section, we present SAFe self-assessment survey results. Figures 2, 3 and 4 show the median score across all participants within each level.

4.2 Team Level

There are nine teams at Ocuco Ltd. Of these, only one team is fully co-located in the home office; the other teams are globally distributed across seven countries. Ocuco Ltd. development teams use Scrum to develop their software, with two-week sprints. Each team consists of 7-9 members comprising the Product Owner, a Senior Developer (who also acts as a Technical Lead), Developers, Quality Assurance (responsible for testing), and a Scrum Master (who also acts as a Project Manager). There are also some cross-trained team members who work on multiple products; Program Management facilitates this resource sharing during “Scrum of Scrums” meetings at the program level.

The SAFe Self-Assessment addresses five team level areas, namely: Product Ownership Health, Program Increment (PI)/Release Health, Sprint Health, Team Health and Technical Health; the self-assessment questionnaire includes five questions assessing each of these areas. A summary of the

Table 2: Product ownership health.

Question	Median ¹	Mode ¹
Q1. Product Owner facilitates user story development, prioritization and negotiation	4	4
Q2. Product Owner collaborates proactively with Product Management and other stakeholders	4	4
Q3. User Stories are small, estimated, functional and vertical	3	4
Q4. Product owner facilitates development of acceptance criteria which are used in planning, review and story acceptance	4	4
Q5. Teams refine the backlog every sprint	4	5

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

results is shown in the radar chart in Figure 2. Of these, we found PI/Release Health, Sprint Health and Technical Health were the weakest areas, whereas Product Ownership and Team Health were strong across all teams (with related sub-practices being performed “very often”).

The results of the Team level assessment are fully reported elsewhere [24]; here, we focus on Product Ownership Health, Sprint Health, and Team Health.

4.2.1 Product Ownership Health. Product Owners maintain the *product backlog*, which is a list of user “stories” that define requirements for the project. Table 2 shows results of Product Ownership Health at Ocuco Ltd..

According to quantitative data, at Ocuco Ltd. (Table 2), the Product Owners use “User Stories” “*Very Often*”, but turning to the associated qualitative results, one of the Product Owners mentioned:

We don’t really use User Stories. We do a lot of prioritization and negotiation. We do a slightly more defined conversation/specification and communicate directly with developers.

As a rationale for not using User Stories, a developer explained:

This is a customer focused project. There is very little user story development in it. All we have are big long documents and specifications. However, they [Product Owners] did a good job in prioritizing and negotiating with the customer.

This statement results in our concluding that the Product Owner “*Very Often*” facilitates prioritization, and negotiation (in Table 2, Q1), and not user story development. But, on the other hand, a Project Manager who also acts as a Scrum Master said:

There is not a lot of negotiation going on for our team as the estimates are done in advance. Due to nature of contract we don’t work with User Stories. We have deliverables that have been defined as part of the contract.

The grooming session ensures that items on the backlog are sufficiently described, and estimated. Then, they can be considered at the sprint planning meeting for implementation during the next sprint. A team usually performs a grooming session once in each sprint. Due to the fact that teams are distributed, we have observed that Product Owners use screen shots, mock-ups, and requirement specifications to enhance team understanding during the grooming session.

Table 3: Sprint Health.

Question	Median ¹	Mode ¹
Q1. Team plans the sprint collaboratively, effectively and efficiently	4	3,5
Q2. Team always has clear sprint goals, in support of PI objectives, and commits to meeting them	4	3
Q3. Teams apply acceptance criteria and Definition of Done to story acceptance	3	4
Q4. Team has a predictable, normalized velocity which is used for estimating and planning	3	2,3,4
Q5. Team regularly delivers on their sprint goals	3	3

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

One of the developers explained how the grooming session starts, and the exchange between the team and the customer:

... grooming [comprises] cleaning up the requirements and making sure that they are clearly understood and there are no questions or grey areas. There is some cleaning involved with regards to the requirements. . . Then if we need any information from the customer themselves then we turn it back to them, saying, we need this information.

Product Owner teams have staff members onshore for discussions with clients, and offshore for disseminating information to development teams.

In Ocuco Ltd., we have seen that the Product Manager spends one or two weeks on-site with customers going through their plan and business processes to document their requirements. Then, the Product Owners translate those requirements into an “epic” comprising features or “tickets” to be implemented, in collaboration with the Product Manager; these are placed on the backlog.

It has been agreed by most of the team members that, geographic separation has both advantages and disadvantages. One of the Technical Leads at the remote office observed:

It would helpful if everybody was in the same room but in our case. . . we have the advantage of people in different time-zones. They are given some work that I am doing at the moment then they will be working on it during the day. Then the [onshore] DBA can handover to me in the next morning.

In contrast, the Product Owner mentioned:

The difficulty is that, when I am here on-site I only get a couple of hours in the morning to deal with Dublin stuff.

The motivation of Ocuco Ltd. is to become a global organization, developing software globally for a global market by using benefits of global software development but one of the Product Owner identified that, geographical separation is massively hindering his project. He cannot talk to the people he needs to talk to urgently. Due to global development difficulties, the Product Owner has considered moving to another project that has the team in same time zone.

4.2.2 Sprint Health. Table 3 shows the aggregated result of the survey on Sprint Health at Ocuco Ltd..

Teams “Often” calculate velocity to plan for the upcoming sprint. Additionally, teams “Very Often” plan the sprint

Table 4: Team health.

Question	Median ¹	Mode ¹
Q1. Team members are self-organized, respect each other, help each other complete sprint goals, manage interdependencies and stay in-sync with each other	5	5
Q2. Scrum Master attends Scrum of Scrums and interacts with RTE as appropriate	3	5
Q3. Stories are iterated through the sprint with multiple define-build-test cycles (e.g. the sprint is not a waterfall)	3	4
Q4. Team holds collaborative, effective and efficient planning and daily meetings where all members participate, status is given clearly, issues are raised, obstacles are removed and information exchanged	5	5
Q5. Team holds a retrospective after each sprint and makes incremental changes to continually improve its performance	4	5

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

collaboratively, effectively, and efficiently. However, one of the team members said:

Sprints are not planned as such as we are at the tail end of the dev cycle. Almost all open tickets go into the sprint.

Though teams “Often” calculate velocity to plan for the upcoming sprint, due to lack of accurate estimation, teams cannot always meet the sprint goals:

We are often behind on doing the estimates, not taking the needed time or missing information enough to do a proper estimate

A Project Manager identified that “Over commitment” and Quality Assurance “Speed” are hindering the team in meeting the sprint goals. However, a Developer said:

It’s a bit up and down, sometimes we succeed. It is like it has become common to always introduce new ‘critical’ issues into the current sprint, instead of letting them wait for the next sprint planning.

4.2.3 Team Health. Table 4 shows the aggregated result of Team health at Ocuco Ltd..

According to Table 4, in Ocuco Ltd. teams “Always” hold collaborative, effective and efficient planning meetings. Daily meetings are in place where all home office and remote members participate, usually by video conferencing; status is given clearly, issues are raised, obstacles are removed, and information is exchanged among team members. Team members are self-organized, respect each other, “Always” help each other to complete sprint goals, manage interdependencies, and stay in-sync with each other.

A Product Owner states:

Teams work well together and everyone is providing their part to making the best product. We just don’t always agree, which is good!

The Retrospective meeting is meant to be a qualitative method to measure the Scrum team’s performance in a sprint by asking, 1) what went well, and 2) what didn’t work well. This meeting helps the team to ensure that all team members contribute equally, and also helps to encourage ‘self-organization.’

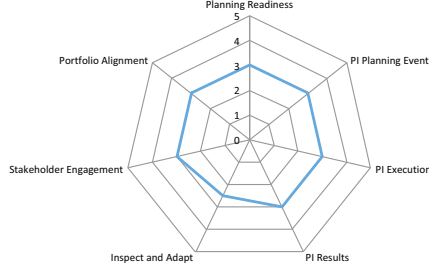


Figure 3: SAFe Program Self-Assessment (values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always).

But the self-assessment results show this practice is not fully adopted: while the predominant response is that they “Always” hold planning and daily meetings, at least one team almost never holds retrospectives after each sprint:

I can only recall one retrospective during the last 2 years, it was done after a release and not after each sprint.

4.3 Program Level

The Program level at Ocuco Ltd. is distributed across seven countries. Our research has identified five important roles at this level: Product Manager, Architectural Lead, Quality Assurance (QA) Lead, Project Manager, and Director of Development, who oversees the Program level. Each Project Manager is responsible for an individual team and is also the Scrum Master at Team level.

SAFe prescribes seven areas at the Program level, namely Planning readiness, Program Increment (PI) planning event, PI execution, PI results, Inspect and Adapt (I&A), Stakeholders engagement, and Portfolio alignment to access Program level. Of these, we found Inspect and Adapt reported “Occasionally” in practice, and rest of the areas are “Often.” In the following sections, we focus on the PI planning event, Inspect and Adapt, and Portfolio alignment.

4.3.1 PI Planning Event. PI planning attempts to align all the teams in the Agile Release Train (ART) to a shared mission and vision. In the SAFe approach, a Release Train Engineer (RTE) facilitates the PI planning event to establish communication channels across all team members. This includes building social networks, aligning development to business goals, identifying dependencies, and fostering cross-team and cross-Agile Release Train collaboration [12]. PI planning is essential to SAFe; according to Leffingwell [12], this is quite a significant event and “if you are not doing it, you are not doing SAFe”.

In Ocuco Ltd., the program management team “Often” performs some elements of PI planning events, which are supported by the Director of Development:

We don’t have a formal PI planning event but we do have elements of this done informally.

As a way forward, program management decided to develop an Agile Release Train (ART), where each team participates at the end of each sprint to sync with other teams.

Table 5: PI Planning Event.

Question	Median ¹	Mode ¹
Q1. PI Planning event has effective facilities and distributed team communications	3	4
Q2. Program stakeholders, business owners, and shared services are in attendance	3	3
Q3. All team members participate in person or remotely	3	4
Q4. Business context, Product vision, and Top 10 features are effectively communicated	3	0
Q5. Timeboxes are adhered to; event starts and ends on time	3	3,4
Q6. Risks are ROAMed (Resolved, Owned, Accepted, Mitigated)	3	3
Q7. Management addresses key scope and quality challenges and makes planning adjustments as necessary	3	3
Q8. Alignment is achieved: all teams leave with agreed upon SMART team PI objectives ranked by business value, accepted by business owners, and communicated to all stakeholders	3	3
Q9. Confidence vote is 3 or higher	3	0

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

Table 6: Inspect and Adapt.

Question	Median ¹	Mode ¹
Q1. Event is held every PI	3	0
Q2. Most team members attend	3	0
Q3. Key stakeholders attend	2	0
Q4. Improvement stories are identified and itemized	0	0
Q5. Improvement stories are incorporated into PI Planning event	0	0

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

4.3.2 Inspect and Adapt. In SAFe, Inspect and Adapt (I&A) is a significant event that is held at the end of each Program Increment (PI), where all program stakeholders participate with the agile teams. This event consists of three parts: the PI system demo, quantitative measurement, and retrospective.

Table 6 shows that, teams “Often” perform I&A workshop at the end of each PI. However, the Director of the Development said, *... We don’t do this yet.* That is also supported by a Project Manager: *... I haven’t attended a PI Planning event in a long time.* As our study identified, the PI planning event has not been implemented at the Program level, and as a consequence of that teams do not need to perform I&A workshop.

4.3.3 Portfolio Alignment. The program vision and roadmap provide a view of the features to be developed and also reflect the needs of stakeholders. Program portfolio management (PPM) holds the primary responsibility to align the portfolio with the Program level to guide program execution and governance.

In the studied organization, PMO “Often” aligns the program vision and roadmap with the strategic themes and portfolio vision. This is consistent with one of the Project Manager’s comments:

Table 7: Portfolio Alignment.

Question	Median ¹	Mode ¹
Q1.Program vision and roadmap are aligned with the strategic themes and portfolio vision	3	3
Q2.Epic Owner works with appropriate stakeholders to split Epics into Features and to prioritize them into the respective Program Backlog	3	3
Q3.Governance adheres to lean-agile principles	3	3
Q4.Objective fact-based measures and milestones are reported to the Program Portfolio Management team	3	3

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

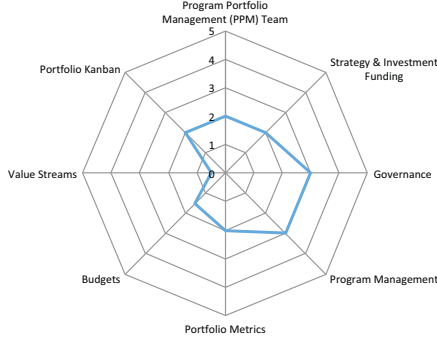


Figure 4: SAFe Portfolio Self-Assessment (values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always).

Product Manager, Chief Executive Officer (CEO), Chief Technical Officer (CTO) and Director of Development are meeting periodically, I have only attended the last portfolio review and it seems to start working on the agile direction at portfolio level.

In regards to whether governance adheres to lean-agile principles, this same Project Manager also mentioned:

In my opinion it has only started, but there is a complete shift in culture that still has to happen.

4.4 Portfolio Level

There are eight areas in Portfolio level, namely Program Portfolio Management (PPM) Team, Strategy & Investment Funding, Governance, Program Management, Portfolio Metrics, Budgets, Value Streams, and Portfolio Kanban. Of these, Governance and Program Management are “Often” implemented in practice, while the rest of the areas were weaker as shown in Figure 4. The Portfolio level is distributed in five countries, but the majority of the Portfolio team members are based at the headquarters in Dublin.

4.4.1 Program Portfolio Management (PPM) Team. In SAFe, Program Portfolio Management (PPM) represents the people who have the highest-level strategy and fiduciary decision-making responsibility. PPM also participates in the establishment and communication of strategic themes that guide the enterprise’s investments and strategy, determine the relevant value streams, and allocate budgets.

Table 8: Program Portfolio Management (PPM) Team.

Question	Median ¹	Mode ¹
Q1.Effectively supports demand management and continuous value flow	2	0,2
Q2.Facilitates lightweight, epic-only business cases	1.5	0,2
Q3.Supports decentralized rolling-wave planning	1	0
Q4.Fosters agile estimating and planning	2	1,2,3
Q5.Drives product and solution strategy; effectively manages investments	3.5	4
Q6.Uses objective, fact-based measures and milestones to evaluate the performance of agile Release Train (ARTs)	0.5	0

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

Table 9: Strategy & Investment Funding.

Question	Median ¹	Mode ¹
Q1.Strategic themes have been created and are used to connect the Portfolio Vision to the enterprise business strategy	3	3
Q2.Strategic themes guides investments and the enterprise architecture needed to support customer and business needs	2.5	2
Q3.Strategic themes are used as decision making filters in the Portfolio Kanban System	1.5	1,2

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

However, in Ocuco Ltd., the PPM team “Occasionally” support demand management and continuous value flow. The CTO mentioned:

We don’t do demand management but rather that it is not particularly done as part of the portfolio process. The demand management we do is more at the resource and project management and resource balancing. The portfolio drives what needs doing and it is the management of the execution which then makes it happen, including resetting the timelines to match production capability.

In respect to epic in Portfolio level, the CTO also mentioned:

We aren’t really epic-based at the PPM level, it is more about projects/deliverables/contracts, from which epics are produced within the team environment after the fact.

The PPM “Occasionally” fosters agile estimating and planning. As a rationale of that, a PPM member mentioned:

We do planning and estimation, but it isn’t specifically done at the PPM level outside of general broad estimates and timelines. But it is the word “fosters” which is important here. The PPM doesn’t specifically or actively ‘foster’ estimation and planning, outside of it being part of the accepted strategic direction to embrace agile.

4.4.2 Strategy & Investment Funding. The purpose of Strategy & Investment Funding in SAFe is to support implementation of the business strategy through programs that develop and maintain the company’s value-added products and services.

The studied organization “Often” use strategic themes to connect the Portfolio Vision to the enterprise business strategy. However, according to the Director of Development,

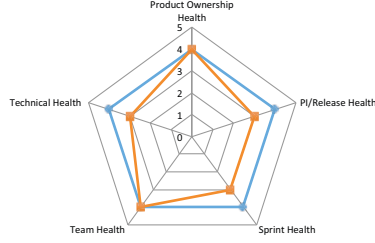


Figure 5: Team Level

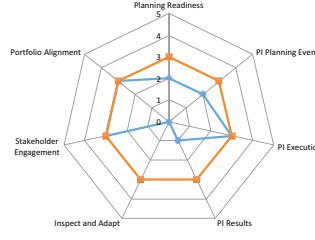


Figure 6: Program Level.

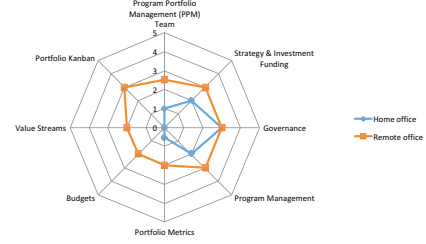


Figure 7: Portfolio Level.

Table 10: Governance.

Question	Median ¹	Mode ¹
Q1.Decisions are decentralized when they are frequent, time critical, and do not have global economies of scale	4	4
Q2.Release Management assist and adjusts scope; communicates expectations internally and externally; and provides necessary governance authority	4	4
Q3.Centralized control is used for decisions that are infrequent, not time critical, and have global economies of scale	3.5	3,4
Q4.The Program Portfolio Management (PPM) Team is respected as having the highest fiduciary decision-making authority for its programs	3.5	3,4
Q5.The PMO is agile and provides operational support to the Program Portfolio Management (PPM) Team	3	3,4
Q6.Each agile Release Train (ART's) program vision and roadmap is aligned with the Portfolio Vision and Strategic Themes	3	2,3,4
Q7.Epic Owners work with Product and Solution Management to split Epics into Capabilities and Features and to prioritize them into the respective backlogs	3	3
Q8.Each Value stream reports objective fact based KPIs and milestones	1.5	1,2
Q9.Guidelines have been established to direct Program Epics to an individual Value Stream or ART or to the Portfolio Kanban, depending upon its estimated size	2	2

¹ Values: 0 - Never, 1 - Rarely, 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always.

this is done in an “*informal*” manner rather than according to the SAFe procedure.

4.4.3 Governance. Governance functions still exist in agile otherwise there would be no Portfolio level feedback on investment spend. In comparison to other areas at the Portfolio level, *Governance* seems to be one of the strongest areas at Ocuco Ltd..

The decision making process at this level is “*Very Often*” decentralized, as reported in Table 10. This has the effect of making distributed development easier through provision of autonomy to global teams.

4.5 Home Office vs Remote Office Work

During this study, we have observed some mismatches between our qualitative and quantitative data. Our data has also allowed us to compare the perceived knowledge between home office and remote team members at the Team, Program, and Portfolio levels (see Figures 5, 6, 7).

At the Team level, we can see that there are three areas where home office team members state that they perform “*Very Often*” compared to remote team members who “*Often*” perform these activities. Product Ownership health and Team health are reported equal by both home office and remote team members. According to Ocuco Ltd.’s Director of Development:

We realized our Product Owners were being pulled in different directions by their multiple responsibilities, and as a result their teams were drifting away from the product roadmap. So we decided to hire additional staff so the Product Owners could focus solely on Product Ownership and keep the long-term product vision in-focus.

Interestingly, at Program level, we have found that, the adoption rate of SAFe by remote team members is higher than home office team members. Home office team members think *Inspect and Adapt* have never been adopted. However, remote team members think they perform this workshop “*Often*”. But, we have observed there is a mismatch between quantitative and qualitative data set as a remote Project Manager said:

At the program level, there are other people doing it [Inspect and Adapt] or is it the Project Manager that is providing. But collectively at the PMO meeting we are not doing this exercise or very informally.

The remote Portfolio personnel indicate that they have adopted more SAFe practices at this level compared to home office portfolio personnel. Both home office and remote personnel think *Governance* has been “*Often*” adopted in both cases. However, a remote managing director admitted, “*SAFe is not fully implemented in all countries.*” that also echo’s operation manager’s voice at home office:

I think it’s just we’ve adopted items that we feel work well for us, such as the Scrum. We don’t do it cause its agile. We just feel it makes sense to meet every morning and discuss what’s going on so people don’t work in isolation. So, I think we have adopted principles but not necessarily under the guise of we are going agile.

Finally, the improvements can be seen if teams at different level become more familiar with their roles. This is especially true for global team members. On the other hand, people may try harder to please (and inflate what they do in the survey), because they know they are being measured.

5 DISCUSSION

Returning to our research question, *To what extent can SAFe practices be implemented in a global SME?* the self-assessment results provide a mixed message. On the one hand, the quantitative results indicate that Ocuco Ltd. has been able to implement SAFe at the *Team* level, but has not so far successfully done so at the *Program* and *Portfolio* levels.

On the other hand, the comments from participants seem to suggest that the numbers do not tell the whole story: while the *Team* level *numbers* indicate many practices are performed “Very Often” or “Always,” comments indicate some teams never perform certain practices. This is echoed at the *Program* level, where for example responses indicate *Inspect and Adapt* is performed “Occasionally” but the *Director of Development* said it’s not performed at all (yet).

The *Portfolio* level assessment presents a third perspective: respondents at the *Portfolio* level seemed to be highly critical of their progress on SAFe implementation. It appears that respondents answered the survey questions according to a very strict comparison of their activities to SAFe. In reality, our observations and interviews at the *Portfolio* level suggest that Ocuco Ltd. does perform many of the activities assessed by the SAFe self assessment, but in a different way and using different terminology than SAFe. For example, in the *Strategy & Investment funding* area, respondents rated Ocuco Ltd.’s use of “strategic themes” in guiding investment as somewhere between “Occasionally” and “Often.” But we have observed that Ocuco Ltd. is quite disciplined in their approach to investing in product development, although they do not use the term “strategic theme.”

Finally, some SAFe practices may simply not apply to SMEs. For example, Ocuco Ltd.’s *Portfolio* level respondents rated their implementation of practices related to *Value Streams* as quite low (Figure 4). This may reflect the fact that *Value Streams*, as conceived in SAFe, are optional, as they are targetted toward large organizations: “The *Value Stream Level* is optional in SAFe. Enterprises that build systems that are largely independent, or that can be built with a few hundred practitioners, may not need the constructs of this level, in which case they can operate from the ‘collapsed view,’ which is called 3-level SAFe [12, value-stream-level page].”

In software development, teams tailor their practices based on the metrics used to measure their system and evaluate their performance [12]. By applying self-assessment, a software organization can understand its current process maturity, identify practices to improve, and identify practices that are missing. But an overemphasis on quantitative results of such self-assessments may obscure the true picture. As we have shown, the numbers do not always match the underlying truth.

A strength of the SAFe self-assessment instrument is that it asks respondents to write comments that illuminate the values they assign to questions. This provides insight into why different metrics have given values. Also, the SAFe self-assessment is intended to be a team exercise: while team

members may complete the assessment individually, they are encouraged to discuss the results as a group. This promotes deeper understanding than the numbers alone can provide.

In light of this observation, the SAFe self-assessment instrument should not be viewed as a management reporting tool, because the values must be filtered through the lens of the qualitative comments that explain why levels are reported as they are.

5.1 Threats to Validity

While we received responses from 92% of staff at the *program* level, and 60% at the *portfolio* level, we only received responses from 27% of *team* level staff. However, qualitative discussion has been included to support the data that we have found during the study. While we have concerns that particular practices are not performed, qualitative responses suggest that there may be genuine reasons for this. The SAFe survey that we used only looks at a sub-set of practices. This supports the company goals to achieve cost savings and increase employee motivation.

To overcome these threats, and to validate our findings, we conducted two validation workshops within Ocuco Ltd., attended by many of the respondents and others who did not answer the questionnaire. We have also circulated the results of all self-assessment survey through Ocuco Ltd.’s internal knowledge sharing repository.

6 CONCLUSIONS

The goal of this study was to answer the research question, *To what extent can SAFe practices be implemented in a global SME?* We employed the SAFe self-assessment instrument to identify and evaluate the adoption rate of SAFe practices, and the health levels of different process areas in a medium-sized Irish-based software company. SAFe provides more roles, events, artefacts and practices compared to other frameworks that aim to support organizations to scale on an enterprise level. But, in smaller organizations, adopting the many different ceremonies as well as dedicated roles may not be possible or necessary to meet their business goals. Ocuco Ltd. also had to implement SAFe within their global software development teams.

We have found that, at the *Team* level, the SAFe self-assessment indicates that most of the process areas are adopted quite strongly. However, home office team members have responded that they have adopted more when compared to responses of remote team members. It could be that more advocacy and training on SAFe is occurring at the home office, as compared to other locations. We observed the reverse at the *Program* level, where remote team members think they have adopted more SAFe practices at this level, compared to home office team members. In a similar fashion, we have also found that those working in the remote *Portfolio* level think they have also adopted more SAFe prescribed processes. In all cases, however, we view the qualitative results of the self-assessment, in the form of open-ended comments made by respondents about their answers to assessment questions,

as both informative, and necessary to interpret the numeric results.

A major goal for Ocuco Ltd. is to standardize their processes across all distributed teams, the program level, and the portfolio level, through transitioning to the SAFE framework. They are starting to achieve this by tailoring SAFE practices, and integrating these with the Global Teaming Model [15, 23].

7 ACKNOWLEDGMENTS

We thank the members of Ocuco Ltd. for their generous and thoughtful collaboration on this study, and for allowing us to study their software development efforts. This work was supported, in part, by Science Foundation Ireland grants 13/RC/2094 to Lero - the Irish Software Research Centre (www.lero.ie).

REFERENCES

- [1] Pekka Abrahamsson, Kieran Conboy, and Xiaofeng Wang. 2009. “Lots done, more to do”: the current state of agile systems development research. (2009).
- [2] Scott W. Ambler. 2008. Agile software development at scale. In *Balancing agility and formalism in software engineering*. Springer, 1–12.
- [3] Christof Ebert and Maria Paasivaara. 2017. Scaling Agile. *IEEE Software* 34, 6 (2017), 98–103.
- [4] Hans-Christian Estler, Martin Nordio, Carlo A Furia, Bertrand Meyer, and Johannes Schneider. 2014. Agile vs. structured distributed software development: A case study. *Empirical Software Engineering* 19, 5 (2014), 1197–1224.
- [5] William A Florac, Robert E Park, and Anita D Carleton. 1997. *Practical software measurement: Measuring for process management and improvement*. Technical Report. Carnegie-Mellon Univ Pittsburgh pa Software Engineering Inst.
- [6] Martin Fowler. 2006. Using an agile software process with offshore development. WWW page, accessed 2016-04-15. (2006). <http://www.martinfowler.com/articles/agileOffshore.html>
- [7] Geir K. Hanssen, Darja Smite, and Nils Brede Moe. 2011. Signs of agile trends in global software engineering research: A tertiary study. In *Sixth International Conference on Global Software Engineering Workshops (ICGSEW)*. IEEE, 17–23.
- [8] Deborah Hartmann and Robin Dymond. 2006. Appropriate agile measurement: using metrics and diagnostics to deliver business value. In *Agile Conference, 2006*. IEEE, 6–pp.
- [9] Tuomo Kähkönen. 2004. Agile methods for large organizations – building communities of practice. In *Agile Development Conference*. IEEE, 2–10.
- [10] Marco Kuhrmann and Sarah Beecham. 2014. Artifact-based software process improvement and management: a method proposal. In *Proceedings of the 2014 International Conference on Software and System Process*. ACM, 119–123.
- [11] Maarit Laanti. 2014. Characteristics and principles of scaled agile. In *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*. Springer, 9–20.
- [12] Dean Leffingwell. 2018. Scaled Agile Framework® 4.0. WWW page, accessed 2018-01-24. (2018). <http://v4.scaledagileframework.com/>
- [13] Bertrand Meyer, Jerzy R. Nawrocki, and Bartosz Walter. 2008. *Balancing Agility and Formalism in Software Engineering*. Vol. 5082. Springer Science & Business Media.
- [14] John Noll, Mohammad Abdur Razzak, and Sarah Beecham. 2017. Motivation and Autonomy in Global Software Development: An Empirical Study. In *Evaluation and Assessment in Software Engineering (EASE '17)*. Karlskrona, Sweden.
- [15] John Noll, Mohammad Abdur Razzak, Ita Richardson, and Sarah Beecham. 2016. Agile Practices for the Global Teaming Model. In *11th International Conference on Global Software Engineering Workshops (ICGSEW)*. IEEE, 13–18.
- [16] Maria Paasivaara. 2017. Adopting SAFE to scale agile in a globally distributed organization. In *Proceedings of the 12th International Conference on Global Software Engineering*. IEEE, 36–40.
- [17] Maria Paasivaara, Benjamin Behm, Casper Lassenius, and Minna Hallikainen. 2018. Large-scale agile transformation at Ericsson: a case study. *Empirical Software Engineering* (2018), 1–47.
- [18] Maria Paasivaara and Casper Lassenius. 2004. Using iterative and incremental processes in global software development. In *3rd International Workshop on Global Software Development*. 42–47.
- [19] Maria Paasivaara and Casper Lassenius. 2006. Could global software development benefit from agile methods?. In *International Conference on Global Software Engineering (ICGSE'06)*. IEEE, 109–113.
- [20] Jan Pries-Heje and Malene M. Krohn. 2017. The SAFE Way to the Agile Organization. In *Proceedings of the XP2017 Scientific Workshops (XP '17)*. ACM, Article 18, 3 pages.
- [21] Balasubramaniam Ramesh, Lan Cao, Kannan Mohan, and Peng Xu. 2006. Can distributed software development be agile? *Commun. ACM* 49, 10 (2006), 41–46.
- [22] Anu Raninen, Jarmo J Ahonen, Hanna-Miina Sihvonen, Paula Savolainen, and Sarah Beecham. 2013. LAPPI: A light-weight technique to practical process modeling and improvement target identification. *Journal of Software: Evolution and Process* 25, 9 (2013), 915–933.
- [23] Mohammad Abdur Razzak. 2016. Transition from Plan-Driven to Agile: An Action Research. In *17th International Conference on Product-Focused Software Process Improvement*. Springer, 746–750.
- [24] Mohammad Abdur Razzak, John Noll, Ita Richardson, Clodagh Nic Canna, and Sarah Beecham. 2017. Transition from Plan Driven to SAFE®: Periodic Team Self-Assessment. In *18th International Conference on Product-Focused Software Process Improvement*. Springer, 573–585.
- [25] CMMI Product Team. 2010. CMMI® for Development, Version 1.3, Improving processes for developing better products and services. no. CMU/SEI-2010-TR-033. Software Engineering Institute (2010).
- [26] Oktay Turetken, Igor Stojanov, and Jos JM Trienekens. 2017. Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. *Journal of Software: Evolution and Process* 29, 6 (2017).
- [27] Dan Turk, Robert France, and Bernhard Rumpe. 2014. Limitations of agile software processes. *Third International Conference on Extreme Programming and Flexible* (2014).
- [28] Han Van Loon. 2004. *Process Assessment and ISO/IEC 15504: a reference book*. Vol. 775. Springer Science & Business Media.
- [29] VersionOne. 2017. 11th Annual State of Agile Report. WWW page, accessed 2017-07-07. (2017). <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2>
- [30] Aurora Vizcaíno, Felix García, Mario Piattini, and Sarah Beecham. 2016. A validated ontology for global software development. *Computer Standards & Interfaces* 46 (2016), 66–78.
- [31] Karl E. Wiegers. 1998. Molding the CMM to your organization. *Software Development* 6 (1998), 49–53.