CHALMERS
UNIVERSITY OF TECHNOLOGY

Scaled Agile Maturity Measurement in Manufacturing

A Study at Volvo Car Corporation

Master's thesis in Management and Economics of Innovation

Rabbani Mahatma Bagaskara
Verawaty Sampe

DEPARTMENT OF TECHNOLOGY MANAGEMENT AND ECONOMICS
DIVISION OF INNOVATION AND R&D MANAGEMENT

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se
Report No. E2021: 076

REPORT NO. E 2021: 076

Scaled Agile Maturity Measurement in Manufacturing

A Study at Volvo Car Corporation

RABBANI MAHATMA BAGASKARA
VERAWATY SAMPE

Department of Technology Management and Economics
Division of Innovation and R&D Management
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Scaled Agile Maturity Measurement in Manufacturing
A Study at Volvo Car Corporation
RABBANI MAHATMA BAGASKARA
VERAWATY SAMPE

© RABBANI MAHATMA BAGASKARA, 2021.
© VERAWATY SAMPE, 2021.

Report no. E2021: 076
Department of Technology Management and Economics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 100

Cover:
C40, new electric car released in March 2021 by Volvo Cars (source: volvocars.com)

Gothenburg, Sweden 2021

Scaled Agile Maturity Measurement in Manufacturing
A Study at Volvo Car Corporation

RABBANI MAHATMA BAGASKARA
VERAWATY SAMPE

Department of Technology Management and Economics
Chalmers University of Technology

ABSTRACT

Agile has been rapidly adopted by organisations, as it is seen as an effective method for dealing with complex tasks and changing environments. However, agile is initially designed for software development and carried out by small and individual teams. Hence, despite the high success rate offered by team works in agile, adopting agile in non-software development, performed by teams on a large-scale poses its own challenges.

The automotive industry is continuously evolving due to many factors, such as technological shifts and changing customer behaviour. In this paper, we present our study at Volvo Cars, as a case study in six divisions under the Manufacturing Engineering department. Agile in manufacturing in this industry consists of hardware and software development and is carried out by large-scale teams. As the teams are varied, so is their agile adoption. Therefore, it is important to measure their different state of art in their agile maturity to plan a better road map in adopting agile in large-scale development. By means of measuring this different scaled agile maturity, a suitable model that fits manufacturing context is necessary.

In this study, we develop a Scaled Agile Maturity Measurement Model in Manufacturing and its metrics from current models in literature. Our model is based on three agile maturity models (Agile Maturity Model, Scrum Maturity Model, Sidky Agile Maturity Model), where we extend those models with agile, SAFe, and agile manufacturing. So, it consists of levels, principles, and practices as measurement metrics. When applying this model in the six divisions, the measurement result shows variation corresponding to different state in their agile adoption. Hindrances, commonly shared and specific to certain divisions, are presented to analyse these differences.

Keywords: Scaled Agile, Agile Maturity Measurement, Scaled Agile Maturity Model, Agile Manufacturing in Automotive

Table of Contents

1 Introduction

This chapter comprises the master thesis's background, followed by the aim, limitations, and the specification of the issue under investigation to provide a comprehensive understanding of the assignment to be done.

1.1 Background

The ubiquitous fact that despite some projects being completed on time and on the budget, they are still considered failures. They were rigidly planned, therefore failed to recognize, and responded to changes during the process (Serrador and Pinto, 2015). These drawbacks of traditional project management motivated the search for new methodologies to counter these drawbacks. The agile method has been seen as a new and effective alternative to dealing with complex tasks and changing environments, in a short time, in software and product development. This method was initially designed for small and individual teams (Boehm and Turner, 2005). When the project is well-executed, the small teams can always deliver high performance, better results faster, and increase teamwork quality (Rigby, Sutherland, & Noble 2018).

The idea of having an agile team that works across the organisation is appealing. Still, there are growing challenges in the company's structure and cognitive inertia when applying agile to large-scale development. Agile requires less bureaucratic intervention, self-governing, and quick response to change. They greatly rely on the entrepreneurial team, people’s feedback, and iterative process.

Industry continuously faces dynamic changes, such as regulations, environmental concerns, and customer preferences. Automotive industry as one example, robustly faces digital disruption, and it requires shifting from hardware-driven into a more software-driven product (Aboagye, Baig, Hensley, Kelly, Padhi, & Shafi, 2017). Thus, the idea of adopting agile principles is attractive to the automotive industry since it is perceived to be more responsive toward changes.

Agile is initially designed for software development (Dybå and Dingsøyr, 2008). Meanwhile, agile in the automotive industry, in their manufacturing, consists of hardware and software

development that need to continuously synchronize their works. Hence, they pose another challenge to adopt agile in manufacturing, as it is less developed in hardware development.

On another hand, as agile is introduced for small scale, hence its adoption needs to be adjusted to fit the larger organisation. Contrary to small organisations, the complexity of scaled agile increases as large organisations have interdependencies between projects and teams and require intensive coordination (Dikert, Paasivaara, & Lassenius, 2016). Misra et al., (2010) stated that adopting agile at a large-scale development usually means changing the entire organisation's culture.

1.1.1 Case Study

Volvo Car Cooperation (henceforth referred as "Volvo Cars") is an automotive company with the headquarter located in Gothenburg, Sweden. Volvo Cars realise that the automotive industry is continuously changing, driven by technological shifts and customer behaviour (Volvo Cars Group Annual Report, 2020). Therefore, Volvo Cars believe they need to change their traditional ways of working to respond to such changes.

As agile has been proven to increase responsiveness to changes in software development, Volvo Cars have been inspired to also adopt scaled agile in their manufacturing. In 2017, Volvo Cars introduced the Volvo Cars Agile Framework (VCAF) as a foundation to adopt agile in large-scale development, built upon the Scaled Agile Framework (SAFe). However, the process of adopting VCAF in Volvo Cars varies in the organization. For example, some divisions in Manufacturing Engineering (ME) Department, the subject of the study, started adopting VCAF for less than a year ago, while other departments across Volvo Cars started adopting VCAF earlier. Hence, these departments need to understand the different state of art in their agile adoption to plan better roadmap in adopting agile in large-scale development.

1.2 Purpose

This thesis aims to study the state of art of agile adoption at Volvo Cars, particularly within the Manufacturing Engineering (ME) department. As the department transforms from traditional to agile setup, it is important to measure its maturity (Gren, Torkar, & Feldt, 2015). To date, for various reasons, divisions in the ME Department have different stages in their agile

adoption. Some divisions are still in the pilot project of adopting scaled agile, while others have become more established in their setting. The nature of the ME Department divisions combines hardware and software environments. It will be interesting to assess the relationship between the different environments with agile adoption and what factors hinder maturity.

1.3 Problem Analysis and Research Questions

As mentioned above, the divisions within the ME Department are at different stages of adopting scaled agile. To succeed in scaled agile adoption, it is necessary to understand their agile maturity (Gren et al., 2015). Thus, understanding these divisions scaled agile adoption is obtained by measuring their scaled agile maturity. Therefore, the first research question (RQ) is designed as follows:

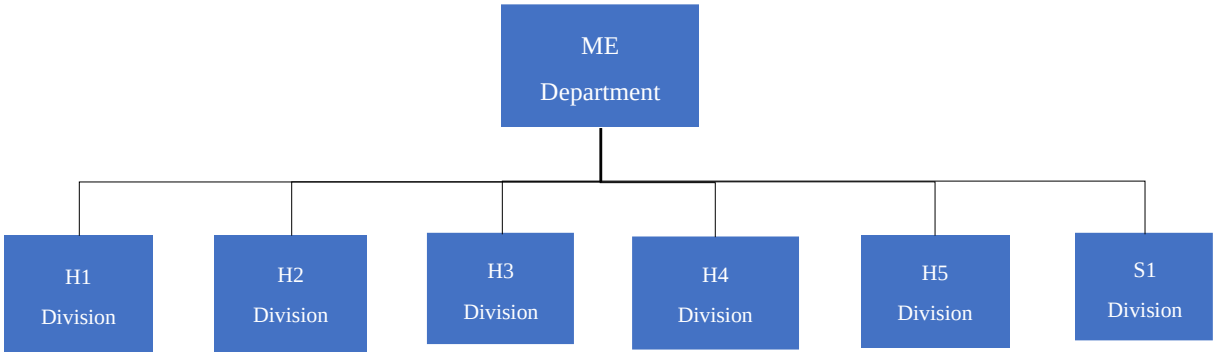
RQ 1: What is the measurement model and its metrics to assess the scaled agile maturity in manufacturing?

The proposed model and its metrics will be used to assess the agile maturity of each division. Apart from the different timeframe of adopting scaled agile, other hindrances may also inhibit the maturity. Therefore, the second research question is designed as follows:

RQ 2: What is the maturity of the current agile practice in each division, and what hinders the agile maturity?

1.4 Delimitations

The ME department consists of several divisions, and the thesis will be conducted only in 6 (six) divisions. The divisions' names are altered to enhance the clarity between two environments, software, and hardware. Figure 1 presents the organisational structure to provide brief information on the limitation of this thesis.



H1, H2, H3, H4, H5: divisions work in hardware development.
S1: division works in software development.

Figure 1 Organisation Structure of ME Department (source: Volvo Cars, 2020)

2 Theoretical Framework

In this chapter, we discuss related literature studies that set the foundation for our study. We begin by introducing the difference between waterfall and agile methodology, followed by agile in manufacturing and agile on a large scale. As Volvo Cars adopt Scaled Agile Framework (SAFe) and Volvo Cars Agile Framework (VCAF) in their scaled agile, the theory of these two frameworks also be described. In the end, we cover several Agile Maturity Models (AMMs) and their components.

2.1 Waterfall Model

For years, a successful project is defined based on time delivery, within budget, and performed agreed functionality and called the "triple constraint" (Pinto, 2016). The triple constraint may have been once the standard by which project performance was routinely assessed. However, these days, it is considered outdated and too simplistic in defining the success or failure of a project because it works only in an internal field. What was lost is the external part, characterized as the customer's acceptance, as projects intend to fulfil the customer's requirement (Pinto and Rouhiainen, 2002). With the new constraint "client acceptance," a new framework is formed, and the project can be considered a success only if it satisfies the customer as intended. This new framework is called quadruple constraint (Pinto, 2016), as seen in Figure 2.

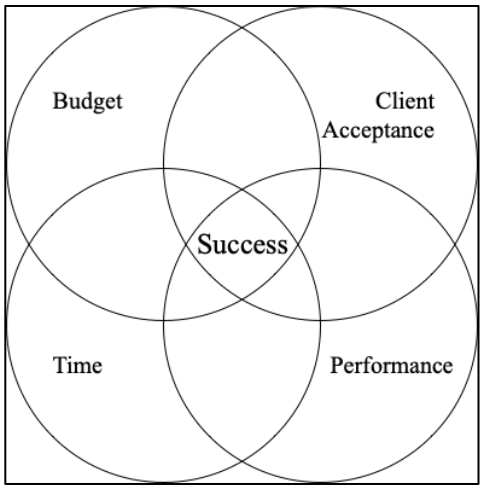


Figure 2 The quadruple constraint (Source: Pinto, 2016)

Due to the triple constraints, the traditional project management will always have rigorous and rigid upfront planning before the projects begin. The required resources and capabilities for the projects are planned and set, and therefore room for changes when they are progressing, unlikely to be expected.

Many traditional project managements, particularly in software development, followed a waterfall methodology in their implementation. The waterfall model introduced in the 1970s by Winston Royce is a sequential process signified by its name. The model provided the structure for the development to progress in order, stage by stage, as seen in Figure 3 (Royce, 1970).

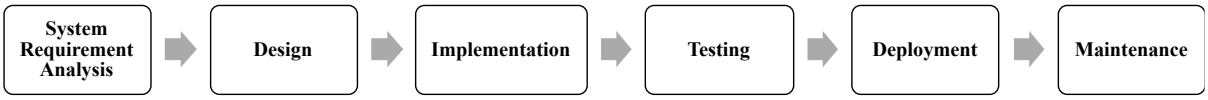


Figure 3 The Waterfall Model (Source: Royce, 1970)

However, having the new constraint "client acceptance" challenged this structured approach. In many cases, especially in software development, numerous factors affect client acceptance. These factors include technical advancement, business environment, or regulations (Serrador and Pinto, 2015). Changes resulting from these factors often occur in the middle of the project, and adjusting the established setup is often too expensive, if not too complicated. The waterfall model has failed to accommodate flexibility towards changes during the project development and motivates the search for new methodologies to cope with this limitation.

2.2 Agile Model

Agile emerged in the 1990s, launched by software developers as a response to the traditional approach. This agile method was set on the understanding that software development is not predictable, and therefore extensive upfront planning is not suitable. In the contrary, as changes are inevitable, they rely on people and the team's creativity rather than processes and heavy documentation, on small and incremental delivery than deliver at once (Boehm and Turner, 2005; Dybå and Dingsøyr, 2008). Agile was quickly accepted by software development as it offers small-frequent delivery, light-weight documentation, incremental and iterative process,

embrace changes, and include customers' feedback. Agile facilitates instead of eliminating changes and feedback in the ongoing project (Williams and Cockburn, 2003).

Originally in 2001, a group of practitioners named themselves "agile alliance" at Utah, set a foundation for agile, called "agile manifesto". The agile manifesto declared four values and 12 agile principles that became the backbones of both study and practices in agile development, as shown in Table 1 and Table 2, respectively. The declared value has two priorities at each point, such as individuals and interactions, working software, customer collaboration, and response to change as the first priorities. The 12 principles derived from values are the essential characteristics that should be adopted for an organisation that is going agile.

Table 1 Agile Values (Source: Agile Alliance, 2001)

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value:

1. *Individuals and interactions* **over** *processes and tools.*
2. *Working software* **over** *comprehensive documentation*.
3. *Customer collaboration* **over** *contract negotiation.*
4. *Responding to change* **over** *following a plan.*

That is, while there is value in the items on the right, we value the items on the left more."

-Agile Alliance, 2001

Table 2 Agile Principles (Source: Agile Alliance, 2001)

"We follow the following principles:

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. Businesspeople and developers 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 maximising the amount of work not done—is essential.
11. The best architectures, requirements and designs emerge from self-organising teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly."

-Agile Alliance, 2001

The study introduced agile practices as the manifest of agile principles and defined as techniques and activities deployed in software development in a manner that reflects and is consistent with agile principles (Sidky, Arthur, & Bohner, 2007). Agile practices embrace changes to deliver value to customers, such as iteration based on customer feedback or evolutionary requirements (Larman, 2004). Other agile practices emerge from facilitating knowledge sharing, such as pair programming, co-location, daily stand up, and minimal documentation (Law and Charron, 2005). There is a high diversity of agile practices that are currently used in the industry because some of the practices developed from other disciplines

to meet the needs of working agile. In general, agility is associated with the agile manifesto and attributes such as *flexibility*, *learning*, *speed*, *leanness*, and *responsiveness* (Qumer and Handerson-Sellers, 2008).

For a deeper understanding of waterfall and agile methods, Dybå and Dingsøyr (2008), in their empirical study of different literatures regarding Agile, presented the main differences between the waterfall and agile, as depicted in the following Table 3.

Table 3 Main differences between Waterfall and Agile Development (Source: Dybå and Dingsøyr, 2008)

	Traditional development	Agile development
Fundamental assumption	Systems are fully specifiable, predictable, and are built through meticulous and extensive planning	High-quality adaptive software is developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change
Management style	Command and control	Leadership and collaboration
Knowledge management	Explicit	Tacit
Communication	Formal	Informal
Development model	Life-cycle model (waterfall, spiral or some variation)	The evolutionary-delivery model
Desired organizational form/structure	Mechanistic (bureaucratic with high formalization), aimed at large organizations	Organic (flexible and participative encouraging cooperative social action), aimed at small and medium-sized organizations
Quality control	Heavy planning and strict control. Late, heavy testing	Continuous control of requirements, design and solutions. Continuous testing

2.3 Agile in Non-software Development

The expectation to imitate the success of software development with the agile model intrigued the adoption of agile in non-software development. Studies reported some benefits of adopting agile in non-software development: *teamwork, customer interaction, increasing productivity and speed* and *increasing flexibility toward changes* (Gustavsson, 2016). However, adopting agile processes in traditional development raises its challenge due to the contradictive nature of traditional and agile development (Boehm and Turner, 2005). Moreover, when adopting agile in non-software development, companies found it difficult to nurture flexibility, lack of process visibility and leadership buy-in (Gustavsson, 2016).

2.3.1 Agile in Manufacturing

With a fast product cycle in the changing market, manufacturing could suffer from the slow response to such dynamics due to their traditional development process (Cooper and Sommer, 2018). The traditional development aims for optimization over a longer period while agile values short and iterative development (Boehm and Turner, 2005).

The advent of agile software development in manufacturing is gaining prominence and arising the needs for synchronization with the rest of the system (Boehm and Turner, 2005). Hence, agile in hardware development within manufacturing is inevitable. However, when it comes to the question of how to adopt agile in manufacturing that combines hardware and software, we argue that agile suits better on product development than product production process.

Iteration process is often used as an example in research studies to highlight the difference of software and hardware development (e.g Laanti, 2016; Cooper, 2016). While software development fluently utilizes agile iteration, hardware is not that flexible (Laanti, 2016). Hardware development often needs longer time than the typical sprint period and requires heavy resources in its iteration process (Punkka, 2012; Cooper and Sommer, 2018). Agile emphasizes frequent iterations for the effective and efficient learning process, and this process can be costly due to different cycles in prototyping processes (Punkka, 2012). Although it could add the cost of the project, one can argue that frequent iterations are aimed at avoiding higher losses at the end of the project (Punkka, 2012).

Therefore, it is vital to find a consensus in adopting agile in manufacturing that still suits hardware and software development settings to improve productivity, faster time to market and innovative offers (Cooper and Sommer, 2018). Cooper (2016) proposed combining agile and traditional development to get the benefit from these two approaches. In his proposal, Cooper (2016) suggested that agile methodology be applied only in the stages related to technical development phases within traditional development. This agile adoption in the stage-gate method for product development has proven to promote positive results (Cooper, 2016).

2.4 Agile in Large-scale Development

Research studies have defined the scaled agile development from different focuses, such as the number of people involved in the development, the number the teams, or the projects' size (Dingsøyr and Moe, 2014). A study by Dingsøyr and Moe (2014) analysed the definition of scaled agile from the workshop participants at XP2014 and defined the scaled agile as an agile development with more than two teams.

The success of agile methods motivated an organisation to implement agile on a larger scale (Dingsøyr and Moe, 2014). However, agile in a larger organisation poses new challenges. For

instance, large organisations tend to have interdependencies between project and team and entail additional communication and coordination (Dikert, 2016). Meanwhile, agile at a large-scale development is a collective action and usually means to change the organisation's culture (Misra, 2010).

As the agile method was initially intended for small scale, scaled agile needs to be adjusted. Different frameworks have been developed to scaling up agile practice in large development, such as Large-Scale Scrum (LeSS), Scaled Agile Framework (SAFe) and Disciplined Agile Delivery (Paasivaara, 2017). This study utilises SAFe because it is the foundation of Volvo Cars Agile Framework (VCAF) and has been acknowledged as a widely used scaled agile framework in the study by Moe and Dingsøyr (2017).

2.4.1 Scaled Agile Framework

Scaled Agile Framework (SAFe) is a framework designed for a larger organisation to achieve business agility by incorporating Lean, Agile, and DevOps (Scaled Agile Inc., 2021). Since its first introduction in 2007, SAFe has been refined to the latest version, SAFe 5, in 2020.

A larger organisation often grounds itself with its existing business model, hierarchy structure, and inertia in technological infrastructure, making it challenging to keep up with the rapid changes (Scaled Agile Inc., 2021). Therefore, SAFe emphasises the importance of business agility for the organisation to stay competitive. SAFe helps organisations adapt and respond to competition threats, identify and deliver incremental value for customers, and sustain the quality of deliverables (Scaled Agile Inc., 2021).

On SAFe 5, there are four different implementation scenarios, which are Full SAFe, Portfolio SAFe, Large Solution SAFe, and Essential SAFe (Scaled Agile Inc., 2021). These scenarios are divided based on the level included in each scenario. Figure 4 shows the level incorporated in each scenario. For example, Large Solution SAFe consists of solution and essential levels. Complete building blocks of SAFe are shown in Figure 5.

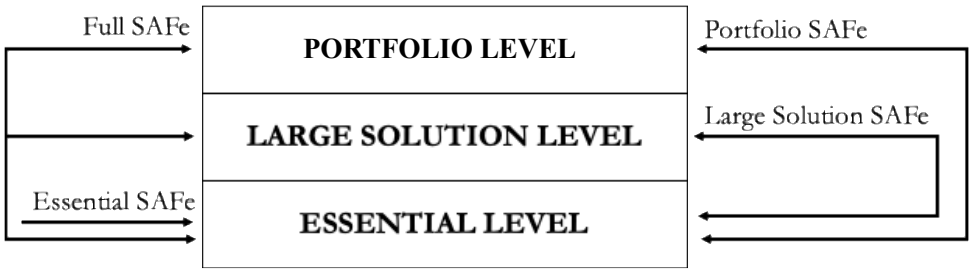


Figure 4 SAFe 5.0 level (Adopted from: Scaled Agile Inc., 2021)

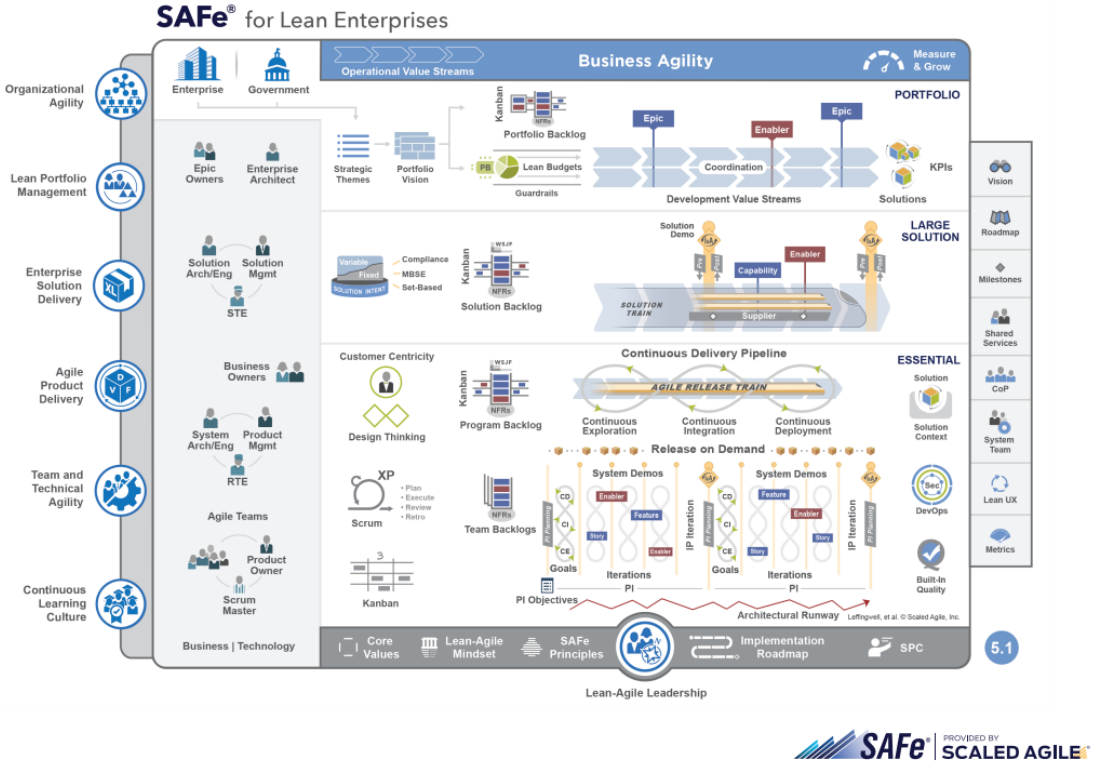


Figure 5 SAFe building blocks (Source: Scaled Agile Inc., 2021)

Each level of SAFe contains a set of roles and practices:

- At the Essential level, Agile Release Train (ART) is the centre of the level, consisting of teams of agile teams. There are two roles at this level, team and ART. Team consists of agile team, product owner and scrum master, who conduct the iteration process of planning, execution, review, and respective. ART consists of several agile teams, product management, system architect/engineering, release train engineer, and business owners, with activities such as program increment (PI) planning and scrum of scrums.

- At the large solution level, the solution train concept is used to describe a set of multiple ARTs. Three main roles introduced to help coordinate multiple ARTs: Solution architect/engineering, solution management, and solution train engineer, with one of the major activities, is pre- and post-PI planning.
- At the portfolio level, lean portfolio management is introduced to align the organisation's strategy and allocate investment and resources.

2.4.1.1 *SAFe Core Values*

SAFe consists of four core values important for the framework's effectiveness: *Alignment*, *Built-in quality*, *Transparency*, and *Program execution* (Scaled Agile Inc., 2021). *Alignment* is related to the direction toward a common goal, but it does not entail command or top-down control (Scaled Agile Inc., 2021). *Built-in* quality corresponds to assurance to meet the quality standards for each solution throughout the development lifecycle (Scaled Agile Inc., 2021). *Transparency* is core to build trust and promotes openness to build high-performance teams (Scaled Agile Inc., 2021). The first three values help last value, *program execution*, to continuously deliver value through increment solution (Scaled Agile Inc., 2021).

2.4.1.2 *SAFe Mindset and Principles*

SAFe underlines the importance of pursuing a growth mindset, in which people can learn new behaviors (Scaled Agile Inc., 2021). In addition to the mindset, SAFe is built upon ten principles, as listed in Table 4 (Scaled Agile Inc., 2021).

Table 4 SAFe Principles (Source: Scaled Agile Inc., 2021)

No	Principle	Description
1.	Take an economic view.	Delivering value for people and community by putting economic perspective, such as operational and development cost.
2.	Apply systems thinking.	Improve the components of the complex system in the SAFe.
3.	Assume variability; preserve options.	Being adaptable for changing and open for different possibilities.

4.	Build incrementally with fast, integrated learning cycles.	Promotes a short time-box development for faster feedback loop from the customer.
5.	Base milestones on objective evaluation of working systems.	Regular evaluation during the life cycle development to ensure financial, technical, and fitness-for-purpose governance in the investment.
6.	Visualise and limit work in process (WIP), reduce batch sizes, and manage queue lengths.	Achieve a continues flow from concept to revenue (or benefit).
7.	Apply cadence, synchronise with cross-domain planning.	Alignment for effective operation in uncertain development.
8.	Unlock the intrinsic motivation of knowledge workers.	Creating better employee engagements, such as through promoting a mutual environment for the employee.
9.	Decentralize decision-making.	Promotes fast deliveries by reduces decision-making's bureaucracy.
10.	Organize around value.	Develop an organisation around changing value flow instead of traditional principles around functional expertise.

2.4.2 Volvo Cars Agile Framework

Volvo Car Agile Framework (VCAF) is a framework used by Volvo Cars in their agile adoption with SAFe as the foundation. However, VCAF is not completely the same as SAFe, as it has been adjusted to fit with Volvo Cars context. For example, VCAF introduces a new role related to procurement position in the large solution level, as an addition to the original SAFe roles: solution architecture/engineer, solution management, and solution train engineer.

Following VCAF as the guideline, some departments have adjusted their structure to better fit the context in the agile ways of working. The nature of work at Volvo Cars organisation varies across departments; for example, the different nature works between software and hardware

developments. Thus, different departments in Volvo Cars might implement VCAF in different ways.

2.5 Agile Maturity Models

Leppänen (2013) described maturity as a situation when a specific subject is ready, complete and in the perfect state. However, different terminologies are used to define agile maturity. Some refer to the level of agile adoption that includes both agile principles and values (Sidky et al., 2007; Leppänen, 2013). Schweigert, Vohwinkel, Korsaa, Nevalainen, & Biro (2013) concluded that agile maturity refers to the adoption of agile practices in the organisation or project management.

The word "adoption" and "implementation" often being used interchangeably in describing agile maturity (ex: Fontana, R. M., Fontana, I. M., da Rosa Garbuio, Reinehr, & Malucelli, 2014). This paper uses agile adoption to emphasise that the process is not only "copy-paste" the established agile setup but also "incorporate" and "adjust" the agile principles and values to the current process.

For an organisation to be fully mature in their agile adoption, a transformation is required, and it takes time and effort. Agile transformation is described as "*an ongoing, dynamic effort to develop an organisation's ability to adapt rapidly within a fast-changing environment and achieve maximum business value by engaging people, improving processes, and enhancing culture* (PMI, 2017)."

Both academia and practitioners agreed to measure maturity, assess their current state, and see which areas need improvement. The maturity models usually consist of levels, descriptions of goals in each level, dimensions, and activities or practices tied to each level and dimension (Leppänen, 2013). Numerous agile maturity models (AMMs) emerged to provide guidelines during the agile adoption for the organisation. 40 AAMs introduced in literature, and the researchers have conducted studies to assess these different models and their relevancy to the industry context (Schweigert et al., 2013; Leppänen 2013; Ozcan-Top and Demiros 2013, Gren et al., 2015). All the studies do not reject the use of AMMs; however, criticisms are suggested for future research.

Ozcan-Top and Demiros (2013) assessed different AMMs according to these following criteria: *fitness for purpose, completeness, the definition of agile levels, objectivity, correctness, and consistency*. They assessed the agile maturity of a software company using all the models studied. Their study presented three models marked as "fully achieved" in at least one of the indicators, which are M1 (Agile Maturity Model by Patel and Ramachandran), M2 (Scrum Maturity Model by Yin, Figuirdedo & Da Silva), and M3 (Agile Adoption Framework, known as Sidky Agile Measurement Index). The assessment results in Table 5 shows that M3 (AAF) by Sidky et al. (2007), has the highest score (Ozcan-Top and Demiros, 2013).

Table 5 AMMs Assessment Result (Source: Ozcan-Top and Demiros, 2013, p140)

Criteria / Models	Fitness for Purpose	Completeness	Definition of A. Levels	Objectivity	Correctness	Consistency
M1 (AMM)	**Fully Achieved**	Partially Achieved	Not Achieved	Largely Achieved	Partially Achieved	Not Achieved
M2 (SMM)	Largely Achieved	Partially Achieved	Partially Achieved	Partially Achieved	**Fully Achieved**	**Fully Achieved**
M3 (AAF)	Largely Achieved	Partially Achieved	**Fully Achieved**	Largely Achieved	**Fully Achieved**	**Fully Achieved**
M4 (BM)	Largely Achieved	Partially Achieved	Not Applicable	Not Applicable	Not Applicable	Not Applicable
M5 (ASM)	Partially Achieved	Partially Achieved	Partially Achieved	Not Applicable	Not Applicable	Not Applicable

Inspired by the study of Ozcan-Top and Demiros (2013), we conduct a literature study of these three AMMs, to have a closer look at the models and their attributes.

2.5.1 Agile Maturity Model

Patel and Ramachandran (2009) created the Agile Maturity Model (AMM), aiming to enhance agile adoption in software development. Patel and Ramachandran (2009) defined the maturity path of their model from the Capability Maturity Model Integration (CMMI). This is a process improvement approach in software development, consists of Software Process Improvement (SPI) practices to be implemented in agile software development environments from the stage of commencement until the sustained maturity (CMMI Product Team, 2010). The AMM measures the agile maturity following the CMMI maturity path, with each level has goals and key process areas and entailed agile software development practices to focus on, as shown in Figure 6.

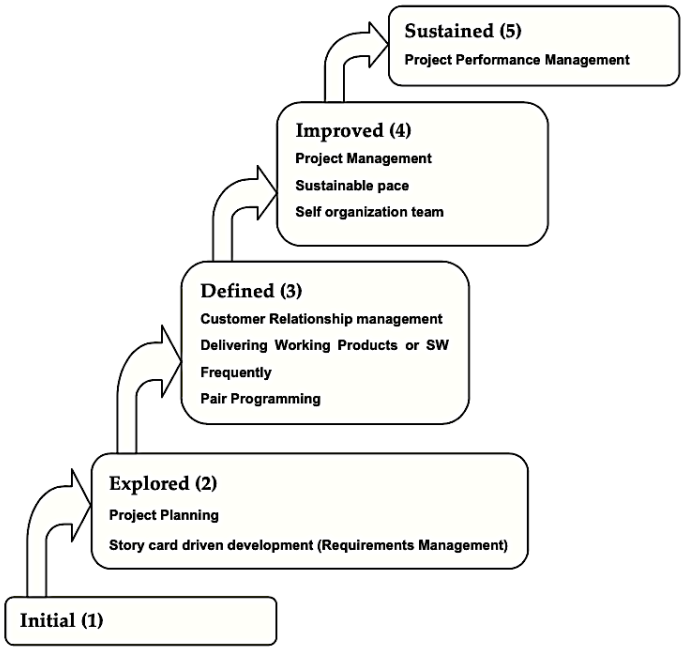


Figure 6 Agile Maturity Model (Source: Patel and Ramachandran, 2009)

Level 1 - Initial: At this level, there is no process improvement goal (Patel and Ramachandran, 2009). The organisation employs the traditional approach to their software development, and the work performs individually rather than in small teams.

Level 2 - Explored: There are two goals at this level: project planning (including monitoring the project progress) and requirement management for engineering driven by the story card from the customer (Patel and Ramachandran, 2009). The assessment of these goals is defined from a questionnaire with 27 SPI practices such as estimation of project's scope, customer or business representative present or invited to all the team estimation session, user stories written for the requirement engineering.

Level 3 - Defined: There are three goals at this level: customer relationship management, delivering working products or software frequently, and pair programming (Patel and Ramachandran, 2009). The organisation focuses on good customer relationship management by improving the test-driven development (TTD) through practices such as pair programming, coding, and testing, frequent software delivery and standards for coding. The assessment of these three goals is defined by a questionnaire with 43 agile software development practices.

Level 4 - Improved: There are three goals at this level: project management, sustainable pace, and self organizing team (Patel and Ramachandran, 2009). This level is called the improved level due to its people and project management focus. It emphasises the self-organising team, improving the code optimisation planning and risk, and assessment of these goals is defined by a questionnaire with 23 agile software development practices.

Level 5- Sustained: This is the mature level, where the organisation focuses on continuous improvement of their process. There are four goals at this level: context improvement, uncertainty management, tuning project performance and defect prevention (Patel and Ramachandran, 2009). This level addresses the customer and development team's satisfaction and the assessment of these four goals is defined by a questionnaire with 12 agile software development practices. Figure 7 shows the example of how AMM is structured at level 5, where goals entitled to key process areas, practices, and metrics. Same structure applied at other levels, with different content.

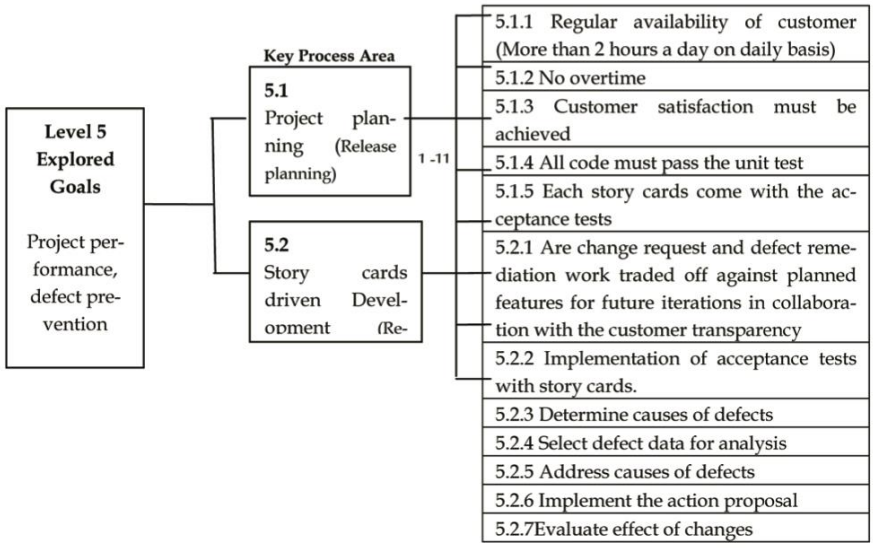


Figure 7 Goals, Key Process Area and Practices at Level 5 (Source: Patel and Ramachandran, 2009)

The AMM is self-assessment and based on four maturity assessment criteria (Patel and Ramachandran, 2009).

- Not Achieved (NA) denotes little or no evidence of achievement of the assessed practice.

- Partially Achieved (PA) denotes the presence of evidence of a convincing systematic approach towards, and a few achievements of the assessed practice.
- Largely Achieved (LA) denotes the presence of evidence of a convincing systematic approach towards, and significant achievements of the assessed practice.
- Fully Achieved (FA) denotes the presence of evidence of a complete and systematic approach towards, and full achievements of the assessed practice.

The model has been validated by assessing three different software companies, however, the authors still suggested evaluation for this model (Patel and Ramachandran, 2009).

To conclude, this AMM has developed its five maturity levels on the SPI practices with a business and good management orientation. However, it has very few agile practices and solely focuses on the software development process. Furthermore, although the authors structured the levels and their goals after the CMMI-DEV, we could not find the argument behind the organisation of the practices at each level. To put this in perspective, several practices related to planning are set at level 3 (customer satisfaction oriented), instead of at level 2 (planning oriented). In addition, the focus on empowering the team is instated at level 4, where we see that it should be at level 2 because the core of agile is the self-organised team that work iteratively in providing value for their customers. These levels were structured more on the software process rather than agile values and principles.

2.5.2 Scrum Maturity Model

Yin, Figueiredo, & da Silva (2011) developed Scrum Maturity Model (SMM) to improve software development based on Scrum practices, emphasising collaboration with clients as the main success factor of a software development project. Yin et al. (2011) further noted that the model aids to provide a roadmap for the organisation that is not familiar with Scrum to adopt this methodology and improve their software development process. Capability Maturity Model Integration (CMMI) inspires the maturity path of SMM, and therefore the model has five maturity levels, with different goals, objectives and scrum practices entitled to each level.

Level 1 - Initial: Organisation at this level has no goal and often faces problems such as overtime, over-budget, poor communications, and inadequate final product's quality (Yin et al., 2011).

Level 2 - Managed: There are two goals at this level: *basic scrum management* and *software requirements engineering* (Yin et al., 2011). Therefore, the organisation is argued to be more structured than at level 1. Basic scrum management directs objectives such as scrum roles, artifacts, and ceremonies despite being used correctly. Software requirements engineering comprises a set of practices to ensure the quality of the final product, such as product backlog and sprint planning. The assessment of these two goals is measured with six scrum objectives.

Level 3 - Defined: This level focuses on customer relationship and deliveries as scheduled; therefore, the two goals are: *customer relationship management* and *iteration management* (Yin et al., 2011). Customer relations management comprises practices emphasizing the definition of "done" for the project and sprint review. Iteration management comprises practices such as sprint backlog and planned terations. The assessment of these two goals is measured with six scrum objectives. The organisation at this level could have some successful projects, but to maintain the consistentency, there is a need for improvement by incorporating a standardized management processes.

Level 4 - Quantitatively Managed: This level aims to provide a standard for the project and processes managements. It has two goals: *standardized project management* and *process performance management* (Yin et al., 2011). Standardized project management provides the same standard for all project management to eliminate the deviation and enable the same quality and performance level for all the projects. Process performance management requires a measurement of all suggested practices and improvements in these practices performance. The assessment of these two goals is measured with two scrum objectives. With the standard in place, the organisation should have more successful projects, and the only thing left is to optimize the processes.

Level 5 - Optimizing: Organisation at this level focuses on continuous improvement and higher satisfaction levels of the customers and all the stakeholders, and the only goal is *performance management* (Yin et al.2011). Performance management enables the organisation to measure its performance and take self-improvement actions. The assessment of these two goals is measured with four scrum objectives. Table 6 is an example of how SMM is structured at level 3, where goals entitled to objectives, practices, and metrics. The same structures apply at other levels, with different content.

Table 6 Goals and Objectives at SMM level 3 (Source: Yin et al., 2011)

Goals	Objectives	Practices	Suggested Metrics
Basic Scrum Management	Scrum roles exist Scrum artifacts exist	(...)	(...)
	Scrum meetings occur and are participated	(...)	(...)
	Scrum process flow is respected	(...)	(...)
Software Requirements Engineering	Clear definition of Product Owner	(...)	(...)
	Product Backlog Management	(...)	(...)
	Successful Sprint Planning Meetings	(...)	(...)

This model was evaluated by interviewing two Scum Masters who are experts in agile and CMMI (Yin et al., 2011). The validation was conducted by having project managers from three different Information Technology (IT) companies filled the self-assessment questionnaire. However, there is no explanation of how the maturity assessment was measured including the criteria.

To conclude, this SMM provides a roadmap for the organisation to adopt scrum methodology in their software development with its five maturity paths. Although it claimed to focus on improving customer satisfaction, the levels also make recommendations to improve the project and process management by including standards and measuring performance. The authors have been clear when they structured the goals as it follows CMMI, but we think they are more process-oriented and missed these agile value: *individuals* and *adaptive to changes.* This SMM is based on scrum context and lacks technical practices. The authors do not clearly distinguish the objectives and scrum practices. Some of the practices are in the objectives; therefore, it will be harder for the organisation to differentiate the practices mixed up with the objectives.

2.5.3 Sidky Agile Measurement Index Model

Sidky et al. (2007) argued that agile maturity is influenced by three things: agile practices, the teams, and the environment. Therefore, they introduced the Sidky Agile Measurement Index (SAMI) model to measure these three factors. They structured the models into five maturity

levels built on the core of the agile manifesto. They divided 12 agile principles into five essences. Lastly, they populated the five levels and five essences with 40 agile practices identified in literature. The complete SAMI model shown in Table 7.

Level 1 - Collaboration: The goal at this level is to foster communication and collaboration between the stakeholders (Sidky et al., 2007), derived from agile value: *collaboration and interaction*, set as the foundation for agile. There are eight agile practices at this level.

Level 2 - Evolutionary: The goal is to improve the process in the evolutionary nature of the developed software (Sidky et al., 2007). Working agile aims for continuous improvement and evolutionary approaches as the result of the changes in the progress is natural. There are seven agile practices at this level.

Level 3 - Effectiveness: The goal is to increase the effectiveness of software development by incorporating engineering practices (Sidky et al., 2007) and preparing the development process to embrace more changes to come without taking risks to the process. There are nine agile practices at this level.

Level 4 - Adaptive: The goal is to build capability to respond to change through multiple levels of feedback mechanism and tracking for the emerging requirements (Sidky et al., 2007). There are nine agile practices at this level.

Level 5 - Encompassing: The organisation is assumed mature, so the goal is to establish the culture and environment that sustain agile in the organisation (Sidky et al., 2007). There are seven agile practices at this level.

The five essences of the agile principles in SAMI (Sidky et al.2007) consist of:

- embrace change to deliver customer value,
- plan and deliver software frequently,
- human-centric,
- technical excellence,
- customer collaboration.

Table 7 Structure of SAMI Model (Sidky et al., 2007)

	Agile Principles				
	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 [33, 39]	Agile project estimation [20]	Ideal agile physical setup [33]	Test driven development [11] Paired programming [49] No/minimal number of level -1 or 1b people on team [17, 15]	Frequent face-to-face interaction between developers & users (collocated) [12]
Level 4 **Adaptive** *Responding to change through multiple levels of feedback*	Client driven iterations [33] Continuous customer satisfaction feedback [35, 43]	Smaller and more frequent releases (*4-8 weeks*) [35] Adaptive planning [33] [20]		Daily progress tracking meetings [6] Agile documentation [40, 31] User stories [21]	Customer immediately accessible [15] Customer contract revolves around commitment of collaboration [26, 35]
Level 3: **Effective** *Developing high quality, working software in an efficient an effective manner*		Risk driven iterations [33] Plan features not tasks. [20] Maintain a list of all features and their status (backlog) [31]	Self organizing teams [33, 39, 31, 18] Frequent face-to-face communication [39, 18, 13]	Continuous integration [33] Continuous improvement (refactoring) [31, 12, 24, 5]. Unit tests [28] 30% of level 2 and level 3 people [17, 15]	
Level 2: **Evolutionary** *Delivering software early and continuously*	Evolutionary requirements [33]	Continuous delivery [33, 31, 26, 12] Planning at different levels [20]		Software configuration management [31] Tracking iteration progress [33] No big design up front (BDUF) [4, 12]	Customer contract reflective of evolutionary development [26, 35]
Level 1: **Collaborative** *Enhancing communication and collaboration*	Reflect and tune process [35, 43]	Collaborative planning [39, 18, 33]	Collaborative teams [46] Empowered and motivated teams [13]	Coding standards [29, 48, 36] Knowledge sharing tools [33] Task volunteering [33]	Customer commitment to work with developing team [13]

SAMI was evaluated with feedback survey by 35 participants from agile community, who have various industry background, role, and years of experience in agile, thus having knowledge about agile adoption and sufficient representation (Sidky, 2007). Sidky (2007) proposed validation of this model by a longitudinal study to gather evidence and empirical benefits of the adopted agile practices.

The maturity assessment is rated by following criteria (Sidky, 2007):

- Not Achieved (NA) denotes little or no evidence of achievement of the assessed practice.
- Partially Achieved (PA) denotes the presence of evidence of a few achievements of the assessed practice.
- Largely Achieved (LA) denotes the presence of evidence of a significant achievement of the assessed practice.
- Fully Achieved (FA) denotes the presence of evidence of a full achievement of the assessed practice.

To conclude, SAMI measures an organisation's agile maturity by the numbers of the agile practices they used, which means the organisation that used more agile practices is more agile than the organisation with fewer practices (Sidky, 2007). There are no sequential levels for the maturity path, contrary to other maturity models that are based on CMMI. However, Sidky (2007) believed that the practices' achievement at the lower level will help lay the foundations for higher levels. We agree that SAMI has exhibited the agile manifesto as the backbone of the agile methodology in their model and added technical practices to cope with software development.

2.5.4 Analysis of Studied AMMs

The previous literature study of three different AMMs gives us an overview of each model's attributes and how the maturity is measured. In this section, we compared and analysed the three models to identify and formulate foundations that can be combined or further developed to meet the needs of our study. Despite the claims of how the three different AMMs have been developed to measure the agile maturity model, there have been sentiments worth noting regarding the models' insufficiency to measure agility. Following is the analysis explained in detail.

2.5.4.1 *Agile Manifesto as the Backbone*

Fowler and Highsmith (2001) emphasised agile values, principles, and practices as agile building blocks. The lack of explicit explanation relates to these dimensions have been criticised in the CMMI-based maturity model (Leppänen, 2013). The assessment of different

AMMs by Ozcan-Top and Demiros (2013) used *correctness* as one of the criteria to highlight the alignment of the model with agile principles where SAMI by Sidky et al. (2007) and SMM by Yin et al. (2011) got "fully achieved" mark on this criterion. Even though SMM by Yin et al. (2011) has a similar structure with CMMI, they included scrum-agile practices to build the model (Ozcan-Top and Demiros, 2013). Therefore, we see that it is important to ensure agile value, principles, and practices as the backbone of the agile maturity measurement.

2.5.4.2 *Staged Levels and Predefined Systems are Waterfall Approaches*

Two studied models based their maturity path on CMMI, a maturity model in software development that aims to measure process maturity and capability requirements. CMMI maturity path is a waterfall approach and does not reflect agility. The staged levels and the predefined systems heavily regulate how maturity progresses from one stage to another and seen as the hindrance for agility, which one of its highlight is adaptability. A criticism from Schweigert et al. (2013) was that capability maturity is about increasing efficiency and risk mitigation, hence this type of measurement using one dimensional scale, while agile is like a spider web that does not have one dimensional scale. Other criticism came from Leppänen (2013) that the capability maturity path is plan-driven development, while agile development is emerging and responsive to changes.

2.5.4.3 *Different Priority in Structure of Levels and Goals*

AMM by Patel and Ramachandran and SMM by Yin et al. are both based on the CMMI maturity model. However, there are differences in how they structured the level description, objectives, and goals. Therefore, it was difficult to regard both models' maturity levels as references due to inconsistency. Moreover, there is no explanation from these models on how they structured their goals and levels. To put it in context, AMM by Patel and Ramachandran listed a self-organised team as one of the goals at level 4 (improved - a mature level). We see that agile started with the small-organised team and is the foundation of agile maturity. Therefore, it should be in level 2 (explored–the initial maturity level) rather than at a higher level.

2.5.4.4 *Process-oriented*

One purpose of CMMI is to measure process maturity, therefore, mainly process-centric. Fontana et al. (2014) argued that agile includes subjective value, such as flexibility, communication, and self-organising. Consequently, they underline the importance of infusing such values in measuring agile maturity. These indicators are not about the process but about people and culture, which are not incorporated in the CMMI-based model.

2.5.4.5 *Context-dependency*

We argue that the three studied models are context-dependent, as they were created based on software development. One example is the model by Sidky et al. (2007) that has *Plan and Deliver Software Frequently* as the agile principle. This principle cannot be directly used in non-software development, one reason is due to the distinct functions and different standards between hardware and software (Gren et al., 2015). Ozcan-Top and Demiros (2013) also noted this dependency problem where a model incorporates a specific agile method. One example, SMM (Yin et al., 2011) only based on the scrum method. We see that, as agile underlines the importance of being adaptive to changes, AMMs should have a broader perspective that enables the adoption of agile in different contexts. Agile teams are expected to be flexible and self-organised. Thus predefined-narrow practices in measuring agile maturity do not align with the goals (Fontana et al., 2014). Therefore, measuring agile maturity should consider various contexts and still versatile enough to allow adjustment when necessary.

2.5.5 Conclusion from the Studied AMMs

After analysing the studied models and the related literature, we conclude that our agile maturity model should be based on the agile manifesto as the backbone, otherwise it will be considered a generic maturity model. The measurement shall not take a waterfall approach with staged level and predefined system.

Furthermore, there has to be a clear explanation behind the organisation of the levels and their objectives that align with the agile manifesto. The agile maturity model should not be process-oriented, as agile values people and interaction with process and response to change over the iterative plan. As it serves as a guideline for agile adoption, the indicators should not fit into specific context and have room to accomodate changes. Finally, the focus of our study is scaled

agile in manufacturing, hence the value, principles and practices corresponding to these new additions should be incorporated in the model.

After analysing the three studied models, a new model is needed since aforementioned foundations are not fully covered in any of them.

Despite the criticism against AMMs as a waterfall approach, there is still a need to measure agile maturity from the practitioners (Gren et al., 2015). Hence, we consider these two aspects, that the model should not follow the waterfall approach and should be able to measure agile adoption and provide guidelines for the organisation. The interplay of these two aspects for our proposed model is to still have levels to demonstrate different practices reflecting the agile manifesto. These levels will not be used as the staged-maturity path. Instead, the maturity will be measured based on the number of indicators used. Hence the indicators can be spread in different levels and dimensions, not solely at the one-dimensional level. However, the utilisation of the lower level is indeed helping the organisation build a foundation in adopting the practices at the next level (Sidky et al., 2007).

3 Research Methodology

This chapter presents the methodology to answer the purpose of the study. It explains the strategy and approach, the process of data collection and data analysis, and finally the ethical consideration in the thesis study.

3.1 Research Strategy and Research Approach

As described by Bell, Bryman, & Harley (2018), a research strategy is defined as the plan on how to perform a study and it can be of either a quantitative or qualitative. For the conducted study, we apply a qualitative research strategy, where the emphasis is on analysing interpretations of words and other non-numerical data (Bell et al., 2018). This is viewed as the most appropriate strategy to answer the research questions since data will be collected through interviews and other literature sources. Furthermore, Bell et al. (2018) stated that a research approach defines the relationship between theory and research within the study. The approach used can be either deductive, inductive, or a combination of both. For the conducted study, the research approach is inductive since the theory results from the findings of the collected data. An inductive approach is often combined with a qualitative research strategy (Bell et al., 2018). Figure 8 depicts a detailed view of the procedures proposed for this study.

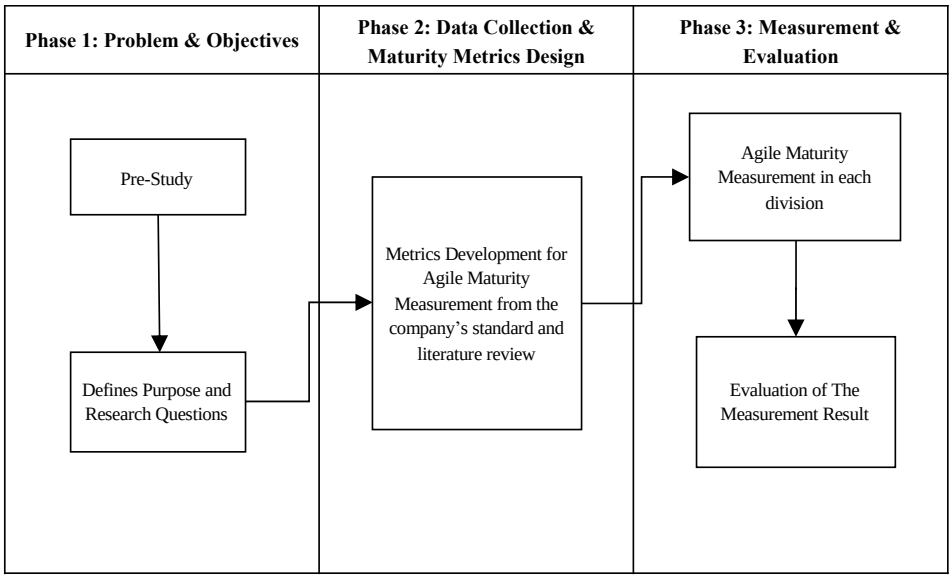


Figure 8 Procedure of the Study

In Phase 1, we identify the purpose and research questions through literature study and interviews. Proceeding into phase 2, we work with the Research Question 1 by developing Scaled Agile Maturity Model in Manufacturing through extensive literature study and several data collection. Phase 3 is the last procedure to answer the Research Question 2, where the assessment is conducted to analyse the agile maturity of each division and what hinders maturity.

3.2 Data Collection

The data collection is based on the literature study, interviews, self-completion questionnaire, and other relevant empirical data.

3.2.1 Literature Review

The literature review of this study is carried out in the field of traditional and agile project management, agile framework, and agile maturity measurement models. The literature review is needed to engage with previous studies and find relevance in the observed topic (Bell et al., 2018). We use the narrative review to understand the previous studies, which allows the researcher to generate the initial idea about the topic while allowing the possibility to conduct an iterative literature review in a broader scope (Bell et al., 2018). Such approaches allow better flexibility to modify research boundaries as the research progresses, which can be more suitable for inductive research (Bell et al., 2018). The literature review is conducted back and forth throughout the research study. The purpose and research questions of this study are developed from the relevant theory from the literature review. The relevant theory is further used for RQ 1 to build a model that suits the purpose of this study.

3.2.2 Interviews

The semi-structure interviews will be held with key persons within the ME department and corporate office. The semi-structured interview is used due to the increased flexibility in the interview process, where the interview has several predetermined questions while giving the interviewee the possibility to go beyond those questions to expand on their answers (Bell et al., 2018). Bell et al. (2018) described that it is important to adopt an appropriate interview structure to obtain valid information. Later the interviews are analysed with the literature

review to answer the research question. The intended information gathered from the interviews is related to Volvo Cars standards or guidelines and each division's agile practices.

3.2.3 Self-completion Questionnaire

The self-completion questionnaire, or sometimes referred to self-administered questionnaire, is a research method where the participants are required to answers the questions by themselves (Bell et al., 2018). We conduct two different self-completion questionnaires in this study. The first questionnaire is for the agile experts to evaluate the initial Scaled Agile Maturity Model, and the second is for self-assessing their division scaled agile maturity.

The first questionnaire comes from a research method of documenting and collecting data from daily life, enabling the participants to relate their experience and opinions to a specific context of the questionnaire (Eckerdal and Hagström, 2017). Some criticisms are addressed to this method, especially due to the less direct interaction between the participants and the researcher. For example, the researcher needs to include detailed instruction on answering the questions and the inability to directly ask a follow-up question (Eckerdal and Hagström, 2017). To address the issues, we conduct meetings with all the participants before filling the questionnaire. Furthermore, two rounds of qualitative questionnaires are conducted for the same participants with clarification sessions during and after each round when is needed. This method is utilized to gather data from the company's experts when evaluating the initial scaled agile maturity model.

The second questionnaire is built based on Bell et al. (2018), where they described four types of variables in data analysis:

1. Nominal/Categorical: where the data cannot be ranked and can only be categorized, such as gender, employment status, sport preferences. Data in this category is mutually exclusive.
2. Ordinal: where data can be categorized and ranked, such as level of agreement (strongly disagree, disagree, neither agree nor disagree, agree, strongly disagree), level of satisfaction (very dissatisfied, dissatisfied, neither dissatisfied nor satisfied, satisfied, very satisfied). Data in this category has a natural order, such as very dissatisfied is

worse than dissatisfied, very satisfied is better than satisfied. However, the distance between rank is unknown.

3. Interval: where data can be categorized, ranked, and evenly spaced, such as temperatures (Celsius, Fahrenheit). Data in this level has equal and known distance, and there is no true zero in interval data that differentiate it from ratio, for example, 0 deg Celsius.
4. Ratio: where data can be categorized, ranked, and evenly spaced and has true zero, such as number of families with kids in a region, years of work experience, numbers of cars own in a household.

The second questionnaire aims to measure the degree of maturity based on numbers of agile practice adopted by the team; hence the questionnaire will be designed to have ordinal output using a 5-points Likert scale. The maturity assessment is a self-questionnaire filled by the members in each division. Some questions are to be answered only by certain roles.

3.2.4 Other Empirical Data

Other information, for example, from internal documents such as organisation structure and job description, will help us understand the current way of working within and among the divisions. Documentation is a essential part of understanding the current situation and drawing an overall picture of the problem (Davidson and Patel, 2003). Information such as documented standards and documented agile retrospective will be used to answer the research questions.

3.3 Data Analysis

Extensive literature related to agile and SAFe is compared with the current practices and agile-related documents in Volvo Cars to develop a model that measures scaled agile maturity in the manufacturing. To evaluate the initial model from the literature review, we gather feedback from the selected agile experts in ME Department by asking them to contemplate the proposed practices in the model with both their knowledge and experiences in the manufacturing context.

According to Bell et al. (2018), there are no clear guidelines for analysing qualitative data. Instead, the process is often an iterative approach, whereas the data collection and analysis simultaneously occur. Therefore, the finding from qualitative data needs to be sorted to understand the phenomena. Findings from qualitative research are classified to understand the

pattern and generated what considered to be the theme (Bell et al., 2018). Transcribed data are codified based on their similarities to generate first-order concepts, in which themes are derived from the first-order concepts based on their relevance (Bell et al., 2018).

For the second questionnaire with ordinal data using the Likert scale, the analysis will be based on data central of tendency by measuring the median. Measuring central tendency helps to identify one value that summarizes a distribution of values and classified into mean, median or mode (Bell et al., 2018). Median is the midpoint in the data distribution, and used in ordinal data (Bell et al., 2018). The outcome of the 5-points Likert scale self-assessment is then translated into four maturity assessment criteria. These four maturity assessment criteria are used to translate the tendency of each agile practices' adoption into level of achievement in agile maturity model.

3.4 Research Ethics

When performing research, there are four ethical issues to consider: avoidance of participants' harm, informed consent, protection of participants' privacy and prevent deception (Bell et al., 2018). All these areas will thoroughly be considered; for example, the interviewee will be informed about the study's purpose and consented to their participation in the report. Furthermore, the interview's content and the collected data are conscientiously and carefully analysed with respect to the correct content.

4 Scaled Agile Maturity Model in Manufacturing

In order to understand the scaled agile adoption, organisation needs to understand its scaled agile maturity in which suitable maturity measurement becomes essential. This chapter will cover our Research Question 1 by presenting the scaled agile maturity model in manufacturing (later referred as "proposed model"), that build upon insights from analysing the AMMs in the literature.

RQ 1: What is the measurement model and its metrics to assess the scaled agile maturity in manufacturing?

4.1 Design Scaled Agile Maturity Model in Manufacturing

4.1.1 Metrics for the Model

The standard maturity model consists of levels, descriptions of the goals in each level, dimensions, and activities or practices tied to each level and dimension (Leppänen, 2013). Since our purpose is to create a model that measures scaled agile maturity, we will follow the standard maturity model. The SAMI model has been recognised for incorporating the agile manifesto with a clear structure in how they are linked. Hence, our model is designed following SAMI, particularly on how it structured its metrics, as presented in Table 8.

Table 8 Agile Dimension in the Model

Metrics	Agile & SAFe Principles
Agile & SAFe Values	Agile & SAFe Practices in manufacturing (HW and SW)

Metrics	Principle 1	Principle 2	Principle 3	Principle 4	Principle n
Value 1	practices	practices	practices	practices	practices
Value 2	practices	practices	practices	practices	practices
Value 3	practices	practices	practices	practices	practices
Value n	practices	practices	practices	practices	practices

Taking the insights from agile maturity models in the literature and SAMI as inspiration, the metrics of the proposed model are values (Agile and SAFe), principles (Agile and SAFe), and practices (Agile and SAFe in manufacturing context).

4.1.2 Values

Following the design of the proposed model, the first step is to identify the values that resemble agile and SAFe values. These values then translated into levels. However, one insight from Chapter 2, levels are not predefined maturity paths as the waterfall approach.

The levels in the proposed model follow SAMI, and we extend each level and its objectives by adding SAFe values. The new levels and objectives presented in Table 9 with *italic* text represent the adjustment we made.

Table 9 Level and Objectives of Scaled Agile Maturity Model in Manufacturing

Agile Level	Level Name	Level's Objective (Agile Value Re-worded)
Level 5	Encompassing	Establishing a vibrant and all-encompassing environment to sustain agility *and incorporating enterprise business objectives*.
Level 4	Adaptive, *alignment*	Responding to change through multiple levels of feedback. *Ensuring alignment in product development and working in the same direction*.
Level 3	Effective	Developing high-quality and working *deliveries* in an efficient and effective manner *in the whole development lifecycle*.
Level 2	Evolutionary	Delivering *increment deliveries* early and continuously
Level 1	Collaborative, *transparency*	Enhancing communication and collaboration; *promote trust and openness*.

To fit agile in manufacturing context, our model uses *deliveries* instead of software or hardware to signify the object of the development, hence it is more general and suits both hardware and software.

Level 1 - Collaborative and transparency: SAMI puts collaborative as a foundation toward agility. Prioritizing communication and collaboration in the development process has proven

to yield a positive result (Sidky, 2007). Furthermore, the agile manifesto also puts people and interaction as the opening sentence, highlights the importance of being collaborative (Sidky, 2007). The perspective toward collaborative action is further extended by incorporating transparency from SAFe value. Transparency is the key to building trust and openness, which are tightly connected with communication and collaboration.

Level 2 - Evolutionary: Evolutionary is listed at the second level, highlights agile practices dependency on incremental development instead of big-leap development (Sidky, 2007). SAFe values also underlined the same direction in promoting better processes through iterative planning and continuous delivery.

Level 3 - Effective: This level ensures the quality of the deliveries in effective and efficient manner, since the incremental processes have already been in place (Sidky, 2007). A similar value is shared in SAFe that emphasises quality assurance in the whole development life cycle.

Level 4 - Adaptive and alignment: Being adaptive is highlighted after quality standards, to ensure that changes will not sacrifice the quality (Sidky, 2007). Moreover, the entire system needs to support the development process to keep up with the changes and preserve the product flow. By having this SAFe value into account, the level's objective from SAMI is extended.

Level 5 - Encompassing: The last level is emphasising sustainable agility in the organisation. Extended with one of SAFe values, it highlights sustainable agility as the needs to aligned the agile adoption with enterprise business objectives, which is the driver of the organisation itself.

4.1.3 Principles

After completing the levels, we work with agile and SAFe principles as the subsequent metrics of the model. Table 10 presents both Agile and SAFe principles.

Table 10 Agile and SAFe Principles

Agile Principles	SAFe Principles
1. Early and continuous delivery 2. Embrace changes 3. Working in cadence and frequent release 4. Close collaboration and alignment between technical and business 5. Motivated people 6. Face to face conversation 7. Less documentation 8. Sustainable development 9. Technical excellence & quality 10. Simplicity 11. Self-organizing team 12. Reflect, tune and adjust	1. Economic view 2. Systems thinking 3. Adapt for changing 4. Incremental building 5. Regular evaluation 6. Continuous flow 7. Work in cadence, sync and align 8. Employee engagement 9. Decentralize decision making 10. Organize around value

We start by analysing and mapping both Agile and SAFe principles into four essences of principles. Figure 9 shows the outcome of these essences, with the grey box resembling agile principles and the white box resembling SAFe principles.

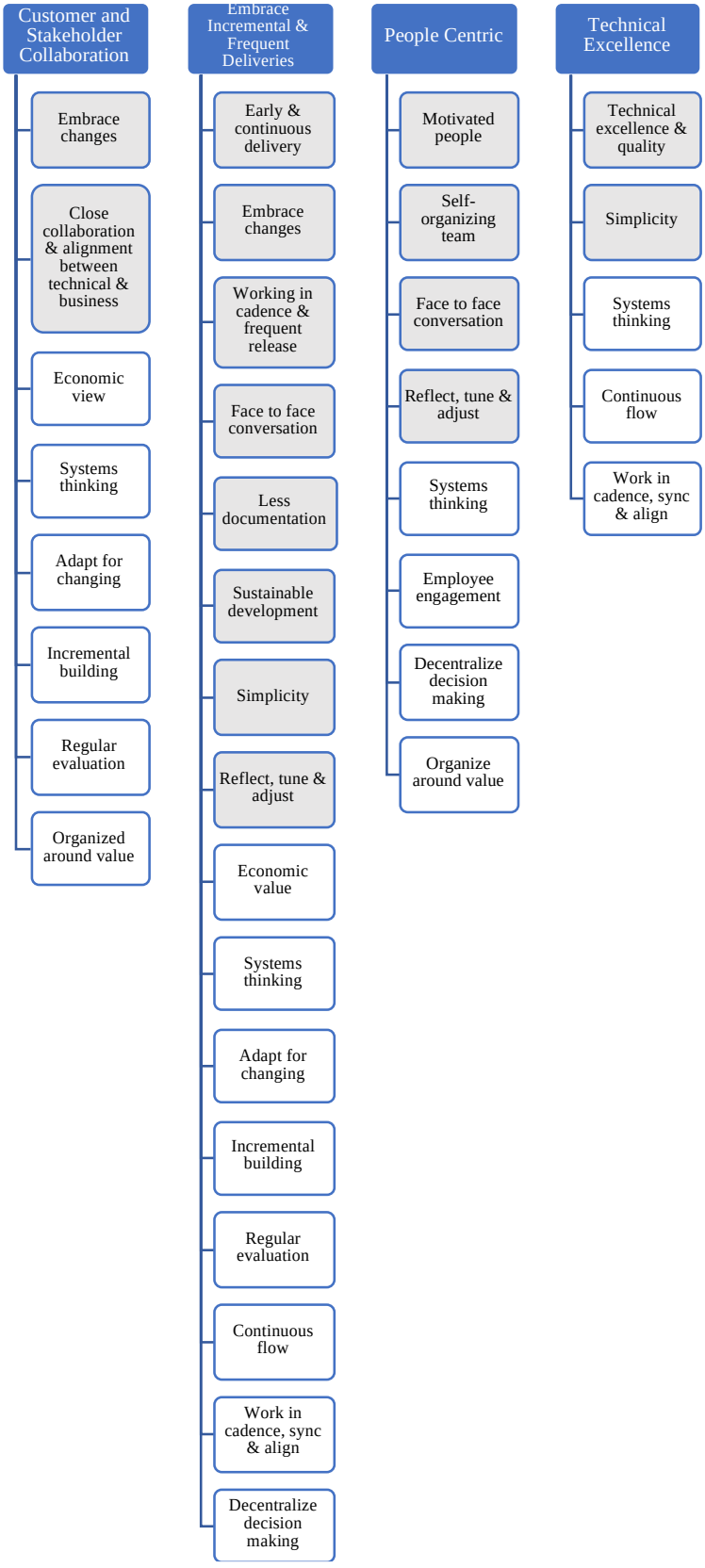


Figure 9 Mapping of Agile and SAFe Principles

Principle 1 - Customer and Stakeholder collaboration: The first essence is about collaboration with customer and stakeholder, as agile is being responsive to their acceptance despite the changing requirements. Close collaboration with customers and stakeholders plays a significant role in the success of a project. In scaled agile, there is also a need to ensure the alignment of the work with the company's objective by taking economic perspective into the context.

Principle 2 - Embrace Incremental and Frequent Deliveries: The second essence is related to inevitable changes, therefore fix planning or process does not suit agile methodology. Agile enables incremental for deliveries and processes, in the shorter cycle through continuous development and improvement. It is derived from working in iteration using feedback mechanism. The incremental is beneficial in improving the quality of deliveries with a more effective and efficient process.

Principle 3 - People-Centric: The third essence is about people and their interaction. As agile is performed by the team(s), both the deliveries and processes result from the collaboration that happens within or between the team(s).

Principle 4 - Technical Excellence: The fourth essence is technical expertise, as one of the agile principles is to deliver an outcome that embraces technical excellence and high quality deliveries.

4.1.4 Practices

In this section, we identify practices of both agile and scaled agile in manufacturing to populate the model corresponding to the values and principles. These practices will be the indicators in this proposed model, as discussed in the previous section.

We identify the practices from the studied models and literature review. Further evaluation then conducted with the six divisions to fit the practices into the manufacturing context, as it explains in the next section. We aim to use practices that fit into scaled agile in manufacturing and still serve as a guideline, rather than strict standards based solely on ceremonial events or processes.

Table 11 presents the proposed model, while Appendix A exhibits the detailed description and objective of each agile practice.

Table 11 Proposed Model for Scaled Agile Maturity Measurement Model in Manufacturing

		Principles			
		Cust & Stakeholder Collaboration	**Embrace Incremental & Frequent Deliveries**	**People-Centric**	**Technical Excellence**
Value	**Level 5: Encompassing**	• Open space	• Low process ceremony • Roadmap & project estimation	• Leadership buy-in	
	Level 4: Adaptive & Alignment	• Measure & grow • CRACK customer	• Lean development at scale • Continuous customer feedback and adaptive iterations	• Organise distributed teams	• Agile architecture • Usability testing
	Level 3: Effective	• DevOps • Vision & features • "Done" definition	• Inspect & adapt • Agile release train • Risk driven iteration • Plan and maintain a list of all features	• Scrum of scrums	• Continuous improvement (Refactoring) • Continuous integration
	Level 2: Evolutionary	• Agreement with customer cover the nature of evolutionary development	• Continuous delivery • Evolutionary requirements • Estimation & velocity measurement • PI Planning	• Stand up meeting	• Test-driven development (TDD) • Configuration management (Version Control)
	Level 1: Collaborative & Transparency	• User stories	• Collaborative planning • Reflect & tune the process	• Self-organised & empowered team	• Coding standard • Acceptance testing • Taskboard • Task volunteering • Knowledge sharing

4.1.4.1 *Practices in Level 1*

Principle 1 - Customer and Stakeholder Collaboration: The practices at this level and principle relate to tool, method or mindset that can enhance the collaboration between the team and the customer and stakeholders. User stories help the team understand the value expected by the customer and stakeholder and the delivery priority. Customer commitment to involve during the development is important to have successful collaboration with the team.

Principle 2 - Embrace Incremental and Frequent Deliveries: The practices at this level and principle relate to tool, method or mindset that can enhance the collaboration and transparency

by raising the awareness of different delivery process between agile and waterfall setup. Collaborative planning establishes the same vision with the stakeholders and helps them to be aware of agile deliveries from the start. Thus, it strenghthen their commitment during the development. The reflect and tune process gives the opportunities for the teams to reflect on the problems they encountered during the process in a regular retrospective. It promotes collaboration and builds transparency by sharing how they can address such issues in the future.

Principle 3 - People-Centric: The practices at this level and principle relate to method or mindset that can enhance self-organised and empowered team as the foundation for collaboration and transparency in the team. The members need to cooperate and trust each other to solve the team's problem. Therefore, it is essential to give authority and motivation for the team to solve the encountered problems by themselves.

Principle 4 - Technical Excellence: The practices at this level and principle relate to tools or engineering practices that can enhance the collaboration, especially at the beginning of the development. Coding standard to ease the written code shared among team members. Acceptance testing is a practice where the development's functionality or technical implementation is tested in the early phase to see whether they have been developed as the intended user story. Knowledge sharing to facilitate knowledge transfer and documentation. Taskboard helps the team enhance their transparency by sharing their work progress. The last practice is task volunteering, where the task should be voluntarily or consensusly distributed within the self-organised team rather than appointed by a manager or external party.

4.1.4.2 *Practices in Level 2*

Principle 1 - Customer and Stakeholder Collaboration: The practice at this level is agreement with customer covers the nature of evolutionary development, relate to practice, tool and mindset that emphasise the evolutionary development. This practice prevents the agreement with customer to predefine the systems and milestones, and it is instead reflecting the evolutionary approach.

Principle 2 - Embrace Incremental and Frequent Deliveries: The practices at this level and principle relate to tool, method or mindset related to continuous deliveries in agile. It highlights the importance of frequent and small deliveries at regular intervals within the timeframes and

not delivering everything at once. Therefore, an evolutionary requirement is proposed to ensure that the requirement is not fully developed in the initial stages but evolve over iterations. The upfront requirement is not supported as agile promotes feedbacks to deliver value for the customers. Estimation and velocity measurement ensures the team has fair workload by estimating their schedule, cost, and productivity through quantitative measurement such asstory point. Scaled agile consists of program increment (PI) to embrace incremental and frequent deliveries, to ensure the deliveries move to the right direction. Thus, PI planning is necessary to prepare the next PI and helps the team to break it into several iterations.

Principle 3 - People-Centric: The practices at this level and principle related to tool, method, or mindset to enhance evolutionary deliveries in working with agile through people approach. The regular meetings become more essential to keep releasing increment deliveries early and continuously. Thus, stand up meeting supports such a need by facilitating a short regular meeting to communicate the completed and ongoing works. It helps keeping the team engaged and senses problem as soon as possible by sharing individual's progress.

Principle 4 - Technical Excellence: The practices at this level and principle relate to tools or engineering practices that can be beneficial to cope with the evolutionary and iterative development approach. Test-driven development on small build or pilot plan to test the idea is proposed to identify potential errors and technical difficulties before the development starts on a large scale. Configuration management is necessary to ensure the integrity and quality of the tools, practices or systems used in the development.

4.1.4.3 *Practices in Level 3*

Principle 1 - Customer and Stakeholder Collaboration: The practices at this level and principle relate to tool, method or mindset that ensure efficiency and effectiveness of the development in customer and stakeholder domain. Development and Operations (DevOps) is mindset, culture, and practices to tighten the collaboration and communication between development and operation through cross-functional teams during the development. Vision and features are the inspirational and achievable vision of the current developed solution in the future. Definition of done helps both the team and customer or stakeholder understand when iteration is done and reduce complexity while working.

Principle 2 - Embrace Incremental and Frequent Deliveries: The practices at this level and principle relate to tool, method or mindset that promote high-quality deliveries. Agile release train (ART) is the heart that constructs deliveries in SAFe. ART aligns teams with overarching goals, schedule, and iteration to effectively deliver the best value for customers. Inspect and adapt underlines the importance to demonstrate and evaluate the state of the deliveries by the end of PI. Reflection is vital to improve the performance of the next PI. Such improvement also supported by the risk-driven iteration, where the riskiest and most challenging element in iteration/development needs to be addressed as early as possible. It essential to hinder the team from building a system that hard to finish and thus becomes a waste. Moreover, to effectively create high-quality deliveries, it is vital to plan and maintain a list of all features. Planning should express customer perspectives and should be reflected in a visible list, including their status. Thus, it ensures the customer's needs are addressed and understood and minimizes the impact of changing requirements.

Principle 3 - People-Centric: The practices at this level and principle relate to tool, method or mindset that increase the effectiveness of the team in overarching deliveries development. Scrum of scrums is a regular meeting of teams' representatives within corresponding ART. It is needed to synchronize and promotes better coordination to develop high-quality deliveries effectively during the development process.

Principle 4 - Technical Excellence: The practices at this level and principles relate to tools or engineering practices that ensure the efficiency and effectiveness of the overarching deliveries development. The continuous improvement relates to improving the existing instructions, operating procedures, code for better understandability and reducing complexity. Continuous integration encourages team members to integrate their work frequently and have the system ready to be delivered at any time, rather than scattered in various places or persons in charge.

4.1.4.4 *Practices in Level 4*

Principle 1 - Customer and Stakeholder Collaboration: The practices at this level and principle relate to method or mindset to align the development with the customer and organisation's strategy. CRACK (Collaborative, Representative, Authorized, Committed and Knowledgeable) positions customers as expert who are available for consultation at any time. Measure and grow evaluates the impact of the development in portfolio level towards sales,

operation, and customer, to determine the improvement steps and have the business agility on track.

Principle 2 - Embrace Incremental and Frequent Deliveries: The practices at this level and principle relate to tool, method or mindset that promotes adaptibility of the development and aligns it with the internal process. Organisation needs to implement lean development on a scale. It needs to address multiple feedback channels and align the impact of feedback across the teams. Therefore, lean thinking encourages an organisation to adapt to the changes with the least amount of resources during the development. Another key practice is customer feedback and adaptive iteration. It utilizes customer feedback to plan and adapt to customers' changes. Addressing the feedback is essential as a takeaway from previous iterations to reduce unwanted deliveries from customers.

Principle 3 - People-Centric: The practices at this level and principle relate to tool, method or mindset that help the organisation to manage highly distributed teams. This need comes from the realization that when the organisation scales up and grows bigger, responsibilities become more distributed and the needs for better coordination among different teams increase. Therefore, tools or procedures are needed to manage such complexity and alignment across the teams.

Principle 4 - Technical Excellence: The practices at this level and principle relate to tool or engineering practices that align the final developed deliveries from multiple teams. Intentional architecture augments the design, performance and usability of the solution and guides the inter-teams from different levels to collaborate and synchronise. Usability testing is to answer the question of how the user responds to the developed solution under realistic condition.

4.1.4.5 *Practices in Level 5*

Principle 1 - Customer and Stakeholder Collaboration: Open space in this level and principle relate to tool, method or mindset that maintain collaboration with cutomer and stakeholder in sustaining the organisation agility. It is a culture and practice to have diverse people who are interested in a specific theme to frequently meet, discuss, and aim to solve a complex problem, improve, and share the knowledge and practice in the organisation.

Principle 2 - Embrace Incremental and Frequent Deliveries: The practices at this level and principle relate to method or mindset that sustain agility while considering business objective through continuous delivery. One of the practices is low process ceremony, which indicated by low paperwork during the development. Low documentation in changing requirement as one of low paperwork application, which indicates that the team expresses high trust and responsibility, and responsive to changes. Alignment with business objective in adopting agile is also supported by establishing the roadmap and project estimation. The practice underlined the scheduled events and milestones of the company to pursue the overarching vision. However, as agile also promotes adaptability to changes, estimation is needed to address the changes while maintaining future goals.

Principle 3 - People-Centric: Leadership buy-in at this level and principle is vital to sustaining the agile adoption in an organisation. As a cultural transformation, substantial changes are required, and top-bottom facilitation enables such changes. It also highlights the importance of aiming the enterprise benefits of the organisation.

4.2 Evaluation of the Model

After the proposed model is ready, the next step is to have it evaluated by the experts from the six divisions. These experts are in charge of agile adoption in their divisions due to their extensive information about their divisions and/or their knowledge of agile.

The roles of the experts are varied and consist of scrum master, manufacturing ART leader, technical leader, and program commodity leader from ME Department. Manufacturing ART leader is a role in ME Department, which similar to the product owner in agile. Technical leader is a solution level role responsible for technical expertise within the department, for example, relating to the systems, components, and capabilities. Program commodity leader is a solution level responsible for aligning ART(s) by conducting and organising planning related to overall ME Department deliveries. Details of the experts' experiences are presented in Table 12.

Table 12 Background of the Experts

No	Role	Division	Experience in Hardware Development	Experience in Software Development	Experience with Agile
1	Scrum master	Hardware	4 years	-	4 months
2	Scrum master	Hardware	6 years	-	2 years
3	Technical leader	Hardware	4 months	3 years	1.5 years
4	Program Commodity Leader	Hardware	14 years	-	1 year
5	Manufacturing ART Leader	Hardware	4 years	-	2 years
6	Manufacturing ART Leader	Hardware	9 years	-	-
7	Scrum master	Hardware	4 years	-	4 months
8	Program Commodity Leader	Hardware	3 years	-	4 months
9	Scrum master	Software	1.5 years	2 years	2 years
10	Scrum master	Software	2 years	6 years	2 years
11.	Scrum master	Software	-	4.5 years	1.5 years

The evaluation for the proposed model focuses on its relevance as it is built and improved upon previous models (Wieringa, Maiden, Mead, & Rolland, 2006). Therefore, two criteria are utilized to evaluate the proposed model: *utility* and *relevancy* (Spencer, Ritchie, Lewis, & Dillon, 2004; Wieringa et al., 2006).

The first criterion is *utility*, whether the component of the model clears the problem and serves its intended purpose (Spencer et al., 2004; Wieringa et al., 2006). Initial presentations and discussions with the experts are held separately to explain the importance of the model and how it is designed and used, including its components.

Second criterion is *relevancy*, whether the model fits with the context (Spencer et al., 2004; Wieringa et al., 2006). As the model aims to fit the manufacturing context, the relevancy of the initial model is evaluated in the ME department. A semi-structured questionnaire is sent out to all the experts asking for their feedback consisting of relevancy and comment, based on their perception of each agile practice. We provide clear explanation with description, objectives and of how the practices affect agility to complement the questionnaire.

4.3 Final Model

Table 13 shows the proposed model after being evaluated by the agile experts (referred as "final model"), with *italic* representing the new or adjusted changes. Substantial changes in the description and goals of the agile practices have also been made based on the assessment, where the manufacturing context, hardware and software development, is taken into account. Appendix B exhibits the detailed description and purpose of each agile practice with italic characters representing the recent changes.

Table 13 Final Model for Scaled Agile Maturity Measurement in Manufacturing

		Principles			
		Cust & Stakeholder Collaboration	**Embrace Incremental & Frequent Deliveries**	**People-Centric**	**Technical Excellence**
Value	**Level 5: Encompassing**	• *Community of Practices (COPs)* (V5P1.1)	• Low process ceremony (V5P2.1) • Roadmap & project estimation (V5P2.2)	• Leadership buy-in (V5P3.1)	
	Level 4: Adaptive & Alignment	• Measure & grow (V4P1.1) • CRACK customer (V4P1.2)	• Lean development at scale (V4P2.1) • Continuous customer feedback and adaptive iterations (V4P2.2)	• Organise distributed teams(V4P3.1)	• Agile architecture (V4P4.1) • Usability testing(V4P4.2)
	Level 3: Effective	• DevOps (V3P1.1) • Vision & features (V3P1.2) • "Done" definition (V3P1.3)	• Inspect & adapt (V3P2.1) • Agile release train (V3P2.2) • Risk driven iteration (V3P2.3) • Plan and maintain a list of all *deliveries* (V3P2.4)	• Scrum of scrums (V3P3.1)	• Continuous improvement (Refactoring) (V3P4.1) • Continuous integration (V4P4.2)
	Level 2: Evolutionary	• Agreement with customer cover the nature of evolutionary development (V2P1.1)	• Continuous delivery (V2P2.1) • Evolutionary requirements (V2P2.2) • Estimation & velocity measurement (V2P2.3) • PI Planning (V2P2.4)	• Stand up meeting (V2P3.1)	• Test-driven development (TDD) (V2P4.1) • Configuration management (Version Control) (V2P4.2)

	Level 1: Collaborative & Transparency	• User stories (V1P1.1)	• Collaborative planning (V1P2.1) • Reflect & tune process (V1P2.2)	• Self-organised & empowered team (V1P3.1)	• *Building instructions/operating procedure/coding standard* (V1P4.1) • Acceptance testing (V1P4.2) • Taskboard (V1P4.3) • Task volunteering (V1P4.4) • Knowledge sharing (V1P4.5)

4.3.1 Practices

4.3.1.1 *Practices in Level 1*

Principle 4 - Technical Excellences: The practices at this level and principle relate to tools or engineering practices that can enhance collaboration, especially at the beginning of the development. Building Instructions/Operating Procedure/Coding Standard refers to standardized common instructions/procedures or code in hardware and software to increase the comprehension and easiness of sharing code or work among team members.

4.3.1.2 *Practices in Level 3*

Principle 2 - Embrace Incremental and Frequent Deliveries: The practices at this level and principle relate to tool, method or mindset that promote high-quality deliveries . Deliveries of the product should be maintained in a visible list with their status, e.g., create and control backlog. The word "feature" for this practice is challenging when interpreted in the context of hardware development. Thus, "deliveries" is used for a better fit with the context.

4.3.1.3 *Practices in Level 5*

Principle 1 - Customer and Stakeholder Collaboration: Community of Practice relates to culture and mindset that sustain organisational agility. A practice to have people across the different teams within the organisation, who have a common interest in a specific domain to gather, communicate and collaborate regularly to share knowledge, improve current practices in the organisation or solve a complex problem in communal.

5 Scaled Agile Maturity Assessment in ME Department

This chapter covers our second research question, since the model developed in Chapter 4 then used to understand the scaled agile adoption in divisions within ME Department by measuring their scaled agile maturity. The result of the measurement is analysed to find the hindraces that inhibit the maturity of their scale agile adoption.

RQ 2: What is the maturity of the current agile practice in each division, and what hinders the agile maturity?

5.1 Assessment with Scaled Agile Maturity Model

This Scaled Agile Maturity Model in Manufacturing is a self-assessment, where each division receives an online questionnaire to be distributed to all employees in the division. The employees individually assess their division. The questionnaire has 38 questions with 5-point Likert scale, corresponding to 38 agile practices in the model as shown in Appendix C.

Two of the previously studied agile maturity models (AMM and SAMI) used four maturity assessment criteria, hence the same criteria are used for this model as follows:

- Not Achieved (NA) denotes little or no evidence of achievement of the assessed practice.
- Partially Achieved (PA) denotes the presence of evidence of a few achievements of the assessed practice.
- Largely Achieved (LA) denotes the presence of evidence of a significant achievement of the assessed practice.
- Fully Achieved (FA) denotes the presence of evidence of a full achievement of the assessed practice.

As the agile maturity is based on the number of agile practices adopted, the maturity of the division is defined by the numbers of agile practices adopted with higher level achievement. The evaluation of each division result is presented, together with the hindrances to provide a sound understanding of each division's maturity, and to create plans for future improvement.

5.2 Scaled Agile Maturity at H1 Division

H1 is one of the divisions working in the hardware environment. The division restructured their organisation less than a year ago, following the ongoing changes within ME department, where the new structure adapts the scaled agile structure: teams, ARTs, and solutions. The new structure is self-initiative within the ME department to suit VCAF and the agile ways of working. H1 has set up a pilot team since the reorganisation to adopt agile, while remaining teams are not yet working agile. The scaled agile maturity self-assessment was filled by 13 respondents, including members of teams that are and are not in agile setup. The result of the assessment is presented in Table 14.

Table 14 Result of H1 Scaled Agile Maturity Assessment

Level 5	V5P1.1			V5P2.2			V5P3.1			V5P2.1
Level 4	V4P1.1		V4P1.2		V4P2.1		V4P2.2	V4P3.1	V4P4.1	V4P4.2
Level 3	V3P1.1	V3P1.2	V3P1.3	V3P2.1	V3P2.3	V3P2.4	V3P3.1	V3P4.1	V3P4.2	V3P2.2
Level 2	V2P1.1	V2P2.1		V2P2.2	V2P2.3	V2P2.4	V2P3.1	V2P4.2		V2P4.1
Level 1	V1P1.1	V1P2.1	V1P3.1	V1P4.1	V1P4.2	V1P4.3	V1P4.4	V1P4.5		V1P2.2

	Total # of Practices	**% of Practices**
Fully Achieved	0	0%
Largely Achieved	0	0%
Partially Achieved	34	89.5%
Not Achieved	4	10.5%
Total	**38**	**100%**

The result shows that 89.5% of the agile practices adoption is "partially achieved," while 10.5% is "not achieved. The "partially achieved" indicates only a few achievements of agile practices, and they are not consistently adopted. The assessment result also indicates that only a small part of H1 has not adopted agile practices, and the agile practices adopters come from the whole division, not only from the pilot team.

There are four agile practices that are "not achieved," which are *reflect and tune process* (Level 1: Collaborative and Transparency), *Test Driven Development* (Level 2: Evolutionary), *Agile release train* (Level 3: Effective) and *Low process ceremony* (Level 5: Encompassing). The "not achieved" indicates that the teams do not or rarely adopt these practices.

The agile adoption is self-initiative and voluntarily carried out by the division through learning by doing. The agile adoption in the division is facilitated by the team members themselves and motivated by the needs to align their work with the R&D department, that has fully adopted agile for several years. From the interview with the managers and the agile experts, we identify the hindrances in adopting agile practices in H1. The hindrances are:

a) *The bottom-up approach:* because agile is self- initiative and not mandatory, results in the following.

   - The absence of institutionalised guideline of how to approach agile adoption in hardware context.
   - Lack of formal facilitation such as limited agile training and no coaching are also identified. Agile mindset and knowledge in hardware development are required to address the inflexibility and inertia of the waterfall approach that has long been used in hardware development.

   Together, the absence of institutionalised guideline and formal facilitation could contribute to the low agile practices familiarity and inconsistency in their agile practices adoption.

b) *Difficulties due to different environment between software and hardware:* Although the agile practices in this model have been altered to suit the manufacturing contexts, there is still difficulties in adopting scaled agile due to the different environment. Processes in hardware development are often resourceful in terms of finance or time, those it hinders H1 from adopting agile practices that are related to evolutionary, such as *test-driven development (TDD).* Eventhough, differences of software and hardware development is expected, still there is a need to help H1 to alter the agile practices into their context, otherwise the perceived differences may continue challenge their agile adoption and inhibit their maturity. One recommendation from Copper (2016) for agile adoption in manufacturing is to combine agile and waterfall in the specific process of the entire deliveries development. However, the combination of waterfall-agile recommendation is arguably not a bottom-up decision.

For a division that just adopted agile with only one pilot team, it is expected if the assessment has revealed that maturity of H1 is mainly partially achieved. It shows that the agile values and principles have been adopted to some extent with lack of consistency, and a small part of H1 has not adopted agile practices. Addressing above hindrances could help H1 in improving their agile maturity.

5.3 Scaled Agile Maturity at H2 Division

H2 is one of the divisions working in the hardware environment. The division restructured their organisation less than a year ago, following the ongoing changes within ME department, where the new structure adapts the scaled agile structure: teams, ARTs, and solutions. The new structure is self- initiative within the ME department to suit VCAF and the agile ways of working. H2 has set up a pilot team since the reorganisation to adopt agile, while remaining teams are not yet working agile. The scaled agile maturity self-assessment was filled by 8 respondents, only from the agile pilot team. The assessment is presented in Table 15.

Table 15 Result of H2 Scaled Agile Maturity Assessment

Level 5	V5P2.1			V5P3.1			V5P1.1		V5P2.2	
Level 4	V4P1.2	V4P2.1		V4P2.2	V4P3.1		V4P4.1		V4P4.2	V4P1.1
Level 3	V3P1.1	V3P1.2	V3P2.1	V3P2.2	V3P2.4	V3P3.1	V3P4.1		V3P4.2	V3P1.3	V3P2.3
Level 2	V2P3.1	V2P1.1	V2P2.1	V2P2.4	V2P4.1	V2P4.2	V2P2.2			V2P2.3
Level 1	V1P3.1	V1P4.3	V1P1.1	V1P2.1	V1P2.2	V1P4.1	V1P4.2		V1P4.4	V1P4.5

	Total # of Practices	**% of Practices**
Fully Achieved	3	7.9%
Largely Achieved	28	73.7%
Partially Achieved	6	15.8%
Not Achieved	1	2.6%
Total	**38**	**100%**

The result shows that 73.7% of the agile practices adoption is “largely achieved. The “largely achieved” indicates significant achievements of agile practices adoption, and they are consistently adopted.

At level 1, there are two “fully achieved” agile practices which are *self-organised and empowered team* and *taskboard,* with remaining practices as “largely achieved”. This result

denotes that H2 is more mature in their Collaborative and Transparency, which is important to set structure for the upper levels. Teams that are good in collaboration and transparency will be easier to adopt practices related to evolutionary, effectiveness, and alignment. As the model is inspired by SAMI, the adoption of agile practices in the lower level is designed to build a foundation to adopt the practices on the next level.

The agile adoption is self-initiative and voluntarily carried out by the division through learning by doing. The agile adoption in the division is facilitated by the team members themselves and motivated by the needs to align their work with the R&D department, that has fully adopted agile for several years. From the interview with the managers and the agile experts, we identify the hindrances in adopting agile practices in H2. The hindrance is: *the bottom-up approach:* because agile is self-initiative and not mandatory, resulting in the following:

- There is no institutionalised guideline of how to approach agile adoption in manufacturing context. The interview reveals that H2 in their early agile adoption focuses on prioritising and planning their expected delivery. Therefore, "partially achieved" and "not achieved" practices in Principles 2 (Embrace Incremental and Frequent Deliveries) indicates lesser priority in infusing evolutionary requirements and risk driven iteration. For the long term, we see there is a need for institutionalised guideline of agile practices adoption within ME departments.
- Lack of formal facilitation such as limited agile training and no coaching are also identified. Agile mindset and knowledge in hardware development are required to address the inflexibility and inertia of the waterfall approach that has long been used in hardware development.

Together, the lack of institutionalised guideline and formal facilitation could contribute to the low agile practices familiarity and inconsistency of their agile practice adoption.

For a pilot team that recently adopted agile, H2 has shown unexpected result with "largely achieved" maturity, although there are still several practices that need to be addressed. Providing directions with guideline and training could help H2 improving their agile maturity.

5.4 Scaled Agile Maturity at H3 Division

H3 is one of the divisions working in the hardware environment. The division restructured their organisation less than a year ago, following the ongoing changes within ME department, where the new structure adapts the scaled agile structure: teams, ARTs, and solutions. All teams in H3 are already working in agile. The H3 division initiated one pilot team one year ago, before the restructuring took place. Meanwhile, the remaining team has just begun to adopt agile since the restructuring. The new structure is self- initiative within the ME department to suit VCAF and the agile ways of working. The scaled agile maturity self-assessment was filled by 18 respondents. The result of the assessment is presented in Table 16.

Table 16 Result of H3 Scaled Agile Maturity Assessment

Level 5	V5P1.1	V5P2.1	V5P2.2	V5P3.1						
Level 4	V4P3.1	V4P1.1	V4P1.2	V4P2.1	V4P2.2	V4P4.1	V4P4.2			
Level 3	V3P1.1	V3P2.1	V3P2.4	V3P4.1	V3P4.2	V3P1.2	V3P1.3	V3P2.2	V3P2.3	V3P3.1
Level 2	V2P3.1	V2P2.2	V2P2.3	V2P2.4	V2P4.2	V2P1.1	V2P2.1	V2P4.1		
Level 1	V1P4.3	V1P2.2	V1P3.1	V1P4.1	V1P4.4	V1P1.1	V1P2.1	V1P4.2	V1P4.5	

	Total # of Practices	**% of Practices**
Fully Achieved	2	5.3%
Largely Achieved	15	39.5%
Partially Achieved	21	55.2%
Not Achieved	0	0
Total	**38**	**100%**

Based on the assessment, half of the agile practices are denoted as "partially achieved," followed by "largely achieved" and "fully achieved." The half "partially achieved" indicates only a few achievements of agile practices adoption, and they are not consistently adopted. The other half of "largely achieved" indicates significant achievements of agile practices adoption, and they are consistently adopted.

In general, the level of achievement decreases as the value moves to the higher levels. For example, the majority of practices at the level 1 and level 2 are denoted with "fully achieved" or "largely achieved" while on the level 4 and level 5 are "partially achieved". Since the model

draws inspiration from SAMI, the adoption of agile practices at lower levels is designed to lay the foundation for adopting practices at the next level. Thus, as expected the practices with "fully achieved" are located on the lower level; level 1 (*Taskboard)* and level 2 (*Stand up meeting)*.

The agile adoption is self-initiative and voluntarily carried out by the division through learning by doing. The agile adoption in the division is facilitated by the team members themselves and motivated by the needs to align their work with the R&D department, that has fully adopted agile for several years. From the interview with the managers and the agile experts, we identify the hindrances in adopting agile practices in H3. The hindrance is: *the bottom-up approach:* because agile is self-initiative and not mandatory, resulting in the following:

- There is no institutionalised guideline of how to approach agile adoption in manufacturing context. The assessment result shows that the majority of "partially achieved" practices (6 out of 8 practices) belong to principle 1 (Customer and Stakeholder collaborations). This result may correlate with the lack of the institutionalised guideline on how to adopt agile, as close colaboration with customers is not a customary practice in hardware development.
- Lack of formal facilitation such as limited agile training and no coaching are also identified. Agile mindset and knowledge in hardware development is required to address the inflexibility and inertia of the waterfall approach that have been long used in hardware development.

Together, the absence of institutionalised guideline and formal facilitation could contribute to the low agile practices familiarity and inconsistency in their agile practices adoption.

H3 with all teams have been in agile setup, has an almost-even result between "partially achieved" and "largely achieved" practices. This result is expected as H3 has team that have adopted agile for over year and teams that just recently adopted it, thus the maturity may vary due to the time adoption differences. Providing directions with guideline and training could help H3 in improving their agile maturity and close the gap between the teams.

5.5 Scaled Agile Maturity at H4 Division

H4 is one of the divisions working in the hardware environment. The division restructured their organisation less than a year ago, following the ongoing changes within ME department, where the new structure adapts the scaled agile structure: teams, ARTs, and solutions. The new structure is self- initiative within the ME department to suit VCAF and the agile ways of working. H4 has set up a pilot team since the reorganisation to adopt agile, while remaining teams are not yet working agile. The scaled agile maturity self-assessment was filled only by 4 respondents across the teams within H4, with no distinction whether the respondents are from the agile team only, or combination of both agile and non-agile. Although Bell et.al (2018) did not emphasize number of minimum acceptable respondents in a qualitative data, and instead highlighted its detail and quality, but 4 respondents is considered small, compared to total individuals within H4. Hence it may pose risk of not presenting the true scaled agile maturity of H4. The result of the assessment is presented in Table 17.

Table 17 Result of H4 Scaled Agile Maturity Assessment

Level 5	V5P3.1		V5P1.1		V5P2.1			V5P2.2		
Level 4	V4P2.1	V4P4.2	V4P1.1	V4P2.2	V4P3.1			V4P4.1	V4P1.2	
Level 3	V3P1.2	V3P1.3	V3P2.1	V3P2.2	V3P2.4	V3P3.1	V3P1.1	V3P2.3	V3P4.1	V3P4.2
Level 2	V2P3.1	V2P2.1	V2P2.4	V2P4.1	V2P1.1	V2P2.2		V2P2.3	V2P4.2	
Level 1	V1P2.1	V1P2.2	V1P3.1	V1P4.3	V1P4.4	V1P4.5	V1P1.1	V1P4.1	V1P4.2	

	Total # of Practices	**% of Practices**
Fully Achieved	1	2.6 %
Largely Achieved	3	7.9 %
Partially Achieved	19	50.0 %
Not Achieved	15	39.5 %
Total	**38**	**100%**

The assessment of H4 has two prominent results which are “partially achieved” and “not achieved”. The half of “partially achieved” indicates only few achievements of agile practices adoption, and they are not consistently adopted. The second result is “not achieved” indicates a significant number of agile practices that have not or rarely adopted.

The only “fully achieved” practice is *stand up meeting* (Level 2: Evolutionary). Whilst *collaborative planning* (Level 1: Collaborative and Transparency), *lean development at scale*

and *usability testing* (Level 4: Adaptive and Alignment) are the "largely achieved" practices. These results indicate that H4 has prioritized collaboration, and monitoring their progress while being adaptive and aligned to the changes within the teams and their corresponding teams.

Practices with "partially achieved" are spread equally across the levels, and practices of Principle 2 (Enhance Incremental and Frequent Deliveries) is the highest "partially achieved" with 7/19 have been adopted, followed by Principle 4 (Technical Excellence) with 5/19. We can argue that H4 has already set their priorities in working in iteratively with small deliveries while embracing technical excellence, however they need consistency in adopting these principles.

We see similiar thing with practices that are "not achieved" that spread equally across the levels. However, if we analyze based on principles, Principle 3 (People Centric) has zero "not achieved" practices. It means self-organised teams and practices to enhance this principles have been adopted by H4, and yet the consistency is needed to enhance the maturity. The remaining three principles: Principle 2 (Enhance Incremental and Frequent Deliveries), Principle 1 (Customer and Stakeholder Collaboration), Principle 4 (Technical Excellence) respectively have not or rarely been adopted.

Moreover, although it is expected to have higher achievement practices at the lower levels, they are spread throughout the levels rather than decline towards the lower achievement as the level increases. For example, Level 4 has more "largely achieved" with only one "not achieved" practice compared to Level 3 with no "largely achieved" and more "not achieved" practices. It indicates that it is not necessary to follow the level of maturity in adopting agile.

The agile adoption is self-initiative and voluntarily carried out by the division through learning by doing. The agile adoption in the division is facilitated by the team members themselves and motivated by the needs to align their work with the R&D department, that has fully adopted agile for several years. From the interview with the managers and the agile experts, we identify the hindrances in adopting agile practices in H4. The hindrances are:

a) *The bottom-up approach:* because agile is self- initiative and not mandatory, results in the following.

- The absence of institutionalised guideline of how to approach agile adoption in hardware context. The interview reveals that H4 focuses on prioritising and planning their expected delivery. Therefore the results display that H4 shows priority in practices relating to planning, monitoring their progress and embraces technical excellence to ensure their quality of deliveries. However practices that relates to scaled agile and evolutionary such as *Agile Release Trains (ARTs)*, *user stories*, *risk driven iteration* and *continuous integration* are "not achieved" or "partially achieved." For the long term, we see there is a need for institutionalised guideline of agile practices adoption within ME departments.
- Lack of formal facilitation such as limited agile training and no coaching are also identified. Agile mindset and knowledge in hardware development is required to address the inflexibility and inertia of the waterfall approach that have been long used in hardware development.

Together, the absence of institutionalised guideline and formal facilitation could contribute to the low agile practices familiarity and inconsistency in their agile practices adoption.

b) Difficulties due to different environment between software and hardware: Although the agile practices in this model have been altered to suit the manufacturing contexts, there is still difficulties in adopting scaled agile due to different environment. Processes in hardware development often resourceful in terms of finance or time, those it hinders H4 to adopt agile practices that related to evolutionary, such as *evolutionary requirements, acceptance testing*, and *test-driven development (TDD)* that are "not achieved". Eventhough, differences of software and hardware development is expected, still there is a need to help H4 to alter the agile practices into their context, otherwise the perceived differences may continue challenge their agile adoption and inhibit their maturity. One recommendation from Copper (2016) for agile adoption in manufacturing is to combine agile and waterfall in the specific process of the entire deliveries development. However, the combination of waterfall-agile recommendation is arguably not a bottom-up decision.

H4 is division with most of the practices at lower-level achievement. The number of "not achieved" and "partially achieved" practices are almost the same, indicates the lack of

consistency in agile adoption and significant numbers of agile practices have not been adopted. The fact that H4 has only one pilot team with the rest of the teams are not in agile setup makes the low agile maturity of this division expected. Addressing above hindrances could help H4 in improving their agile maturity. However, due to the small sample size, the result might not represent the true condition of the H4 scaled agile maturity and measurement with bigger sample size is recommended.

5.6 Scaled Agile Maturity at H5 Division

H5 is one of the divisions working in the hardware environment. The division restructured their organisation less than a year ago, following the ongoing changes within ME department, where the new structure adapts the scaled agile structure: teams, ARTs, and solutions. The new structure is self- initiative within the ME department to suit VCAF and the agile ways of working. The scaled agile maturity self-assessment was filled by 6 respondents. Although Bell et.al (2018) did not emphasize number of minimum acceptable respondents in a qualitative data, and instead highlighted its detail and quality, but 6 respondents is considered small compared to total individuals within H5. Hence it may pose risk of not presenting the true scaled agile maturity of H5. The result of the assessment is presented in Table 18.

Table 18 Result of H5 Scaled Agile Maturity Assessment

Level 5	V5P2.1		V5P3.1		V5P1.1		V5P2.2		
Level 4	V4P2.2	V4P1.1	V4P1.2	V4P2.1	V4P3.1	V4P4.1	V4P4.2		
Level 3	V3P1.3	V3P2.1	V3P2.4	V3P3.1	V3P1.1	V3P2.2	V3P2.3	V3P4.2	V3P1.2	V3P4.1
Level 2	V2P2.3	V2P2.4	V2P3.1	V2P1.1	V2P2.1	V2P2.2	V2P4.1	V2P4.2		
Level 1	V1P4.3	V1P1.1	V1P2.1	V1P2.2	V1P3.1	V1P4.1	V1P4.2	V1P4.5	V1P4.4	

	Total # of Practices	**% of Practices**
Fully Achieved	1	2.6%
Largely Achieved	8	21%
Partially Achieved	24	53.2%
Not Achieved	5	13.2%
Total	**38**	**100%**

The assessment result shows 53.2% "partially adopted," that indicates only few achievements of agile practices adoption, and they are not consistently adopted. The 21 % "largely achieved" indicates significant achievements of agile practices adoption and they have been consistently adopted. The remaining 13.2% "not achieved" indicates agile practices that have not or rarely adopted.

There is one "fully achieved" agile practice, which is *taskboard* from Level 1 (Collaborative and Transparency*)*. On the other hand, there are five agile practices that are "not achieved," which are *task volunteering* (Level 1: Collaborative and Transparency), *vision and features* (Level 3: Effective), *continuous improvement* (Level 3: Effective), *community of practice* (Level 5: Encompassing), and *roadmap and project estimation* (Level 5: Encompassing). Moreover, although it is expected to have higher achievement practices at the lower levels, they are spread throughout the levels rather than decline towards the lower achievement as the level increases For example, the number of "not achieved" practices is level 3 is higher than level 4. Therefore, it indicates that it is not necessary to follow the level of maturity in adopting agile.

The agile adoption is self-initiative and voluntarily carried out by the division through learning by doing. The agile adoption in the division is facilitated by the team memberm themselves and motivated by the needs to align their work with the R&D department, that has fully adopted agile for several years. From the interview with the managers and the agile experts, we identify the hindrances in adopting agile practices in H5. The hindrance: is *the bottom-up approach*, because agile is self-initiative and not mandatory, resulting in the following:

- There is no institutionalised guideline of how to approach agile adoption in manufacturing context. The division is focusing on the practices that have a direct impact on achieving their deliveries. Their full achievement in *taskboard* indicates their concern about the progress of the deliveries. Furthermore, the majority of "largely achieved" practices are related to Principle 2 (Enhance Incremental and Frequent Deliveries) which exhibit the concern toward continuous deliveries. Therefore, the division has less priority in the other practices. For the long term, we see there is a need for institutionalised guideline of agile practices adoption within ME departments.

- Lack of formal facilitation such as limited agile training and no coaching are also identified. Agile mindset and knowledge in hardware development is required to address the inflexibility and inertia of the waterfall approach that have been long used in hardware development.

Together, the absence of institutionalised guideline and formal facilitation could contribute to the low agile practices familiarity and inconsistency in their agile practices adoption.

Since H5 just started to adopt agile, it is understandable that the majority of the practices are "partially achieved". It shows that agile values and principles have been adopted to some extent and lack consistency. However, due to the small sample size, the result may not represent the true condition of H5 scaled agile maturity and measurement with bigger sample size is recommended.

5.7 Scaled Agile Maturity at S1 Division

S1 is one of the divisions working in the software environment. The division restructured their organisation less than a year ago, following the ongoing changes within ME department, where the new structure adapts the scaled agile structure: teams, ARTs, and solutions. However, S1 has started to adopt agile more than two years ago, before the restructuring has taken place. The new structure is self- initiative within the ME department to suit VCAF and the agile ways of working. The scaled agile maturity self-assessment was filled by 19 respondents. The result of the assessment is presented in Table 19.

Table 19 Result of S1 Scaled Agile Maturity Assessment

Level 5	V5P3.1		V5P2.1		V5P2.2		V5P3.1		
Level 4	V4P1.2	V4P2.1	V4P2.2	V4P3.1	V4P4.1	V4P4.2	V4P1.1		
Level 3	V3P1.1	V3P1.1	V3P1.3	V3P2.1	V3P2.2	V3P2.3	V3P2.4	V3P4.1	V3P4.2	V3P4.2
Level 2	V2P2.4	V2P3.1	V2P2.1	V2P2.2	V2P2.3	V2P4.1	V2P4.2	V2P4.2	
Level 1	V1P4.3	V1P1.1	V1P2.1	V1P2.2	V1P3.1	V1P4.1	V1P4.2	V1P4.4	V1P4.5

	Total # of Practices	**% of Practices**
Fully Achieved	4	10.5%
Largely Achieved	29	76.3%
Partially Achieved	5	13.2%
Not Achieved	0	0
Total	**38**	**100%**

Based on the assessment, most of the agile practices are denoted as "largely achieved" followed by "partially achieved" and "fully achieved." As agile was initially introduced in the software development, it is expected to have S1 has the highest level of maturity compared to other divisions. It is also supported with the fact that S1 is the first agile adopter in ME department.

The number of practices with "fully achieved" are higher at level 2 compared to level 1. Although the adoption of agile practices in the lower levels help building a foundation to adopt the practices on the next level, the result shows that it is not necessary to start from the lower levels to increase the maturity. This result supports the argument that agility should not follow the staged level and predefined system and therefore S1 can fit their context in adopting scaled agile.

The agile adoption is self-initiative and voluntarily carried out by the division through learning by doing. The agile adoption in the division is facilitated by the team members themselves and motivated by the needs to align their work with the R&D department, that has fully adopted agile for several years. From the interview with the managers and the agile experts, we identify the hindrances in adopting agile practices in S1. The hindrances are:

a) *The bottom-up approach:* because agile is self- initiative and not mandatory, results in the following.

- The absence of institutionalised guideline of how to approach agile adoption in hardware context.
- Lack of formal facilitation such as limited agile training and no coaching are also identified. Agile mindset and knowledge in hardware development is required to address the inflexibility and inertia of the waterfall approach that have been long used in hardware development.

Together, the absence of institutionalised guideline and formal facilitation could contribute to the low agile practices familiarity and inconsistency in their agile practices adoption.

b) Dependency with other divisions that have different phases in adopting scaled agile hinders S1 from fully adopting these practices. As agile adoption among divisions within ME department is varied, so does with their maturity. Consequently, it affects their understanding and familiarity of those their agile practices and how to adopt them. Thus, there is difficulty in synchronizing the project between divisions when a collaboration work is needed.

S1 is considered more mature in adopting scaled agile and it is expected as they work in software environment and as the early agile adopter. Agile practices in S1 are dominantly in "largely achieved" with only a few that have been "fully achieved", by addressing above hindrances could help S1 improve their agile maturity.

6 Conclusion

6.1 Summary of the Findings

The purpose of this study is to understand the current scaled agile adoption in Volvo Cars, focusing in ME Departement that combines software and hardware development. The divisions within the ME Department have different stages in their agile adoption, which is related to different maturity levels in these divisions. Two research questions are studied to help understand the scaled agile maturity within these divisions, and their answers are as follows.

RQ 1: What is the measurement model and its metrics to assess the scaled agile maturity in manufacturing?

Based on the literature review, we concluded that there was no perfectly fit model to measure scaled agile maturity in manufacturing. Thus, we developed a model for these contexts as described in detail in Chapter 4.

We studied three agile maturity models by analysing their metrics and how they are structured. Five conclusions are derived from the three studied agile maturity models: *agile manifesto as the backbone of the model, criticism toward staged levels and predefined systems as a waterfall approach, different priority in levels and goal's structure, process-oriented maturity measurement and excluded people and culture, and sentiment toward context-dependence.* These conclusions are critical and become the foundations for formulating our model and its metrics. One of the conclusions is agile manifesto as the backbone, hence values, principles, and practices from agile and SAFe are populated into a model that is inspired by SAMI, one of the studied models.

The initial model is proposed and evaluated based on two criteria: *utility* and *relevancy*. *Utility* is to evaluate whether the model addresses the intended purposes. *Relevancy* is to evaluate whether the model fits with the contexts of manufacturing in large development. The evaluation is conducted by agile experts from each division. Significant changes have been made based on the feedback, and a final model is proposed as shown in Table 13.

RQ 2: What is the maturity of the current agile practice in each division, and what hinders the agile maturity?

The scaled agile maturity of each division is then measured by using the final model to understand the current agile adoption and identifying the hindrances. The agile adoption in each division varied, including the numbers of teams that have and have not been adopting agile. This difference is clearly seen in their assessment result. Each division has different maturity results, despite the facts that some divisions have the same number of agile teams or adopted agile at the same period. We identified several hindrances to their maturity, where some hindrances are similar and some are specific to certain divisions.

The similar shared hindrance is *the bottom-up approach*. Each division decided to adopt agile based on their self-innitiative, and it is not mandatory. From the interviews, we noticed that each division adopts agile with learning by doing and there is absence of institutionalized guideline and formal facilitation such as limited agile training and no coaching. The absence of institutionalised guidelines on how to approach agile, especially in manufacturing context, leads the divisions into different prioritisation in their agile practice adoption. Some divisions prioritise only planning and "temporarily" disregard evolutionary development, while some have already adopted evolutionary development but are struggling with customer and stakeholder collaboration. *A guideline will help the teams set the appropriate priority in their agile adoption that fits their manufacturing context best.* The lack of formal agile training and coaching results in teams within these divisions being unfamiliar with agile practices and unable to utilise or have them adopted in their ways of working. *Facilitating the teams with agile training and coaching will help them mitigate different agile practices and have them adopted into their manufacturing context.*

Two divisions also share the same difficulties in their scaled agile adoption *due to the different environment between software and hardware.* For example, in adopting *test-driven development (TDD),* the divisions require more financial and time than testing in waterfall approach. Therefore, there has to be a guideline on how to address such challenges by balancing iterations in scaled agile and optimisation in waterfall approaches, otherwise it will always be perceived as challenge and henceforth hinders their agile adoption.

One division raises the *dependence with other divisions that have different stages in adopting scaled agile* has a negative impact on them from fully adopting practices. As some divisions are still new in adopting scaled agile, arises a different level of understanding and familiarity with agile practices. Thus, to ensure that cooperation between them still works, there is a need to balance their settings.

6.2 Contribution to the Study and Future Research

The purpose of this study is to understand different maturity of scaled agile adoption in manufacturing, which enables the organisation to plan a better roadmap for their scaled agile adoption. However, there is no AMMs in the literature that properly fits and can be used to measure scaled agile in the manufacturing context. Therefore, we developed a model for scaled agile maturity measurement in manufacturing context that built on the AMMs in the literature. Henceforth, this study contributes to adding scaled agile in manufacturing context into the maturity measurement model.

However, as this study is conducted under a department that is greenhorn in adopting scaled agile in manufacturing context, there could be a gap in understanding the relationship between scaled agile and the development process in manufacturing. As the scaled maturity model aims to fit manufacturing in a broader organization, further evaluation with manufacturing organizations that have more experience in working with agile is needed for further study.

The data collection with small number of respondents might poses a risk of not presenting the general scaled agile maturity of its corresponding division. Bell et al., (2018) emphasised detail, depth, and quality of the qualitative data than the quantity. Therefore, when facing an issue related to small number of respondents, different data collection methods that could enhance the former criteria by Bell et al (2018) despite the small numbers of data need to be exercised for future study. One suggestion is to apply generic purposive sampling to collect the data and gain wide range of insights from different roles within organisation (Bell et al., 2018).

The scaled agile maturity measurement with our proposed model is the first conducted in these divisions. We have conducted evaluation for this model, but further research regarding assessing its reliability, of how consistent the model measuring maturity and delivering the same results under the same conditions is needed. Especially if the assessment is performed by

different assessors, there might be result's discrepancy among them. Although we have provided detailed of description and objectives of each agile practices and how this model measure the scaled agile maturity, but for future usage of this measurement model, training about this model is suggested to minimize the bias and enhance the understanding of these assessors about the model.

References

Aboagye, A., Baig, A., Hensley, R., Kelly, R., Padhi, A., & Shafi, D. (2017). Facing digital disruption in mobility as a traditional auto player. McKinsey Center for Future Mobility. Retrieved February 2021 from https://www.mckinsey.com/~/media/McKinsey/Industries/Automotive%20and%20Assembly/Our%20Insights/Facing%20digital%20disruption%20in%20mobility%20as%20a%20traditional%20auto%20player/Facing-digital-disruption-in-mobility-as-a-traditional-auto-player.pdf

Agile Alliance (n.d). Subway Map to Agile Practices. Retrieved February 2021 from https://www.agilealliance.org/agile101/subway-map-to-agile-practices/

Bell, E., Bryman, A., & Harley, B. (2018). Business research methods. Oxford university press.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organisations. *IEEE Software*, *22*(5), 30-39.

CMMI Product Team. (2010). CMMI for Development, Version 1.3. Carnegie Mellon University, (November), 482. http://doi.org/CMU/SEI-2010-TR-033 ESC-TR-2010-033

Cooper, R. G. (2016). Agile–Stage-Gate Hybrids: The Next Stage for Product Development Blending Agile and Stage-Gate methods can provide flexibility, speed, and improved communication in new-product development. *Research-Technology Management*, *59*(1), 21-29.

Cooper, R. G., & Sommer, A. F. (2018). Agile–Stage-Gate for Manufacturers: Changing the Way New Products Are Developed Integrating Agile project management methods into a Stage-Gate system offers both opportunities and challenges. *Research-Technology Management*, *61*(2), 17–26.

Davidson, B., Patel, R., (2003). *Forskningsmetodikens grunder: Att planera, genomföra och rapportera en undersökning*. Lund: Studentlitteratur AB.

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, *119*, 87-108.

Dingsøyr, T., & Moe, N. B. (2014, May). Towards principles of large-scale agile development. In *International Conference on Agile Software Development* (pp. 1-8). Springer, Cham.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, *50*(9-10), 833-859.

Eckerdal, J. R., & Hagström, C. (2017). Qualitative questionnaires as a method for information studies research.

Fontana, R. M., Fontana, I. M., da Rosa Garbuio, P. A., Reinehr, S., & Malucelli, A. (2014). Processes versus people: How should agile software development maturity be defined? *Journal of Systems and Software*, *97*, 140-155.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, *9*(8), 28-35.

Gren, L., Torkar, R., & Feldt, R. (2015). The prospects of a quantitative measurement of agility: A validation study on an agile maturity model. *Journal of Systems and Software*, *107*, 38-49.

Gustavsson, T. (2016). Benefits of agile project management in a non-software development context: A literature review. In *Fifth International Scientific Conference on Project Management in the Baltic Countries, April 14-15, 2016, Riga, University of Latvia* (pp. 114-124). Latvijas Universitate.

Laanti, M. (2016, May). Piloting Lean-Agile Hardware Development. In *Proceedings of the Scientific Workshop Proceedings of XP2016* (pp. 1-6).

Larman, C. (2004). Agile and iterative development: a manager's guide. Addison-Wesley Professional.

Law, A., & Charron, R. (2005, May). Effects of agile practices on social factors. In Proceedings of the 2005 workshop on Human and social factors of software engineering (pp. 1-5).

Leppänen, M. (2013). A comparative analysis of agile maturity models. In *Information Systems Development* (pp. 329-343). Springer, New York, NY.

Misra, S. C., Kumar, V., & Kumar, U. (2010). Identifying some critical changes required in adopting agile practices in traditional software development projects. *International Journal of Quality & Reliability Management*.

Moe, N. B., & Dingsøyr, T. (2017, May). Emerging research themes and updated research agenda for large-scale agile development: a summary of the 5th international workshop at XP2017. In *Proceedings of the XP2017 Scientific Workshops* (pp. 1-4).

Nurdiani, I., Börstler, J., Fricker, S., & Petersen, K. (2019). Usage, retention, and abandonment of agile practices: a survey and interviews results. *e-Informatica Software Engineering Journal*, *13*(1).

Ozcan-Top, O., & Demirörs, O. (2013, June). Assessment of agile maturity models: a multiple case study. In *International Conference on Software Process Improvement and Capability Determination* (pp. 130-141). Springer, Berlin, Heidelberg.

Paasivaara, M. (2017, May). Adopting SAFe to scale agile in a globally distributed organisation. In *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)* (pp. 36-40). IEEE.

Patel, C., & Ramachandran, M. (2009). Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices. International Journal of Software Engineering, IJSE, 2(1), 3-28.

Pinto, Jeffery K. (2016). Project management: Achieving competitive advantage (4th-ed). Upper Saddle River, NJ: Pearson/Prentice Hall.

Pinto, J. K., & Rouhiainen, P. (2002). *Building customer-based project organizations*. John Wiley & Sons.

Project Management Institute (PMI), (2017). Achieving greater agility, the people and process drivers that accelerate results. In *PMI's Pulse of the Profession In-Depth report*.

Punkka, T. (2012). Agile hardware and co-design. In *Embedded Systems Conference* (pp. 1-8).

Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption, and improvement of agile methods in practice. Journal of systems and software, 81(11), 1899-1919.

Rigby, D. K., Sutherland, J., & Noble, A. (2018). Agile at scale. *Harvard Business Review*, *96*(3), 88-96.

Royce, W. W. (1970) Managing the Development of Large Software Systems, *Proceedings of IEEE WESCON 26 (August*), pp. 1–9.

SAFe. (n.d.). *Various pages*. Retrieved February 2021, from http://www.scaledagileframework.com/glossary

Scaled Agile Framework. (n.d.). *Various pages*. Retrieved February 2021, from http://www.scaledagileframework.com/#

Schweigert, T., Vohwinkel, D., Korsaa, M., Nevalainen, R., & Biro, M. (2013, June). Agile maturity model: a synopsis as a first step to synthesis. In *European Conference on Software Process Improvement* (pp. 214-227). Springer, Berlin, Heidelberg.

Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. International Journal of Project Management, 33(5), 1040-1051.

Sidky, A. S. (2007). *A structured approach to adopting agile practices: The agile adoption framework* (Doctoral dissertation, Virginia Tech).

Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting agile practices: the agile adoption framework. *Innovations in systems and software engineering*, *3*(3), 203-216.

Spencer, L., Ritchie, J., Lewis, J., & Dillon, L. (2004). Quality in qualitative evaluation: a framework for assessing research evidence.

Volvo Car Group. (2020). Volvo Car Group Annual Report 2019. Retrieved April 2021, from https://investors.volvocars.com/annualreport2019/assets/pdf/VCG_AR_ENG_20200326.pdf

Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements engineering*, *11*(1), 102–107.

Williams, L., & Cockburn, A. (2003). Agile software development: it's about feedback and change. *IEEE computer*, *36*(6), 39-43.

Yin, A., Figueiredo, S., & da Silva, M. M. (2011). Scrum maturity model. Proceedings of the ICSEA, 20-29.

Appendix A

Practices in Initial Scaled Agile Maturity Model

No	Agile Practices	Source	Description	Objective	Level	Principle
1	User stories (V1P1.1)	Nurdiani, Börstler, Fricker, & Petersen (2018), Agile Alliance	A goal of the development from the end user's perspective and in simple sentences. It is general explanation with the least possible documentation. It usually is describing the expected values and general features with the least documentation.	Minimize requirement ambiguity.	1	Cust & Stakeholder Collaboration
2	Collaborative planning (V1P2.1)	Sidky (2007)	Put together all stakeholders in the planning phase.	To enhance project visibility, buy-in, loyalty, and commitment from stakeholders.	1	Embrace Incremental & Frequent Deliveries
3	Reflect and tune process (V1P2.2)	Sidky (2007)	Conduct regular retrospective during the development process. A forum where related stakeholders reflect on the challenges during the previous iteration and suggest future suggestions.	To prepare for the future's obstacle based on the obstacles or challenges encountered so far.	1	Embrace Incremental & Frequent Deliveries

4	Self-organised and empowered team (V1P3.1)	Nurdiani et al. (2018), Sidky (2007), Agile Alliance	Small team with 3 - 10 members and consists of various competencies. Team is fully responsible for the task and being independent to solve problems.	Effective communication and cooperation with each other.	1	People-Centric
5	Coding standard (V1P4.1)	Patel and Ramachandran (2009), Nurdiani et al. (2019)	A set of common language (coding) and practices shared among the developers during the collaboration.	- Knowledge retention, - increases the comprehension and ease of sharing code among team members.	1	Technical Excellency
6	Acceptance testing (V1P4.2)	Agile Alliance	Refers to the early phase functionality test to evaluate whether the developed system has met the initial requirements or intended story or not. To avoid a large volume of testing, the team automates the testing.	It is an advance test to verify the user's point of view (three amigos): - What is the problem? - How might we solve the problem? - Testing	1	Technical Excellency
7	Taskboard (V1P4.3)	Nurdiani et al. (2019), Agile Alliance	A visual display of the current sprint progress shows which stories are in progress, done or upcoming in the backlog. The board is commonly restarting in every sprint planning.	It is information for the team, to keep them focused on progress and obstacles during the sprint.	1	Technical Excellency

8	Task volunteering (V1P4.4)	Agile Alliance	As a self-organizing team, the members voluntarily sign up for the tasks rather than having someone outside the team assign them.	To increases motivation, ownership, and performance for the team members. If there is no volunteer, the team should take collective responsibility to complete the task.	1	Technical Excellency
9	Knowledge sharing (V1P4.5)	Sidky (2007), SAFe	Knowledge sharing is crucial in agile, transferring information, skills and understanding among the members. The tools will facilitate this transfer and enhance the collaboration by helping to store and maintain the information to be used by others.	Tools for knowledge sharing can be electronics (e.g., wiki, blogs) or simple whiteboards and walls.	1	Technical Excellency
10	Agreement with customer cover the nature of evolutionary development (V2P1.1)	Sidky (2007)	Although agile values collaboration than strict contract with customer, the agreement with customer is still necessary. Customers aware of the evolutionary development, that work iteratively and incrementally.	To prevents the agreement from predefined the systems and milestones, instead it is reflecting the evolutionary approach.	2	Cust & Stakeholder Collaboration

11	Continuous delivery (V2P2.1)	Sidky (2007), Agile Alliance	Team delivers its product to customer in small and frequent releases at regular intervals.	- Ensure that team not deliver the whole deliveries at once. - Encourage organisation with short timeframe releases to adapt to changes.	2	Embrace Incremental & Frequent Deliveries
12	Evolutionary requirements (V2P2.2)	Sidky (2007)	Requirement should not be fully developed in the first place, instead evolve over iterations. A further approach is undertaken to understand customer's requirement deeply.	To be adaptive with customer feedback and understand what and why it is needed to be accommodated.	2	Embrace Incremental & Frequent Deliveries
13	Estimation and velocity measurement (V2P2.3)	Patel & Ramachandran (2009), Agile Alliance	A quantitative practice for estimating the workload, for example, with story point. However, estimation is not an absolute decision and can be adjusted over progress.	- To estimate the schedule, cost, and productivity of the teams. - Also, ensure a fair workload to promote sustainable deliveries pace.	2	Embrace Incremental & Frequent Deliveries
14	Program Increment (PI) planning (V2P2.4)	SAFe	Plan incremental releases for the upcoming PI and facilitate the team to break it into several iterations regularly.	Ensure the deliveries is moving into the right direction while allowing adjustment along the way.	2	Embrace Incremental & Frequent Deliveries

15	Stand up meeting (V2P3.1)	Nurdiani et al. (2019)	Short regular meeting where the whole team communicate and reflect on the completed and ongoing work. The meeting addresses three issues: what has been done yesterday, what will be done today, and the issue that hinders progress.	- To help the teamwork with the development issue, and - to keep the team excited and engaged by sharing individual's progress.	2	People-Centric
16	Test-driven development (TDD) (V2P4.1)	Nurdiani et al. (2019), Agile Alliance	Refer to a style of programming where the developers write, test, and refactor the unit tests to identify the defects before they write the code.	To ensure that when code is written, it has passed the necessary test.	2	Technical Excellency
17	Configuration management (Version Control) (V2P4.2)	Agile Alliance	A practice or tool to control various versions of hardware or software being used and developed, to ensure that hardware or software is known and can be tracked.	To ensure the integrity and quality of that version.	2	Technical Excellency
18	DevOps (Development and Operations) (V3P1.1)	SAFe	A mindset, culture, and practices to tighten the collaboration and communication between the development & operation.	The goal is to deliver value whenever there is a business need.	3	Cust & Stakeholder Collaboration

19	Vision and features (V3P1.2)	SAFe	A description of the stakeholder's view of the solution to be developed regarding their needs and proposed features.	To capture envisioned solution and provides an overview of the system to be developed. It describes the market, customer segments and needs, in the form of new features, non-functional requirements and design constraints.	3	Cust & Stakeholder Collaboration
20	"Done" definition (V3P1.3)	Cooper and Sommer (2018), Agile Alliance	A clear definition or a list of criteria to be fulfilled for a sprint iteration is considered done, that need to be agreed upon and presented to customers. An example of a done feature is an MVP or a design.	- To reduce the complexity of a complete sprint. - To limit the cost of rework once a feature has been accepted as "done".	3	Cust & Stakeholder Collaboration
21	Inspect and adapt (V3P1.4)	SAFe	Perform demonstration and evaluation of the state of the solution by the end of each PI. It includes: - PI demo - Qualitative & quantitative measurement - Retrospective & problem-solving workshop.	Identify and promote measurable actions to improve multiple teams in ART.	3	Cust & Stakeholder Collaboration

22	Agile release train (ART) (V3P2.1)	SAFe	ART is the main value that constructs deliveries in SAFe. It is long-lived agile teams that organised around the enterprise value streams.	Align teams with common goals, schedule, and iterations in implementing continuous development flow.	3	Embrace Incremental and Frequent Deliveries
23	Risk driven iteration (V3P2.2)	Sidky (2007), Patel and Ramachandran (2009)	Identify and prioritize the riskiest and most difficult elements as early as possible in the iterations.	- To tackle and address the vital risk/issue as early as possible. - To mitigates the team to build a system that hard to complete. - Promotes more effective development.	3	Embrace Incremental & Frequent Deliveries
24	Plan and maintain a list of all features (V3P2.3)	Sidky (2007)	Planning should be based on customer's lens. Features are the terminology that expresses customer's needs and their perspectives. Features are maintained in a visible list with their status, e.g., create and control backlog (to add, remove, specify, update, and prioritize).	To ensure customer's needs are addressed and minimizing the impact of changing requirements from customer.	3	Embrace Incremental & Frequent Deliveries
25	Scrum of scrums (V3P2.4)	SAFe	A regular meeting attended by the appointed members (team's representative) from each agile team in corresponding ART.	- To synchronization multiple teams	3	Embrace Incremental &

				- To promotes better process coordination, for example, the dependencies among the teams.		Frequent Deliveries
26	Continuous improvement (Refactoring) (V3P4.1)	Nurdiani et al. (2018), Agile Alliance	Improving the existing code for better understandability and reduced complexity by improving the internal structure without changing the external behavior.	- Refactoring focuses on removing code duplication. - Making the code more understandable and readable.	3	Technical Excellency
27	Continuous integration (V3P4.2)	Sidky (2007), Agile Alliance, SAFe	Continuous integration is an agile practice that encourages members of a team to integrate their work frequently. Often assist by tools.	It is a matter of attitude than the tools, - to lessen the pain of integration by doing it frequently and - to enable updated product delivery at any moment.	3	Technical Excellency
28	Measure and grow (V4P1.1)	SAFe	Evaluate the progress in the development towards business agility to determine the improvement steps.	Measuring the impact /performance on sales, operations and customer such as time to market, quality and productivity, customer' & stakeholder's satisfaction etc.	4	Cust & Stakeholder Collaboration

29	CRACK customer (V4P1.2)	Sidky (2007), Nurdiani et al. (2019)	Customer is CRACK (Collaborative, Representative, Authorized, Committed and Knowledgeable) and continuously involved in the development process and available for consultation anytime they are needed.	Improves coordination with customers & requirements prioritization.	4	Cust & Stakeholder Collaboration
30	Lean development at scale (V4P2.1)	Nurdiani et al. (2019), SAFe	Implementation of lean thinking: - Specify and identify the value stream for each product. - Organize uninterrupted value flow. - Help the customer use/receive the value. - Aiming perfection	Invest the least number of resources in creating a short lead time with best value and quality for customer.	4	Embrace Incremental & Frequent Deliveries
31	Continuous customer feedback and adaptive iterations (V4P2.2)	Sidky (2007)	Utilize customer's feedback in the development process as the foundation to plan and adapt to customer's requirements.	Maximize the lesson learned from previous iterations to reduce the risk of "wasted" deliveries for the customer.	4	Embrace Incremental & Frequent Deliveries
32	Organize distributed teams (V4P3.1)	SAFe	Managing suitable communication and networking channel across distributed teams in large organisation by providing necessary tools.	Create proximity and align vision across distributed teams.	4	People-Centric

33	Agile architecture (V4P4.1)	Agile Alliance, SAFe	A generic set of strategic architecture has been planned to some extent and should reflect value to deliver, capabilities to develop, or constraints to tackle. To balance intentionality and emergence of the architecture, it must emerge and built incrementally during the iterations.	To enhance the solution's design, performance, and usability, and guide the inter- teams from different levels' collaboration and synchronization.	4	Technical Excellency
34	Usability testing (V4P4.2)	Agile Alliance	A testing to answer how the user responds to the hardware or software under realistic condition.	- To focus on difficulties, fac by the customer/user, - to identify the differences between the team assumption and actual user/customer behavior.	4	Technical Excellency
35	Open space (V5P1.1)	Nurdiani et al. (2019), Agile Alliance, SAFe	A straightforward interaction between individuals (that normally may not have chance to talk or work with) about specific theme. Set as a self-organised meeting and agenda, attended by anyone, gather opinion, discuss, and collectively work together to solve complex themes.	To enhance instant feedback and incredible communication from different people, and based on these principles: - Whoever comes are the right people. - Whatever happens, is the only thing that could have.	5	Cust & Stakeholder Collaboration

				- Whenever it starts is the right time. - Whenever it is over, it is over. - Wherever it happens is the right place.		
36	Low process ceremony (V5P2.1)	Sidky (2007)	Low-level paperwork involved during the process, for example low documentation needed for changing requirement request.	Promote high responsiveness toward changes. Low-level paperwork expresses a high level of trust and responsibility within the team.	5	Embrace Incremental & Frequent Deliveries
37	Roadmap and project estimation (V5P2.2)	SAFe	Roadmap is scheduled events and milestones that connect organisation vision with strategies. It proposed to embody near-term roadmap (PI), long-term roadmap (solution) and portfolio solution. Roadmap might span monthly or multiple years, thus it is essential to forecast the long-term goal carefully. Such estimation is needed to reflect changes while maintaining the future goal.	Promote high adaptability towards changes and pursues organisation objective.	5	Embrace Incremental & Frequent Deliveries

38	Leadership buy-in (V5P3.1)	Cooper and Sommer (2018), SAFe	Organisations and the executive undertake substantial changes for the enterprise benefits, e.g., restructure organisation. Top-bottom facilitation is provided by the executive, including leadership by example.	Establish supportive environment in overarching organisation.	5	People-Centric

Appendix B

Practices in Final Scaled Agile Maturity Model

No	Agile Practices	Source	Description	Objective	Level	Principle
1	User stories (V1P1.1)	Nurdiani et al. (2019), Agile Alliance	The hardware/software development goal from the customers/users' perspective and in written simple sentences. It is normally describing the expected values and general features with less documentation. *Product Owner and team formulate the user stories through discussion with stakeholders before organizing them into product backlog*.	*To help the team see how the software/hardware features will give value to customer/end-user*.	1	Cust & Stakeholder Collaboration
2	Collaborative planning (V1P2.1)	Sidky (2007)	Put together all stakeholders in the planning phase.	- *To shape a direction toward the same goals* - To enhance project visibility, buy-in, loyalty, and commitment from stakeholders.	1	Embrace Incremental & Frequent Deliveries

3	Reflect and tune process (V1P2.2)	Sidky (2007)	Conduct regular retrospective during development process. A forum where related stakeholders reflect the challenges during the previous iteration and discuss future suggestions.	To prepare for the future's obstacle based on the obstacles or challenges encountered so far.	1	Embrace Incremental & Frequent Deliveries
4	Self-organised and empowered team (V1P3.1)	Nurdiani et al. (2019), Sidky (2007), Agile Alliance	Small team with 3 - 10 members and consists of various competencies. Team is fully responsible for the task and being independent to solve problem.	- Effective communication and cooperation with each other within the team. - *It also motivates the team to improve their competencies*.	1	People-Centric
5	*Building instructions/operating procedure /coding standard* (V1P4.1)	Patel and Ramachandran (2009), Nurdiani et al. (2019)	A set of common *building instructions, operating procedures (OP) or coding shared* among the developers during the collaboration.	- Knowledge retention, - increase the comprehension and easiness of sharing code/*work* among team members.	1	Technical Excellency
6	Acceptance testing (V1P4.2)	Agile Alliance	Refers to functionality or technical implementation test, *held in the early phase by the team and customer expert.*	- To evaluate the developed system, whether it has/has not met the initial requirements or intended story.	1	Technical Excellency

				- *To align early on with plan and mitigate the element of surprise that may evolve.*		
7	Taskboard (V1P4.3)	Nurdiani et al. (2019), Agile Alliance	A visual display of the current sprint progress shows which stories are in progress, done or upcoming in the backlog. The board is commonly restarting in every sprint planning.	It is an information for the team, to keep them focus on the progress and obstacles during the sprint.	1	Technical Excellency
8	Task volunteering (V1P4.4)	Agile Alliance	As a self-organizing team, the members are voluntarily signing up for the tasks rather than having someone outside the team assign them. If there is no volunteer, the team should take collective responsibility to complete the task.	This practice increases motivation, ownership, and performance for the team members.	1	Technical Excellency
9	Knowledge sharing (V1P4.5)	Sidky (2007), SAFe	Knowledge sharing is crucial in agile, to transfer information, skills and understanding among the members. The team owns tool that will facilitate this transfer and enhance the collaboration by	Tools for knowledge sharing can be electronics (e.g., wiki, blogs) or simple whiteboards and walls.	1	Technical Excellency

			helping to store and maintain the information to be used by others.			
10	Agreement with customer cover the nature of evolutionary development (V2P1.1)	Sidky (2007)	Although agile values collaboration than strict contract with customer, the agreement *(such as request for inquiry (RFQ), contract, minutes of meeting etc.)* with customer is still necessary. Customers aware of the evolutionary nature, that development works iteratively and incrementally.	This practice prevents the agreement from predefined the systems developed and its milestones, instead reflecting the evolutionary approach.	2	Cust & Stakeholder Collaboration
11	Continuous delivery (V2P2.1)	Sidky (2007), Agile Alliance	Team delivers its product to customer in small and frequent releases. *Iteration between releases depends on the scope of the process. For example, production phase requires initial capital and time-consuming preparation, thus "too often" changes might not come in handy.*	- To ensure that team does not deliver the whole deliveries at once. - To encourage organisation with short timeframe releases to adapt to changes.	2	Embrace Incremental & Frequent Deliveries

12	Evolutionary requirements (V2P2.2)	Sidky (2007)	Requirement should not be fully developed in the first place, but instead evolves over iterations. A further approach is undertaken to understand various customer's requirement deeply. *However, commonality with different customers and consensus on frequency/period of changing requirements needs to be highlighted. For example, pursuing economic of scale as manufacturing is a heavy-invested process.*	- To be adaptive with customer feedback - To understand what and why it is needed to be accommodated.	2	Embrace Incremental & Frequent Deliveries
13	Estimation and velocity measurement (V2P2.3)	Patel and Ramachandran (2009), Agile Alliance	A quantitative practice for estimating the workload, for example with story point. However, estimation is not an absolute decision and can be adjusted over progress.	- To estimate the schedule, cost, and productivity of the teams. - To ensure a fair workload to promote sustainable deliveries pace *and foresee potential problem in each iteration*.	2	Embrace Incremental & Frequent Deliveries
14	Program Increment (PI) planning	SAFe	Plan incremental releases for the upcoming PI and facilitate the team to break into several iterations in regular basis.	Ensure the deliveries is moving in the right direction while allowing adjustment along the way.	2	Embrace Incremental &

	(V2P2.4)					Frequent Deliveries
15	Stand up meeting (V2P3.1)	Nurdiani et al. (2019)	Short regular meeting where the whole team communicate and reflect on the completed and ongoing work. The meeting addresses three issues: what has been done yesterday, what will be done today, and the issue that hinders progress.	- To help the teamwork with the development issue, - To keep the team excited and engaged by sharing individual's progress.	2	People-Centric
16	Test-driven development (TDD) (V2P4.1)	Nurdiani et al. (2019), Agile Alliance	**Software**: refer to a style of programming where the developers write, test, and refactor the unit tests, to identify the defects before they write the code. **Hardware**: *refer to a prior small build such as pilot plan to test the plan, identify possible errors and technical difficulties.*	To ensure that when the team continues the *development, they have the necessary information in advance.*	2	Technical Excellency
17	Configuration management (Version Control)	Agile Alliance	A practice to control various versions of *tools, practices, or systems being used in hardware or software development ensures*	To ensure the integrity and quality of the *tools, practices, or systems to ease the integration and*	2	Technical Excellency

	(V2P4.2)		that *the tools, practices, or systems* are known, approved, and can be tracked.	*synchronization of developed hardware or software*.		
18	DevOps (Development and Operations) (V3P1.1)	SAFe	A mindset, culture, and practices to tighten the collaboration and communication between *departments (for example R&D and ME) who are in charge of development & operation of hardware/software development.*	The goal is to deliver value whenever there is a business need *by having all teams on board and collaborate*.	3	Cust & Stakeholder Collaboration
19	Vision and features (V3P1.2)	SAFe	A description *of the future plan for the currently developed solution from the stakeholder/customer's perspective. It describes the market, customer segments and needs, in the form of new features, non-functional requirements and design constraints.*	*It is inspirational and achievable vision of the developed solution future state. It inspires people to keep working when seeing what product comes next. It also sets context and boundaries for the related work.*	3	Cust & Stakeholder Collaboration

20	"Done" definition (V3P1.3)	Cooper and Sommer (2018), Agile Alliance	A clear definition or a list of criteria to fulfil for a sprint iteration is considered to be done, that need to be agreed upon and presented to customers. Example of a done feature is an MVP (Minimum Viable Product) or a design.	- To reduce the complexity of a complete sprint - to limit the cost of rework once a feature has been accepted as "done". - *Reduce uncertainty of what criteria to complete when planning and working in iterations.* - *Serve as guideline for the new team members of how the task is going to be fulfilled.*	3	Cust & Stakeholder Collaboration
21	Inspect and adapt (V3P1.4)	SAFe	Perform demonstration and evaluation of the state of the solution by the end of each PI. It includes: - PI demo - Qualitative & quantitative measurement	- Identify and promote measurable actions to improve multiple teams in ART. - *It also entails PI retrospective; thus, it aims to improve the future processes based on what has been learned.*	3	Embrace Incremental & Frequent Deliveries

			- Retrospective & problem-solving workshop.			
22	Agile release train (ART) (V3P2.1)	SAFe	ART is the main value that constructs deliveries in SAFe. It is long-lived agile teams that organised around the enterprise value streams.	Align teams with common goals, schedule, and iterations which in implementing continuous development flow.	3	Embrace Incremental & Frequent Deliveries
23	Risk driven iteration (V3P2.2)	Sidky (2007), Patel and Ramachandran (2009)	Identify and prioritize the riskiest and most difficult elements as early as possible in the iterations.	- To tackle and address the vital risk/issue as early as possible. - To mitigates the team to build a system that hard to complete. - Promotes more effective development.	3	Embrace Incremental & Frequent Deliveries
24	Plan and maintain a list of all *deliveries* (V3P2.3)	Sidky (2007)	Planning should be based on customer's lens and express customer's needs and perspectives. *Deliveries* are maintained in a visible list with their status, e.g., create and control backlog (to add, remove, specify, update, and prioritize).	- To ensure customer's needs are addressed. - To minimize the impact of changing requirements from customer.	3	Embrace Incremental & Frequent Deliveries

25	Scrum of scrums (V3P3.1)	SAFe	A regular meeting intended by the appointed members (team's representative) from each agile team in corresponding ART.	- Synchronization of multiple teams - promotes better process coordination, for example the dependencies among the teams.	3	People-Centric
26	Continuous improvement (Refactoring) (V3P4.1)	Nurdiani et al. (2019), Agile Alliance	Improving the existing *instructions, operating procedures, code* for better understandability and reduced complexity by improving the internal structure without changing the external behavior.	- Refactoring focuses on removing *instructions, operating procedures, code duplicatio*n, - and making them more understandable and readable.	3	Technical Excellency
27	Continuous integration (V3P4.2)	Sidky (2007) Agile Alliance SAFe	Continuous integration is an agile practice that encourages team members to integrate their work frequently, *especially for new features built by different developers or teams*. Often assist by tools.	It is a matter of attitude than the tools, - to lessen the pain of integration by doing it frequently and - to enable updated product delivery at any moment.	3	Technical Excellency

28	Measure and grow (V4P1.1)	SAFe	Evaluate *the solution & portfolio progress in the development* towards business agility to determine the improvement steps.	Measuring the impact /performance of *the solution & portfolio on agreed metrics* in sales, operations and customer such as time to market, quality and productivity, customer' & stakeholder's satisfaction, etc.	4	Cust & Stakeholder Collaboration
29	CRACK customer (V4P1.2)	Sidky (2007), Nurdiani et al. (2019)	Customer is CRACK (Collaborative, Representative, Authorized, Committed and Knowledgeable) and continuously involved in the development process and available for consultation anytime they are needed.	- Improves coordination with customers & requirements prioritization. - Shorten the feedback loop.	4	Cust & Stakeholder Collaboration
30	Lean development at scale (V4P2.1)	Nurdiani et al. (2019), SAFe	Implementation of lean thinking: - Specify and identify value stream of each product. - Organize uninterrupted value flow. - Help the customer use/receive the value.	Invest the least number of resources in creating short lead time with best value and quality for customer.	4	Embrace Incremental & Frequent Deliveries

			- Aiming perfection			
31	Continuous customer feedback and adaptive iterations (V4P2.2)	Sidky (2007)	Utilize customer's feedback in the development process as the foundation to plan and adapt to customer's requirements. *However, commonality with different customers' needs to be highlighted, as specific customization to address each customer can be costly in resources.*	Maximize the lesson learned from previous iterations to reduce the risk of "wasted" deliveries for the customer.	4	Embrace Incremental & Frequent Deliveries
32	Organize distributed teams (V4P3.1)	SAFe	Managing suitable communication and networking channel across distributed teams in large organisation by providing necessary tools/*procedure*.	Create proximity and align vision across distributed teams.	4	People-Centric
33	Agile architecture (V4P4.1)	Agile Alliance, SAFe	A generic set of strategic architecture *(system, solution, and enterprise)* has been planned to some extent and should include value to deliver, capabilities to develop, or constraints to tackle. It has to emerge and built incrementally during the iterations to	To enhance the design, performance, and usability of the solution, and guide the inter-teams from different levels to collaborate and synchronize.	4	Technical Excellency

			balance intentionality and emergence of the architecture.			
34	Usability testing (V4P4.2)	Agile Alliance	A testing to answer the question of how the user responds to the developed hardware or software under realistic condition.	To focus on difficulties, face by the customer/user, to identify the differences between the team assumption and actual user/customer behavior.	4	Technical Excellency
35	*Community of practices (CoP)* (V5P1.1)	Nurdiani et al., (2019), Agile Alliance SAFe	*A community that consisted of people who have common interest in technical or business domain, where they communicate & regularly collaborate to share knowledge, practices or solve related problems within interested domain.*	To enhance instant feedback, *knowledge & practices, and incredible communication from different people across the organisation.*	5	Cust & Stakeholder Collaboration
36	Low process ceremony (V5P2.1)	Sidky (2007)	Low level of paperwork involved during the process, for example low documentation needed for changing requirement request.	Promote high responsiveness toward changes. Low level paperwork expresses high level of	5	Embrace Incremental & Frequent Deliveries

				trust and responsibility within the team.		
37	Roadmap and project estimation (V5P2.2)	SAFe	Roadmap is scheduled events and milestones that connect organisation vision with strategies. It proposed to embody near-term roadmap (PI), long-term roadmap (solution) and portfolio solution. Roadmap might span monthly or multiple years, thus it is essential to forecast the long-term goal carefully. Such estimation is needed to reflect changes while maintaining the future goal.	Promote high adaptability towards changes and pursue organisation objective.	5	Embrace Incremental & Frequent Deliveries
38	Leadership buy-in (V5P3.1)	Cooper and Sommer (2018) SAFe	Organisations and the executive undertake substantial changes for the enterprise benefits, e.g., restructure organisation. Top-bottom facilitation is provided by the executive, including leadership by example.	Establish supportive environment in overarching organisation.	5	People-Centric

Appendix C

Self – Assessment Questionnaire

No	Agile Practices	Level	Principle	Filled by	Statement	Value				
1	User stories	1	Cust & Stakeholder Collaboration	All	The project starts by having the user story as the guideline for the product owner and team.	Never True	Rarely True	Sometimes True	Frequently True	Always True
2	Collaborative planning	1	Embrace Incremental & Frequent Deliveries	All	All related stakeholders are involved and considered during the planning phase.	Never True	Rarely True	Sometimes True	Frequently True	Always True
3	Reflect and tune process.	1	Embrace Incremental & Frequent Deliveries	All	Regular retrospective with stakeholders is conducted at the end of each iteration and uses feedback for future improvement.	Never True	Rarely True	Sometimes True	Frequently True	Always True

4	Self-organised and empowered team	1	People-Centric	All	Team with 3-10 members who responsible and independent in solving team's problem.	Never True	Rarely True	Sometimes True	Frequently True	Always True
5	Building instructions/operating procedure/ coding standard	1	Technical Excellency	All	The team works with standardized building instruction/SOP/coding to enhance collaboration & knowledge retention.	Never True	Rarely True	Sometimes True	Frequently True	Always True
6	Acceptance testing	1	Technical Excellency	All	The functionality & technical implementation of the developed HW/SW has always tested against user story before starting the main construction/development.	Never True	Rarely True	Sometimes True	Frequently True	Always True
7	Task board	1	Technical Excellency	All	Every sprint has taskboard to share the team progress, to help the team focus and mitigate the obstacles.	Never True	Rarely True	Sometimes True	Frequently True	Always True
8	Task volunteering	1	Technical Excellency	All	The work is self-assigned unless there is collective	Never True	Rarely True	Sometimes True	Frequently True	Always True

					decision within the team regarding the task distribution.					
9	Knowledge sharing	1	Technical Excellency	All	Knowledge sharing is facilitated and encouraged within the team, some supporting tools are provided to ease the knowledge transfer & documentation.	Never True	Rarely True	Sometimes True	Frequently True	Always True
10	Agreement with customer cover the nature of evolutionary development.	2	Cust & Stakeholder Collaboration	All	Customers /End User understands the evolutionary nature of the development, and no prior upfront planning has written in the agreement (such as request for inquiry (RFQ), contract, minutes of meeting etc.).	Never True	Rarely True	Sometimes True	Frequently True	Always True
11	Continuous delivery	2	Embrace Incremental & Frequent Deliveries	All	Product is not delivered at once but with small and frequent releases.	Never True	Rarely True	Sometimes True	Frequently True	Always True

12	Evolutionary requirements	2	Embrace Incremental & Frequent Deliveries	All	Requirement of the product is to evolve over iterations based on customers' feedbacks and not fully developed in the first place.	Never True	Rarely True	Sometimes True	Frequently True	Always True
13	Estimation and velocity measurement	2	Embrace Incremental & Frequent Deliveries	All	Quantitatively estimates the team's workload, e.g., story point, to promote sustainable deliveries pace.	Never True	Rarely True	Sometimes True	Frequently True	Always True
14	Program increment (PI) planning	2	Embrace Incremental & Frequent Deliveries	All	Utilize program increment (PI) planning for the upcoming product and break it into several regular iterations.	Never True	Rarely True	Sometimes True	Frequently True	Always True
15	Stand up meeting.	2	People-Centric	All	Regularly conducted a short meeting with the whole team to reflect upon the completed and ongoing work.	Never True	Rarely True	Sometimes True	Frequently True	Always True
16	Test-driven development (TDD)	2	Technical Excellency	All	The team has small build or unit test to test the building instruction or coding before	Never True	Rarely True	Sometimes True	Frequently True	Always True

					the main development / construction begins.					
17	Configuration management (Version Control)	2	Technical Excellency	All	Tools, Building Instruction, Operating Procedure, Software is always updated, and all team members work with the same version.	Never True	Rarely True	Sometimes True	Frequently True	Always True
18	DevOps (Development and Operations)	3	Cust & Stakeholder Collaboration	All	Collaboration between development and operation is encouraged to enhance the SW/HW development & delivery.	Never True	Rarely True	Sometimes True	Frequently True	Always True
19	Vision and features	3	Cust & Stakeholder Collaboration	All	There is clear vision or future plan of the ongoing project that visible for all SAFe (teams, ARTs, solution, portfolio).	Never True	Rarely True	Sometimes True	Frequently True	Always True
20	"Done" definition.	3	Cust & Stakeholder Collaboration	All	Definition of Done (DOD) is clear and agreed upon, between the team &	Never True	Rarely True	Sometimes True	Frequently True	Always True

					customer/user before the sprint is started.					
21	Inspect and adapt.	3	Embrace Incremental & Frequent Deliveries	All	Evaluate the performance of the previous PI and identify a future improvement by the end of each PI.	Never True	Rarely True	Sometimes True	Frequently True	Always True
22	Agile release train (ART)	3	Embrace Incremental & Frequent Deliveries	All	SAFe implemented with ART that organised around enterprise value stream.	Never True	Rarely True	Sometimes True	Frequently True	Always True
23	Risk driven iteration.	3	Embrace Incremental & Frequent Deliveries	All	Identify and address the riskiest and most challenging element of the work as early as possible to avoid "impossible" deliveries and promote effectiveness.	Never True	Rarely True	Sometimes True	Frequently True	Always True
24	Plan and maintain a list of all deliveries.	3	Embrace Incremental & Frequent Deliveries	All	Deliveries are planned based on customers' needs and perspectives. The list of the deliveries should be	Never True	Rarely True	Sometimes True	Frequently True	Always True

					maintained in a visible list with their status, e.g., with backlog.					
25	Scrum of scrums	3	People-Centric	All	Appointed team's member for multiple agile teams in corresponding ART is synchronized through a regular meeting.	Never True	Rarely True	Sometimes True	Frequently True	Always True
26	Continuous improvement (Refactoring)	3	Technical Excellency	All	Existing building instruction, operating procedure, code is continuously reviewed and improved, to make them more understandable & less complex.	Never True	Rarely True	Sometimes True	Frequently True	Always True
27	Continuous integration	3	Technical Excellency	All	The teams that work with different features in one sprint, continuous integration & work synchronized are encouraged, to have the	Never True	Rarely True	Sometimes True	Frequently True	Always True

					delivery ready incomplete set up anytime needed.					
28	Measure and grow.	4	Cust & Stakeholder Collaboration	Solution Level	At the solution & portfolio level, impacts of the development are measured. Metrics are used to assess the change in operation (i.e., efficiency, performance, quality, ergonometric), sales & customer (satisfaction).	Never True	Rarely True	Sometimes True	Frequently True	Always True
29	CRACK customer	4	Cust & Stakeholder Collaboration	All	Customers/Users are collaborative and actively participate in the development process. Their feedback is easy to get.	Never True	Rarely True	Sometimes True	Frequently True	Always True
30	Lean development at scale	4	Embrace Incremental & Frequent Deliveries	All	Implemented lean thinking during the development.	Never True	Rarely True	Sometimes True	Frequently True	Always True

31	Continuous customer feedback and adaptive iterations	4	Embrace Incremental & Frequent Deliveries	All	Continuously utilize customer's feedback as a foundation to plan and adapt with customer feedback.	Never True	Rarely True	Sometimes True	Frequently True	Always True
32	Organize distributed teams.	4	People-Centric	All	Using tools/procedures to organize distributed teams.	Never True	Rarely True	Sometimes True	Frequently True	Always True
33	Agile architecture	4	Technical Excellency	All	The Safe (teams, ARTs, solutions, portfolio) has a generic architect of the intended solution build across the levels. Everyone works toward the same end result.	Never True	Rarely True	Sometimes True	Frequently True	Always True
34	Usability testing	4	Technical Excellency	All	Customer/user tested the developed HW/SW together with the team, to see if there is different assumption occurs from the expected delivered HW/SW.	Never True	Rarely True	Sometimes True	Frequently True	Always True

35	Community of practices (COPs)	5	Cust & Stakeholder Collaboration	All	There is community for the teams across or and share their practices, skill, and problems. This community is encouraged and facilitated by the management.	Never True	Rarely True	Sometimes True	Frequently True	Always True
36	Low process ceremony	5	Embrace Incremental & Frequent Deliveries	All	Low paperwork needed to adapt to changing development process, e.g., requirement.	Never True	Rarely True	Sometimes True	Frequently True	Always True
37	Roadmap and project estimation	5	Embrace Incremental & Frequent Deliveries	All	Establish and estimate a roadmap that connects organisation vision with strategies. Roadmap is proposed for near-term, long-term, and portfolio development.	Never True	Rarely True	Sometimes True	Frequently True	Always True
38	Leadership buy-in	5	People-Centric	All	The top management supports the agile ways of working across the organisation by	Never True	Rarely True	Sometimes True	Frequently True	Always True

					establishing environment, culture, and training to sustain the agile practice.					

DEPARTMENT OF TECHNOLOGY MANAGEMENT AND ECONOMICS
DIVISION OF INNOVATION AND R&D MANAGEMENT
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY