Eindhoven University of Technology

MASTER

Scaling agile using scaled agile framework

Stojanov, I.

Award date:
2015

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Technische Universiteit Eindhoven University of Technology

Department of Mathematics and Computer Science
Department of Industrial Engineering & Innovation Sciences

Scaling Agile using Scaled Agile Framework

Master Thesis

Igor Stojanov

in partial fulfilment of the requirements for the degree of
Master of Science in Business Information Systems

Supervisors:
dr. O. (Oktay) Türetken
dr.ir. J.J.M. (Jos) Trienekens
Frederik Kooistra (Phillips)
Martine Vleeshouwers (Philips)

Version 1.0

Eindhoven, August 2014

Abstract

Nowadays IT organizations are facing dynamic business environment. This dynamicity demands from organizations to deliver software at much faster pace and be more tolerable to changing requirements during the project development cycle. The agile software development, as an iterative and incremental approach for software development, has emerged as an alternative to the traditional development methodologies to address these challenges. Among the several different agile methods is the Scaled Agile Framework (SAFe) which arose in the past decade. SAFe is a novel framework for implementing agile practices at enterprise scale. The focus of SAFe is to highlight the roles, activities and artifacts necessary to scale agile to teams, programs and enterprises. This Master Thesis provides in-depth investigation on the benefits of adopting practices from the Scaled Agile Framework. Furthermore, due to lack of an appropriate implementation strategy for SAFe, an evolutionary implementation strategy is proposed based on existing software process improvement and maturity model.

Keywords: Agile; Agile Adoption, Scaled Agile Framework, SAFe, Benefits

Dedication

To the loving memory of my father.

To my mother, who sacrificed all to set me on the right path in life.

I owe every bit of my existence to you both.

Acknowledgements

First and foremost I would like to thank my parents for all their support and guidance throughout my life. I am grateful for their patience, love and attention without which I would have never reached this point in my life.

I would like to thank my supervisors, Dr. Türetken, Dr. Trienekens, Frederik Kooistra and Martine Vleeshouwers for their advice, professionalism and guidance during my master thesis project. Their dedication, kindness and assistance were vital for the completion of my thesis work.

During my graduate internship I was also fortunate to work at Philips with the management team of Martine Vleeshouwers. They made my internship at Philips a positive and fun place to do research and study. These are: Peter Voetman, Henk Mooijweer, Zsuzsanna Rozsahegyi, Maria Lara, Wim Welberg, Skelte Bergsma, Abirami Ganesan, Leendert van Achteren, Fred Jacobs, Gareth Mulligan, Andriy Paliychuk, Rohit Gupta.

I would also want to extend my gratitude to my close friends and relatives. I am especially thankful for the support and assistance of my fellow graduate students. These are: Juby Joseph, Ekaterina Sabelnikova, Kostas Traganos, Goppy Gustaman. Those who I forgot to mention, please forgive me.

Last by not least I am thankful to the department of Mathematics and Computer Science at TU/e and the faculty for their contribution to my learning experience.

Contents

Abstract 3
Dedication 4
Acknowledgements 5
List of Figures 9
List of Tables 10
Abbreviations 11
1 Introduction 12
1.1. Problem description 13
1.2. Research questions 14
1.3. Structure 16
Chapter 2 17
2. Background 17
2.1. Traditional waterfall model 17
2.2. Agile methodologies 18
2.3. Comparison of waterfall and agile methodologies 18
2.4. Scrum 19
2.5. Extreme programming (XP) 20
2.6. Lean software development 21
2.7. Product development flow 21
2.8. Agile enterprise big picture 22
2.8.1. Team level 23
2.8.2. Program level 23
2.8.3. Portfolio level 23
Chapter 3 25
3. Research Methodology 25
3.1. Research approach 26
3.2. Data collection and analysis 27
Chapter 4 29
4. Implementation strategy for SAFe 29
4.1. Proposed implementation strategy 30
4.1.1. IDEAL model 31
4.1.2. Barriers and risks to agile adoption 34
4.2. SAFe maturity model 37
4.2.1. Sidky agile measurement index (SAMI) 38

4.2.2. Developing the SAFe maturity model 40
4.2.1. Development of SAFe maturity model with Delphi study 43
4.2.2. Panel of experts 43
4.2.3. Results 45
Chapter 5 48
5. Case study 48
5.1. Introduction of case company (Royal Philips) 48
5.1.1. Brief description 48
5.1.2. Transformation of Philips 49
5.1.3. Adoption of SAFe at Philips 50
5.2. Related case studies 51
5.3. Time-to-market 52
5.4. Predictability 54
5.4.1. Velocity reliability 55
5.4.2. Cost predictability 55
5.5. Customer satisfaction 57
5.6. Quality 57
Chapter 6 59
6. Conclusions 59
6.1. Research conclusions 59
6.2. Validity and reliability 60
6.3. Limitations 60
6.4. Future research 61
Bibliography 63
Appendix A. Graphs and tables 67
Appendix B. Agile Manifesto principles [3] 71
Appendix C. SAMI populated with agile practices (adapted from [28]) 72
Appendix D. Description of SAFe maturity model practices 74
Appendix E. Indicators per level 81
Appendix F. Delphi study questions and responses 92

List of Figures

Figure 1. Waterfall model 17
Figure 2. SCRUM sprint cycle 20
Figure 3. Enterprise Big Picture v 2.5 22
Figure 4. Research methodology 27
Figure 5. SAFe implementation strategy 29
Figure 6. IDEAL model 31
Figure 7. Components in SAMI Adapted from [28] (Indicators not shown) 38
Figure 8. Delphi method structure 43
Figure 9. Characteristics of the panel of experts 44
Figure 10. Results from evaluation of Maturity Model 45
Figure 11. Scatterplot of time to market projects (in days) 52
Figure 12. Scatterplot of delayed projects (in days) 54
Figure 13. Average velocity of team/sprint 55
Figure 14. Velocity of teams/sprint 55
Figure 15. Box plot representation of cost predictability 56
Figure 16. Bar chart customer satisfaction 57
Figure 17. Pie chart customer satisfaction 57
Figure 18. Scatterplot of defect density 58

List of Tables

Table 1. Comparison of Traditional and agilemethodologies (adapted from [23]) 19
Table 2. Agile levels in SAMI 39
Table 3. SAFe maturity model 47
Table 4. Time to market statistics in days 53
Table 5. Delayed project statistics (in days) 54
Table 6. Cost predictability statistics (in EAC/planned cost) 56
Table 7. Defect rate statistics 58

Abbreviations

AAF – Agile Adoption Framework

ART – Agile Release Train

AWG- Agile Working Group

CMMI – Capability Maturity Model Integration

CoD – Cost of Delay

DoD – Definition of Done

IT- Information Technology

MDT – Multi-Disciplinary Teams

PDF – Product Development Flow

PIL – Philips Integrate Landscape

PSI – Potentially Shippable Increment

QIP - Quality Improvement Paradigm

RAD – Rapid Application Development

RTE- Release Train Engineer

RUP – Rational Unified Process

SAFe – Scaled Agile Framework

SAMI – Sidky Agile Measurement Index

SEI – Software Engineering Institute

SPC – SAFe Program Consultant

SPI – Software Process Improvement

TPS- Toyota Production System

UX- User Experience

WIP – Work in Process

WSJF – Weighted Shortest Job First

XP – eXtreme Programming

Chapter 1

Introduction

Over the past few decades, IT organizations have witnessed increased importance and complexity in software development. The process of building software has evolved from simple ad-hoc executions to multifaceted, multifarious heuristics which continue to rapidly grow and improve. This increase in complexity is particularly evident in larger organizations which are required to design, develop, deploy and maintain systems on large scale. In an attempt to discover a repeatable, predictable process that provides quality and productivity in software development, many different software methodologies have been developed over the years.

The traditional waterfall approach used to be the dominant methodology employed by many organizations [1]. This predictive, heavily-documented methodology is effective when there is no ambiguity in the software requirements (they are fixed, clear and understood by all stakeholders), the project definition is stable, and resources and experts are freely available. However, the modern reality of software development is that changes in requirements are inevitable during the development life-cycle. Furthermore, there are increasing competitive pressures from organizations to deliver high value and high quality software at faster rates [2]. This led to the increase in popularity of new leaner and more agile software development methodologies which embrace changes in requirements and put high emphasis on customer collaboration and working software [3].

The evolution of software development methodologies has evolved from predictive (Waterfall) to iterative and incremental (e.g. Spiral, RAD, RUP), and Agile Processes (e.g. Scrum, XP, Lean, Kanban) [Appendix A, Figure 1]. Even though organizations started to understand and adopt the advantages of these new methods, they were primarily intended for small teams and organizations rather than large enterprises with several hundreds or thousands of development teams. In an attempt to scale the advantages of these new methodologies, Dean Leffingwell in 2011 created a new agile, scalable model commonly known as the Agile Enterprise Big Picture: Scaled Agile Delivery Model [2], [4]. Notwithstanding the novelty of the model, it has already been applied in large scale development in software enterprises worldwide with encouraging results and several success stories. In fact, the Agile Enterprise Big Picture framework is rapidly emerging as the industry standard approach for scaling agile in the enterprise. This case study aims at exploring the challenges and benefits of the proposed framework in a large international company.

1.1. Problem description

The Agile Enterprise Big Picture Framework, commonly referred as Scaled Agile Framework (SAFe), was primarily developed for organizing and managing agile practices in large enterprises. However the implementation of the Scaled Agile Framework in practice is not a trivial task. Currently, there is a lack of well-structured gradual implementation approach for implementing and scaling SAFe [2], [5]. Furthermore, the benefits of SAFe adoption are scarcely documented in few mainly self-reported case studies. Therefore the main problems that are addressed in this thesis are the challenges of SAFe adoption and the potential benefits of such endeavor. These two problems have not been addressed in academic literature in great depth, mainly due to the recent development of the SAFe model. The emerging growth of SAFe in industry and practice requires an academic investigation which will address the current challenges and benefits of SAFe adoption.

The Scaled Agile Framework (SAFe) itself is a product of continuous work. Currently SAFe is being documented in version 2.5 of the Big Picture Framework which contains the current set of best practices of agile and lean principles. An updated version 3.0 of the SAFe Big Picture is scheduled for release in the summer of 2014 with new changes primarily in the portfolio section. The current state of the framework has been attained through multiyear successful and failed implementations and experimenting in many companies. As stated by the creators and contributors of SAFe, the framework is publicly available in order to help individuals, teams and enterprises achieve their potential and realize the business benefits that this innovative framework provides [5].

Although the theory behind SAFe is getting wider acceptance in industry, in practice companies tend to experience numerous challenges when transitioning and scaling agility [see Section 4.1.2]. As stated in the Scaled Agile Framework, SAFe is "focused solely on describing the best practices, roles, activities, and artifacts that enterprises have used to achieve significant business, economic and individual benefits that result from successful implementation of Lean|Agile methods at enterprise scale" [5]. Moreover, the founders of the framework go on to claim that "SAFe does not implement itself and makes no attempt to describe the significant organizational change management, cultural impacts, implementation strategies…that are typically required for successful implementation" [5]. Several studies have already acknowledged the potential risks and challenges of adopting agile practices [22], [23], [24]. These risks can range from lack of implementation strategy, lack of management support, or technological and process limitations. Challenges can be also posed by a specific organizational culture or IT-governance practices which are not aligned with principles of SAFe. Moreover, since the framework puts very high emphasis on co-located teams, face-to-face communication and small teams size, these restrictions may have negative implications for large companies which want to leverage global IT-outsourcing and dynamic partnerships. Based on the above summarized description the following problem statement is formulated:

What are the benefits and challenges of adopting Scaled Agile Framework in a large Enterprise?

The interest of this research effort is to investigate the practical challenges of adopting SAFe as well as report on the practical benefits of this endeavor. The challenges refer to the needed foundation that large organizations should possess in order to successfully adopt SAFe. Besides, as stated previously, the current state of the framework does not provide guidance for change and transition management, implementation strategies and desired organizational culture which undoubtedly have impact on the rate and success of agile adoption. The lack of proper implementation strategy is seen as the key challenge for successful implementation of the framework. This challenge will be addressed by proposing a gradual implementation approach which is suitable for enactment of SAFe in large and complex enterprises. Furthermore, there are handful and limited number of studies that investigate the benefits of adopting the Scaled Agile Framework. To further elaborate on the scope of the problem statement, this Master Thesis project will also give new insights into the potential benefits of adopting the framework. These benefits will be documented though a case study analysis in a large international company. The benefits of interest are reduction in time-to-market and increases in predictability, quality and customer satisfaction.

1.2. Research questions

From the problem description and the problem statement in the previous section, several research questions arise which will be elaborated in this section.

As the popularity of agile adoption increases, the questions organizations ask themselves shifts from why to adopt agile practices to the how to adopt and scale these practices. SAFe tends to focus merely on describing the best practices, roles and artefacts of agile and lean principles but makes no attempt to describe any implementation strategies [5]. Although there are several examples of successful SAFe adoptions, these success stories are typically narrowly focused to the specific organizations [See Chapter 5.2]. In the current context, there is a lack of proper broad guidelines which can guide enterprises on the necessary preparation for agile and SAFe, outlining the organizational change management and implementation strategies. Several studies [21], [22], [23], [24] have attempted to outline the risks involved in agile transformations. However, these studies do not attempt to propose implementation strategies which will assess and mitigate those risks. Other studies have proposed some adoption strategies and maturity models for agile adoption [27], [28], [55], but none of them have been widely accepted or address the problem of scaling agile practices to enterprise level. Therefore, there is a need for a more structured approach for adopting and the Scaled Agile Framework in large enterprises. The first research question is formed to answer the problem mentioned above:

RQ1: How can the Scaled Agile Framework be implemented in larger enterprises?

As a continuation to the research question above, this research effort will also investigate the needed foundation large enterprises necessitate for effective transformation to Agile and SAFe. The term foundation refers primarily to the organizational characteristics such as organizational culture, current processes, people, technology and incentive structures which have to be aligned with the values of Agile and SAFe. These barriers and risks of adopting agile way of working will be documented through a literature research. For instance, some papers [23], [25] indicate that culture and cultural support is an important factor for embracing agile ways of working. An implementation of a framework and its underlying processes in every organization is to some extent unique due to the differences within each organization, and in that respect the transformational process design is untested and unverified. Nonetheless a broad, gradual and structural approach of implementation of SAFe is needed for better managing and supporting the transformation to agile and SAFe.

The Scaled Agile Framework claims multiple comprehensive and sustainable benefits which accelerate time-to-market, increase productivity and quality and reduce risks and project costs. Although several case studies in several companies [see Section 5.2] have verified some of these benefits, this thesis report will provide more extensive investigation on the benefits of SAFe adoption. A more holistic method of evaluation will be applied while looking at attributes such as time-to-market, on-time delivery, velocity reliability, cost-predictability, customer satisfaction and quality. These benefits will be examined through a case study in a large international enterprise which is adopting the Scaled Agile Framework. Thus, the second research question is stated as follows:

RQ2: What are the effects of adopting SAFe in terms of time-to-market, predictability, customer satisfaction and quality?

To answer these questions, first an extensive review of current literature was performed. Next to the literature review, a case study was conducted in large enterprise which at the time of writing of this thesis, was in the process of adopting Agile|Lean principles as well as the Agile Enterprise Big Picture of Dean Leffingwell. The research approach that was followed to answer these questions is described in Chapter 3.

1.3. Structure

This thesis is structured as follows.

Chapter 1 provides a brief introduction to the subject. Furthermore, the problem statement is elaborated and several research questions are formulated to address the problem statement.

Chapter 2 presents the background information. The intent of this chapter is to give to the reader clear understanding of the Scaled Agile Framework and of the different aspects that play a role in this thesis. The background information describes the Scaled Agile Framework and the principles of Agile, Lean, Product Development flow that are embedded in the framework itself. Furthermore, waterfall and agile software methodologies are briefly explained and compared.

Chapter 3 elaborates on the research approach which is used during the research.

Chapter 4 addresses the first research question. This chapter proposes implementation strategy for SAFe and describes how the maturity model for SAFe is developed. The implementation strategy is based on existing SPI framework and maturity model.

In Chapter 5 the second research question is addressed and the benefits of adopting the framework are empirically validated through a case study. The SAFe maturity model developed in the previous chapter is applied in the case organization. Short review of similar case studies on the benefits of adopting SAFe is also provided.

The report ends with conclusion and limitations of the study in Chapter 6.

Chapter 2

Background

This chapter is concerned with the concepts and terminology used throughout this thesis. Background information about the different principles and aspects upon which SAFe is constructed is provided. Furthermore, several aspects which play an important role in this research are described.

2.1. Traditional waterfall model

The waterfall model was formally introduced in a published article by Winston W. Royce in the 1970's. In this article Royce presented his own personal view on managing large software developments [46]. The waterfall model is essentially a sequential design process in which each life-cycle of software development phase flows downwards, similarly to a waterfall [Figure 1]. The main life-cycle phases have to be planned up front and the whole project is considered to be predictable. Also, the customer requirements are considered to be understood and unchanging. The strengths of this approach are the facts that it clearly outlines the phases of development and stresses the importance of requirements. However, the main limitations are that software projects rarely follow sequential flow and clients rarely can declare all the requirements up-front [45]. Even Royce in his conclusion of the proposal of this simpler model for software development writes the following "In my experience, however, the proposed model has never worked on large software development efforts…" [46]. As Royce had predicted, the waterfall model has been shown to be unsuitable for large software development efforts. Due to these limitations, several other software development methodologies have been developed over the years. Among those are the agile methodologies which are elaborated in the section below.

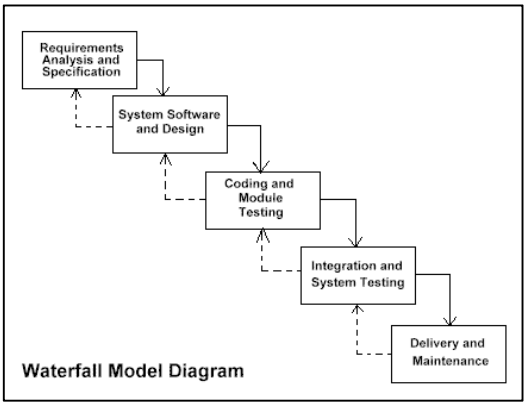


Figure 1. Waterfall model

2.2. Agile methodologies

In the 1990's, agile methodologies initially arose as an opposition movement to the traditional, heavy-process oriented methodologies which were dominating the software industry during that time. They emerged due to the evolving and changing software requirements, which called for a much lighter, effective, and human oriented software development techniques [8]. The movement toward more lightweight and agile methodologies gave rise to multitude of new approaches to software development (e.g. Extreme Programming, Scrum, Adaptive software development, Feature-Driven Development, Dynamic Systems Development Methodology). In an attempt to unify the movement towards the agile approaches on the 11th of February 2001, 17 agile proponents met in order to establish a common ground. As a result of the meeting, they created an Agile Manifesto which outlined the guiding principles and values of agile software development [3]. The values of the Agile, as outlined in the Agile Manifesto are as follows:[3]

- *Individuals and interactions* over *Processes and tools.*
- *Working software* over *Comprehensive documentation.*
- *Customer collaboration* over *Contract negotiation.*
- *Responding to change* over *Following a plan.*

These value statements have a specific form i.e. in each bullet point the first segment indicates a preference, while the latter segment describes an item that, though important, it is of lesser priority [3]. The Agile Manifesto also includes 12 principles which provide guideline for using agile methods in order to achieve the values outlined above. These 12 agile principles are listed in Appendix B.

Since SAFe conglomerates several different agile methodologies (e.g. Scrum, XP, Kanban), elements of Lean thinking and Product Development Flow, in this section of the thesis a brief description of these elements is also given.

2.3. Comparison of waterfall and agile methodologies

Both the waterfall and agile methodologies have the same common goal which is to deliver a quality product in a predictable and efficient manner. Also, both the waterfall and agile methodologies depend on the same programmatic building blocks: scope, cost, schedule and performance. In waterfall the scope of the project is fixed and the cost, schedule and performance are allowed to vary. In the agile methodology however, the scope is estimated and the cost, schedule and performance are fixed. This is due to the difference in fundamental assumptions of the two methodologies. In waterfall development it is assumed that the software requirements can be fully specified beforehand through extensive planning. Agile development recognizes that requirements cannot be fully predicted beforehand and they might change throughout the development process. This is one of the fundamental differences between these

two software methodologies. There are other differences in respect to the way projects are being managed and developed in both methodologies. These differences are summarized by a study of Nerur et al. [23] and displayed in the Table 1.

Table 1. Comparison of Traditional and agile methodologies (adapted from [23])

	Traditional methodologies	Agile methodologies
Fundamental Assumptions	Systems are fully specifiable, predictable, and can be built through meticulous and extensive planning.	High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change.
Control	Process centric	People centric
Management Style	Command-and-control	Leadership-and-collaboration
Knowledge Management	Explicit	Tacit
Role Assignment	Individual—favors specialization	Self-organizing teams—encourages role interchangeability
Communication	Formal	Informal
Customer's Role	Important	Critical
Project Cycle	Guided by tasks or activities	Guided by product features
Development Model	Life cycle model (Waterfall, Spiral, or some variation)	The evolutionary-delivery model
Desired Organizational Form/Structure	Mechanistic (bureaucratic with high formalization)	Organic (flexible and participative encouraging cooperative social action)
Technology	No restriction	Favors object-oriented technology

We must mention that both waterfall and agile development have their own advantages and disadvantages. However, there are particular settings which make one methodology more suitable than the other. The current dynamic environment tends to favor more agile approaches of software development.

2.4. Scrum

Scrum is an iterative, incremental process of software planning, development, testing, release and deployment [9]. It was originally developed by Jeff Sutherland and Ken Schwaber in 1995 and is currently one of the most popular agile methodologies [10]. The Scaled Agile Framework borrows several scrum practices, especially at the Team Level. Thus in Scrum, as well and

SAFe, the development process is divided in time-boxed cycles called sprints [Figure 2] which can last from 1 up to 4 weeks. At the beginning of each sprint, a sprint planning session takes place in which a multi-disciplinary team selects requirements which can be implemented in the upcoming sprint. The required functionality is managed with a product backlog which contains prioritized backlog items [2], [11]. Each day, the team has daily stand-up session during which the progress of the team is discussed and impediments are resolved. At the end of each sprint, the team and the stakeholders review the results together. Periodically, the team has retrospective meeting during which it reflects on the development process to figure out how to improve it [10]. An example of Scrum Framework is displayed in Figure 2 below.

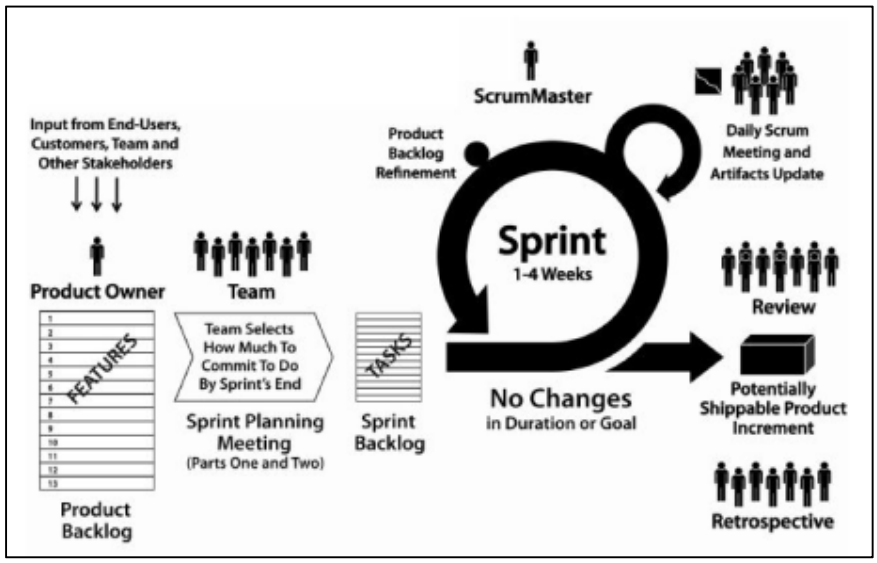


Figure 2.Scrum sprint cycle

The Scrum methodology introduces several roles which compose a multi-disciplinary team which are also adopted in SAFe. These roles are described as follows:[11]

- Product Owner – person which works closely with the team and is responsible for the vision, budget and prioritizing the product backlog items.
- Development Team – cross-functional, self-organizing team responsible for delivery and testing of incremental software.
- Scrum Master – person who is supporting the team in achieving its goals and eliminates impediments.

2.5. Extreme programming (XP)

Extreme Programming, as one of many agile methodologies, has also been widely used in the past decade. It was invented in 1999 by Kent Beck and refined later in 2004 [13]. This methodology also tries to address the changing customer requirements while improving software quality and responsiveness. XP, as a prescriptive methodology, is described in terms of thirteen

primary practices and eleven corollary practices [14]. Out of these practices, several are adopted in the SAFe such as: [2]

- Team of five to ten programmers work at one location with customer represented on site;
- Development occurs in frequent build of iteration and delivers incremental functionality;
- Requirements are specified using user stories;
- Programmers do their own unit testing and customers participate in acceptance testing;
- Requirements, architecture and design emerge over the course of the project.

2.6. Lean software development

The lean software movement originated from the success of Toyota and its production system (TPS) [15]. This lean production system is about trying to preserve value while minimizing work. This is achieved by getting the right things to the right place at the right time while reducing waste and being open to change [15]. The success from the Toyota Production System initiated a lean movement towards applying the practices and principles of lean in software development. Dean Leffingwell incorporates several of these principles in SAFe while applying the framework for lean software thinking of Larman and Vodde [2], [16]. The goal of providing maximum value to the customers is achieved by continuous improvement, having respect for people and utilizing a product development flow which prescribes how software should be developed and deployed to end users. The management support plays an essential role in the implementation of lean. Unlike agile where management mainly supports and eliminates impediments, in lean software development management is expected to have a leading role in driving continuous improvement [2].

2.7. Product development flow

As we have already mentioned SAFe incorporates various principles from Agile, Scrum, Extreme Programming, and Lean in order to provide substantial benefits to the enterprises which adopt this framework. SAFe is also built upon principles of Product Development Flow which teams can use to develop and deliver working software. Among these principles are the eight major themes (principles) of Product Development Flow of Reinertsen. [47] These principles are: [2]

1. Take an economic view.
2. Actively manage queues.
3. Understand and exploit variability.
4. Reduce batch sizes.

5. Apply WIP constraints.
6. Control flow under uncertainty- cadence and synchronization.
7. Get fast feedback
8. Decentralize control.

These eight principles are incorporated into the House of Lean based on the model of Larman and Vodde [2], [16]. They provide economic and quantitative basis for the Scaled Agile Framework and these principles are instantiated in several SAFe practices throughout the framework.

2.8. Agile enterprise big picture

The SAFe Enterprise Big Picture [Figure 3] is a visual representation of the framework developed by Dean Leffingwell and it is meant to serve as both organization and process model for agile requirements practices [2]. SAFe uses already existing body of work in terms of Scrum, XP, Lean, and Product Development Flow as described in the previous section. In essence the framework is separated into 3 separate perspectives, namely Team Level, Program Level, and Portfolio Level. The boundaries between these three levels are arbitrary and serve as a model for abstraction of scope and scale between levels [2].

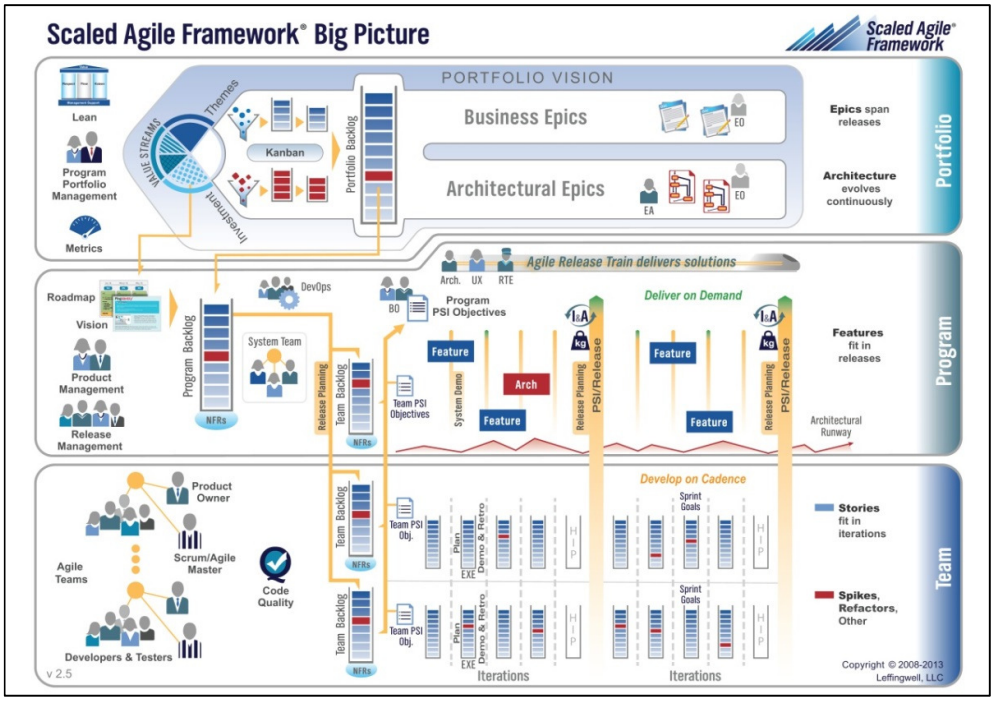


Figure 3.Enterprise Big Picture v 2.5 (Copyright Scaled Agile, Inc. 2014. Used with permission)

2.8.1. Team level

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 in order to define the software requirements which are placed in the Team Backlog. Also, Scrum components such as Sprint Planning, Daily Stand-ups. Definition of Done (DoD) and Retrospectives are adopted at each iterations. 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. The teams are self-organizing and typically consist of: Developers, Scrum Master, Product Owner, Testers, and Tech Leads. [5]

2.8.2. Program level

The primary goal of 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 the 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 the Program Level of SAFe is the Agile Release Train (ART) which provides cadence and synchronization. The ART produces releases or Potentially Shippable Increments (PSI's) at fixed time boundaries, typically 60 to 120 days [2]. The releases are planned during a two day Release Planning event which involves all relevant stakeholders. An evolving architecture is planned and developed ahead of business functionality. Furthermore, a System Team is formed in order to establish initial infrastructure and support continuous integration and end-to-end testing efforts. Roles and teams which are involved on Program level include DevOps/Systems Team, Release Management, Product Management, System Architect, Business Owners, Release Train Engineer (RTE) and User Experience (UX).

2.8.3. Portfolio level

The Portfolio Level, as higher level of abstraction, is needed for very large enterprises which employ thousands of practitioners and developers. For these enterprises SAFe introduces new governance and management models with new artifacts. The portfolio level in essence is about achieving balance between four potentially conflicting goals [12]. The first goal is to maximize the financial value of the portfolio by identifying Value Streams using Kanban Systems. The second goal is about linking the portfolio to the strategy of an organization through investment themes. Third goal is about ensuring that the scope of activities is feasible by measuring appropriate metrics. The last goal tries to balance the portfolio on relevant dimensions by defining and managing business and architectural epics which run across value streams. These goals are achieved through several artifacts: Portfolio Vision, Lean thinking paradigm, Program

Portfolio Management and Metrics. Roles on this level are more loosely defined and typically include Portfolio Management, Enterprise Architects and Enterprise Owners.

Chapter 3

Research Methodology

Research methodology provides a systematic approach and defines the path to solving the predefined research objectives. The purpose of this study is to gain new insights into the prospective benefits and challenges of adopting the Agile Enterprise Big Picture Framework in a large enterprise. Due to the novelty of the framework there exist only a handful number of studies which examine and outline the process of implementing this novel organizational and process framework. Since theoretical models on implementation and benefits of SAFe are scarcely available, the scope of this research effort is of exploratory nature rather than hypothesis testing. By exploring the proposed research problems in more depth, this research will contribute to creating the basis for future research by building theoretical models and potential hypothesis to be tested.

For these reasons an exploratory research methodology in the form of case analysis will be used for answering the research questions. Exploratory research typically builds on secondary research such as review of literature or data, or qualitative and formal approaches such as in-depth interviews, focus groups and cases studies [50]. The research strategy used in this master thesis triangulates between three different research methods: literature research, case study and Delphi method research. This triangulation enables us to obtain richer insight of reality since different research methods focus on different aspects of reality and therefore a richer understating of a research topic will be gained by combining several methods together in a single piece of research or research program [51].

The focus of a case study research is to gain objective understanding of the dynamics of a real process, evaluating it in its natural context. Case study research is appropriate in settings where the researcher is confronted with unfamiliar situations for which there is no consolidated theoretical base [19] or during early stages of research of a new phenomenon [20]. Case study methodology is also used in the initial stages of a development of a new theory, although in some situations case studies can be used to extend and support existing theories or to generate questions on these [18]. In this research, case study is used for observing the benefits of SAFe in a large international enterprise.

New knowledge does not emerge out of nothing. This research effort, as many others which preceded it, builds on existing findings and discoveries found in existing literature. The focus of a literature review is to survey academic articles, conference papers, books, and others sources which are relevant to our topic of interest. In this research, literature review is used when developing the objectives of this research, formulating the theoretical models, and when trying to better understand the subject under consideration.

Delphi study is particularly useful for this research since it is well aligned with exploratory theory building on complex, interdisciplinary issues, often involving a number of new or future trends [52]. The Delphi method itself, as a structured approach for soliciting expert opinion on a particular topic, is used for development and limited validation of the proposed implementation strategy and maturity model for SAFe.

3.1. Research approach

Research approaches are plans and procedures for research that span the steps from broad assumptions to detailed methods of data collection, analysis and interpretation. In this section, the broad steps are defined. First, the specific research problems were identified which required further exploration. These problems are related to the lack of implementation strategy for the Scaled Agile Framework and the scarcity of empirical evidence for the benefits of SAFe. Next, an extensive review of current available literature was performed in order to get clarity and focus when developing the research objectives. During the literature review several approaches of implementation strategies was explored, including various existing implementation strategies, current case studies on the benefits of SAFe, Software Process improvement (SPI) methods and agile maturity models. After completing the literature research, clear objectives and goals are outlined in the form of research questions which address the main research problems statement. The first problem statement pertaining to the lack of proper SAFe implementation strategy initially was answered by developing initial implementation strategy based on the extensive literature review. To bring cogency and increase the applicability to the implementation strategy and maturity model, a Delphi study was utilized consisting of two rounds of feedback. After the second round, the SAFe maturity model and implementation strategy was finalized. In the next step, a case study was conducted in the case organization. During this step, data was collected and analyzed and interpretation of the obtained results was presented. Previously developed SAFe maturity model was also applied in the case organization in order to measure SAFe maturity. The research methodology concludes with generalization and interpretation of the observed results. Limitations to the study are deliberated and threats to reliability and validity are also discussed. The path which is used in this master thesis is presented in Figure 4.

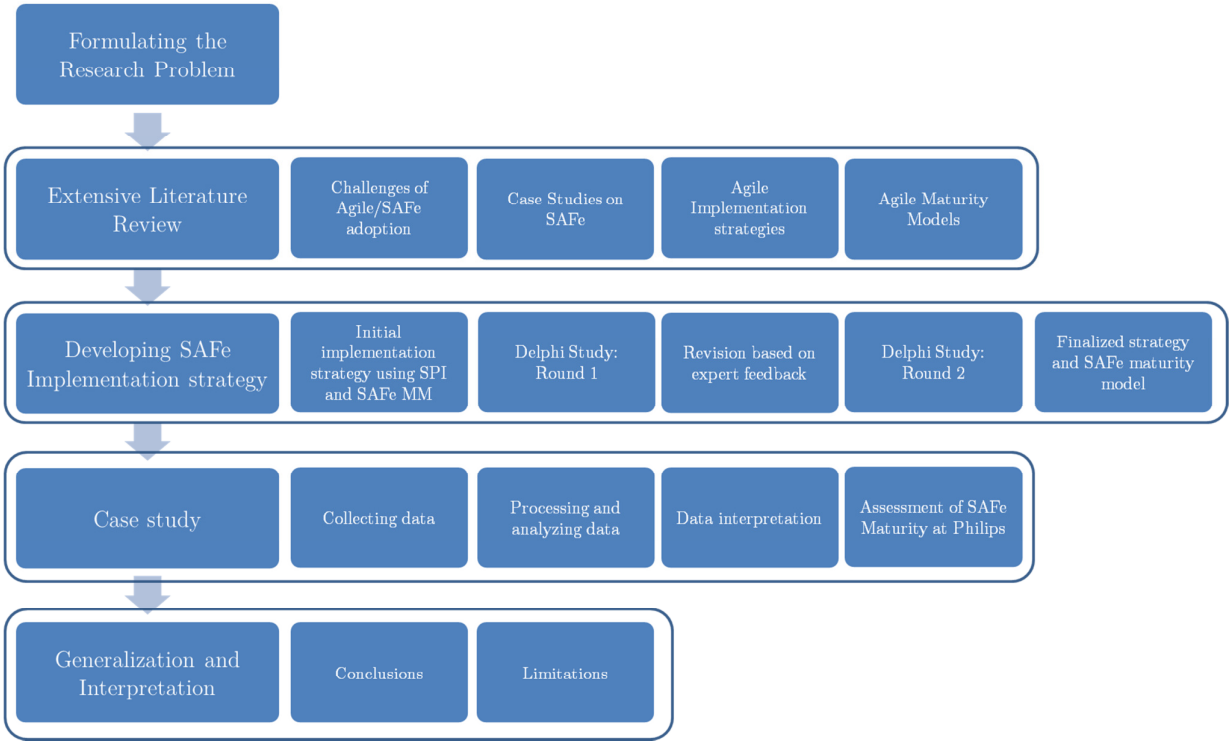


Figure 4.Research approach

At each operational step of the research process, as shown in Figure 4, we have to choose appropriate methods and procedures which will solve our research questions. Since our research questions differ in many aspects, they require different methods for collecting, analysis and interpretation of data.

3.2. Data collection and analysis

The process of collecting and preparing data for analysis is essential for addressing any research objective. Proposing an implementation strategy and maturity model is the first research objective in this master thesis. Commonly used method for establishing validity of a new framework is a longitudinal study. During the longitudinal study, the researcher has an opportunity to observe, gather data and compare the newly proposed implementation strategy to an organization which is not using the proposed strategy. Due to the long timespan, this type of longitudinal study was not viable for this research effort. Instead, the proposed implementation framework was developed and evaluated using a Delphi method with two rounds of feedback. A panel of experts in the case company was gathered in order to collect ideas and facilitate consensus among individuals who have expert domain knowledge in the subject. Online questionnaires were used for collecting data during the two rounds of the Delphi study.

The assessment of the current level of maturity in terms of adopted SAFe practices was done with the assessment model shown in Appendix E. The appraisal of the benefits from SAFe in terms of time-to market, predictability, quality and customer satisfaction adoption required collecting quantitative measurements observed in the case study organization, which is in the process of adopting the framework. Information was acquired through collection and analysis of existing data. This data was obtained through the Project Office Insight Center in the case company, which contains confidential information of the company performance and project metrics. The data was analyzed using SPSS which is software package often used for statistical analysis. The characteristics of the data were assessed by looking at descriptive statistics of important variables, scatter plots, correlations and associations.

Chapter 4

Implementation strategy for SAFe

Past research shows that changes in software development are followed by complex organizational changes since adopting a new methodology cannot simply be accomplished by replacing old tools and techniques with new ones [30]. In practice very few organizations are able, psychologically or technically, to take on agile development approaches immediately and adopt them over a short period of time and full transition often takes few years [38]. As pointed out previously, SAFe lacks proper implementation strategy which can guide and assist organizations during the agile adoption. In that sense transitioning an organization to SAFe is currently an unpredictable process. The only available implementation approach as prescribed by SAFe is to train, certify and coach everyone on the philosophies behind SAFe [Figure 5]. However, many improvement efforts which introduce new development environment are complex, and their effects are so far reaching that they require a specialized, systematic approach for managing the adoption cycle [29].

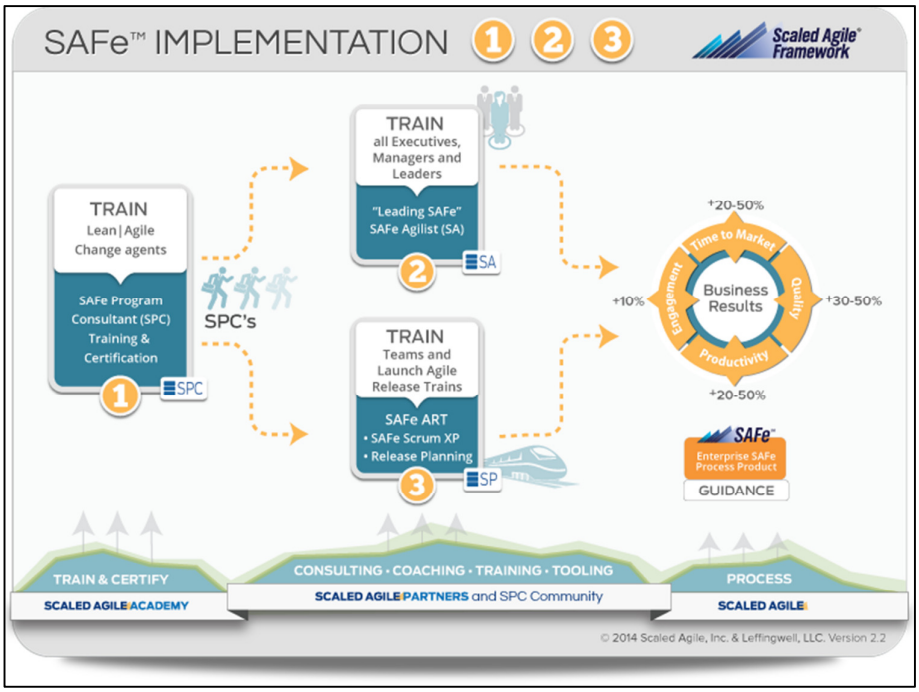


Figure 5.SAFe implementation strategy (Copyright Scaled Agile, Inc. 2014. Used with permission)

The overall process of transforming to an agile organization while applying the framework involves radical changes to an organization in terms of processes, people, culture and similar

aspects. Therefore, the current revolutionary approach could be replaced with a more evolutionary approach of adopting and scaling agile practices which allows for more incremental changes while testing for fit and effectiveness. The wholesale adoption approach which is currently suggested by SAFe, could be replaced with a more gradual adoption approach which is recommended in this thesis report.

Agile adoption is a complex process which has to be conducted effectively through some guiding principles. Since each organization is described by distinctive set of characteristics which makes them unique, these guiding principles should have narrow focus in order to be applicable in diverse settings. Also, the guiding principles have to be repeatable in order to increase their reliability. In the past there have been several attempts to provide a generalizable framework for agile adoption [27], [28], [55]. A comprehensive literature analysis on agile adoption strategies has revealed that currently there are twelve gradual implementation strategies for agile adoption, and only one wholesale strategy where entire agile process is adopted at once [55]. Moreover, case studies on the wholesale strategy have confirmed that wholesale strategy cannot be successful due to lack of substantial preparation for agile adoption and lack of consensus from the team [56]. The wholesale strategy is also considered a more risky approach when compared to the incremental adoption approach [55], [56]. To uphold the values and principles of SAFe, organizations could use a strategic and gradual approach which will guide organizations toward an effective transition to an agile mindset. This research will use the current available literature and best practices in order to develop a new structural implementation strategy for SAFe adoption. The implementation strategy is based on existing SPI and maturity models.

4.1. Proposed implementation strategy

In many cases, organizations want to adopt agile values and SAFe as a way to improve performance, time-to-market, adaptability, reducing risk, etc. Since the adoption of SAFe involves re-design of the current organizational process, adopting SAFe can be seen as a software process improvement effort. Software process is defined as a set of activities, methods, practices, and transformations that people use to develop and maintain software and its associated products [26]. Process improvement on the other hand, as described by Deming, is an iterative cycle of improvement consisting of six steps: [26]

1. Understand the status of a development process.
2. Develop a vision of the desired states.
3. List improvements actions in priority order.
4. Generate a plan to accomplish the required actions.
5. Commit the resources to execute the plan.
6. Start over at step 1.

This improvement process cycle of Deming influenced a number of similar software process improvement (SPI) models e.g. Stewart-Deming's cycle, QIP, IDEAL, ISO/IEC 15504. There are number of similarities between these models and several of them have already been used for deploying agile practices [27], [28]. The implementation strategy proposed in this Master Thesis is built based on the IDEAL SPI model. The IDEAL Model is chosen due to its higher usage, recognition, applicability and maturity over the other similar SPI models. It also provides the basis for continuous improvement during iterations of the process improvement cycle. There are other existing SPI models e.g. Capability Maturity Model Integration (CMMI), a process improvement model with high recognition and usage, but is not utilized in this research due to the limitation of its iterative approach. The CMMI doesn't provide continuous process improvement and ways for sustaining those improvements.

4.1.1. IDEAL model

The IDEAL model, as developed by the Software Engineering Institute (SEI) at Carnegie Mellon, is an organizational improvement model that serves as a roadmap for initiating, planning, and implementing improvement actions [see Figure 6], [29].

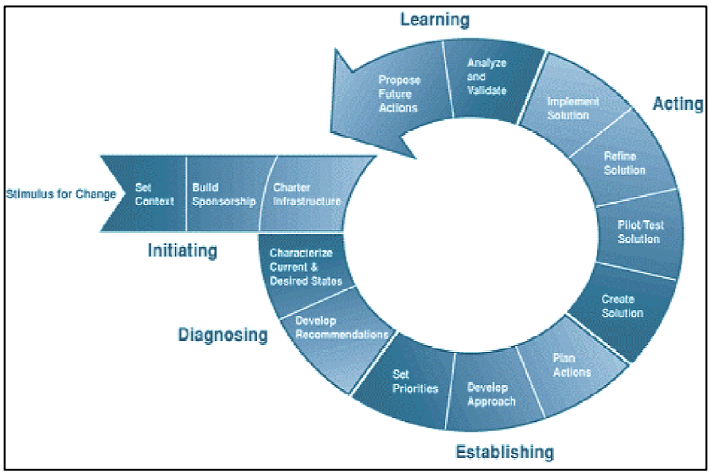


Figure 6.IDEAL model (adapted from [29])

The word IDEAL is just an acronym for the five phases it describes, namely: [29]

I – Initiating: Laying the groundwork for a successful improvement effort.

D – Diagnosing: Determining where you are relative to where you want to be.

E – Establishing: Planning the specifics of how you will reach your destination.

A – Acting: Doing the work according to the plan.

L – Learning: Learning from the experience and improving your ability to adopt new technologies in the future.

Sidky proposed the IDEAL model to be used as a disciplined approach for software improvement and adopting agile practices [28]. In this thesis report the IDEAL model was also used as bases for guiding organizations in adopting SAFe. The five phases and sub-phases of IDEAL are clearly described in the available SEI publication on IDEAL [29]. This publication was adapted and used to describe the five phases used for implementing SAFe. The usability of IDEAL was authenticated during the Delphi study. In the sections below a succinct description of the five phases of IDEAL is given in the context of SAFe adoption.

Initiating

During the initiation stage an organization should evaluate the business and practical reasons for adopting SAFe. Typically, there should be some external or internal reasons which prompt organizations to apply SAFe. The reasons can range from improving operational performance (e.g. time-to-market, productivity, employee engagement) to increasing overall quality of software, effectiveness and scalability. If the stimulus for change is self-evident and not forced or involuntary, there should be greater buy-in from all levels of the organization [29]. During the initiation stage an organization should also set clear context in terms of how SAFe aligns with the specific business goals, objectives and other ongoing initiatives in the organization. Having management support which can build sponsorship is important in the initial stage to sustain early uncertainties. Once all these aspects are in place, an organization should set up initial mechanism i.e. charter infrastructure for managing implementation details. Depending on the size of the organization, this chartered infrastructure can be composed of e.g. one certified SAFe consultant to a full Agile Working Group (AWG). Charting infrastructure, according to the IDEAL model, also involves the development of written agreements which describe clear responsibilities for these people/groups [29]. Ideally, the responsibilities the AWG would include overall organizational change management efforts, establishing strong support from executives and management as well as diagnosing and deployment of agile and SAFe practices.

As shown in Figure 6, the initiation phase is not cyclical and it only happens once. Therefore, it is of vital importance that the initiation phase is done completely and effectively. If that is not the case, all subsequent activities will suffer greater disruptions and needlessly waste time, effort and resources. The main focus in the initiation phase should be on the barriers to agile adoption. These barriers to agility are elaborated in Section 4.2.2.

Diagnosing

The diagnosing phase is characterized by determining the current and desired state of an organization and developing recommendations on how to bridge the gap. The diagnosis is accomplished by using the SAFe Maturity Model which is proposed in Section 4.3. As organization matures, more and more SAFe and agile practices are becoming better defined and implemented throughout the organization. The current level of maturity has to be diagnosed

since it will serve as a reference point from where the organization started. Also, it will provide insight of how agile an organization is in terms of the agile and SAFe practices it has already implemented. Based on the assessment of the current situation, recommendations should be developed with subsequent activities which will push organizations toward their desired maturity level. The desired maturity level might be full or partial implementation of the SAFe and agile practices.

Establishing

The next stage phase of the IDEAL model is the establishing phase. During this stage a detailed action plan should be formulated by AWG on how to adopt and implement agile practices and scale those practices to the specific enterprise. The data gathered during the diagnosing phase should be used when developing the action plan. Since the context to which each organization applies SAFe practices might differ, we won't go into too many details on what the action plan should contain. But typically, an action plan defines what actions are needed; the stakeholders responsible for carrying out those actions; the deadlines and resources needed for those actions and the preferred way of communicating those actions. Furthermore, the organization should set priorities for the change effort based on the current available resources and internal organizational factors. After developing the approach, a rollout strategy specific to each organization can be realized. The strategy includes schedule, tasks, responsibilities, measurements, risks and other elements as required by the specific organization [29].

Acting

The most time and resource consuming phase is the acting phase. During this phase all previous planned activities are taken into action and are implemented. The first step of the acting phase is to create best-guess solution on how to implement and scale SAFe practices. This solution should be created by the previously developed competencies of the AWG or SAFe consulting partners. Most companies typically adopt agile practices on small pilot programs to minimize risks of negative impacts. The IDEAL model also suggests that the proposed solution first be tested via pilot execution [29]. Based on the successes and failures of the pilot implementation, the organization should refine the existent solution plan to reflect the knowledge, experience and lessons learned from the pilot. Once there is a fully workable solution the organization should proceed with implementation of SAFe practices throughout the organization. There are several possible roll-out solutions for SAFe practices e.g. top-down, bottom-up and just-in-time. The decision on which implementation solution is appropriate should be left to the individual organization based on the available resources and priorities.

Learning

As mentioned previously, the IDEAL model was chosen, among other reasons, due to its iterative cycle of improvement which is aligned with the agile/lean principle of continuous improvement. Therefore in the last phase i.e. learning phase the organization should analyze and validate the accomplishments and lessons learned of implementing SAFe practices. Also, the business needs which were identified during the initiation phase should be re-examined to check if SAFe practices truly provided the needed benefits. Based on the analysis and validation, future proposed actions should be undertaken by the appropriate levels of management [29]. After completing the learning phase, the organizations should proceed with diagnosis and reiterate through the IDEAL cycle until the desired state of SAFe maturity is reached.

This IDEAL model offers a high-level approach to SPI and should form the infrastructure for planning and guiding deployment of agile and SAFe practices. Therefore, The IDEAL model is used as foundation for the new proposed implementation strategy. However, an SPI lifecycle alone cannot be used as implementation strategy for SAFe. The IDEAL model is supplemented with maturity model which will guide the organization in terms of the appropriate agile techniques and SAFe practices that need to be attained.

The scope of this research is focused mainly on the initiation and diagnosing phases of the IDEAL cycle. During the initiation phase several barriers and risks to agile adoption should be assessed which are discussed in Section 4.1.2. During the Diagnosing phase of IDEAL the maturity model for SAFe could be used for diagnosing the current maturity in terms of the SAFe practices which have already been adopted by the enterprise. The proposed SAFe maturity model, which is constructed by extending the agile maturity model of Sidky is elaborated in Section 4.3.

4.1.2. Barriers and risks to agile adoption

During the initiation stage of IDEAL model, an organization should asses several characteristics and capabilities of the organization which may act as barriers to SAFe adoption. Organizations must assess their readiness for implementation of SAFe before practically implementing it. It is clearly ineffective to waste time, effort and resources in transitioning an enterprise to SAFe if the enterprise is not ready for such endeavor [28]. Furthermore, the success of deploying agile practices in the context of process improvement should occur only when there is commitment to change by the organization on overall organizational and project level. As mentioned previously, the initiation stage is important since if it is not conducted effectively it could lead to greater disruptions later on while implementing SAFe practices. Sidky claimed that organizations which are not ready for agile adoption can experience technical chaos in which the partial adoption of agile can leave the organization in unstable state until it reverts back to the previous software

development methodology [28]. Therefore, the initiation phase of IDEAL and assessing barriers and risks is an important part of SAFe adoption.

Several studies have already investigated the challenges of migrating to agile methodologies. The Scaled Agile Framework in essence aggregates multiple agile software development practices which are scaled into one consistent framework. Therefore the challenges to adopting general agile methodologies are comparable to the challenges of adopting SAFe. The focus on this section would be to stipulate the risks and challenges for SAFe adoption. Based on the current risks of agile adaption, as evidenced in research and practice, several aspects are presented which should be assessed during imitation phase of the IDEAL cycle. It is important to mention that the barriers to SAFe adoption are only acknowledged. No attempt is made to propose ways of assessing these barriers since this is out of scope of this research.

Based on extensive review of literature [4], [23], [24], [31], [32], [33], [34], [35] several main aspects were identified which have major impact on agile adoption. The aim of this section is to outline and generalize the risks of agile adoption.

People: As written in the agile manifesto, the main agile values consist of *Individuals and Interactions* and C*ustomer Collaboration* which place high emphasis on people in the organization [3]. People issues are at the heart of the agile movement, and much of the paradigm change has to do with empowering individuals by supporting ownership, shorter feedback cycles, reasonable goals and flexibility [24]. Furthermore, the skillset and quality of people become a primary concern for the would-be agile team [31]. The main people related issues which may prevent organizations from successfully adopting SAFe are: [23]

- Ability of people to work effectively in teams
- Lack of desired skills and low level of competence
- Closeness of customer relationship

Process: Since agile way of working introduces a new way of working, new practices and values, this has an impact on many of the old organizational processes. Therefore, old outdated processes have to be replaced with new agile ones [32]. The new agile processes place less emphasis on formal planning and documentation [3] and more emphasis on the tolerance to change and iterative development. Such a change entails major alterations to processes and work procedures [23]. These alterations are the following: [23]

- Change from process-centric to people-centric approach
- Managing large, scalable projects
- Selecting appropriate agile practices
- Short, iterative and adaptive development

Technology: Organizations which intend to adopt SAFe practices should invest in the technological infrastructure that support and facilitate rapid software development. If the

existing technology within an organization is not appropriate for these developments, technology can act as a roadblock for SAFe adoption. There are several existing tools on the market that support SAFe practices i.e Agilefant, codeBeamer, Rally and others. However, tools alone cannot make software development successful. People must be trained on how to use them correctly [23].Therefore the following technological issues have to be addressed before adopting SAFe: [23]

- Appropriateness of existing technology and tools
- New skill sets- e.g. refactoring, configuration management

Culture: When transitioning to agility and SAFe, some of the current cultural values within the organization might conflict with the newly introduced agile values. Since agile is mostly about the mindset of people in an organization, and the collective mindsets shapes the organizational culture it is of vital importance that the organizational culture is aligned with agile values. Several studies have already highlighted the importance of compatibility between the organizational culture and the software development methodology [33], [34], [35]. Furthermore, organization culture cannot easily be changed, and this may be a multi-year effort. The following cultural and managerial aspects require special attention during the initiation phase of the IDEAL model: [23]

- Leadership and collaborative management style
- Organizational Form
- Reward System and Incentive structures
- Organizational culture

Dean Leffingwell also acknowledges that there are significant challenges to adopting and scaling agility to an enterprise level [4]. He categorizes these challenges into two classes; the first is due to the challenges inherit in agile itself and the second is due to the challenges imposed by the enterprise. The challenges imposed by the agile method arise due to the fixed rules and assumptions of agile which often contradict the typical enterprise environment. These limits include: [4]

- Close customer involvement
- Small team size
- Collocation
- Emerging architecture
- Lack of requirements analysis and documentation
- Culture
- Physical environment

The challenges imposed by enterprise arise due to the fact that, as Dean Leffingwell argues, enterprises are living systems and as all living systems they have learned to defend themselves and know how to maintain status quo. The adoption of SAFe challenges the status quo which causes resistance to change. This resistance to change can come due to several aspects namely:[4]

- Formalized policies and procedures
- Corporate culture
- High degree of distribution
- Organization around functions rather than product line
- Friction between departments and teams
- Fixed functionality mandates.

4.2. SAFe maturity model

In practice, organizations are unable to fully adopt agile development practices immediately or over a short period of time [38]. The difficulty of adopting agile practices increases when there is a need to scale those practices. As organizations transform and adopt agile and SAFe they require a uniform model for measuring maturity. In this section a maturity model is proposed for the purpose of identifying the current state and determining the maturity of an organization in terms of adopting the Scaled Agile Framework. A maturity model is a conceptual framework, with constituent parts, which defines maturity in the area of interest – in this case adoption of SAFe [57]. The proposed maturity model can also be used as roadmap for SAFe adoption for enterprises which have not adopted agile way of working and intend to adopt SAFe in the future.

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 [53]. In general maturity models are characterized by a limited and ordered number of maturity levels (usually four to six), and each maturity level defines characteristics or practices which have to be achieved at each level.

In a systematic review of literature, Schweigert et al. found approximately forty currently available agile maturity models [42]. Based on their analysis Schweigert et al. conclude that even though there is a need for agile maturity models, currently none of the proposed models is widely accepted. In another case study assessment of agile maturity models, nine agile maturity models were compared in terms of fitness of purpose, completeness, definition of agile levels, objectivity, correctness and consistency [36]. The assessment of this case study found that the Agile Adoption Framework [AAF] of Sidky has obtained the best assessment results. The well-

defined structure of AAF, based on the assessment criteria, can be extended to cover agile best practices [36]. In this thesis the structural foundation of the maturity model of Sidky (SAMI) will be used and extended for scaling agile practices for the SAFe.

4.2.1. Sidky agile measurement index (SAMI)

The Agile Adoption Framework (AAF) of Sidky uses a maturity model of agile practices called Sidky Agile Measurement Index (SAMI) to measure the agile potential of an organization. In this section, SAMI is briefly explained to get a better understanding of the structure behind the proposed SAFe maturity model. The measurement index of Sidky is composed of four main components: agile levels, agile principles, agile practices and concepts, and indicators [Figure 7].

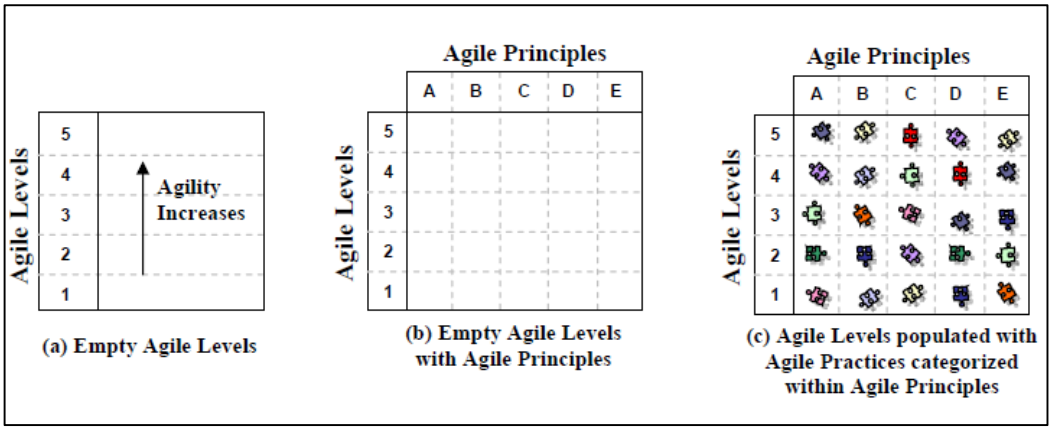


Figure 7.Components in SAMI (adapted from [28]) (Indicators not shown)

Agile Levels:

Most maturity models contain levels of maturity which enumerate the level of maturity in specific domain. In SAMI, the agile levels measure and enumerate the agile maturity level of an organization. These levels are not associated with any particular agile method e.g. Scrum, XP, Pair Programming but rather are based on the core values and qualities of agility. Sidky derived the agile levels from the four values and twelve principles of the Agile Manifesto. The identified five levels of agility, which are based on agile values and qualities, are shown in Table 2. Collaboration is one of the essential value and quality of agile and thus it is enumerated as Level 1. Developing software through an evolutionary approach, as shown at Level 2, is the second most important value of agility. Being effective, adaptive and establishing all-encompassing environments incrementally increase the agile maturity of organizations as well. The development of these maturity levels prioritizes the improvement actions for increasing agile maturity. These five levels of agility have been found to be comprehensive enough and therefore they will also be used in the SAFe maturity model.

Table 2. Agile levels in SAMI (adapted from [28])

Agile Level	Level Name	Level's Objective (Agile Value Reworded)
Level 5	Encompassing	Establishing a vibrant and all-encompassing environment to sustain agility.
Level 4	Adaptive	Responding to change through multiple levels of feedback
Level 3	Effective	Developing high quality, working software in an efficient and effective manner.
Level 2	Evolutionary	Developing software early and continuously.
Level 1	Collaborative	Enhancing communication and collaboration.

Agile Principles:

The agile manifesto describes twelve agile principles [Appendix B].These agile principles can be used as a way to differentiate between dissimilar agile aspects in order to ensure that an organization does not only adopt only one or few of these principles. Since twelve principles are far too many, Sidky grouped and summarized these twelve agile principles into five principles in order to avoid unnecessary complications. The five comprehensive principles he identified and which are used in SAFe Maturity Model are: [28]

- Embracing change to deliver customer value
- Plan and deliver software frequently
- Human-centric
- Technical excellence
- Customer collaboration

As described in Chapter 2, SAFe also incorporates principles from Product Development Flow (PDF) and lean. However, since PDF and lean principles are instantiated in various agile practices of SAFe for simplicity reasons these principles won't be enumerated in the SAFe maturity model. The five comprehensive principles shown above were also used in the SAFe maturity model.

Agile Practices:

Agile Practices are a set of techniques or methods which are used for developing software in a manner which is consistent with the agile principles [28]. In the previous two sections agile levels and principles were introduced. The agile practices and concepts are categorized under the agile principle they belong to and the agile level which they help to achieve. Each agile level contains practices which when adopted collectively leads to significant improvements in agility. Organization should adopt agile practices on lower levels first, since the agile practices on higher level are dependent on the practices introduced at lower level. When organizations adopt

more agile practices they in fact increase their agile maturity. Also, when organizations adopt these practices they realize the value of each level. For instance, if an organization adopts all the practices from level 1 it will be considered to be a collaborative organization.

Agile Indicators:

Indicators are sets of questions which are used to assess a specific set of characteristics of an organization or project, in order to determine the current level of agility in an organization. SAMI contains over 300 indicators for each agile practice. In SAMI indicators are also used to identify the readiness of an organization to adopt a specific agile practice. In the SAFe maturity model however, the readiness of organization should be assessed before overall adoption of SAFe and not for each practice individually. The reasoning for this simplification lies in the fact that there is major overlapping of the factors or barriers which prevent practices to be implemented. Therefore the focus of indicators in the SAFe maturity model will be only on existence of the listed practices and concepts.

In conclusion, the Sidky Agile Measurement Index (SAMI) is the most comprehensive and generalizable maturity model which can be extended with the addition of other scalable SAFe practices. This Measurement index, together with the populated agile practices is shown in Appendix C.

4.2.2. Developing the SAFe maturity model

The SAFe Maturity Model is constructed based on the structure of SAMI. Thus, the five levels of maturity of SAMI and five comprehensive agile principles are being re-used for mapping SAFe practices at each maturity level and in according to agile principles. In addition to the existing agile practices already present in SAMI [Appendix C], several other SAFe practices are identified. These practices originated from the two books of Dean Leffingwell in which he introduced SAFe practices for scaling agility. These books are: *Scaling Software Agility: Best Practices for Large Enterprises* (2007) and *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (*2010). These identified practices from SAFe which natively scale to enterprise level are:

- Acceptance testing [2]
- Agile Estimating and Velocity[2]
- Agile Release Train [2]
- Changing the organizations [4]
- Concurrent testing [4]
- Continuous integration [4]

- Impact on customers and operations [4]
- Intentional architecture [4]
- Iterating [2]
- Lean requirements at scale [4]
- Managing highly distributed teams [4]
- Mastering the iteration [4]
- Measuring business performance [4]
- Product Backlog[2]
- Regular reflection and adaptation [4]
- Release Planning[2]
- Requirements Discovery [2]
- Roadmap [2]
- Smaller, more frequent releases [4]
- System of Systems [4] [Appendix F]
- Software Kanban Systems [2]
- The define/build/test team [4][Appendix F]
- Two-level planning and tracking [4]
- User stories [2]
- Vision, features [2]

Description of these SAFe practices, as well as the agile practices from SAMI is provided in Appendix D. In addition to the practices listed above, additional practice is introduced at level 5 called *Continuous SAFe Capability Improvement*. Although this is not a recognized and documented SAFe practice, the aim of this practice is to provide a foundation for enterprises to strive for continuously improving the capabilities of SAFe. As stated previously, SAFe is a product of continuous work and therefore an organization which is at the highest level of maturity should be striving to contribute and improve the practices, artifacts of the SAFe itself. Moreover, similar practices which are focused on overall capability improvements are present in several other maturity models such as P-CMM, ISO/IEC 15504 [61], [62].

All of these practices are populated in the SAFe Maturity Model based on several guiding rules. The first rule of mapping SAFe practices states that each practice must realize the objective of the agile level. For example, having *Collaborative Planning* at level 1 helps the collaboration goal of level 1. The second rule states that each practice is listed under the agile principle which is related to. For example, *Collaborative Planning* is related to the principle for *Plan and Deliver Software Frequently*. Last, but also a very important rule is that the practice(s) on higher agile levels depends on agile practice(s) on lower levels. For instance, the SAFe practice of *Release Planning* introduced at level 2 is dependent on having *Collaborative Teams* and

Collaborative Planning both of which are at level 1. *Test Driven Development* at level 5 depends on the presence of *Unit Tests* at level 3. Having *Self Organizing Teams* at level 3 depends on the having *Empowered and Motivated Teams* at level 1. There are many other similar dependencies between practices in the maturity model and therefore it is recommended that enterprises first adopt practices on a lower level of maturity before proceeding to adopt practices on a higher level.

In addition to the introduced SAFe practices, two of the practices identified in SAMI i.e. *Paired Programming* and *Agile Documentation* were removed due to inconsistency with SAFe practices. These two practices are not used in SAFe, and cannot be used to measure SAFe maturity. Two practices are also moved to different level of maturity because they are considered more rudimentary. One of those was the concept of *User Stories which* in SAMI was introduced at Level 4, but several practices at lower level rely on User Story development e.g. *Adaptive Planning* and *Planning at Different Levels*. Thus the practice of *User Stories* has been moved to Level 1. Cohn also suggests that *User Stories* are introduced at first level of agility because, based on his experience, they enhance collaboration and communication between the stakeholders in regards to requirements [28]. Product backlog which is introduced on Level 3 in SAMI is considered more rudimentary practice in SAFe and thus moved to Level 2. SAFe makes use of product backlogs on all team, program and portfolio level and thus has defined team, program and portfolio backlogs [2]. The presence of product backlogs in SAFe also fulfills the goal of Level 2 of early and continuously delivery of software by defining the epics, features and task in the corresponding backlogs.

In order to complete the adjustments to SAMI, two practices were renamed based on the naming convention used in SAFe. The SAMI practice called *Planning at Different Levels* was renamed to *Two Level Planning and Tracking* based on the naming convention used in Dean Dean Leffingwell book [4]. Also, the concept of backlog which in SAMI is referred to as *Maintain a List of All Features and Their Status* is simplified to the concept of *Product Backlog*[2]. Although these concepts refer to the same practices, it was more appropriate to use the naming convention used in SAFe. All these alterations conclude the initial revisions to SAMI.

In a more mature domain the identification of components and practices which are to be used in a maturity model can be established only through an extensive literature review. However, in a relatively new domain such as scaling agile practices at enterprise level it was not feasible to gather adequate and complete knowledge through review of literature alone [54]. In this research effort literature can only be used as a theoretical starting point for identification of the components and practices. In order to obtain a more complete, practical and useful maturity model it is recommended to use more appropriate exploratory research methods such as Delphi method, interviews, or focus groups. For reasons elaborated in next section, Delphi method was most suitable approach for this research which consisted of eliciting expert opinions through two rounds of feedback.

4.2.1. Development of SAFe maturity model with Delphi study

The initial maturity model for SAFe was constructed through a synthesis of various concepts, best practices and broad literature review. However, to increase the cogency and applicability of the proposed model a Delphi study was used. Delphi studies are characterized by several features which differentiates them from similar group interrogative methods. These three characteristics are: multiple iterations of group responses with interspersed feedback; anonymous group interaction; the presentation of statistical analysis. The main advantages of this method are the fact that it uses group decision-making techniques while involving experts from the field which increases the validity of the research. Furthermore, the anonymity of the participants resolves the difficulties commonly associated with group interviews e.g. deference of authority, impact of oral facility. However, there are several disadvantages of Delphi studies, namely: the length of the process, the researcher influence on the responses due to a specific way of formulation questions [48]. The Delphi study consisted of two rounds, with a timespan between the two rounds of about thirty days. Seven SAFe and agile experts participated in both the first and second round of the Delphi study.

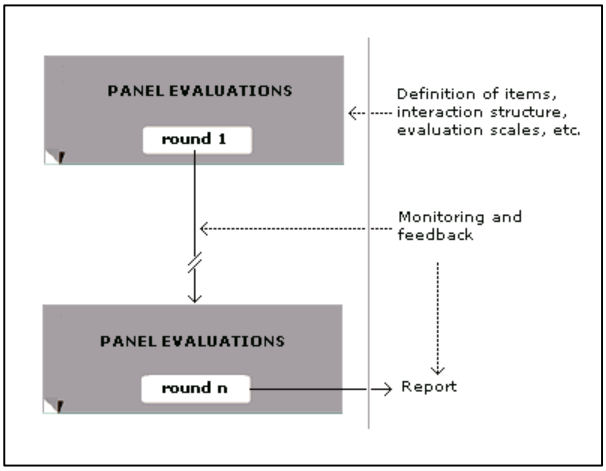


Figure 8. Delphi method structure

4.2.2. Panel of experts

An important aspect of the Delphi study is the selection of the panel of experts. Typically the panel of experts consists of academic and industrial experts in order to balance the views from both theoretical and practical perspectives. In this research the panel of experts (N=7) was internally selected and comprised of industry experts from a large multinational company. The panel of experts consisted of agile coaches with multi-year experience in practical implementation of agile practices. 85 % of the experts were also SAFe Program Consultants (SPC). SPC are internal change agents in the enterprise which have domain knowledge on how to implement SAFe and are qualified to launch Agile Release Trains. They are also certified experts on SAFe who can train management and practitioners in the organization on SAFe

practices and artefacts. Due to the homogeneity and internal nature of the group two rounds were deemed appropriate to reach consensus. The level of experience and expertise of the panel of experts is displayed in Figure 9.

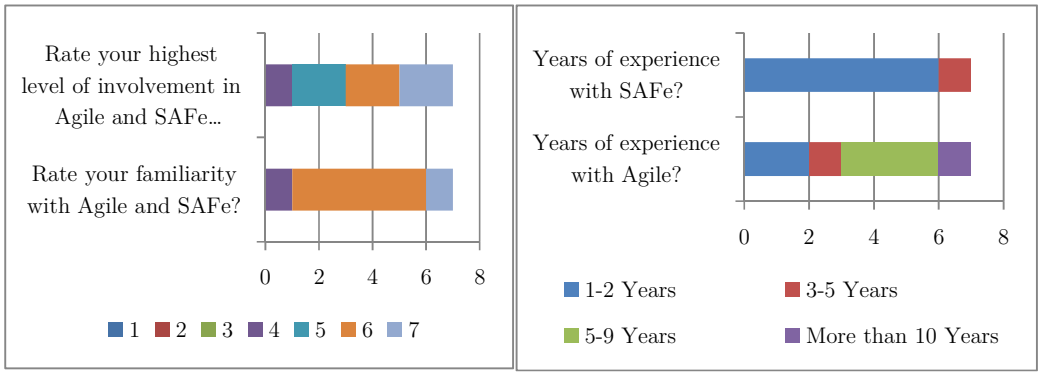


Figure 9.Characteristics of the panel of experts

Round 1

The aim of the first round of the Delphi study was to elicit broad comments from the panel of experts. The implementation strategy together with the proposed maturity model was presented and a draft report was sent to the participants of the panel. An online questionnaire was also used to gather responses to the different type of questions. Likert 5 point type questions were used to determine whether there is a need for a gradual adoption approach for SAFe; the IDEAL model is useful for guiding the adoption and if barriers/risk of the adoption need to be assessed. Also open-ended type questions ware used to check whether the five agile levels are sufficient and if the SAFe and agile practices are complete and aligned appropriately. In cases when experts disagreed with the proposed implementation strategy or maturity model, they were prompted to give their response based on their practical experience and expertise. The full questionnaire together with the feedback and results obtained from the first round is shown in Appendix F.

Round 2

The primary benefits 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 previous round and reach consensus. Also, since 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 presented with results from the first round and requested to reach agreement on the proposed improvements. The improvement revisions were grouped into three distinctive types: fundamental, additive and corrective revisions. Fundamental revisions addressed the fundamental changes proposed by the experts and further clarifying the existing concepts. The fundamental revisions done to the model only included expending the description of the agile and SAFe practices. Additive revisions covered the additions of SAFe practices

which were proposed during Round 1. The practices which were added were the following: *Automated Testing*, *PSI/Release*, *DevOps*, *Scrum of Scrum* and *Continuous SAFe Capability Improvement* [Appendix F]. The corrective revisions included the corrections of the existing elements which mainly focused on aligning practices under the correct maturity levels and principles. The corrected practices were the following: *User Stories; Agile Release Train; Vision, features; Intentional Architecture; Impact on Customers and Operations; Measuring Business Performance* and removing *Systems of Systems*. At the end of the second round, the panel of experts was asked to reach consensus on these revisions and also provide a limited evaluation of the maturity model. The Questionnaire from the second round together with the obtained results can be found in Appendix F.

4.2.3. Results

After the second round of the Delphi study, the panel of experts was asked to evaluate the necessity, practicality, completeness and understandability of the maturity model and implementation strategy [Figure 10]. Most of the experts agree that the model is: practical and can be used in the industry; complete since it contains all the necessary components; has high understandability and is easy to understand. However, the experts were reserved regarding the necessity of the model since five out of the seven experts stated that they neither agree nor disagree that the maturity model is beneficial and necessary for the industry. Although four of the experts agree with the completeness of the implementation strategy, one expert disagreed stating that the description of SAFe and agile practices in the maturity model are not complete, which could reduce the usability as well [Appendix F]. After the completion of the Delphi study the descriptions of practices were further expended and the sources of information were properly cited to address this remark [Appendix D].

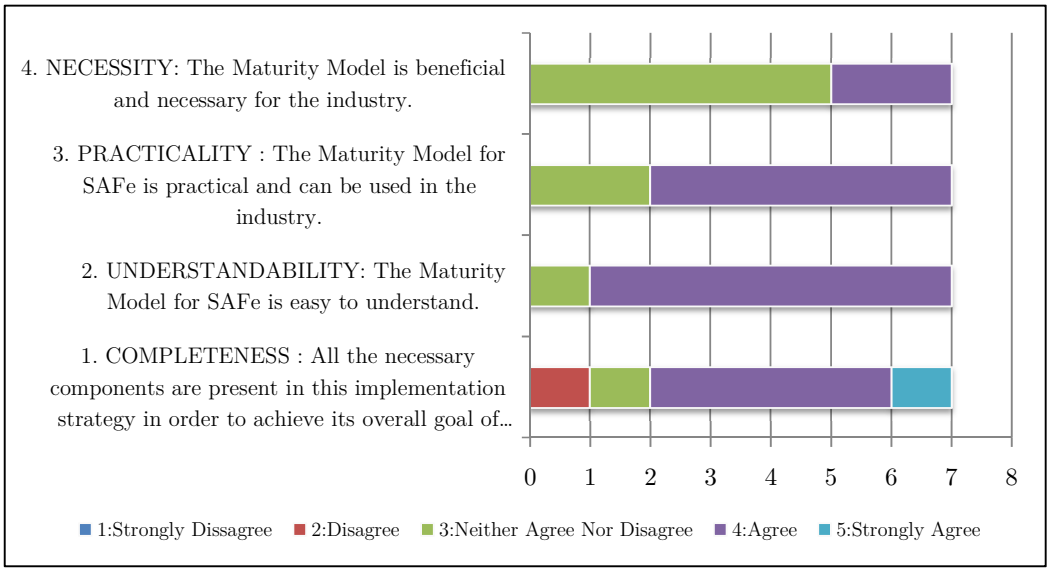


Figure 10. Results from evaluation of Maturity Model

The final SAFe Maturity Model obtained after the validation through the Delphi study is shown in Table 3. The practices from SAMI that remained unchanged in the SAFe maturity model are displayed in black color. SAMI practices that were changed in the current model are displayed in red color. Newly added SAFe practices which scale to enterprise level and were introduced and partially validated with the Delphi study are shown in blue color.

Although the rationale for populating the SAFe and agile practices in their appropriate levels is explained above, with time and experience the positioning of the practices might change. The focus of the SAFe Maturity Model is mainly on providing a structure for guiding the SAFe adoption. Sidky had similar claims about the tailorability of the SAMI. He states that the science on agile measurement indexes is still in its infancy stage and more refinement is needed to establish accurate relations between the different instances in SAMI [28]. Furthermore, since there are constant changes and improvements in the SAFe Big Picture Framework, the SAFe maturity model also needs to constantly evolve and adapt to the changes made in SAFe.

Since the SAFe Maturity Model describes the essential practices that organizations should possess at a particular level of maturity, the SAF Maturity model can be classified as a descriptive model. The full descriptive model, updated and validated through a Delphi study is shown in the Table 3 below. The description of the practices listed in SAFe Maturity Model is shown in Appendix D, while the indicators which can be used for assessment of the level of maturity are shown in Appendix E.

Table 3. SAFe maturity model

	Embrace change to deliver customer value	Plan and deliver Software Frequently	Human-centric	Technical Excellence	Customer Collaboration
Level 5 Encompassing	-Low process ceremony -Continuous SAFe Capability Improvement	-Agile project estimation	-Ideal Agile physical setup -Changing the organization	-Test driven development -No/minimal number of level -1 or 1b people on team -Concurrent testing	-Frequent face-to-face interaction between developers and users (collocated)
Level 4 Adaptive	-Client driven iterations -Continuous customer satisfaction -Lean requirements at scale	-Smaller and more frequent releases -Adaptive planning -Measuring business performance	-Managing highly distributed teams	-Daily progress tracking meetings -Intentional architecture	-Customer immediately accessible -Customer contract revolves around commitment of collaboration
Level 3 Effective	-Regular reflection and adaptation	-Risk driven iterations -Plan features not tasks -Roadmap -Mastering the iteration -Software Kanban Systems -PSI/Release -Agile Release Train	-Self-organizing teams -Frequent face to face communication -Scrum of Scrum	-Continuous integration -Continuous improvement (refactoring) -Unit tests -30% of level 2 and level 3 people	-DevOps -Vision, features -Impact on customers and operations
Level 2 Evolutionary	-Evolutionary requirements -Smaller, more frequent releases -Requirements discovery	-Continuous delivery -Two-level planning and tracking -Agile estimating and velocity -Release planning	-The define/build/test team	-Software configuration management -Automated testing -Tracking iteration progress -No big design up front(BDUF) -Product Backlog	-Customer contract reflective of evolutionary development
Level 1 Collaborative	-Reflect and tune process	-Collaborative planning	-Empowered and motivated teams -Collaborative teams	-Coding standards -Knowledge sharing -Task volunteering -Acceptance testing	-Customer commitment to work with development team -User stories

Chapter 5

Case study

In the previous chapter of this report a new implementation strategy was proposed as a solution to the most imperative challenge of adopting agility at scale. In this chapter the focus shifts to investigating the potential benefits of adopting SAFe and its practices. The benefits will be examined and empirically measured through a case study in a large international company which is transitioning towards an agile way of working. An empirical case study is conducted and quantitative approach is employed to measure and analyze the benefits of adopting SAFe practices. The SAFe maturity model which was developed in the previous section was also applied to measure current maturity in the case organization.

The empirical study is done mainly through comparison of waterfall and agile development, and partially by examining only agile (SAFe) features. In Section 2.3 of this report, the main differences between these two methodologies were shown. The traditional waterfall approach assumes that "systems are fully specifiable, predictable, and are built through meticulous and extensive planning". In Agile software development on the other hand "high quality software is developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change" [44]. Since these two approaches of software development rely on dissimilar underlying principles, it can be expected that they tend to provide different benefits and performance in organizations. The case study was conducted in an organization which currently employs both software methodologies but is also attempting to uphold agile values and SAFe. A brief description of the organization of interest i.e. Royal Philips is provided below.

5.1. Introduction of case company (Royal Philips)

5.1.1. Brief description

Royal Philips, commonly known as Philips, is a diversified health and well-being company focused on improving people's lives through meaningful innovation in the areas of Healthcare, Consumer Lifestyle and Lighting. Through its three divisions, the company provides multitude of diversified products, ranging from CT scanners and X-Ray equipment to shavers, lamps and similar other domestic appliances. The company is headquartered in Amsterdam and currently employs more the 115,000 employees in more than 100 countries. In 2013, Philips reported revenue of EUR 23.3 billion [6]. As one of the largest electronics companies in the world, Philips manages a dynamic portfolio of more than 40 businesses with over 59,000 patents.

5.1.2. Transformation of Philips

In 1996, Peter Drucker suggested that "Big companies have no future… By and large there are no more advantages to big business. There are only disadvantages" [17]. Economy of scale, brand recognition and greater human resources are not sufficient benefits of large enterprises to diminish the increased importance of reducing time-to-market, production times, and innovation cycles. The bureaucratic hierarchical structure of big enterprises in a sense prevents them to be competitive in today's fast-paced environment. Moreover large organizations typically operate in command and control mode, where decisions are being made on senior level and those decisions are being executed by lower level employees. In a fast changing environment, where requirements change frequently, getting approval of all this changes from senior level can cause large delays. As an example of this type of large and hierarchical organization, Philips(in 2011) when starting new projects required 72 signatures which led to average project initiation time of 6 months [41].

Since 2011, under a new leadership lead by a new CEO Frans van Houten, Philips experienced several transformations. The *Accelerate!* Program, launched in 2011 aimed at redefining the way Philips conducts business, making fundamental changes to its products and processes with the intention of making Philips a high-performing company. The aim of *Accelerate!* was also to create an agile and entrepreneurial Philips [41].In order to support the business transformation Philips also needed radical changes in its IT infrastructure. Furthermore, Philips had to reduce the operating costs of IT which was above that of its competitors. Several IT initiatives were launched in order to support, harmonize and standardize the new way of doing business. The Philips Integrated Landscape (PIL), as one of those initiatives, was put in place in order to streamline IT operations by reusing common components. The PIL initiative also introduced new practices in the organization which laid the foundation for the transition of Philips to Agile way of working. The company introduced multi-disciplinary teams (MDT's) which adopted incremental and iterative approach to software development instead of the traditional waterfall model. Furthermore, Philips created output-based partnerships with several output-based partners which were responsible for the software development process. This led to the creation of over 100 scrum teams which consisted of Philips employees (Business Analysts and Tech Leads) and output-based partners (Developers and Testers).

Since 2011, Philips has made a significant progress in its transformation. For instance, if in 2011 Philips required 72 signatures to start projects, in 2013 only 4 signatures were required [41]. This decreased the project initiation time from 6 months to 40 days in 2013 [41]. There are also significant decreases in IT operational costs. It is evident that the transformation to a more agile, lean company has provided abundant benefits so far. The question which remains to be answered is how successful will Philips be in scaling the benefits upon fully adopting SAFe.

5.1.3. Adoption of SAFe at Philips

Philips IT started adopting SAFe practices towards the end of 2013. Previous to deciding to adopt SAFe, Philips experienced changes in its operational model which now required for projects to be developed in a new agile way. By 2013, due to these changes, Philips was running 150 multi-disciplinary Scrum teams which formed the basis for SAFe adoption [41]. The majority of these Scrum teams were geographically dispersed and contracted from output-based partners. In an attempt to better organize and streamline these Scrum teams, Philips decided to adopt SAFe. Initially, Program and Delivery Managers took the initiative to implement varying aspects of SAFe without the benefit of having formal training on the framework. In February of 2014, with the formation of AWG, Philips started formal training of SPC and SA consultants. Currently the AWG has several internal and external agile and SAFe consultants which are implementing SAFe using the current available strategy [Figure 5].

Due to the ad-hoc wholesale adoption, different programs and teams within Philips have varied level of SAFe maturity. For the purpose of assessing the current SAFe maturity of Philips, an assessment was conducted in one program. The program was randomly selected and served as a representative random sample of Philips. The assessment was done by RTE and a Scrum Master. The RTE has knowledge on the existing SAFe practices on program and portfolio level, while the Scrum Master has full knowledge on the practices employed within teams.

The results showed that 13 % of SAFe practices are fully achieved, mostly on lower levels of maturity. Furthermore, 28 % of the practices are largely achieved in the assessed program. The largest percentage was in the practices which were neutrally adopted at 47 %. This refers to practices that are neither largely achieved nor not achieved, but rather occur on some level within the program. 12 % of the practices in the SAFe maturity model were not achieved and 0% of the practices were fully not achieved. More detailed results from the assessments are displayed in Figure 11.

In reflection to the performed assessment, we have to acknowledge that although 41 % of SAFe practices are fully or largely adopted, there is still a lot of partially existing practices within the program which are not utilized properly. This could be attributed to the wholesale adoption strategy which is used in Philips which does not provide proper guidance on how to deploy these practices. Also, there could be some barriers which were discussed in Section 4.1.2 which are preventing the further adoption of these practices. Only 12% of the SAFe practices were not adopted. Even though the adoption of SAFe is not full in the case organization, it is important to measure the effects of the adoption which is discussed in the rest of this thesis report.

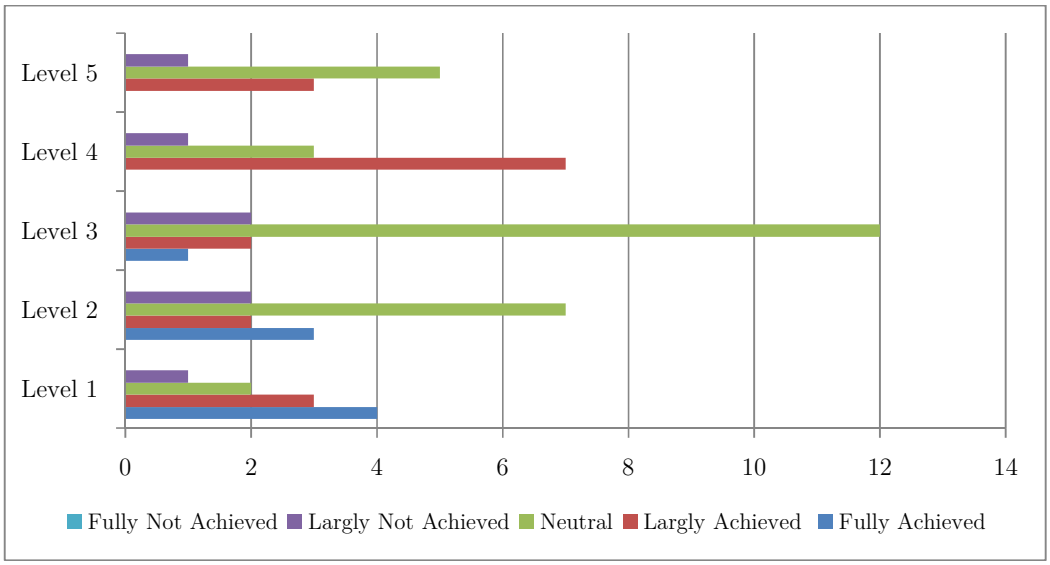


Figure 11. Assessment of SAFe maturity at Philips

5.2. Related case studies

Since the creation of the SAFe in 2011, several companies have applied the framework within their own organizations. Few case studies have documented and reported their experiences on SAFe adoption. Among the names of the companies which have adopted the framework are: Nokia, TradeStation, John Deere, Telstra, BMC software, Infogain Corporation. The majority of these case studies are self-reported which raise question about their validity and accuracy. Nonetheless, in this section few case studies are deliberated.

A well-documented agile transformation [5], [58] at BMC Software gives insight into several improvements in key metrics such as: higher ROI, better alignment with customer needs, increase in productivity of 20 – 50 %, delivering high quality releases in a fraction of the time. This was achieved after twelve month transformation which involved about 300 people. The study also outlines the challenges such as staying releasable throughout the development cycle due to late discovery of defects; defining the right level of requirement detail at the right time during the life cycle [58].

Another case study at Infogain Corporation examined the benefits of having distributed release planning event [7]. The study claims that a proper preparation, orchestration and facilitation of distributed program events are essential for successful release planning. Among the benefits of release planning they list the following: alignment between all stakeholders of the vision and plan for the release; improvement in communication with peer client teams, technical experts and other program stakeholders; clear commitment to realistic plan with estimated dependencies and risks. However, the study concludes with confirmation that geographically distributed teams experience lower productivity due to lack of alignment and solid program execution [7].

Self-reported case study at John Deere claims that SAFe provided manifold improvement, namely: 20% faster time to market, 20% improvement in time to production, 50% decrease in

warranty expenses and increase in employee engagement of 9.8% [5]. Mitchell International experienced 44% decrease in post release defects and 76% decrease in response time to customers [5]. While using an earlier variant of SAFe, Discount Tire increased productivity by 20-25% and experienced dramatic improvements in time-to-market at quality. Telstra was successful at implementing Program-level SAFe theory in to practice and as result experienced 95% decrease in project defects, 50% reduction in delivery cost and average delivery cycle from twelve months to three months. There are several other studies which indicate that SAFe provides practical benefits to organizations which adopt them. However, since most of these studies are self-reported there is a need for more independent assessment of SAFe. This Master Thesis provides objective and unbiased evaluation of a framework which promises abundant benefits.

5.3. Time-to-market

CONFIDENTIAL
CONFIDENTIAL
CONFIDENTIAL

Figure 12. Scatterplot of time to market projects (in days)

Table 4. Time to market statistics in days

CONFIDENTIAL
CONFIDENTIAL
CONFIDENTIAL

Figure 13. Scatterplot of delayed projects (in days)

Table 5. Delayed project statistics (in days)

5.4. Predictability

CONFIDENTIAL
CONFIDENTIAL
CONFIDENTIAL

5.4.1. Velocity reliability

Figure 14.Average velocity of team/sprint

Figure 15.Velocity of teams/sprint

5.4.2. Cost predictability

CONFIDENTIAL

CONFIDENTIAL
CONFIDENTIAL
CONFIDENTIAL

Figure 16. Box plot representation of cost predictability

Table 6. Cost predictability statistics (in EAC/planned cost)

5.5. Customer satisfaction

Figure 17. Bar chart customer satisfaction

Figure 18. Pie chart customer satisfaction

5.6. Quality

CONFIDENTIAL
CONFIDENTIAL

Figure 19. Scatterplot of defect density

Table 7. Defect rate statistics

CONFIDENTIAL
CONFIDENTIAL
CONFIDENTIAL

Chapter 6

Conclusions

6.1. Research conclusions

The current exploratory study was driven by two main motivations. The first motivation came from the case company which intended to develop an implementation roadmap for SAFe adoption. The current wholesale adoption approach of the Scaled Agile Framework, based on the review of available literature, is also deemed too risky and complex in order to be successfully applied in practice. The second motivation was to increase the body of knowledge on SAFe and the effects of its adoption. Currently there are very scarce academic publications on this topic, mainly due to the novelty of the framework. Given the fact that SAFe is rapidly emerging as the industry standard for scaling agile practices, there was a need for an academic evaluation of the effects of SAFe adoption.

The two main contributions of this research effort are both practical and scientific in nature, and are directly associated with these two motivations. The main purpose of the master thesis was to answer the two research questions.

RQ1: How can the Scaled Agile Framework be implemented in larger enterprises?

The first research question was answered in Chapter 4 by developing an incremental implementation strategy for the Scaled Agile Framework. This implementation strategy was developed based on existing SPI and maturity model. The IDEAL model is to be used as a foundation for the software process improvement efforts. In attempt to achieve lasting process improvement, the SAFe maturity model was developed to serve as an evolutionary path which increases organization's SAFe maturity in stages. The implementation strategy and SAFe maturity model were developed through a Delphi Study with participation of seven agile/SAFe experts and two rounds of feedback. Most of the experts agreed that the proposed implementation strategy and maturity model is complete, understandable and practical. The application of SAFe and the Delphi study provided limited validation to the proposed strategy.

RQ2: What are the effects of adopting SAFe in terms of time-to-market, predictability, customer satisfaction and quality?

The second research question was answered in Chapter 5 through a case study analysis in an organization which is adopting SAFe. Based on the empirical evidence, there were found several effects caused by SAFe adoption. These effects can be summarized as follows:

- Average reduction time in time-to-market from waterfall to SAFe of 52.59%.
- Lower time delay of delayed projects in SAFe compared to waterfall projects.
- In SAFe, teams on average deliver 94.31% of planned functionality per sprint. However, per sprint there is also high rate of dispersion around the average.
- There is higher cost predictability for SAFe project than waterfall projects.
- Net Promoter Score in SAFe for customer satisfaction is 30.
- Average defect rate for SAFe projects is 0.0299.

6.2. Validity and reliability

The first research question was developed and partially validated through a Delphi study and therefore the reliability and validity comes from combining the opinions of the experts. The rounds of feedback were also conducted anonymously which allowed for the Delphi participants to rethink their initial opinions without the influence of the other participants. However, the anonymity also can lead to exclusion of group interaction which in some cases may reduce the accuracy of group judgments [59]. During the two rounds of the Delphi study, two questionnaires were used which consisted of mostly open-ended questions. Here the reliability of the open-ended responses depends on whether the Delphi participants interpret and respond to the questions in the same way as intended by the researcher. In order to increase the reliability of the responses, the Delphi participants were informed that all their answers will be kept strictly confidential. Furthermore, to make questions clearer and easy to answer, some questions were clarified with additional description and further explanations.

The second research question was answered based on current available quantitative metrics from the case organization. Internal validity and reliability is assured by the fact that the data was collected from internal metrics that objectively reflect reality. However, there is lower external validity of the empirical study and therefore these results cannot be generalized to other enterprises. The lower generalizability effect is often associated with case study research.

To ensure higher reliability and validity of the entire research, a triangulation strategy was used which typically is used for improving the validity and reliability of research or evaluation of findings [60].

6.3. Limitations

One limitation in respect to the newly proposed implementation approach is the fact that there is certain reluctance in the industry to use plan-driven SPI models for the purpose of agile adoption. The reason lies in the view that plan-driven improvement approach is seen as

incompatible with the end-state agile development method since agile and SAFe are more about embracing change than following a plan-driven implementation or improvement strategy. This might be the reason why the majority of the experts neither agreed nor disagreed with the necessity of the SAFe maturity model [Figure 10]. This is acknowledged in the proposed framework to some extent since the existing IDEAL model is modified to fit the agile adoption efforts. Also, the IDEAL model is extended and made applicable not only for software process improvement, but to any kind of improvement effort [29].

Another limitation arises from the fact that the current SAFe maturity in the case organization may vary. Due to the current wholesale implementation strategy there is inconsistent maturity of SAFe practices throughout programs and teams. The random sample assessment based on the developed SAFe maturity model revealed that 41 % of the SAFe practices are fully or largely adopted in the assessed program. Although this sample is seen as representative of the organization, a full assessment of all programs might reveal more accurate information about SAFe maturity.

The process of gathering performance and quality data, especially in the case of defect rates, was recently developed which might cause inaccuracy of measurements due to missing data.

Other limitations arise due to the validation approaches. For instance, typical limitation of a case study is that the results are context specific. Also, if the number of cases is low, no generalizations can be made. Given that the research was done and pertains to one case study organization, the limitation that arises from this research is the lack of generalizability. Even though it is appropriate to draw conclusions from the data and the obtained results, these results cannot be generalized to other organizations without further studies.

6.4. Future research

This exploratory research contributed to the initial challenges of implementing and measuring SAFe maturity and should pave the way for future research in this area. In this section a few recommendation for future research are made. Among the primary recommendations is to conduct a longitudinal study of an organization(s) which adopt SAFe through a gradual implementation approach as suggested in this research effort. The successes and challenges of this gradual implementation approach should be compared with the current wholesale approach which mainly involves implementation through training and certification.

Due to the current infancy status of SAFe itself, there needs to be further exploration of the risks behind SAFe adoption possibly through case studies in several organizations. Although the barriers and risk of adopting SAFe are comparable to those of adopting agile, typically large enterprises deal with additional complexities i.e. distributed environment, outsourcing and complex hierarchical structures which can lead to additional challenges. The reorganizations enterprises are involved in when adopting SAFe are complex and this can have significant impacts on their financial performance. Therefore, the risks of this effort should be more clearly elaborated with focus on the assessment of these risks and barriers. The development of a risk assessment model might be appropriate.

Further refinements are also needed of the SAFe maturity model due to continuous changes of the SAFe itself. With each new updated version of SAFe, there is introduction of new practices which might replace the old SAFe practices. Thus, the maturity model has to continuously updated and further validated in future studies. Also, the current indicators which check for existence of SAFe practices is too simplistic and should further be developed in order to have a more accurate assessments of maturity.

Bibliography

[1] Laplante, Phillip A., and Colin J. Neill. "The demise of the Waterfall model is imminent." Queue 1.10 (2004): 10.

[2] Leffingwell, Dean. Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Addison-Wesley Professional, 2010.

[3] Fowler, Martin, and Jim Highsmith. "The agile manifesto." Software Development 9.8 (2001): 28-35.

[4] Leffingwell, Dean. Scaling software agility: best practices for large enterprises. Pearson Education, 2007.

[5] Scaled Agile Framework. Retrieved from http://scaledagileframework.com

[6] Philips, Company Profile. Retrived from http://www.philips.com/about/company/companyprofile.page

[7] Applying the Scaled Agile Framework in Outsorcing Context: Infogain Case Study. Retrieved from http://scaledagileframework.com/infogain-case-study

[8] Omar, Mazni, Sharifah-Lailee Syed-Abdullah, and Azman Yasin. "The impact of agile approach on software engineering teams." American Journal of Economics and Business Administration 3.1 (2011): 12.

[9] Puneet Agarwal. 2011. Continuous SCRUM: agile management of SAAS products. In Proceedings of the 4th India Software Engineering Conference (ISEC '11). ACM, New York, NY, USA, 51-60.

[10] C. J. Torrecilla Salinas, M. J. Escalona, and M. Mejías. 2012. A scrum-based approach to CMMI maturity level 2 in web development environments. In Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services (IIWAS '12). ACM, New York, NY, USA, 282-285.

[11] Jessica Rubart and Frank Freykamp. 2009. Supporting daily scrum meetings with change structure. In Proceedings of the 20th ACM conference on Hypertext and hypermedia (HT '09). ACM, New York, NY, USA, 57-62.

[12] Rautiainen, K.; von Schantz, J.; Vähäniitty, J., "Supporting Scaling Agile with Portfolio Management: Case Paf.com," System Sciences (HICSS), 2011 44th Hawaii International Conference on , vol., no., pp.1,10,

[13] K. Beck, Extreme Programming Explained: Embrace Change, Second ed. Reading, Mass.: Addison-Wesley, 2005.

[14] Abdullah, E.; Abdelsatir, E.-T.B., "Extreme programming applied in a large-scale distributed system," Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on , vol., no., pp.442,446

[15] Raman, S., "Lean software development: is it feasible?," Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE , vol.1

[16] Larman, Craig, and Bas Vodde. 'Scaling Lean Agile Development. Thinking and Organizational Tools for Large-Scale Scrum' Addison-Wesley, 2008.

[17] Dove, Rick. Response ability: the language, structure, and culture of the agile enterprise. John Wiley & Sons, 2002.

[18] Vázquez-Bustelo, Daniel, and Lucía Avella. "Agile manufacturing: Industrial case studies in Spain." Technovation 26.10 (2006): 1147-1161.

[19] Yin, R. K. 1989. Case study research: Design and methods. Applied Social Research Series, Vol. 5. London: Sage.

[20] Eisenhardt, K. M. 1989. Building theories from case study research. Academy of Management Review14 (4): 532–50

[21] J. Highsmith, Agile: From Rogue Teams to Enterprise Acceptance Cutter Consortium: Business Technology Trends and Impacts, 2006.

[22] B. Boehm, Get Ready for Agile Methods, with Care, Computer, Vol. 35 (2002),pp. pp. 64-69

[23] S. Nerur, R. Mahapatra and G. Mangalaraj, Challenges of migrating to agile methodologies, Commun. ACM, 48 (2005), pp. 72-78.]

[24] Boehm, Barry, and Richard Turner. "Management challenges to implementing agile processes in traditional development organizations." Software, ieee 22.5 (2005): 30-39.

[25] Lindvall, Mikael, et al. "Empirical findings in agile methods." Extreme Programming and Agile Methods—XP/Agile Universe 2002. Springer Berlin Heidelberg, 2002. 197-207.

[26] Werth, Laurie H. Lecture Notes on Software Process Improvement. No. CMU/SEI-93-EM-8. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 1993.

[27] Pikkarainen, Minna, Outi Salo, and Jari Still. "Deploying agile practices in organizations: a case study." Software Process Improvement. Springer Berlin Heidelberg, 2005. 16-27.

[28] Sidky, Ahmed. A structured approach to adopting agile practices: The agile adoption framework. Diss. Virginia Polytechnic Institute and State University, 2007.

[29] R. McFeeley, IDEAL: A User's Guide for Software Process Improvement, Software Engineering Institute (SEI), 1996, pp. 222.

[30] Sircar, S., Nerur, S.P., and Mahapatra, R. Revolution or Evolution? A Comparison of Object-Oriented and Structured Systems Development Methods. MIS Quarterly 25, 4 (Dec. 2001), 457–471

[31] Cockburn, Alistair, and Jim Highsmith. "Agile software development, the people factor." Computer 34.11 (2001): 131-133.

[32] Coram, Michael, and Shawn Bohner. "The impact of agile methods on software project management." Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the. IEEE, 2005.

[33] Gallivan, Michael, and Mark Srite. "Information technology and culture: Identifying fragmentary and holistic perspectives of culture." Information and organization 15.4 (2005): 295-338.

[34] Leidner, Dorothy E., and Timothy Kayworth. "Review: a review of culture in information systems research: toward a theory of information technology culture conflict." MIS quarterly 30.2 (2006): 357-399.

[35] Iivari, Netta. "Culturally compatible usability work: An interpretive case study on the relationship between usability work and its cultural context in software product development organizations." Journal of Organizational and End User Computing (JOEUC) 22.3 (2010): 40-65.

[36] Ozcan-Top, Ozden, and Onur Demirörs. "Assessment of Agile Maturity Models: A Multiple Case Study." Software Process Improvement and Capability Determination. Springer Berlin Heidelberg, 2013. 130-141.

[37] Soares, Felipe, and Silvio Meira. "An Agile Maturity Model for Software Development Organizations." ICSEA 2013, The Eighth International Conference on Software Engineering Advances. 2013.

[38] Qumer, Asif, and Brian Henderson-Sellers. "A framework to support the evaluation, adoption and improvement of agile methods in practice." Journal of Systems and Software 81.11 (2008): 1899-1919.

[39] Patel, Chetankumar, and Muthu Ramachandran. "Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices." Int. J. of Software Engineering, IJSE 2.1 (2009): 3-28.

[40] Goldratt, E. M. "The Haystack Syndrome: Sifting Information out of the Data Ocean, 1990."

[41] Mocker, M., Ross, J.W. & Heck, E. van (2014). Transforming Royal Philips: Seeking Local Relevance While Leveraging Global Scale.

[42] Schweigert, Tomas, et al. "Agile maturity model: a synopsis as a first step to synthesis." Systems, Software and Services Process Improvement. Springer Berlin Heidelberg, 2013. 214-227.

[43] Glazer, Hillel, et al. "Cmmi or agile: Why not embrace both!." (2008).

[44] Moniruzzaman, A. B. M., and Dr Syed Akhter Hossain. "Comparative Study on Agile software development methodologies." arXiv preprint arXiv:1307.3356 (2013).

[45] Hass, Kathleen B. "The blending of traditional and agile project management." PM world today 9.5 (2007): 1-8.

[46] Royce, Winston W. "Managing the development of large software systems." proceedings of IEEE WESCON. Vol. 26. No. 8. 1970.

[47] Reinertsen, Donald G. The principles of product development flow: second generation lean product development. Redondo Beach,, Canada: Celeritas, 2009.

[48] Osborne, Jonathan, et al. "What “ideas-about-science” should be taught in school science? A Delphi study of the expert community." Journal of research in science teaching 40.7 (2003): 692-720.

[49] Röglinger, Maximilian, Jens Pöppelbuß, and Jörg Becker. "Maturity models in business process management." Business Process Management Journal 18.2 (2012): 328-346.

[50] Ryerson Exploratory Research. Retrieved at www.ryerson.ca/~mjoppe/ResearchProcess/ExploratoryResearch.htm, (2007).

[51] Mingers J."Combining IT research methods: toward a pluralist methodology", Information Systems Research, vol 12, no 3 (2001)

[52] Talor-Powell, E. "Quick tips collecting group data: Delphi technique ", University of Wisconsin, (2002)

[53] Röglinger, M., Pöppelbuß, J. & Becker, J. (2012). Maturity Models in Business Process Management. Business Process Management Journal, 18(2), Preprint.

[54] De Bruin, Tonia, et al. "Understanding the main phases of developing a maturity assessment model." (2005): 8-19.

[55] Rohunen, Anna, et al. "Approaches to agile adoption in large settings: a comparison of the results from a literature analysis and an industrial inventory." Product-Focused Software Process Improvement. Springer Berlin Heidelberg, 2010. 77-91.

[56] Hodgetts, Paul. "Refactoring the development process: experiences with the incremental adoption of agile practices." Agile Development Conference, 2004. IEEE, 2004.

[57] OPM3, "Organizational project management maturity model", Newtown Square, Pennsylvania USA, Project Management Institute.2003.

[58] Case Study – BMC : Scaling Agile Development (2013), Retrieved June, 2014 from http://www.rallydev.com/sites/default/files/Case%20Study-BMC.pdf

[59] Amal, K. A. (2005). "Using the Delphi Technique to Search for Empirical Measures of Local Planning Agency Power." The Qualitative Report 10(4).

[60] Golafshani, N. (2003). Understanding reliability and validity in qualitative research. The Qualitative Report, 8(4), 597–607.

[61] Curtis, Bill, Bill Hefley, and Sally Miller. People Capability Maturity Model (P-CMM) Version 2.0. No. CMU/SEI-2009-TR-003. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2009.

[62] El Emam, Khaled. "The internal consistency of the ISO/IEC 15504 software process capability scale." Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International. IEEE, 1998.

Appendix A. Graphs and tables

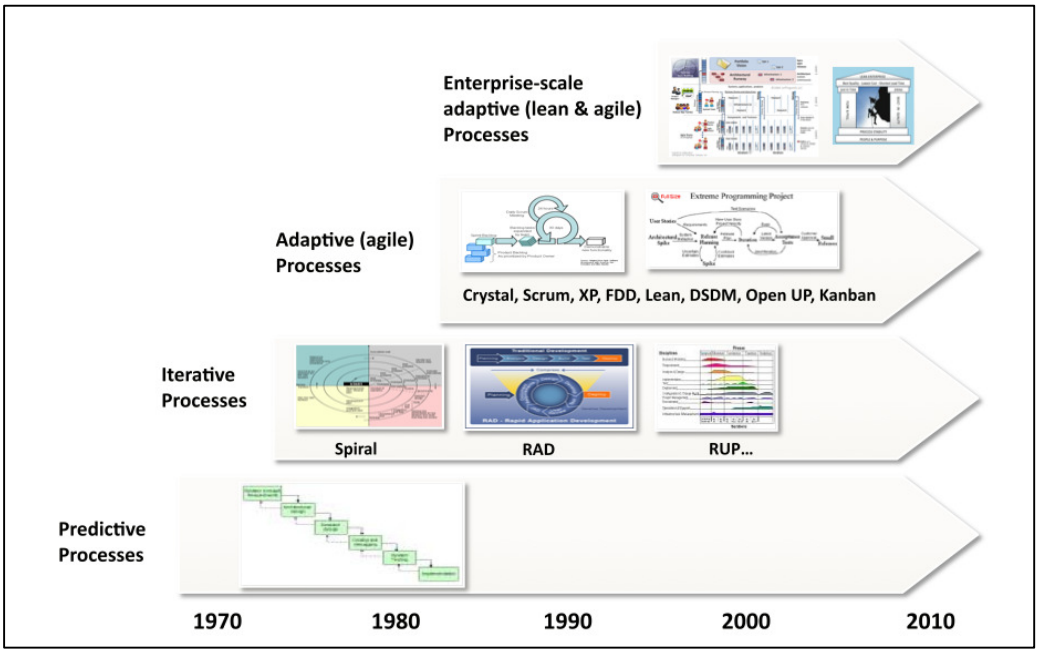


Figure 1. Adapted from Trial Ridge Consulting, LLC [2]

Table 1. Cockburn's Levels

Level	Characteristics
3	Able to revise a method (break its rules) to fit an unprecedented new situation.
2	Able to tailor a method to fit a precedented new situation.
1A	With training, able to perform discretionary method steps (e.g., sizing stories to fit increments, composing patterns, compound refactoring, complex COTS integration). With experience can become Level 2.
1B	With training, able to perform procedural method steps (e.g. coding a simple method, simple refactoring, following coding standards and CM procedures, running tests). Withexperience can master some Level 1A skills.
-1	May have technical skills, but unable or unwilling to collaborate or follow shared methods.

Table 2. Independent samples T Test for time-to-market and project cost

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Time_to_Market	Equal variances assumed	22.039	.000	-10.434	233	.000	-334.267	32.036	-397.385	-271.150
	Equal variances not assumed			-9.331	129.296	.000	-334.267	35.825	-405.147	-263.388
Total_project_cost	Equal variances assumed	3.475	.064	-.528	220	.598	-84024.918	159051.565	-397484.632	229434.796
	Equal variances not assumed			-.492	143.711	.623	-84024.918	170778.221	-421586.647	253536.811

Table 3. Statistics of velocity reliability per sprint

CONFIDENTIAL
CONFIDENTIAL
CONFIDENTIAL

CONFIDENTIAL
CONFIDENTIAL
CONFIDENTIAL

Appendix B. Agile Manifesto principles [3]

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Appendix C. SAMI populated with agile practices (adapted from [28])

	Embrace change to deliver customer value	Plan and deliver Software Frequently	Human-centric	Technical Excellence	Customer Collaboration
Level 5 Encompassing Establishing a vibrant environment to sustain agility	Low process ceremony	-Agile project estimation	-Ideal agile physical setup	-Test driven development -Paired programming -No/minimal number of level -1 or 1b people on team	-Frequent face-to-face interaction between developers and users (collocated)
Level 4 Adaptive Responding to change through multiple levels of feedback	-Client Driven iterations -Continuous customer satisfaction	-Smaller and more frequent releases -Adaptive planning		-Daily progress tracking meetings -Agile Documentation -User stories	-CRACK Customer immediately accessible -Customer contract revolves around commitment of collaboration
Level 3 Effective Developing high quality, working software in an efficient and effective manner		-Risk driven iterations -Plan features not tasks -Maintain a list of all features and their status (backlog)	-Self-organizing teams -Frequent face to face communication	-Continuous integration -Continuous improvement (refactoring) -Unit tests -30% of level 2 and level 3 people	
Level 2 Evolutionary Delivering software early and continuously	-Evolutionary requirements	-Continuous delivery -Planning at different levels		-Software configuration management -Tracking iteration progress -No big design up front(BDUF)	-Customer contract reflective of evolutionary development

	Embrace change to deliver customer value	Plan and deliver Software Frequently	Human-centric	Technical Excellence	Customer Collaboration
Level 1 Collaborative	-Reflect and tune process	-Collaborative Planning	-Collaborative teams -Empowered and motivated teams	-Coding standards -Knowledge sharing -Task volunteering	-Customer commitment to work with development team

Appendix D. Description of SAFe maturity model practices

	Practice	Description of Practice
Level 1	Reflect and tune process	Holding retrospectives at regular intervals of the development process. The objective of this practice is to overcome process challenges that have been faced thus far. [28]
	Collaborative planning	Collaborative Planning encourages all stakeholders to come together during the planning phase. The collaborative efforts increase visibility, loyalty, and acceptance and buy in from all stakeholders. [28]
	Empowered and motivated teams	Supervisors must empower and equip the development teams with the authority to make decisions on their own. This authority helps motivate the team members to solve problems and tasks on their own. [28]
	Collaborative teams	Team members must communicate and cooperate with each other and other teams.[28]
	Coding standards	Collaboration through code by creating a common language (coding standards) among developers. Coding standard increase the comprehension and ease of sharing code among team members. [28]
	Knowledge sharing	Knowledge sharing tools enhance collaboration by helping to document and maintain the information and knowledge exchange among team members. Knowledge sharing tools can be electronics (e.g. wiki, blogs) or simple whiteboards and walls. [28]
	Task volunteering	During the planning meeting the developers should volunteer for tasks, rather than tasks being assigned to them by managers. This practices increases motivation, job satisfaction and quality of performance from team members. If there are no volunteers, the team should take collective responsibility to complete the task. [28]
	Acceptance testing	Acceptance tests are functional tests that verify that the system implements the story as intended. To avoid large volume of manual tests they are automated wherever possible. [2]
	Customer commitment to work with development team.	The customer has to commit to working with the development team in order to establish collaboration with development team. [28]
	User stories	An agile practice where software requirements are formulated as one or two sentences in the every-day language of the user. Users stories are a quick way of handling customer requirements with minimal amount of documentation. [28]

	Practice	Description of Practice
Level 2	Evolutionary requirements	Requirements should be evolving iteratively rather than being fully developed in one major specification effort. The requirements in agile tend to evolve and change based on customer feedback and thus this agile practice is important. [28]
	Smaller, more frequent releases	Agile teams should deliver smaller and more frequent releases to customers and end users. This provides increased responsiveness and reduces risks to the enterprise. This practice does not specify how often an organization should schedule a release which is the purpose of a similar practice at Level 4. [4]
	Requirements Discovery	Agile practice which is used to better understand what needs to be build and why. There are variety of software requirements techniques which are recommended and can be used by teams e,g, Spikes, Use-Case modeling, User Experience Mock-ups, etc. [2]
	Continuous delivery	Delivering software in small iteration at regular intervals. These regular intervals ensure that the team has to divide the product for each iteration. At this level the duration of the iteration is irrelevant but it is later addressed at level 4.[28]
	Two-level planning and tracking	This agile practice suggests development of iteration planning and release planning. Iterations plans are with short term focus and are estimated with great precision and confidence. Releases have long term focus and are more coarse-grained, less comprehensive and less precise. [2]
	Agile estimating and velocity	Agile estimating is a practice for estimating the workload to be delivered and is critical to the productivity and reliability. Estimating is done using story points. The number of story points a team delivers per iteration defines the team velocity. This velocity estimate can then be used for estimating schedule and estimating cost. [4]
	Release planning	Release planning is the seminal, cadence based synchronization point of the Agile Release train (ART). It is a routine, program-scale, face-to-face event which typically includes all members of the program. It takes place over a two day period at each PSI boundary. [5]
	The define/build/test team	A cross functional group of individuals that has the ability and authority to Define (elaborate and prioritize requirements and design the solution), Build (write the code) and Test (run tests cases) – all in a short iteration time-box. The define/build/test team is self-organize, empowered and self-managing and thus depends on practices from lower level. [5]
	Software configuration management	SCM tools help control the different versions of the software being developed. [28]
	Automated testing	By automating test cases, software engineers can easily run their test cases often. At this level of maturity, automated testing should replace some manual testing efforts which will increase the

	Practice	Description of Practice
		speed of delivery and satisfy the goal of level 2 which is to deliver software early and continuously. [Appendix F]
	Tracking iteration progress	Agile practice which is concerned with the team having the means by which they can measure the progress of the development effort within iteration. This concept does not dictate a particular method to fulfill this tracking but its latter defined at level 4 (Daily progress tracking meetings). [28]
	No big design up front(BDUF)	Agile practice which ensures that the product is being developed using an evolutionary approach. BDUF is where a "big" design is created before coding and testing takes place and is typical for waterfall development process. In agile design occurs throughout the development process. [28]
	Product backlog	The product backlog is a repository for all the upcoming work which is anticipated to be delivered. The product backlog can be utilized at different levels of granularity, e.g. team, program and portfolio backlog, [2]
	Customer contract reflective of evolutionary development	The customer understands the evolutionary nature of software development and the contract reflects this evolutionary approach. This practices prevents the contract to define the dates when milestones should be completed but its rather reflecting of the evolutionary approach. [28]
Level 3	Regular reflection and adaptation	Teams adopt regular reflection and adaptation at each iteration and release levels. These reflections are accompanied by quantitative assessment which measure iteration and release metrics. [4]
	Risk driven Iterations	Risk driven iterations help tackling risk elements as early as possible. Mitigating these risks early ensures that the project team does not spend a considerable amount of time building a system that they cannot complete. By catching these issues the development becomes more effective. [28]
	Plan features not tasks	Customer expresses their needs In terms of features, so when the feature changes the impact on the related tasks is minimized. The planning should be done in terms of features in order to prepare the development process for Client Driven Iterations at Level 4. [28]
	Roadmap	Agile practice used to establish alignment across all the teams involved in the ART while also providing predictability to the deliverables over an established timeline horizon.[5]
	Mastering the iteration	The base construct of agile and iterative development is the iteration- the ability of a team to create working, tested software in a short time-boxed interval. The aim of this agile practice is effective production of an increment of working software at each iteration. [4] In essence, this practice validates of the effectiveness

	Practice	Description of Practice
		of the practices established on lower levels.
	Software Kanban Systems	SAFe suggests the development and implementation of Kanban systems for business and architectural portfolio epics. The Kanban system describes four queues that an epic passes through on the way to implementation: Funnel, Backlog, Analysis, and Implementation. [5] The Kanban Systems are used for visualizing workflow, limiting the work in progress, measuring and managing flow, making process policies explicit and using models to recognize improvements. [2]
	PSI/Release	PSI is a development time box (super sprint) that uses cadence and synchronization to facilitate planning, provide for aggregation of newsworthy value, and provide a quantum unit of thinking for portfolio level consideration and road mapping. The goal of PSI/Release is continuous integration and system validation and reducing the risk od deferred integration. [5]
	Agile Release Train	ART is a long-lived team of agile teams that serves the program-level value delivery mechanism in SAFe. Release trains are organized around the enterprise Value Streams and ART aligns teams to common mission, schedule and cadence which helps implement continuous product development flow. [5]
	Self-organized teams	Team is empowered to make decisions without waiting for approval from management. Teams are cross-functional, roles and responsibilities of team members are not or loosely defined and the whole team is responsible for delivery of working software. The importance of self-organizing teams is highlighted in the Agile Manifesto which states that the best architectures, requirements and designs emerge from self-organizing teams. [28]
	Frequent face to face communication	For establishing efficient and effective development process frequent communication among team members is necessary. [28]
	Scrum of Scrum	Scrum of Scrum (SoS) is an agile practice which scales Scrum to large programs consisting of several teams. In this shorts stand-up meeting, Scrum Masters of each team work out interdependencies, report status and state any looming impediments. [2]
	Continuous integration	Continuous integration is an agile which encourages members of a team to integrate their work frequently. It is preferred that each integration is verified by an automated build tool in order to detect any integration errors as quickly as possible. [28]
	Continuous improvement (Refactoring)	Refactoring is an essential practice to be adopted at level 3 because of the evolutionary development process assumed at Level 2. Refactoring involves rewriting the code to improve its structure while preserving the behavior. In general refactoring focuses on removing code duplication.[28]
	Unit tests	Unit tests are code procedures used to validate that the individual

	Practice	Description of Practice
		units of source code are working properly. A unit of source code is the smallest testable part of an application. It provides a strict, written contract that the code must satisfy. It is recommended that unit tests are automated. [28]
	30% of level 2 and level 3 people	Cockburn people levels are directly related to the amount of experience the developer has. Cockburn identified three levels of understanding when they approach new material and has argued that a person level of understanding is directly linked to his experience in the domain. At Level 3 the team needs developers who can handle new unexpected problems and that is why Cockburn Level 2 and Level 3 people are needed. [28]
	DevOps	An agile practice that is used to ensure a faster flow of value to the user by tighter integration of development and operations. This is accomplished by integration personnel from operations into the agile teams or by continuously maintaining deployment readiness.[5]
	Vision, features	The vision describes the stakeholder's view of the solution to be developed in terms of stakeholder's needs and proposed features. It captures the essence of the envisaged solution in the form of high-level features, non-functional requirements and design constraints, and provides an overview of the system to be developed. [5] Features are directly connected to the vision since the vision is expressed in terms of features. [Appendix F]
	Impact on customers and operations	Measuring the impact on sales, operations and customer due to the changes in the development model. [4]
Level 4	Client driven iterations	The client is in control and can request and prioritize features per iteration. The client steers the project, iteration by iteration, requesting the features that are of highest business value to them. [28]
	Continuous customer satisfaction feedback	Continuous customer feedback is crucial to ensure the customer is satisfied with what is being developed. If customer feedback is only sought at the end of the project then there is a significant risk that what has been developed ins not what the customer needed. [28]
	Lean requirements at scale	For larger organizations, it becomes increasing more difficult to make the whole team working toward a common purpose. There is a need to create a scalable requirements pattern consisting of vision, roadmap and just-in-time elaboration as a method that brings the benefits of agility to larger scale teams. [4]
	Smaller and more frequent releases	This encourages organizations to keep small timeframe for the releases and limit them to 8 weeks. Usually building a release will include multiple iterations. Having shorter releases helps the development process to embrace change. [28]

	Practice	Description of Practice
	Adaptive planning	Adaptive planning delays developing the iteration details until immediately before the following iteration, and therefore incorporates all the feedback obtained including what is learned during previous iteration. Adaptive planning helps organizations embrace change because the focus shifts from following a plan to adaptive planning based on latest feedback. [28]
	Measuring business performance	Implementing a flexible, automate and meaningful BSC for the enterprise that measures performance in terms of efficiency, value delivery, quality and agility. The organization also measures team and ART performance. [4][5]
	Managing highly distributed teams	Large corporates are distributed. They should be managed with proper communication and the necessary networking and tooling architecture.[4]
	Daily progress tracking meetings	This is a more specific version of tracking iteration progress practice which was introduced at Level 2. This practice emphasizes that the team should be informed on a daily bases regarding the status of the iteration. [28]
	Intentional architecture	Intentional architecture is a set of purposeful, planned architectural initiatives to enhance solution design, performance and usability and provides guidance for inter-team design and implementation synchronization. [5]Architecture that has been planned to some extent, has been built incrementally, has emerged over the course of prior iterations or has evolved to adequately support the needs of the customers. [4]
	CRACK Customer immediately accessible	The customer is Collaborative, Representative, Authorized, Committed and Knowledgeable. At this level the concern is not with the location of the customer but rather how responsive they are. [28]
	Customer contact revolve around commitment of collaboration	The customer agrees to contract the degree and amount of collaboration and not the requirements and features. This is one of the ultimate factors that enables organizations to embrace change. [28]
Level 5	Low process ceremony	Lower process ceremony allow for being responsive to changes. The changes don't have to be approved by a least 3 levels of management. Process ceremony is also the level of paperwork involved in the process.[48]
	Continuous SAFe capability improvement	This is nor a recognized SAFe and agile practices. The practices was introduced with the aim of providing a foundation for enterprises to strive for continuously improving the capabilities of SAFe. SAFe is a product of continuous work and therefore an organization which is at the highest level of maturity should be striving to contribute and improve the practices, artifacts of the SAFe itself. Similar practices which are focused on overall capability improvements are present in several other maturity

	Practice	Description of Practice
		models such as P-CMM, ISO/IEC 15504.
	Agile project estimation	Under the "Planning and Deliver Software Frequently" agile principles most practices are related to planning. Agile estimation is important since plans are only as good as the estimates they are based on. [28] SAFe uses several estimation techniques ex. Cost of Delay (CoD), WSJF, Story points for estimating. [2]
	Ideal agile physical setup	Ideal agile physical setup helps establish the right environment for the agile software development to thrive in. The key in this setup is that the team is co-located and knowledge can be shared among team members instantly. [25]
	Changing the organizations	Organizations must make substantive changes to achieve full benefits of agile at enterprise level. These changes can mean changes in engineering practices, managing the impact on operations, reorganizations, etc. The executive sponsor has a critical role and his dedication and commitment to the process, as well as leadership by example, is the key factor in unlicking great benefits. [4]
	Test driven development	Software development technique that involves repeatedly first writing a test case and then implementing only the code necessary to pass the test.
	No or minimal number of Cockburn Level 1B or -1	Cockburn Level 1B and Level -1 developers have the least experience and are not able or willing to collaborate which can hamper the transition to agility. This is why their presence is discouraged at Level 5. [28]
	Concurrent testing	The practice of incorporating test-development, test automation, and test execution practices within the course of the iteration. Concurrent testing involves several Agile Testing strategies such as: Unit Testing, Acceptance Testing, Component Testing, System, Performance and Reliability testing. This practices is listed at Level 5 since all the tests should be automated, run frequently and all these efforts take long time to master [4]
	Face to face interaction between develops and users	It is ideal to have not just the developers collocated but also have the customers and users in the same rooms. This ensures almost instant feedback and incredible communication. [28]

Appendix E. Indicators per level

Table 1. Indicators for Level 1

Indicator	Practice	Statement	Nominal Values				
			1	2	3	4	5
LV1_1	Reflect and tune process	Developers and management are committed to reflecting and tuning the process after each iteration and release.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV1_2	Collaborative planning	The customers, developers and management collectively plan the projects.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV1_3	Empowered and motivated teams	Management trusts and empowers teams with decision making authority.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV1_4	Collaborative teams	Developers are willing to work in a team, interact with other team members and help each other.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV1_5	Coding standards	The developers use coding standards.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV1_6	Knowledge sharing	Knowledge sharing tools are available and accessible (e.g. Wikis, blogs, etc.)	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

LV1_7	Task volunteering	Tasks are volunteered by developers and are not being assigned to them.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV1_8	User stories	User stories are used as elicitation method/form for high level requirements	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV1_9	Acceptance testing	Acceptance tests are conducted to determine if the requirements of the user stories are met.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV1_10	Customer commitment to work with development team.	The customer is committed to work with the development team.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Table 2. Indicators for level 2

Indicator	Practice	Statement	Nominal Values				
			1	2	3	4	5
LV2_1	Evolutionary requirements	The requirements are not defined upfront and they allowed to evolve during the iterations.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_2	Smaller, more frequent releases	The systems team delivers frequent and small releases to customers and end users during the Systems team Demo.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_3	Requirements Discovery	Agile teams use up-front analysis and variety of requirements discovery techniques to better understand what needs to be build and why.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_4	Continuous delivery	Software is being developed in incremental-iterative fashion every 1-4 weeks.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_5	Two-level planning and tracking	Planning and keeping track of progress happens on release and iteration level	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_6	Agile estimating and velocity	Teams estimate the workload using story points and each team has established velocity.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_7	Release planning	Release planning session is organized in which all relevant stakeholders participate.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

LV2_8	The define/build/test team	The multidisciplinary teams define, build and test software at each iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_9	Software configuration management	The organization has sufficient and usable software configuration tools for agile development.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_10	Automated testing	The development teams has automated their testing efforts.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_11	Tracking iteration progress	The organization has usable and efficient mechanism to monitor and track iteration progress.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_12	No big design up front(BDUF)	The design is not done once at the beginning of the development process but rather is it's a continuous process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_13	Product backlog	The organization uses product backlog on team, program and portfolio level to maintain an up-to date list of all the requirements.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV2_14	Customer contract reflective of evolutionary development	The customer signs a contract to start development of a product whose requirements are not know fully ahead of time and evolve throughout the development.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Table 3. Indicators for Level 3

Indicator	Practice	Statement	Nominal Values				
			1	2	3	4	5
LV3_1	Regular reflection and adaptation	The organization has regular and effective retrospectives at release and iteration levels.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_2	Risk driven Iterations	The organization performs risk assessment to drive the scope of each iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_3	Plan features not tasks	The customer expresses their needs in terms of features and not tasks.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_4	Roadmap	The organization has program roadmap which provides view of the intended deliverables over a time horizon of three to six months.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_5	Mastering the iteration	The development team has effective iterations consisting of sprint planning, tracking, execution and retrospectives.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_6	Software Kanban Systems	Organization uses Kanban Systems for visualizing the workflow, limiting WIP and measuring and managing flow for business and architectural epics.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_7	PSI/Release	There is established development time box or PSI with fixed cadence and synchronization to facilitate planning.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

LV3_8	Agile Release Train	Release trains are organized around value streams and produce system level PSI at every 8-12 weeks in accordance with PSI Objectives	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_9	Self-organized teams	The teams are competent and empowered to work by self-organization without direct management supervision	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_10	Frequent face to face communication	Development teams are co-located and have frequent face-to-face communication	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_11	Continuous integration	The development team/system team have the tools and processes to support continuous integration	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_12	Scrum of Scrum	Scrum of Scrum is used in the organization to successfully manage interdependencies between teams.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_13	Continuous improvement (Refactoring)	The development team often revisits a working component to improve its design or code structure.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_14	Unit tests	The development team is focused on doing unit tests for their methods and functions while they code.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_15	30% of level 2 and level 3 people	What percentage of developers is of Cockburn level 2 or level 3 experts	0-5%	5-10%	10-15%	15-30%	30% or higher

LV3_16	DevOps	There is tight integration of development and operations. (DevOps)	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_17	Vision, features	High level requirements are presented in terms of Vision Document which contains the features to be developed.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV3_18	Impact on customers and operations	The shorter development cycle is aligned with customers and other departments within organization (sales, marketing, and operations).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Table 4. Indicators for Level 4

Indicator	Practice	Statement	Nominal Values				
			1	2	3	4	5
LV4_1	Client driven iterations	The organization allows for the customer to dictate the scope of the iterations.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_2	Continuous customer satisfaction feedback	The customers have the opportunity to give his feedback throughout the development process by means of interacting with working software or prototype.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_3	Lean requirements at scale	The requirements are scaled effectively from investment themes to epics to features to stories and to tasks.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_4	Smaller and more frequent releases	The organization develops fully functional releases every 4-6 sprints	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_5	Adaptive planning	The planning for the next iteration is based on clients feedback from the previous iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_6	Measuring business performance	The organization has implemented flexible, automated and meaningful methods for measuring performance on team, program and enterprise level.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_7	Intentional architecture	The organization has a set of purposeful, planned architectural initiatives to enhance solution design, performance and usability	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

LV4_8	Managing highly distributed teams	The organization has appropriate networking and tooling infrastructure that enhances communication when managing distributed teams.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_9	Daily progress tracking meetings	The managers meet daily with developers for progress updates of a project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_10	CRACK Customer immediately accessible	The customer or customer representative is collaborative, representative, authorized, committed, knowledgeable and immediately accessible when needed.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV4_11	Customer contact revolve around commitment of collaboration	The customer contract revolves around interaction and collaboration, and not detailed features to be delivered.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Table 5. Indicators for level 5

Indicator	Practice	Statement	Nominal Values				
			1	2	3	4	5
LV5_1	Low process ceremony	There is low level of standardized processes which have to be followed in the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV5_2	Continuous SAFe capability improvement	The organization is continuously contributing to improving the capabilities and practices of the SAFe itself.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV5_3	Agile project estimation	Both managers and teams are experienced and competent in making accurate estimates of effort.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV5_4	Ideal agile physical setup	The organization is willing to have all of the development personal collocated in a common room.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV5_5	Changing the organizations	The Scrum/Agile process is used to drive organization change in incremental fashion where an executive sponsor is dedicated and committed to the process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV5_6	Test driven development	The development team combines test-first development where you write a test before you write just enough production code to fulfill that test and refactoring.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

LV5_7	No or minimal number of Cockburn Level 1B or -1	What is the percentage of Cockburn level 1b or -1	30% or higher	15-30%	10-15%	5-10%	0-5%
LV5_8	Concurrent testing	Automated tests are used for unit, acceptance, component, system and performance testing within the course of the iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
LV5_9	Face to face interaction between develops and users	The developers and users have frequent-face-to face communication	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

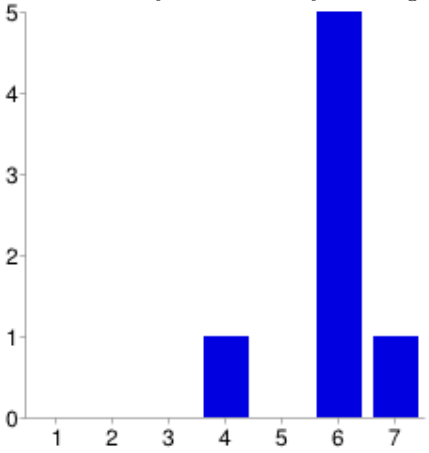
Appendix F. Delphi study questions and responses

The agile and SAFe experts which contributed to the Delphi study development are the following:

- Eelco Rustenburg - Author of Power of Scrum, Partner at Blinklane Consulting, Management Consultant for Agile Adoptions in large companies (e.g Philips, Rabobank)
- Bas Willemsen - Agile coach at Philips
- Maria Lara - Agile Coach at Philips
- Michaël Maurer - Agile coach at Philips, Agile consultant at Blinklane Consulting
- Henk Mooijweer - Agile and Scaled Agile coach at Philips
- (Agile coach wanted to remain anonymous)
- WIm Welberg, Delivery Manager at Philips, SAFe SPC certified

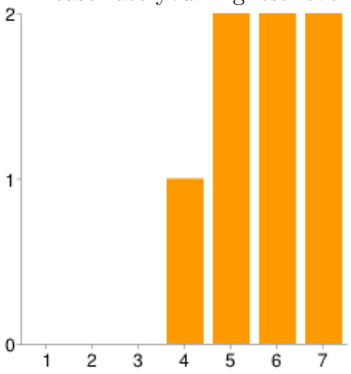
Round 1

The purpose of this questionnaire is to elicit broad feedback from a panel of experts on the proposed implementation strategy and SAFe Maturity Model. The questionnaire is separated in two parts:
-Page 1: Questions regarding your expertise of SAFe and Agile.
-Page 2: Questions on improvement points for the proposed Maturity model.
1. Please rate your familiarity with Agile and SAFe?



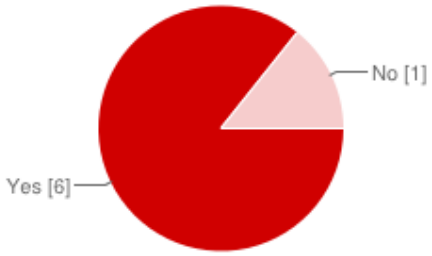
1	0	0%
2	0	0%
3	0	0%
4	1	14%
5	0	0%
6	5	71%
7	1	14%

2. Please rate your highest level of involvement in Agile and SAFe practices?



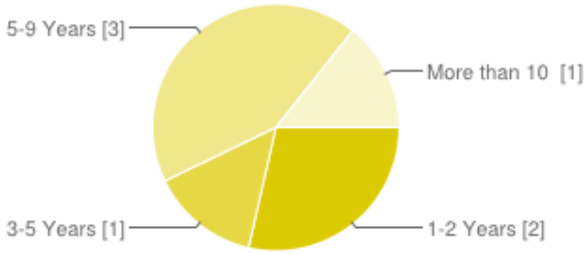
1	0	0%
2	0	0%
3	0	0%
4	1	14%
5	2	29%
6	2	29%
7	2	29%

3. Are you SPC Certified?



Yes	6	86%
No	1	14%

4. How many years of experience do you have with Agile?

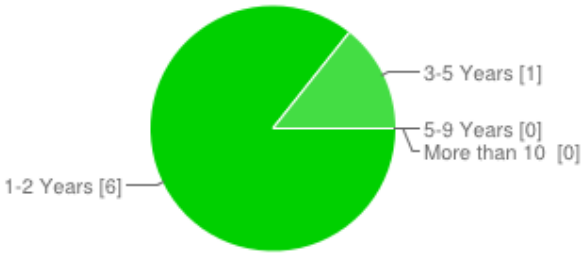


1-2 Years	2	29%
3-5 Years	1	14%
5-9 Years	3	43%
More than 10 Years	1	14%

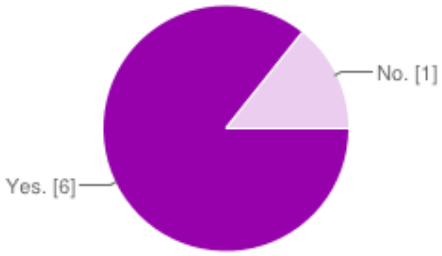
5. How many years of experience do you have with SAFe?

1-2 Years	6	86%
3-5 Years	1	14%
5-9 Years	0	0%

More than 10 Years	0	0%



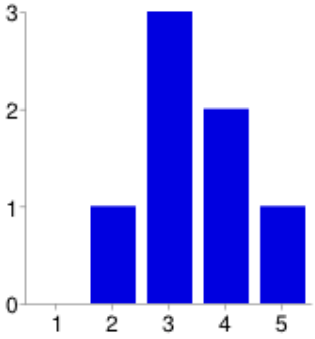
6. Do you want to be listed as contributor to this research effort?



Yes.	6	86%
No.	1	14%

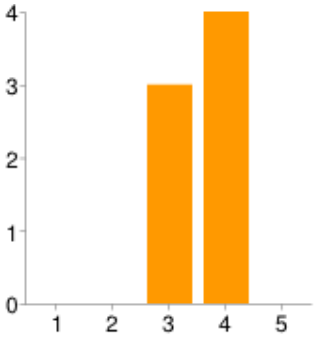
Page 2

1. There is a need for a more structured implementation strategy/Maturity Model for SAFe adoption?



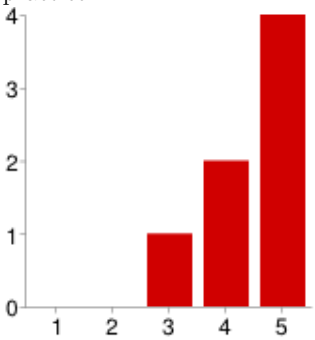
1	0	0%
2	1	14%
3	3	43%
4	2	29%
5	1	14%

2. The IDEAL model is useful for guiding SAFe adoption?



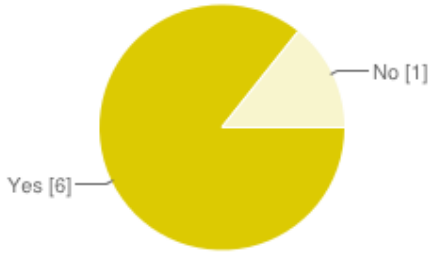
1	0	0%
2	0	0%
3	3	43%
4	4	57%
5	0	0%

3. An enterprise should assess the barriers/risks of adopting SAFe before implementing it in practice?



1	0	0%
2	0	0%
3	1	14%
4	2	29%
5	4	57%

4. Are the agile levels of the proposed SAFe Maturity Model comprehensive enough?



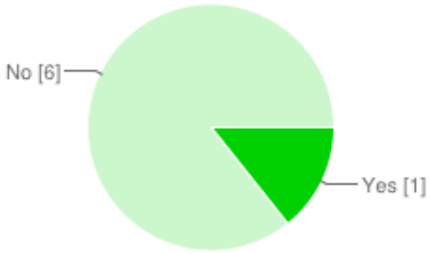
Yes	6	86%
No	1	14%

5. If you answered No, which level(s) should be added?

- No extra levels needed.

- Think about how to make CEO's and CIO's really want to go all the way to five... More engaging level 5 in wording and message?
- I am not clear on how to translate these levels to maturity. For instance, does Level 5 means that Level 1 to 4 are covered? I miss the relation between levels. Also, in my opinion, couple of items in low levels could show more agile maturity that items in higher levels.

6. Are all SAFe practices reflected in the proposed SAFe Maturity Model?



Yes	1	14%
No	6	86%

7. If you answered No, please list the practices which are missing and suggest the level they should be introduced.
- Continuous improvement on team/program/portfolio level
- DevOps, Portfolio Kanban
- PSI - the concept of a fixed date to plan towards (usually 3 months ahead) during the Release Planning
- PSI/ Release, level 3
- Emergent Architecture > SoS is not a recognized practice > Include operations during development and move towards Devops > Measuring relative business performance improvement (value delivery) > User stories = Agile requirements > Agile planning, estimation and velocity > Software Kanban Systems = Portfolio? >
- Scrum of Scrums (of is this the System of System?) Architectural Roadmap?

8. From the already listed SAFe and Agile Practices, please indicate which practices should be introduced on lower/higher level of maturity based on your experience and knowledge? Please explain

-Both, DevOps and Portfolio Kanban, should be in a higher level of Maturity. So, it is not part of the basics. The basics: Product Ownership Scrum team maturity Quality of the Product backlog Changing the organizations (Behavioral change) Scrum of scrums Value thinking Continuous Improvement.
- not known yet
-Human centric: Level 1: more focus on habitual change (sharing responsibilities, forcing iterative working) Level 2: Readyflow (working together to get requirement from nothing to done) Level 3: empowerment and motivation (understanding the bigger picture, feel freedom to do something Level 4: self-organizing teams Level 5: self-organizing highly distributed teams Embrace change: Level 1/2: Add Disciplined team level process + add synchronized team level process
- Automated testing and related practices should be in L2 already.
- User stories: I would not place those under 'Technical Excellence'. To me they could go into 'Embrace change', 'Plan and deliver' of 'Customer Collaboration', depending on what you feel is the most important aspect of US. I would put the under 'Customer Collaboration'. The Define/build/Test Component team: I don't component teams. The goal is feature teams and

only when the situation does not allow for them, component teams are an option. I would do away with the word 'component'. Vision, features: Would not place them under Technical Excellence. Vision is about 'Customer collaboration' and 'customer value'. Features go in the same category as user stories but probably on a higher level. ART: not about technical excellence, would place it in 'Plan and deliver'. Intentional architecture: to me this is about 'Technical excellence' but I understand the 'Plan and deliver' aspect of it. Would move it though since it is closely related to 'no BDUF'. Impact on customer: Would place the at least one, but probably two levels lower in the 'Customer Collaboration' column.
9. Do you have other suggestions for improving the Maturity Model? Please explain
-L4: managing highly distributed teams: you explain this is for large companies but at L5 you say Ideal agile physical set up is to be co-located. Looks a bit contradicting: as big company you will never reach L5 Explanation of Software Kanban Systems: "....for economic decision making"
- I do not understand this explanation "Mastering the iteration": ....."increment of PSI..."-double: the I in PSI = increment Explanation of Requirements Discovery: my question: is this the same as a SPIKE?
-Add Organizational Change management elements in it. One of the prime risks of organizations adopting SAFe is that the agale mindset is not mature enough to effectively adopt SAFe. To adopt a new mindset in an organization, change management is needed.
- Describe all the practices more clearly using verbs (actionable) and MECE. Make the descriptions more specific and maybe even a definition.
- The danger with maturity models is that people start chasing levels, instead of focusing on the improvements that make the team and cooperation great. It then becomes a more mathematical exercise which it completely not your intention. I would in the accompanying text describe what the advantages are of the items in the specific cells and how to recognize a team that uses the objects and practices effectively (what are the characteristics).
- Think should have a stronger focus on continuous improvement. It is the core of lean, and the core of Agile. (We continuously strive for better ways of delivering....) Propose: either add 6th column or make first column more focused on it. Add lean principles like think about improvement of tomorrow. Lean leadership: improve the process, prioritize the value.
-It will help to clarify the relations between the different levels. Also, what is needed to grow to the next level. There are some practices that could be phrase different to avoid wrong interpretations. For the rest, it is a great work.
-Do not get it alright in the first place but start with the first setup, use it in practice, learn and adapt, so it is a continuous change.

Round 2

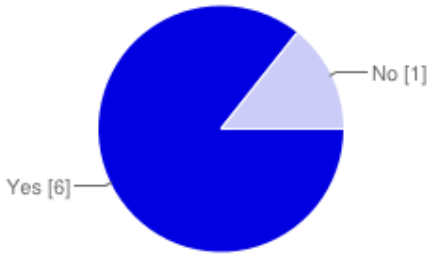
The purpose of this questionnaire is to elicit feedback from a panel of experts on the final proposed SAFe Maturity Model.

The questionnaire is separated in two parts:
-Page 1: Reaching consensus on improvement points from Round 1.
-Page 2: Questions on completeness and practicality of the Maturity Model.

The improvement points from the first round are grouped in three groups:
- Fundamental revisions: Contain fundamental changes to the model, detailing and further characterizing the existing concepts and further justifying the elements.
-Additive revisions: Adding new elements and practices which were proposed during the first round of the Delphi study.
-Corrective revisions: Correcting existing elements based on initial feedback from experts.

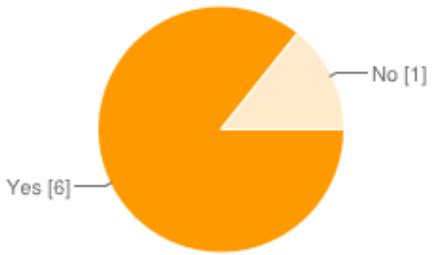
1. Please list your Name and Job Title
Bas Willemsen - Agile coach ;Maria Lara - Agile Coach ;Eelco. Agilist ;Michaël Maurer - Agile coach ;Henk Mooijweer Agile and Scaled Agile coach; [redacted] WIm Welberg, Delivery Manager.

2. Do you agree with the fundamental revisions from Round 1.



Yes	6	86%
No	1	14%

3. If you answered No to the question above, please list other fundamental revisions to be addressed
As a fundament of the framework I find the practices are not well defined. Putting this out of seriously reduces the usability.
4. Do you agree with the additive revisions from Round 1.

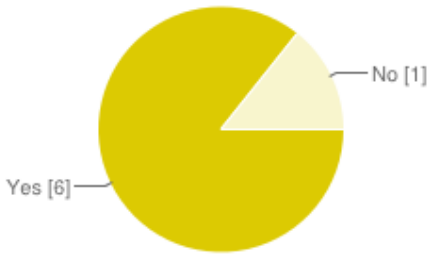


Yes	6	86%
No	1	14%

5. If you answered No to the question above, please mention the practice(s) that should NOT be added.

Automated Testing (Level 2)	0	0%
PSI/Release (Level 3)	0	0%
DevOps (Level 3)	1	100%
Scrum of Scrum (Level 3)	0	0%
Continuous SAFe capability improvement(Level 5)	0	0%

6. Do you agree with the corrective revisions from Round 1?



Yes	6	86%
No	1	14%

7. If you answered No to the question above, please mention the practice(s) which is misaligned and suggest the correct alignment.

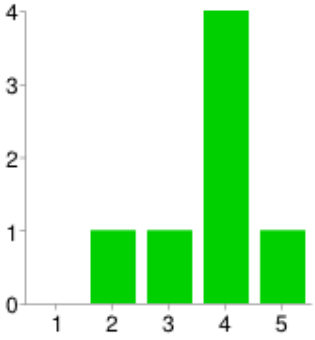
- I agree, but as an additional remark, you can add something about of you can directly start with SAFe or that you need to become first level 3 e.g. and with that you can addapt SAFe.. Just a thought.

-Vision, features is unclear. Think is meant as: "Features connected to the vision explicitly"

-Point 1: Why is Dev Ops team separately mentioned ? Then you can also start mentioning a Systems team in my view. Point 2: I think level 3 should be all about PSI/ ART/ SoS: so baiscs of SAFe. Then in level 4 you can add DevOPs teams, System teams. In other word, if you decide to keep DevOps in, I would suggest to move it to level 4 (and then also add Systems Team)

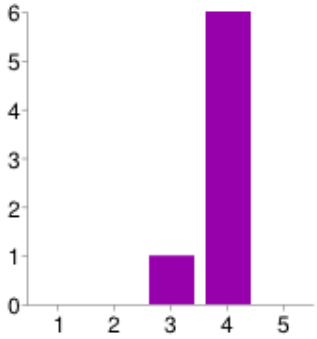
Page 2

1. COMPLETENESS: All the necessary components are present in this implementation strategy in order to achieve its overall goal of aiding an organization in the adoption of SAFe.



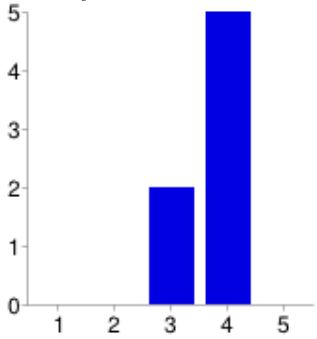
1	0	0%
2	1	14%
3	1	14%
4	4	57%
5	1	14%

2. UNDERSTANDABILITY: The Maturity Model for SAFe is easy to understand.



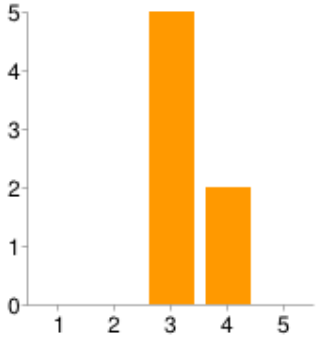
1	0	0%
2	0	0%
3	1	14%
4	6	86%
5	0	0%

3. PRACTICALITY: The Maturity Model for SAFe is practical and can be used in the industry.



1	0	0%
2	0	0%
3	2	29%
4	5	71%
5	0	0%

4. NECESSITY: The Maturity Model is beneficial and necessary for the industry.



1	0	0%
2	0	0%
3	5	71%
4	2	29%
5	0	0%