CHALMERS
UNIVERSITY OF TECHNOLOGY

Agile Hardware Development

Exploring Agility in a Hardware Development Organization

Master's Thesis in the Program Quality and Operations Management

OSCAR PALMQVIST
MARTIN TRIFUNOV

Department of Technology Management and Economics
Division of Innovation and R&D Management
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019
Report No. E2019:068

Master's thesis E 2019:068

Agile Hardware Development

Exploring Agility in a Hardware Development Organization

OSCAR PALMQVIST

MARTIN TRIFUNOV

Supervisor at Chalmers: Johannes Berglind Söderqvist

Supervisor at Volvo Cars: Lucas Gren

Examiner: Lars Trygg

Department of Technology Management and Economics
Division of Innovation and R&D Management
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2019

Agile Hardware Development: Exploring Agility in a Hardware Development Organization

OSCAR PALMQVIST
MARTIN TRIFUNOV

Master's Thesis E 2019: 068

Department of Technology Management and Economics
Division of Innovation and R&D Management
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone: + 46 (0)31-772 1000

Chalmers Reproservice
Göteborg, Sweden 2019

Abstract

Agile product development is seen with considerable interest in both academia and industry. The methodology promises faster development cycles, higher quality, and increased flexibility. However, as agile product development's roots lay within software development, there are inherent differences when compared to hardware development. Research within agile hardware development is underdeveloped and lacks a definite basis upon which to build. This study aimed to identify key aspects of agility within agile hardware development, through the use of a case study at Volvo Cars Efficient Structure department.

An abductive research approach was used to explore this topic, in which a recurrent matching and comparison between literature and empirical findings were used to guide the research-process. A thorough examination of existing literature was conducted, which produced an operationalization of agility. The operationalization provided a basis for both empirical and theoretical research. The empirical research strategy was focused on a case-study, consisting of interviews and participant observation.

Through the systematic matching between the case study, the operationalization of agility, and agile product development literature, key underlying key aspects were identified to be central for agility within a general agile hardware development. These aspects were found to be either beneficial or detrimental for the agility at Efficient Structure or to otherwise inherently affect the organization's ability to promote agility.

In conclusion, a central theme for agility within agile hardware development was the balance of fixed versus flexible development. This study provides insight into agile hardware development. Within an academic realm, the operationalization provides a centralized basis around which general agility can be defined, and the key aspects provide a basis for further exploration of agility within hardware development. Industrially, the key aspects can be used to inform potential frameworks and the implementation of existing frameworks.

Key Words: Agile Hardware Development, Agility, Agile Product Development, Large-Scale Agile, Complexity, Visual Communication, Communication, Informal Networks

Acknowledgments

Working on this master's thesis has been a journey. For having been both a guiding hand and a source of inspiration, an enormous thank you must be given to Johannes Berglind Söderqvist. Thank you for guiding us through the confusion and being willing to challenge us and "take the discussion". We both feel that this helped us grow tremendously, and as a result, has made the report better. Furthermore, we would like to thank Lars Trygg, both for being our examiner and for being an early guide in our master's thesis.

We would also like to thank our supervisor at Volvo Cars, Lucas Gren. Thank you for allowing us to explore a topic we both find engaging and giving us the mandate to explore it fully. Within Volvo we would also like to thank Malin Kihlman, who opened up Efficient Structures department at Volvo for us and proved to be an excellent sounding board throughout our journey.

Oscar Palmqvist Martin Trifunov

Gothenburg, June 2019

Table of contents

List of Figures

List of Tables

1 Introduction

"We used to be more agile before the implementation of SAFe," stated an engineer at Efficient Structures. This sentiment was shared by several of their colleagues at Volvo Cars in Gothenburg. The implementation of agile development, guided by the SAFe approach, which was developed as a response to the traditional plan-driven product development, has resulted, according to some employees, in a less agile organization. The underlying reasons for the undesirable and contradictory experiences may not prove to be as surprising as one might initially believe.

1.1 Background

Agile development is described as one of the most popular trends within project management – both in practice and academia (Hobbs & Petit, 2017). It is a product development method which was consolidated in 2001 through *the agile manifesto* when 17 software developers set out to define methods and practices that were successfully used in meeting customer demands in a rapidly changing market environment (Rigby et al., 2016b). Agile was a response to traditional plan-driven development and combined teamwork with flexibility (Highsmith & Cockburn, 2001). The resulting agile manifesto described the characteristics and spirit of agile product development through four key values and twelve principles (Rigby et al., 2016b). One of the core values – Working software over comprehensive documentation (Beck et al., 2001a) – and the full name of the agile manifesto – Manifesto for Software Development – demonstrates conformity toward software development.

Research in agile development has, according to Hobbs and Petit (2017), likewise focused on software development – mirroring how agile has been applied in practice. While research is gaining popularity and primarily dedicated to software development, agile method research is based on industry-driven and commercialized methods due to the early and leading role the industry has had on the agile community (Conboy, 2009). Agile within other contexts has been researched to a lesser degree but has shown promising potential (e.g., Kuusinen et al., 2016; Schuh et al. 2016). Cooper's (2016) research revealed similar results for combining agile methods with the stage-gate model in manufacturing companies. Although there are results in favor of agile in a broad context, Abrahamsson et al. (2009) describe a need for extending the context of agile methods beyond their software application. More recently, Schuh et al. (2018, p. 24) came to the same conclusion *"[...] a systematic transfer towards product development is still missing."*.

The small extent of research on agile in alternative areas has not stopped a wide range of companies from adopting agile practices (Cooper, 2016). Agile development is enacted in organizations through several practices. *Scrum* is the most common methodology and consists of explicit iterative processes and roles (Hobbs & Petit, 2017). Other methodologies include Extreme Programming (XP) and Feature Driven Development (FDD) (Dybå & Dingsøyr, 2008). Several frameworks have been developed, e.g., *LeSS* (Large-Scale Scrum) and *SAFe* (Scaled Agile Framework), to solve prevalent scaling issues with agile (Dikert et al., 2016; Hobbs & Petit, 2017).

SAFe became the chosen framework when Volvo decided to implement agile development into their organization. The transformation has internally been described as the largest organizational change in the history of Swedish industry. SAFe is a holistic enterprise solution framework with extensive descriptions of the steps and processes necessary for creating an organization that can keep up with the continuously changing market requirements of today. It was developed for the software industry with the aim to provide an answer to the scalability issues of existing agile practices (Scaled Agile, 2018).

Figure 1: The body of a Volvo XC90 developed by Efficient Structures (Volvo Cars, 2013)

At Volvo Cars and the department of Efficient Structures – the subject of the case study in this thesis – the implementation of SAFe that started in the fall of 2018 has in some ways yielded contradictory results. At the time of the study, the change process had not yet been fully realized, meaning there were two parallel organizations – the previous traditional one and the SAFe one. It provided insight into both ways of working. The department develops the body and structure (only physical/hardware parts and systems) of the Volvo cars. The car body they develop is integrated with the development of other functions, such as *Engine, Packing, and Interior Design*, in order to make a whole complete car. They have internally been described as an agile department, long before any mentions of SAFe, and have continuously delivered exceptional products within budget and deadline at a Best in Industry standard, even though they work according to plan-driven development.

1.2 Purpose

Due to agile product development's roots in software development, there is a lack of effective transfer to hardware development. As indicated by Efficient Structures trouble with their SAFe implementation, this divide is significant. This is furthered due to the lacking exploration of agility within hardware development. Therefore, this study aims to develop theory, with the central purpose to:

Identify key aspects for agility within agile hardware development.

The identification of key aspects can be used to develop a basis for future research, and models of agile hardware development can be created.

1.3 Research Questions

Three research questions were created to guide the thesis. The first is based on Conboy's (2009) reasoning that in order to be able to adapt agile development to other contexts and gain a deeper understanding beyond industry-driven renditions of agile, one needs to use the concept of agility. It is described as the core of agile (Conboy & Fitzgerald, 2004).

RQ1: *What is agility within the context of agile product development theory?*

Traditional plan-driven hardware development is often described as the opposite of agile development (Hobbs & Petit, 2017). Efficient Structures which develop hardware has contradictorily been described as one of the most agile functions within research and development at Volvo. Therefore, in order to understand agility within a hardware context and the specific case, the activities affecting agility within Efficient Structures were researched.

RQ2: *What factors affect agility at Efficient Structures at Volvo Cars?*

The last research question was aimed at guiding our discussions in translating our results to a general context. It combines aspects of RQ1 and RQ2 to fulfill the purpose of this study.

RQ3: *How do the results relate to a general agile hardware development context?*

1.4 Delimitations

The research context is confined to complex, large-scale automotive hardware development. When referring to hardware development, the study refers to the development and creation of physical products, that are made in an environment characterized by many interdependencies.

As previously mentioned, agile method research is almost completely based on industry-driven practices such as Scrum and XP (Conboy, 2009). Due to both their commercial nature and varied implementation, this study has therefore chosen to exclude the exploration of such frameworks.

2 Theoretical Framework

The theory is based primarily on research within agile development with added support from other fields. Three topics within agile development were found to be relevant and needed for translating agile to a complex hardware context, namely the core of agile, agile in practice, and other aspects of agile beyond software development.

2.1 The Core of Agile

It is essential to understand the core and the fundamental principles of agile before one can transfer it to another context (Conboy, 2009). The core can be divided into two areas, as the first area, the agile manifesto, is at the heart of many practices used by companies and research papers (Dingsøyr et al., 2012). The other part is the concept of agility, which is described as the core of agile development (Conboy & Fitzgerald, 2004). Although agility and the agile manifesto are separate concepts, e.g., Gren and Lenberg (2019) linked them together to provide a basis for an altered definition of agility (presented in chapter 2.1.2).

2.1.1 The Agile Manifesto

The traditional product development methods that relied upon being pre-determined and bureaucratized had proven to be unsuccessful within software development where changes in requirements are imminent (Highsmith & Cockburn, 2001). Project goals were not being met, deadlines were missed, and the quality was subpar. The agile manifesto was created as a response to those shortcomings (Highsmith & Cockburn, 2001). It was a synthesis of lightweight product development methods and practices that were used at the time (Figure 2) and performed well within software development (Rigby et al., 2016b).

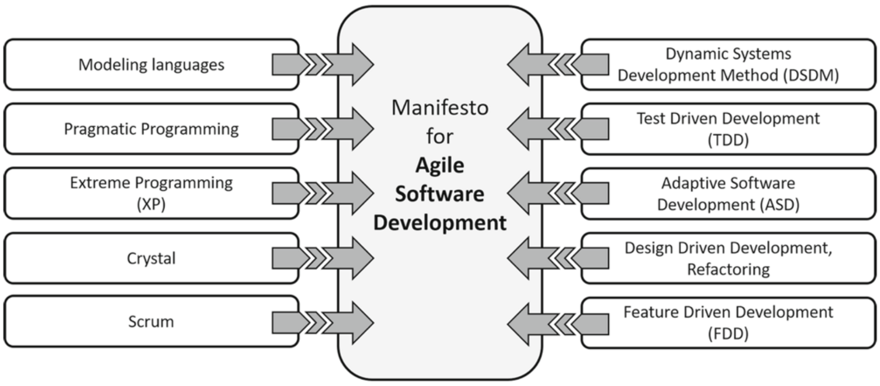

Figure 2: The heritage of the agile manifesto (Hohl et al., 2018, p.11)

The manifesto was welcomed by the software community and is considered by some as the epitome of successful software development, while others view it is as a commercial gimmick (Hohl et al., 2018). Since its creation, the agile manifesto has had a strong influence in guiding agile research (Dingsøyr et al., 2012) and the development of agile practices (Conboy, 2009). The influence has led it to be the subject of criticism (e.g., Conboy & Fitzgerald, 2004; Laanti et al., 2013) and alterations due to its argued inadequacy (e.g., Rigby et al., 2016a; Williams, 2012). The criticism stems, according to Laanti et al. (2013), from the agile manifesto being too inexplicit and not scientifically scrutinized. The first part of the agile manifesto (Beck et al., 2001a) consists of four core values which were framed as dichotomies where the values on the left should be maximized, and the others minimized:

> The Agile Manifesto
>
> *We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*
>
> **Individuals and interactions** over processes and tools
>
> **Working software** over comprehensive documentation
>
> **Customer collaboration** over contract negotiation
>
> **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.*

Figure 3. The agile manifesto (Beck et al., 2001a)

An adapted and more recent version of the manifesto, by Rigby et al. (2016a) (one of the original contributors), was based on the same core values but with further descriptions and a broader context. The first core value is changed to *people over processes and tools* and is built upon motivated and empowered teams and individuals. According to Rigby et al. (2016a), teamwork should be fun, creative, and challenging, and simultaneously encourage a viable workload. Communication should be face-to-face, and management needs to support the teams and collaboration across functions. The second value was altered to *working prototypes over excessive documentation*. Rigby et al. (2016a) stated that it is about seeing the progress of one's work and being able to test products and services toward the end user and in doing so end all team discussions of what the best solution is. The third value is similarly generalized; *customer collaboration over rigid contracts*. As customers have difficulty in predicting what they want, it is imperative to work together through prototyping and market tests to create as much value as possible. The fourth value – *respond to change rather than follow a plan* – described the need for embracing change within product development processes (Rigby et al., 2016a)

The second part of the agile manifesto consists of twelve principles that were developed some months after the original part (Rigby et al., 2016b). The principles are to be followed by agile practitioners. According to Rigby et al. (2016b), frameworks that follow the principles, and the initial four values were to be called agile techniques. Similar to the four values, the twelve principles have also experienced alterations. In an attempt by Williams (2012), the principles were revised – e.g., the third principle was removed due to it being too similar to the first one and face-to-face communication was changed to synchronous communication. Williams (2012) stated that the term synchronous better encompasses the underlying meaning of 'face-to-face' communication, e.g., instant messaging and Skype. The core part is to communicate directly. After a review and comparison were conducted, it was ultimately concluded that the original principles more than adequately described the spirit of agile development but that, contrastingly, some principles, e.g., communication, were not emphasized and described sufficiently (Williams, 2012). The principles (Beck et al., 2001b) are presented below:

The Agile Principles
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Business people and developers must work together daily throughout the project.
Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
Working software is the primary measure of progress.
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Continuous attention to technical excellence and good design enhances agility.
Simplicity-the art of maximizing the amount of work not done-is essential.
The best architectures, requirements, and designs emerge from self-organizing teams.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Figure 4: The agile principles (Beck et al., 2001b)

From an analysis of the twelve principles, Laanti et al. (2013) asserted what each principle emphasized. The results consisted of 22 words and phrases that represent the principles; (1) *customer satisfaction/benefit,* (2) *continuous delivery,* (3) *value,* (4) *early/frequent deliveries,* (5) *adaptability,* (6) *competitiveness, (7) collaboration,* (8) *motivated individuals,* (9) *good environment,* (10) *support,* (11) *trust,* (12) *efficiency,* (13) *communication,* (14) *measure progress via deliverables,* (15) *sustainability,* (16) *people,* (17) *focus on technical excellence,* (18) *good design as an enabler of agility,* (19) *simplicity,* (20) *optimize work,* (21) *self-organization* and (22) *built-in improvement of efficiency and behavior* (Laanti et al., 2013, p.248).

2.1.2 The Concept of Agility

In 1991, agile manufacturing was described as the philosophy that was the way forward for manufacturing companies in order to become successful in a changing market environment, and from it came the concept of agility (Conboy & Fitzgerald, 2004). Agile manufacturing has had the same issues as agile software development – the areas lack a theoretical basis, and definitions are contradicting each other (Conboy & Fitzgerald, 2004). While the topic has been thoroughly covered and the core elements of agility are generally agreed upon, a consensus on a definition (within the context of integrated software development) has not yet been reached (Sarker et al., 2009). The lack of consensus can be problematic as, according to Abrahamsson et al. (2009) there exists a need for understanding agility – not only defining it but made measurable and assessable as well – as it is essential for a transfer of agile methods to broader contexts. Conboy and Fitzgerald (2004) likewise describe the need to use agility as the basis for agile development and propose a definition that is built upon flexibility and leanness.

Flexibility, according to Conboy and Fitzgerald (2004), is linked to embracing or enduring change effectively. It is also linked to speed and the ability to quickly respond to change (Conboy & Fitzgerald, 2004). Speed is central for agility but in the aspect of rapidly updating the context in which changes occur and are needed to be adjusted to (Gren & Lenberg, 2019). Taking six months to receive feedback and act on it is not the agile way (Highsmith & Cockburn, 2001), but optimizing for speed can result in lost value due to loss of customer focus (Gren & Lenberg, 2019).

Leanness comes from *lean* and the Toyota Productions System (Conboy, 2009). Leanness is defined as *"the maximization of simplicity, quality and economy,"* (Conboy & Fitzgerald, 2004, p. 39). From the two concepts, a definition was proposed but later improved upon by Conboy (2009). The new definition was based on same reasoning and was refined to *"the continual readiness of an ISD [information systems development] method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment,"* (Conboy, 2009, p. 340).

Gren and Lenberg (2019) argued against the notion of relating agility to lean as it is about doing things right, i.e., process-oriented, while agility is linked to responding to change and doing the right things. Gren and Lenberg's (2019) definition is instead related to the agile manifesto, the agile principles, and agile methods. The proposed definition – *responsiveness to change* – is a condensation of what agile is at its core (Gren & Lenberg, 2019). Their concise definition is similar to Highsmith and Cockburn's (2001, p. 122) explanation of the concept *"Agility, ultimately, is about creating and responding to change."*

Dove and LaBarge (2014, p. 5) define agility in systems engineering context as *"Agility is the ability of a system to thrive in an uncertain and unpredictably evolving environment; deploying effective response to both opportunity and threat, within mission. Effective response has four metrics: timely (fast enough to deliver value), affordable (at a cost that can be repeated as often as necessary), predictable (can be counted on to meet the need), and comprehensive (anything and everything within the system mission boundary)."* The definition consists of a positive value where a system is to *thrive*, acknowledgment of ever-present change and to respond to the changes. Similar to Conboy and Fitzgerald (2004), Dove and LaBarge (2014) related agility to speed and leanness.

Agility has been discussed in other contexts as well. Schuh et al. (2017a) have another approach when defining agility that is linked to iterative development within hardware development. The degree of agility is not constant but changes depending on which product development phase a project is in (Schuh et al., 2017a). When determining if a team is agile, some organizations use agility as a placeholder for any *"[...] cool, liberated form of undocumented software creativity,"* (Ambler, 2009, p. 8). Agility has been used by Gren et al. (2017) as a means of connecting agile teams to group maturity and development.

2.2 Agile in Practice

The core of agile has now been explained through the agile manifesto and definitions of agility. They provide an understanding of the agile spirit and present the core elements of how agile development is enacted in organizations. Beyond the manifesto, there are more in-depth explanations of the characteristics of agile. The different aspects overlap in some regard, but they can be categorized into teamwork, iterative processes, and improvement (Nerur & Balijepally, 2007; Takeuchi & Nonaka, 1986).

2.2.1 Self-organizing and Cross-Functional Teamwork

Teamwork is essential as teams and people are the main drivers of project success (Highsmith & Cockburn, 2001). In agile development, the focus on teams is greater than in traditional product development methods (Gren et al., 2017). Two sub-themes within teamwork are prevalent with the first being the self-organizing aspect of teams. It is described as the heart of agile development (Hoda et al., 2010). The second part is team cross-functionality, and the reason for it is its positive effect on the overall responsiveness of teams and organizations (Nerur & Balijepally, 2007). All of the popular practices, e.g., Scrum and XP, have teamwork as one of their tenets and emphasize the importance of teams (Dybå & Dingsøyr, 2008). The team-sizes can vary from 3 to 15 members (Hoda et al., 2010). Wheelan (2009) found a significance of team-size in relation to productivity and group maturity.

Self-organizing teams – one of the principles in the agile manifesto (Beck et al., 2001b) – have become popular in the advent of agile development (Hoda et al., 2010). Speed and quality of teams benefit from self-organizing with operative developers being able to effectively solve problems by shifting the decision-making closer to the source, i.e., the teams (Hoda et al., 2010). It removes the bureaucracy related to escalating issues in the organizational hierarchy (Hoda et al., 2010). Similarly, Nerur and Balijepally (2007) found that responsiveness and flexibility are achieved through self-organizing teams. Takeuchi and Nonaka (1986) present three conditions that were found to be essential for a successful self-organizing team. Autonomy, the first condition, revolves around letting the teams govern themselves without any interference from management (Takeuchi & Nonaka, 1986), where management sets as few project specifications as possible (Nerur & Balijepally, 2007). Teams should have the ability to choose their way of working per the challenges they face and the possibility to make decisions (Takeuchi & Nonaka, 1986). Ambler (2009) found that it is necessary to set a framework or boundaries in which teams in a large-scale context can operate within. The second condition is self-transcendence, according to Takeuchi and Nonaka (1986). The most successful teams are elevated beyond themselves and together strive toward a higher purpose. The examples provided by Takeuchi and Nonaka (1986) consist of teams having to overcome seemingly impossible tasks. The last condition is cross-fertilization and revolves around creating teams with a broad set of competencies (Takeuchi & Nonaka, 1986).

While it may yield satisfactory results, self-organization can be challenging to set up (Moe et al., 2009). Barriers to do so exist on two levels – team and organizational. Team barriers are connected to personal commitment, failure to learn, and individual leadership, while organizational barriers are linked to shared resources, organizational control, and specialist cultures. One way to overcome the barriers is to co-locate the members, and another is to empower the team. (Moe, et al., 2009). Empowered teams and individuals are imperative for agile development (Rigby et al., 2016a), and empowerment creates a sense of ownership leading to motivation and overall better performance of teams (Nerur & Balijepally, 2007).

The cross-functional aspect is vital for teams and organizations (Rigby et al., 2016a), and already touched upon by Takeuchi and Nonaka's (1986) concept of cross-fertilization. Highsmith and Cockburn (2001) describe cross-functionality and teamwork in terms of collaboration internally within teams and externally toward customers. Customer involvement is a challenge for agile teams, as the degree of involvement is often lower than what is required of agile methods (Hoda et al., 2011). Adversative results due to lack of customer collaboration may appear as problems in securing customer feedback, difficulty to set requirements, and loss of productivity (Hoda et al., 2011).

Ambler (2009) presented agile team cross-functionality as having all the necessary knowledge in a team to successfully meet the set goals and therefore, a strong need for active stakeholder participation. Kim and Chai (2017) stated that an increase in supplier involvement and informational flow improves agility. Cross-functionality in self-organizing teams, through stakeholder involvement, redundancy, and overlapping of skills and a wide range of team capabilities, results in an ability to quickly respond to changing demands without delay (Nerur & Balijepally, 2007).

2.2.2 Continuous Improvement and Learning

Agile and lean philosophy share similarities, and one of them is continuous improvement (Conboy & Fitzgerald, 2004; Gren & Lenberg, 2019). Continuous improvement is a central part of agile processes (Ambler, 2009), and it is often enacted at a team-level by the removal of hindrances and development of team members through learning (Nerur & Balijepally, 2007). Nerur and Balijepally (2007) explain different types of learning processes (Figure 5) where the best one revolves around iterative problem solving and double-loop learning, i.e., deeper learning through a reflection of learnings. It is achieved through, e.g., experimentation, stand-up meetings, and reflection workshops (Nerur & Balijepally, 2007). Learning is essential for agility as it increases a team's skills to faster and better solve complex problems (Nerur & Balijepally, 2007). Ambler (2009, p. 8) stated that *"Agile teams regularly reflect on, and disciplined teams also measure, how they work together and then act to improve on their findings in a timely manner, "* and added the importance of measurements for the improvement of teams.

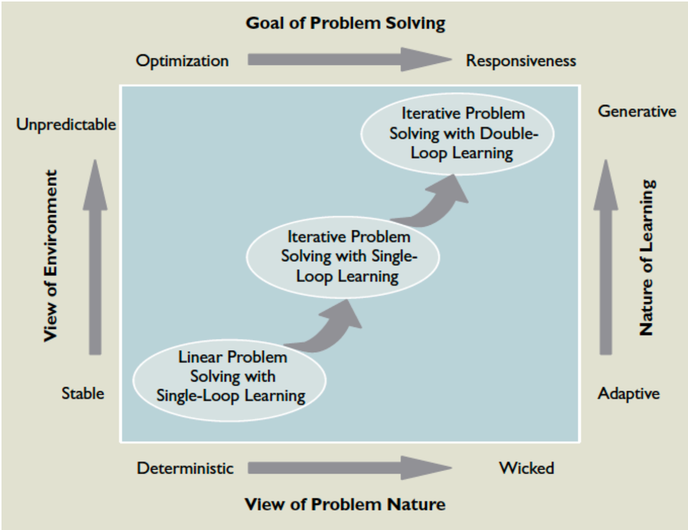

Figure 5: Different types of learning processes within product development (Nerur & Balijepally, 2007, p.82)

Takeuchi and Nonaka (1986) describe two types of learning – multilevel and multifunctional. Multilevel learning distinguishes the different types of learning in an organization. It occurs at an individual, group, and organizational level. Multifunctional learning is aimed at encouraging employees to gain knowledge of other areas beyond their function, as it was demonstrated to improve the performance of teams. Transferring learnings and knowledge across the organization is equally important and can be done through, e.g., standardization of improved ways of working (Takeuchi & Nonaka, 1986).

In a study by Lawson et al. (2009), it was found that informal socialization mechanisms were more effective in transferring knowledge and building relationships than formal activities. The formal activities were still needed as they were antecedent of informal ones and created the basis for personal connections (Lawson et al., 2009).

2.2.3 Iterative and Incremental Development

All of the agile practices, e.g., Scrum or XP, contain an iterative and incremental aspect to their development processes (Dybå & Dingsøyr, 2008). It is a crucial component to agile and revolves around having a set development cycle in which teams develop features that at the end of an iteration are presented for the customer (Ambler, 2009). Highsmith and Cockburn (2001) discuss it from a feedback, planning, and prioritization perspective. Feedback from customer allows dynamic prioritization, i.e., planning and prioritizing what needs to be done until the next feedback session (Highsmith & Cockburn, 2001), and the processes of going back and forth between development and customer feedback are crucial for teams to deliver optimal value (Gren & Lenberg, 2019).

Williams (2012) found that short iterations – meaning 30 days or less – is one of the most important of the agile principles for development teams in their efforts. The incremental aspect is related to developing a product or solution and at the end of each iteration, send out a finished feature resulting in a continuous stream of value instead of the traditional plan-driven development process (Ambler, 2009). In practice, iterative and incremental development can be done through prototyping, experimentation, market test, and software delivery, coupled together with a high degree of customer collaboration (Rigby et al., 2016a).

2.3 New Contexts for Agile Development

The majority of literature covering the topic of agile is linked to either software development or commercial practices, or a combination of both (e.g., Ambler, 2009; Conboy, 2009). Agile practices are usually confined to the context of small innovation teams within software development (e.g., Rigby et al., 2016a). More recently, however, agile research has expanded beyond the contexts mentioned above to other areas including agile in manufacturing companies (Cooper, 2016), large-scale agile (Dikert et al., 2016) and agile free of any contexts (Schuh et al., 2018). Schuh et al. (2018) provided a basis for a transfer of agile to product development through the identification of underlying effects of activities in software development that enable agility and how they should be (Table 1).

Target Content	*Target Level*	**Target Content**	*Target Level*
Customer satisfaction	Maximize	Misdeterminations	Minimize
Target orientation	Maximize	Execution errors	Minimize
Profitability	Maximize	Relation to reality	Maximize
Productivity	Maximize	Motivation of people	Maximize
Complexity	Minimize	Organizational knowledge	Maximize
Reaction rate	Maximize	Overload of people	Minimize
Uncertainty	Minimize	Local knowledge	Maximize
Share of knowledge	Maximize	Personal independency	Maximize

Table 1: Identified effects in agile software development (Schuh et al., (2018, p. 23)

Implementing agile into traditional organizations has its challenges, according to Boehm & Turner (2005). The challenges are both perceptual and technical and are further divided into three categories – development process conflicts, business process conflicts, and people conflicts. The issue consists of having to merge agile's lightweight development with standard industrial processes without decreasing agility and undermining years' work of creating and improving development processes (Boehm & Turner, 2005).

2.3.1 Agile Development in Manufacturing Companies

The inherent differences between software and hardware development, such as software being immaterial, creates a difficulty for transferring agile development directly to hardware development (Schuh et al., 2018). Similarly, Cooper and Sommer (2018) stated that implementing agile development in a non-software environment has its challenges, due to prototyping and incremental development being slower and resource heavier. In software development, companies can continually provide value to their customers through incremental development and sending out updates – a reason why many companies choose to adopt agile practices as it creates a basis for continuous integration of updates (Dikert et al., 2016). Agility in hardware development is affected by product maturity – in the early concept phases, the ability to make changes is high, while in later stages, it is arduous due to more documentation and set supplier contracts (Schuh et al., 2017a).

Laanti (2016) found that agile development coupled with lean development can bring benefits to companies, but it is important to consider the generally applicable development process of hardware and use feedback to increase the maturity of the product. Incremental development has clear benefits for software development, and Laanti (2016) argued that the effects of iterative and incremental development within hardware could be as beneficial. Using a cadence (a set tempo of, e.g., six weeks á iteration) and synchronization further enhances agility (Laanti, 2016). Synchronization is described as a cross-functional integration point where all stakeholders, including suppliers, participate (Laanti, 2016).

Agile development has many benefits, and so does traditional development, and combining them could create synergies that overcome their shortcomings (Cooper & Sommer, 2016). Combining agile development with, e.g., the stage-gate model will solve the issue (Cooper, 2016). According to Cooper and Sommer (2018), the hybrid was revealed to improve team communication compared to traditional ways of working (Figure 6).

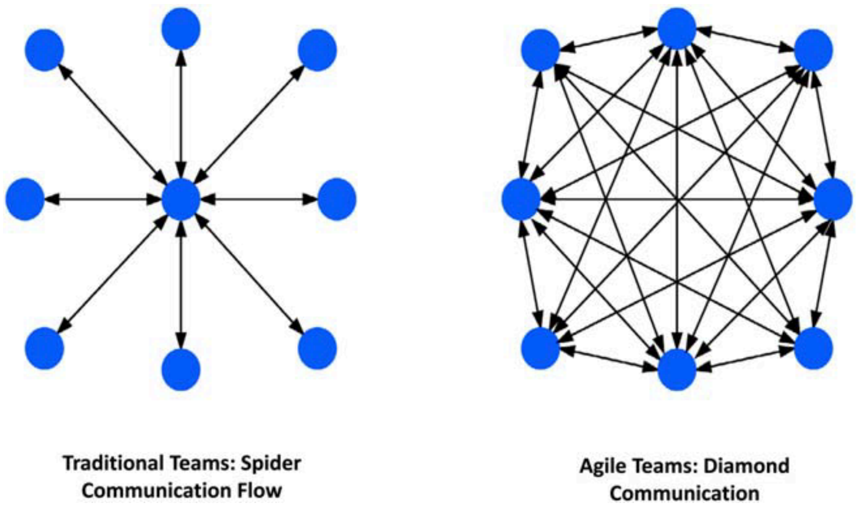

Figure 6: Communication in traditional teams versus agile teams (Cooper & Sommers, 2018, p.9)

Manufacturing companies can overcome their hardware limitations through virtual development (Rauh, 2003). Virtual development is to use software simulation and other tools to develop a product in a virtual environment and is prevalent in the automotive industry (Rauh, 2003). According to Becker et al. (2005), the introduction of virtual simulations into an organization can have an impact on its product development through improved problem-solving, early virtual prototype experiments, and feedback loops. Simulations can be produced and iterated numerous times, allowing for experimentation that otherwise would be impossible

(Becker et al., 2005). Schuh et al. (2017b) likewise found that virtual tools have an overall positive impact on product development for manufacturing companies by enabling iterative development.

Manufacturing companies can also utilize platform and modularization technologies within product development. Platforms were described by Robertson and Ulrich (1998) as a collection of properties shared between products. Modularization was described as a way to divide sections of development into smaller, interchangeable, and independent modules (Robertson & Ulrich, 1998). The benefits of using these technologies and techniques were stated to be: decreased complexity, faster derivative development, and cost-efficiency (Magnusson & Pasche, 2014).

2.3.2 Large-Scale Agile

Automotive companies are often large companies with several thousands of employees. As previously mentioned, implementing agile at a large scale has its challenges (Dikert et al., 2016). Dingsøyr et al. (2014) define large-scale agile in relation to the number of teams in an organization where 2-9 teams constitute as large-scale, and 10 or more teams constitute as very large-scale agile. According to Rigby et al. (2016a), agile is often confined to small co-located teams with a focus on innovation. The right conditions for agile consist of unstable market environments; ability to closely collaborate with the customer; complex tasks; ability to modularize work; and an environment suitable for experimentation (Rigby et al., (2016a). Increasing the number of agile teams in an organization increases the agility, but it is imperative to integrate the teams to the rest of the organization (Rigby et al., 2018).

Scaling up agile has its challenges and of the most notable being coordination in a multi-team environment, according to Dikert et al. (2016). It was found that challenges arose when teams were to work with other teams in a larger context. Although responsiveness is improved at a team-level, the rest of the organization can have a difficulty to keep up, leading to challenges with inter-team collaboration. Another challenge was described within the context of self-organizing teams – they are initially given a high degree of autonomy which improves responsiveness, but in a large-scale setting can lead to sub-optimization on a team-level and less focus on over-arching organizational strategic goals (Dikert et al., 2016).

3 Case Description

The Volvo Car Group is a major automobile manufacturer, with a history spanning almost 100 years (Volvo Cars, 2019). To compete on a global scale, Volvo had to be able to develop new cars rapidly, with high-quality, and in alignment with their customers' needs. In this development process, the department of Efficient Structures played a major role. Specialized in developing the internal frames and exterior design of the car, the department also played a central role for all other functions. Almost all other development areas physically connected to the frame, and due to this needed to coordinate with Efficient Structures. What made efficient structures unique however was their hardware specificity, with minimal work focused on the development of software. This presented a unique opportunity, allowing a unique insight into a large-scale complex hardware developing organization up-close.

3.1 Efficient Structures

The organization had a long history, with roots back to the origins of Volvo. In this history, the development of car bodies had changed drastically, initially working with physical clay and wood models, and with the advent of powerful computation, virtual car development. A significant point of pride within the organization was the fact that almost all body development was done in-house, which made the organization unique in the Volvo Car Group. Furthermore, the emphasis on in-house development meant that what is now known as Efficient Structures had long experience in the development of car bodies, and the staff members could be considered specialists. Communication and collaboration were central to their work, and many other departments considered them highly agile. As expressed by developers, much pride was taken in their work, and the unofficial motto for the organization was that they always deliver.

3.2 Agile Transformation

During the fall of 2018, Efficient Structures initiated large agile transformation, transitioning from a traditional matrix-organization to an organizational structure based on SAFe. The transformation was part of a larger Agile transformation at the Volvo Car Group, in which all developmental areas took part. The transition was one of the largest Agile transformations in Swedish industrial history and lacked clear parallels.

Prior to the transformation, the organization focused on projects based on the development of whole cars. Developers were organized according to specialization areas, such as the front floor, rear, and top-hat. Developers worked in several projects simultaneously, and their area of specialization gave them an area of focus within the project. These projects were long-running and only closed upon the completion of the car.

Following the transformation, there was a shift towards Agile Release Trains (ART), in which specialization areas were grouped into teams. The teams utilized SAFe tools, such as PI-planning, sprints, and Kanban-boards to plan, prioritize, and coordinate work with other stakeholders. It is worth noting that not all tools were found to be useful for all teams, and therefore some adaptions were made to better suit their work-environment.

At the time of writing, the organization had not fully adapted to the SAFe, and those involved expressed a variation in attitudes. A major point of contention was the presence of parallel organizations, with traces of both the new Agile organization existing alongside the older line-organization. However, many of those involved saw clear benefits in the transition and noted both compatible and incompatible areas between the traditional organization and the new, SAFe-organization.

3.3 Development Process

In the development of a new car there were three distinct phases, as seen in Figure 7; the concept-phase, the engineering-phase, and the industrialization-phase. Each phase could be distinguished by the main focus of the work, and the stakeholders involved. Furthermore, in the later phases, the designs became increasingly more concrete and less open to change.

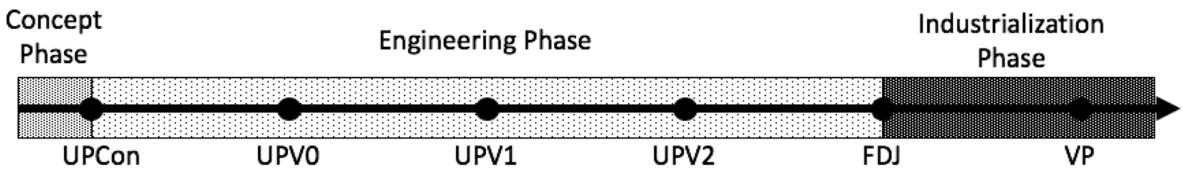

Figure 7: A general visual representation of the phases that development phases

In this work, developers, designers, and other stakeholders iteratively worked in their development, following the general development cycle that can be seen in Figure 8. This cycle ran over most phases but was most heavily followed up until FDJ. This process started when the design team released a design (DSM), which the template team thereafter implemented and synced into templates. These design-templates were general directions for development each ART, and developers used the templates to adapt and update parts. Throughout these updates, technical information was given back to design, which eventually leads to a new design release.

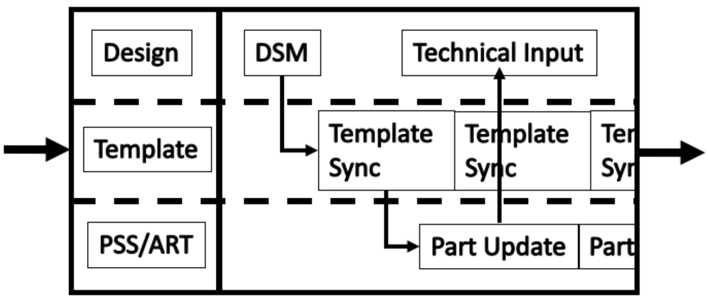

Figure 8: General development cycle.

During the concept phase, the development focused on the general concepts and designs and ended at UPCon. UPCon was a set gate, at which point no conflicts between parts should exist. For Efficient Structures, this meant that a "complete" body was released, with all necessary attributes covered, including manufacturing. The structure is then released to manufacturing engineering and CAE for simulation and testing. Several weeks after this release, Efficient Structures were given feedback regarding the structure.

Simultaneous to UPCon, the design-department released a new design of the car. This new release is based on the technical input given during part update, and then became the new "main design" for which development was done. This iterative process followed the development cycle shown in Figure 8, and continued, in some form, through all of the following phases.

Following the concept-phase, the engineering-phase was initiated. Throughout this phase, the focus lay on detailed engineering, verification of concept, and working against manufacturing. This phase contained four major loops, during which the maturity of the product increased, and different aspects of the design were engineered. Leading up to UPV0, the concept is verified, and its fit is ensured with the body-factories and plants. Then, during the UPV1-loop Efficient Structures adapt designs for die-pressing, work is in testified with stakeholders, including body-factories, and final suppliers are chosen. Following this, UPV2 is when suppliers start ordering tooling, and issues are dealt with. Throughout this process, continuous iterations are used to ensure correct and usable feedback is available for all stakeholders.

After several iterations, a final data judgment (FDJ) is made and the industrialization-phase starts, at which point all design work is over. At this point, tool creation is started to ensure that tools are ready for the start of production, VP. Within Efficient Structures, this was a particularly time-consuming process, as die-casting tools needed several weeks to cooldown correctly. During this time, changes were incredibly difficult to do and were heavily discouraged. Following FDJ, the majority of the development teams moved onto other development project or ARTs, leaving a skeleton crew to manage any future changes or updates necessary.

3.4 Stakeholders

A noticeable feature of Efficient Structures and the teams within was the number of Stakeholders that needed to be taken into account. These can generally be divided into three overarching stakeholders. It is important to note that sometimes these stakeholders can overlap, for example with internal suppliers that compete for contracts, or outside consultants that function as internal stakeholders.

Internal stakeholders work closely with developers, both geographically and organizationally. These tend to be other developers and attributes such as Noise, Vibration, and Harshness (NVH), Structural Calculation, and Safety. There are also stakeholders that fall under the Volvo Car Group-umbrella but are not integrated within the Efficient Structures organization. Examples of these are Volvo Cars Body Components (VCBC), Electronics, Lights, and other functions that connect to the frame but operate relatively independently. It is also worth noting that while the direct stakeholder might exist under the Volvo Car Group umbrella, work might be outsourced to external suppliers.

External Stakeholders are entities that exist wholly outside of the Volvo Car Group-umbrella and usually suppliers, customers, or governmental agencies. There is much variation in the relationship between Efficient Structures and external stakeholders, with some working intimately, almost like an internal stakeholder, while other external stakeholders might be quite distant.

In their work with stakeholders, a significant area for Efficient Structures were the ‘preparation-meetings.’ These weekly meetings were held weekly and lasted several hours. Each article was analyzed for prospective issues, with their respective development teams and stakeholders present. Visual representations of the articles were available for manipulation and change, and any issues or corrections were documented in “living documents,” which were PowerPoint slides with pictures of changes.

4 Research Methodology

In this section, we present the frameworks and methods that were used to accommodate our research process to fulfill the purpose of our study.

4.1 Research Design

This study utilized the systematic combining approach. Systematic combining is *"a process where theoretical framework, empirical fieldwork, and case analysis evolve simultaneously, and it is particularly useful for development of new theories,"* (Dubois & Gadde, 2002, p.554). It features a non-linear and non-positivistic approach (Dubois & Gadde, 2013). Research is done through matching, directing and redirecting, i.e., going back and forth between framework, empirical fieldwork, case, and theory (Figure 9). Case studies are a central part of systematic combining – they provide the possibility to develop theory from a specific context as theory cannot be understood without empirical data and vice versa (Dubois & Gadde, 2002).

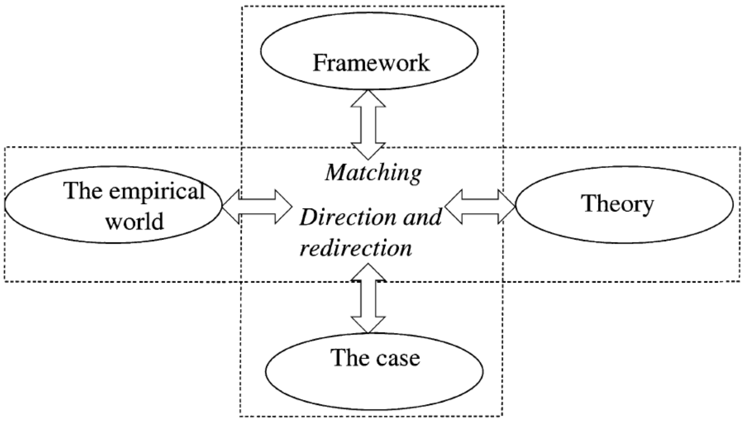

Figure 9: The process of directing, re-directing and matching (Dubois and Gadde, 2002, p. 555)

Furthermore, the emphasis is on single case studies where researchers should use a context's uniqueness as a strength by interpreting specific situations (Dubois & Gadde, 2002). Yin (1994) stated that cases do not provide a sufficient basis for generalization and that multiple case studies are preferable, but Dubois and Gadde (2013) argued that the usefulness of a case lies in its specific context. Similarly, Alvesson and Kärreman (2007, p.1265) stated that *"We emphasize the potential of empirical material as a resource for developing theoretical ideas through the active mobilization and problematization of existing frameworks. In particular, we point to the ways empirical material can be used to facilitate and encourage critical reflection: to enhance our ability to challenge, rethink, and illustrate theory,"* Systematic combining is used to refine existing theories through development rather than creating new ones through generation (Dubois & Gadde, 2002) – in line with the purpose of our study.

4.1.1 Summary of Research Process

The research process can be described as two distinct phases, with a theoretical beginning via empirical inquiry to an analytical ending. A visual representation of this can be seen in Figure 10.

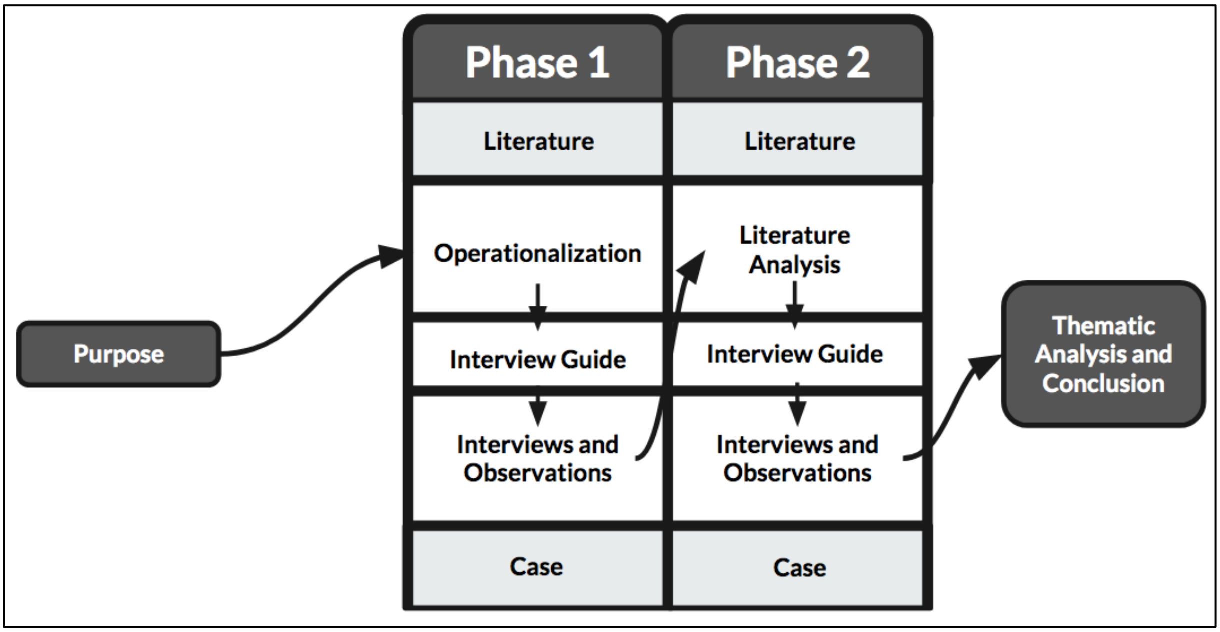

Figure 10: A visual representation of the research process

We set out to explore the area of agile hardware development due to a perceived contrast to the typical application of agile within software development. However, to create a sufficient knowledge-base and lens through which to view a case, a thorough exploration of existing agile development theory was conducted. During this exploration, neither a clear connection between agile and hardware or a clear definition of agility could not be found. It was clear that agile embraced a humanistic approach to development, with people and their creativity in the center, but most articles within the subject used a unique definition or forsook a definition and simply referred to the agile manifesto. Thus, to be able to approach agility effectively, a clear operationalization was needed.

As shown in Figure 10, the first step in phase 1 began with an operationalization. By synthesizing existing definitions of agile and agility within agile development literature, an effective operationalization of agility could be created. With this operationalization as a central tenant of the study, a relevant case-study was necessary to start gathering data. Efficient Structures at Volvo Cars was chosen as they solely develop hardware components in the cars but were described as one of the most agile functions at the company.

After establishing contact with Efficient Structures, preparations for data collection were made. Following an unstructured interview with IF6, we were invited to observe and take part in a large planning event, serving as an introduction to the organization. Thereafter, by utilizing the operationalization, an interview guide, that centered on the agility of Efficient Structures was constructed, alongside some general questions regarding work-processes and the agile transformation (the initial interview guide can be seen in Appendix A). When conducting the interviews, a deliberate choice was made to keep the interviews semi-structured, as we felt that by limiting the areas of discussion would inhibit a free exploration of agility. We encouraged the interviewees to freely discuss pain-points and tangential areas regarding the questions in an effort to capture as much information as possible.

During the interviews, when the interviewees were asked about agility, we noticed a trend in their answers. They would often raise certain aspects, such as teams or communication, that enabled or hindered agility, and that agility was the product of several factors, rather than simply an inherent trait. Therefore, parallel to conducting the interviews, we felt that it was prudent to return to literature and analyze what agile development literature considered central to agility. This began phase two, as seen in Figure 10. We chose to analyze the literature through the lens of our operationalization, where statements from literature were compared to "How does this affect agility?" and if relevant, was noted. These themes were thereafter grouped according to several categories, which we felt constituted the base which agile development literature considered central to agility.

Utilizing this basis, we updated the interview guide to focus on the areas described in literature (as can be seen in Appendix A.3). With this realignment of the interview guide, we chose to remove questions regarding the general opinions of agility and the agile transformation. These were considered to fall outside of the scope of the study. During the following interviews we noticed that the areas raised by literature often elicited good discussions, and other areas that the respondents felt were more relevant were raised.

Following the interviews, it was necessary to systematically capture the discussions surrounding agility, which led us to utilize a thematic analysis. This thematic analysis yielded several areas we felt were noteworthy and were worthy of comparison to literature. The problematization and discussion of the overlap and discrepancies constitute our contribution to the discussion of agile hardware and provides some useful areas for firms to consider when adopting agile frameworks.

4.2 Relation to Previous Theory

A study of the previous theory of agile development was continually conducted to develop the theoretical framework. The following search engines were used: Chalmers Library, Google Scholar, Harvard Business Review and Research Gate. The supervisors also provided articles in relevant areas. Chaining, i.e., searching for papers in the reference list, was used. The sources used during this study consisted of scientific papers and the agile manifesto (web page). An active choice was made exclude existing agile practices due to their low degree of academic scrutiny (Conboy, 2009). The study could be divided into several minor studies.

The first study consisted of finding the core of agile development using the search words: *agility, definition of agile, core of agile,* and *review of agile*. The results consisted of many definitions of agility and guided the operationalization of agility.

The next study consisted of finding common practices used in agile. The search words were *agile practices, self-organizing teams,* and *iterative development.* The findings were used to guide the interviews by creating questions based on the found characteristics and their effect on agility at Efficient Structures.

The last study consisted of finding agile in other contexts, using the search words *large-scale agile, agile in manufacturing,* and *agile hardware development*. The results guided the discussion (chapter 6) in describing the challenges and possibilities there for agile in other contexts.

4.3 Data Collection and Analysis

Interviews provide a way to capture personal experiences and reflections (Bryman & Bell, 2011) and were, therefore, the primary data collection source, as agile is heavily focused on softer values such as people, informal processes and motivation (Ambler, 2009). Observations were conducted parallel to complement the interview data. The data collection focused on how employees work in practice and their experiences. The data were systematically coded and used as a means to confront and develop theory (Dubois and Gadde, 2002). Consideration was made to active and passive types of data – active data is associated with exploration and passive data comes from what the research set out to find (Dubois & Gadde, 2002).

4.3.1 Interviews

The interviews were held at the departments of Efficient Structures and Mechanical Integration. All interviews were pre-booked, and both researchers were present at the interviews. One was responsible for asking questions while the other took notes. All interviews except the two with IF6 and IF10 were recorded.

As the systematic combining approach was utilized, the interviews evolved alongside the matching and redirection between literature and empirical data. In general, it can be said that two types of interviews were held, with the first type being unstructured interviews. An unstructured interview is similar to a conversation with very few prepared questions (Bryman & Bell, 2011). The interview format allows for passive interviewers whilst an active interview subject provides information. They were used as a means to develop a better understanding of the organization of Efficient Structures and gain access to potential interview subjects. Furthermore, the unstructured interviews provided general knowledge of how the organization worked in practice, a basis for the semi-structured interviews, and information on factors affecting agility at the department. Unstructured interviews were held with IF6, and IF10 during the early phases of the empirical data gathering, with only a few prepared guiding questions. The interviews were generally limited to 1 hour and were always conducted face-to-face in order to facilitate better discussions.

The unstructured interview with IF6 was held early in phase 1, as an introductory meeting for the researchers into the organization. An explanation of the purpose of the study was given, alongside a proposed method, and a discussion surrounding secrecy, ethical considerations, and other factors that could affect both the study and the organization. Furthermore, a discussion was held regarding willing and interesting interview-subjects, which was vital for the commencement of the data collection.

The unstructured interview with IF10 was organized with the aim of gaining the perspective of a stakeholder at the end of phase 1. The choice to use the unstructured interview form was made to enable organic discussion of agility, and the probing of aspects IF10 might have considered. Aside from this information, recommendations of future interview subjects were made.

Following the unstructured interview with IF6, semi-structured interviews were held with IF1, IF2, IF8, and IF11. As described by Bryman and Bell (2011), semi-structured interviews are guided by prepared questions, but are unstructured enough to allow follow-up questions. This was considered appropriate, as open discussion was needed to capture experiences, knowledge, and reflections of what the informants considered affected agility in their department. The format makes the interviewers more active, without significantly hindering the interview subject from also actively providing information. In this vein a general interview guide was constructed to ensure similar structure across different interviews (an example can be seen in Appendix A). The interviews were booked to take between 30 minutes and one hour, and outside of a few rare cases, this time-limit was strictly kept to. Like in the unstructured interviews, only face-to-face interviews were conducted, as this was the most conducive to open discussion, and also important to capture the atmosphere and demeanor of the interview subject.

During the initial semi-structured interviews, certain aspects of the interview guide were noted to be superfluous, such as questions regarding their perception of the agile change process. In phase 2, following a return to theory as described in chapter 4.1.1, an update of the interview guide was necessary. The updated interview guide added questions regarding agile factors found in theory, such as teams, iterative and incremental development (the interview guide in Appendix 8 displays this update). These questions were used in interviews with IF3, IF4, IF5, IF7, IF9, and IF12.

A total of 12 interviews were held with as many informants. A brief description of the informants is provided in Table 2. Potential informants were chosen together with a contact person at Volvo to ensure a suitable mix of viewpoints and were contacted by email. Those who volunteered were interviewed. Ten informants were from Efficient Structures and two from Mechanical Integration. Due to time restrictions, no other major stakeholders were studied regarding their work toward Efficient Structures. This potentially led to a less varied view concerning factors affecting agility at Efficient Structures.

Informant 1 (IF1) was a Scrum Master at Efficient Structures.
Informant 2 (IF2) was a Product Owner at Efficient Structures.
Informant 3 (IF3) worked with Mechanical Integration.
Informant 4 (IF4) was a Scrum Master at Efficient Structures.
Informant 5 (IF5) was a Change Leader and worked toward Efficient Structures.
Informant 6 (IF6) was a Manager at Efficient Structures.
Informant 7 (IF7) was an Agile Coach and worked toward Efficient Structures
Informant 8 (IF8) was a Developer at Efficient Structures.
Informant 9 (IF9) was a Developer at Efficient Structures.
Informant 10 (IF10) worked with Mechanical Integration.
Informant 11 (IF11) was a Product Owner at Efficient Structures.
Informant 12 (IF12) was a Product Owner at Efficient Structures.

Table 2: List of informants

The interviews were coded and analyzed using a systematic approach called thematic analysis – more specifically; an interpretative thematic analysis was conducted. It is a systematic coding method and analysis tool used to capture experiences and reoccurring themes and interpret them beyond their description to find an underlying meaning (Smith, 2015). The interview data were coded by following Braun and Clarke's (2005) six steps for conducting a thematic analysis. The steps can be summarized as (1) familiarization of data, (2) coding, (3) theming, (4) reviewing, (5) defining and naming, and (6) writing the report (Braun & Clarke, 2005). Table 3 exemplifies how the coding was conducted.

Quote	**Simplified Meaning**	**Sub-Theme**	**Theme**
"We have a lot of CAE-engineers that are close to us, very close. That cooperation is very important for our quick feedback loops."	Collaboration with CAE-engineers enables fast feedback.	Collaboration across functions	Collaboration

Table 3: Thematic analysis coding example

4.3.2 Participant Observations

Participant observations were described by Bryman and Bell (2011) as a qualitative research method where researchers immerse themselves in a case during a period of time by conducting observations, being part of an organization, have conversations with people, listen to conversations between employees, and actively ask questions on the investigated topic. There are many parallels between participant observations and ethnography (Bryman & Bell, 2011), while Van Maanen (1979) and Burawoy (1998) claimed that ethnography describes a broader and more thorough research approach.

This study used an ethnographic-inspired approach guided by the participant observation research method as described by Bryman and Bell (2011). We immersed ourselves in the research and development organization within Volvo Cars in Gothenburg. Unlike the interviews, participant observations were not exclusive to Efficient Structures and neighboring functions. Approximately three days per week were spent at the company from January to May 2019. Face-to-face conversations and email correspondences were had with many employees on different levels of the organizational hierarchy. In many cases, we acted as confidants for the employees where they saw a possibility to vent out their concerns and experiences of the SAFe implementation. These discussions and conversations added to our insights of the agile transformation at Volvo and the subsequent challenges, as well as guided our interview questions.

Observations were conducted during meetings, planning events, and everyday work, which led to an added understanding of how interdependencies were managed and how virtual tools were being used to simulate the manufacturing processes. This potentially led to a lesser understanding of the effect of the meetings. During the observations of specific events, field notes were taken and reviewed.

Participant observations are often combined with collection of documents (Bryman & Bell, 2011). Through our formal access, we were able to take part in internal documents concerning agile development at Volvo. These documents added to our knowledge of the development processes. They described the different roles within the organizations, the purpose of the agile transformation, and so on. Finally, the participant observations were conducted by the two of us. We were able to discuss our experiences and findings with each other throughout the study. It led to improved problematization and critical reflection of the study that otherwise would not have been possible.

4.4 Quality of Research

Trustworthiness can be used to measure the quality of a qualitative study (Shenton, 2004). It is divided into four criteria – credibility, transferability, dependability, and conformability. Credibility is a measure of the believability of the findings, and was ensured by following Shenton’s (2004) examples. Well-established research methods and analysis tools were used. Interviews were conducted using an interview-guide which was continuously refined, and observations further supported the collected data. Regular meetings with supervisors were held to discuss the progress and direction of the thesis. An examination of existing theory was also conducted.

Transferability is the extent to which the research can be applied to other contexts (Shenton, 2004). Although transferability is connected to a positivistic research approach (Dubois & Gadde, 2013; Shenton, 2004) and that qualitative research results are limited to their specific context (Shenton, 2004) – transferability was achieved through the use of the operationalization of *agility* in guiding the research. The concept of agility is needed for the adaption of agile to other contexts than software (Conboy, 2009).

Dependability was ensured by describing the methods used and reflecting upon their strengths and limitations. The appendices show how the interview guide changed during the study. The used methods were verified with a supervisor. It is to provide a possibility for other researchers to assess the quality of the research process (Shenton, 2004).

Conformability relates to the subjectivity of qualitative research as complete objectivity is impossible (Shenton, 2004). To ensure conformability, researchers should acknowledge their biases and describe ways it could have affected the research. Although a specific analysis method was used, the results have been colored by our own beliefs and interpretations. Similarly, the interviews were translated to English resulting in another level of interpretations. The informants were contacted and allowed to confirm or reject our interpretations and translations. Further confirmability was achieved through the supporting observations and combining the interpretations of two researchers.

4.5 Ethical considerations

Diener and Crandall (1978) describe four types of ethical considerations in research – harm to participants, lack of informed consent, invasion of privacy, and deception. Harm to participants concerning physical health, career development and so on is unacceptable (Diener & Crandall, 1978). To ensure this was fulfilled, all the informants' anonymities were protected. The informants volunteered and consented to the interviews, the recording of the interviews, and how the data would be used. They were allowed to end the interviews early or withdraw their contribution. The topics discussed during the interviews were limited to work processes and not on any matters of privacy. Lastly, to avoid deception, the purpose of the interviews and study were openly described in the email conversations preceding the interviews and at the start of each interview (Diener & Crandall, 1978).

5 Results

Research questions 1 and 2 are answered in this chapter. The research resulted in a definition and operationalization of the concept of agility and the finding of six central themes affecting agility at Efficient Structures.

5.1 Defining and Operationalizing Agility

A central idea of this thesis is agility within the context of management theory. Its importance in adapting and translating agile to other contexts has repetitively been pronounced. Several definitions and argumentations have been presented in the theoretical framework. After consideration of each of these, a decision was made to define and operationalize agility as **responsiveness to change**. The definition is the same as Gren and Lenberg's (2019) and similar to their argumentation, we believe all agile practices and principles aim to increase an organization's or a processes' responsiveness to change. Therefore, it can be concluded that agility is not a binary property but rather a spectrum. Defining agility was needed as a way to translate agile development to another context and to operationalize what constitutes as agility for the next major part of the results. The terms agility and responsiveness to change are in the following sections interchangeable.

5.2 Factors Affecting Agility at Efficient Structures

Six themes, as presented in Figure 11, were found using thematic analysis as exemplified in Table 3 that seemed to affect the responsiveness to change at Efficient Structures. A selection of quotes and a description of each theme are presented below.

Figure 11: The six themes affecting agility at Efficient Structures

5.2.1 Communication

Communication is one of the more prevalent themes, with most other themes tying into it. It includes formal and informal communication on all organizational levels and supplier contact. Several informants described it as a vital part for their agility as it enabled and improved feedback, collaboration, and problem-solving. Communication was needed in order to manage the complexity of developing a car. IF9 stated, *"Communication is paramount. It would not work without it. You have to communicate with your stakeholders. The cars are so complex that it would not work otherwise."* Communication was discussed in terms of both being crucial for the overall development and other minor areas such as problem-solving.

Communication – and the resulting agility – was affected by geographical distances and organizational aspects. IF3 stated, *"Communication is best when it occurs eye-to-eye,"* and described a benefit of closeness. Similarly, IF8 stated, *"That's why it is good that [developers and manufacturing engineering] sit close to each other, it is very easy to have direct communication."* Coupled with a statement by IF8, *"It is due to geographical distances [that communication is handled through official channels],"* it indicated that informal communication occurred when geographical distances were short and vice versa regarding greater distances. The distances were also described from a supplier perspective. As Efficient Structures have many internal suppliers, e.g., VCBC, they could directly communicate and collaborate with them. Many informants stated that it was one of the factors to enable agility the most. Communication with external suppliers was described as always being slower.

The informal pathways were described to improve their agility through quicker feedback loops were described as a crucial part of communication. IF5 stated, *"Within the teams, it eventually boils down to 'who do you know?' 'how do we get these things done?'… [At Volvo] A lot of people have been here for a long time, people know each other, you solve problems because you know each other, which is also a collaboration or stable-structure question. If you know someone, you can call NVH and explain your problem."* IF5 discussed that issues could be solved quicker if one has the knowledge of whom to talk to, which many do due to having worked at the company for a long time.

Several informants described the introduction of daily stand-up meetings as an improvement to their communication. To communicate daily on the tasks at hand led to less rework and improved problem-solving. IF8 stated, *"One [developer] got the question 'What are you working on?', and he replied, 'I'm working on this, and am concentrating on these things'. Then someone else said 'hey, I did that yesterday'."* However, this type of information sharing was not as valuable when not all members participated. IF3 stated, *"If not everyone is present [at the meetings] when we discuss problems, we will have to do it one more time,"* leading to rework and loss of responsiveness to change.

Another recurring theme was visual communication. It entailed the ability to show, discuss, and visually alter issues, generally through the usage of CAD-software. Throughout the interviews, it was seen as a significant benefit for communication and collaboration, allowing changes to occur in 'real-time'. IF9 stated, *"You usually look at the articles. If you make a change, you see it instantly. During the packing meetings, almost everything is visual… it is controllable* *pictures. You can touch and move things, testing, and so on. You get a whole other understanding when you see the things. It speeds up communication."* It was further emphasized by IF5, who presented an advantage they saw in hardware development, *"... we don't develop ones and zeroes. We develop things that can be seen."* To be able to visually present work in progress and instantaneously receive feedback was described by many to improve their agility.

5.2.2 Iterative and Incremental Development

Throughout the design process of a car, a common theme is the iterative work process, in which a model is created, tested, revised iteratively. As is mentioned by IF9: *"You loop quite a lot with CAE. You make a small change here, send it to them, they calculate. 'Does it look better?' 'Does it look worse?' That's how you continue until you have reached the desired results."* Iterative development affected Efficient Structures responsiveness to change by enabling a way to receive feedback on their work and provide an opportunity to alter the course.

There were different levels of looping. For example, IF4 stated, *"We used to have large deliveries to [stakeholders]. We would send a large packet, and they would look it over during a couple of weeks, and then came back with feedback."* It was discussed concerning the different 'releases' which acted as fixed iteration points were the progress is assembled digitally and tested. However, with the new SAFe way of working, they instead focused on smaller feedback loops. This was described to improve agility as the smaller volumes of information made it simpler to make changes earlier. For smaller, more local issues, the looping occurred with quicker feedback, as stated by IF11, *"We have a whole number of [CAE Analysts] who sit close to us, very tight. This collaboration is very important for our quick feedback loops."* In this example, smaller changes are handled informally, without large releases, and it led to improved agility.

5.2.3 Complex Hardware Development

In the data gathering, several areas were identified that affect the responsiveness to change which could be derived from the complexity of automotive development. The first area consisted of the time plans which guide the work at Efficient Structures. The time plans can be altered very slightly. IF2 stated, *"because the articles are so large and heavy within efficient structures, with long lead-times and so on, having a time-plan is very important for us."* Many informants shared this sentiment, as IF3 *"we are very locked in our time-plans and projects."* The time plans were described to have dual effects on agility. The first was that the time plans had set dates in which a car project enter different phases or gates. The different phases have a predetermined amount of agility of how much a car can be altered. The set time plans also did not allow for dynamic work processes. The time plans were rigid and difficult to change due to interdependencies, such as the time it takes to cast a tool die (up to 20 weeks).

Many of the described interdependencies relate to the need for communication – developing a car is an incredibly complex task. IF3 stated: *"A car is one of the most complex consumer products you can develop."* IF5 further emphasized this, *"it is the size and dependencies that is our challenge."* The informants stated that complexity impairs overall agility. IF2 said, *"If* *we get an issue, for example in a crash test, where we get deformation characteristics we do not like, then it is not always certain that [the issue we want to solve] is located there. It might be located somewhere else."* However, the complexity was mitigated within Efficient Structures by, as IF3 stated*, "the benefit at Efficient Structures is that they have always had full control of their processes, resources, very short paths to their suppliers and toolmakers. They get rapid feedback... [and] have long experience."*

It was also described that complexity was decreased and agility increased, through the use of platform and modularization technology. According to the informants, platforms allowed for a flexible design approach and provided a basis for the development. IF1 described that platforms enabled faster development as they only needed to scale components across car models, rather than developing them anew for every car. This is further emphasized by IF8, who claimed that it was vital to ensure that "*certain areas are platforms,*" as it allows a fixed point in a changing environment.

The different phases within their product development were described to affect agility indirectly, as mentioned above. IF8 stated that in in the early phases *"everything is possible."* However, as the development progresses, IF2 stated, *"it is difficult, as we need to update all [earlier work that is both done by the own team and done in collaboration with other stakeholders]."* In this vein, IF5 said that *"there are many benefits [of working agile], especially early on. [However] of course, in the end, when we manage to get a car into production, then we are not at all agile."* Summarized, in the early phases when everything is digital and conceptual, the agility is high. In the later stages when the cars enter pre-production, it is almost impossible to make any significant changes.

5.2.4 Knowledge and Learning

Organizational knowledge management was raised as a theme. In the discussions with informants, the ability to respond to change was often connected to having certain information or knowledge. Greater competence led to greater agility. This connection was exemplified in the example of an engineer with many years of experience could answer questions at the spot, while inexperienced engineers needed time to research the issue. The knowledge-level at Efficient Structures was described to be quite high – many of the issues they faced were reoccurring ones, and the car body components they developed did not change significantly from year to year. It has led them to build up a foundation of many knowledgeable workers. IF8 stated, *"usually we reuse an existing solution, with the attitude that we don't need to reinvent the wheel."*

Knowledge was described as a main factor and learning, i.e., developing knowledge and competence, was described as equally important. The consensus seemed to indicate that interpersonal learning was vital. As IF5 mentioned, *"When I do something and learn from a colleague it sticks so much better than when I sit in a 3-hour lecture."* This was further emphasized by IF12 and IF4 *"It is this thing with informal communication ... it is here the people learn. When you are a part of the daily stand-up, there can be a bit of [learning], that you pick up small things."* In this vein, the introduced daily stand-up meetings seemed to be a great forum where people, as IF9 stated, *"gain insight into what the [other team members] are* *working on. You raise things that you might not know, because other people are sitting on that knowledge, and haven't raised it because there was no appropriate forum."*

IF8 described the importance of receiving feedback for learning. During their work process, if suggestions relating to the article were communicated back to the original development team, improvements could be implemented. While this was not a problem for one-time mistakes, IF8 mentioned that this could be an issue for recurring solutions. Without the feedback, the issue could not be handled, leading IF8 to conclude that "*quality issues need to be easier to find."* The improvement of processes was described to be closely related to learning.

When discussing the role of information and knowledge spreading, the theme of t-shaped individuals was described, i.e., having both broad, overlapping competencies as well as specialized knowledge IF1 stated: *"We have moved away from [everyone can work on every part]. We have leading areas, where you have your expertise, but then there are other tasks others can do. You still have the case that every article is very specific. And you have experience within [the article]."* T-shaped competence was described as positive for agility. Employees with knowledge within many different areas combined with specialized competence were able to respond to change faster.

5.2.5 Teamwork

This theme includes the effect teamwork, and other people-centered aspects have on agility at Efficient Structures. Informants described teamwork as a means of enhancing agility by improving problem-solving and feedback loops. Important to note, SAFe was being implemented and enacted during the period when the interviews were conducted. It led to a more team-oriented development. IF8 described how they worked in teams, *"Usually it is the person you are sitting next to, so I can do an update [to a model] and if I am a bit uncertain, 'Does it look good?' Then I can tap them on the shoulder and ask him to check."* IF8 further stated that closeness affects the probability of teamwork and the ability to solve problems.

Another change that was introduced with SAFe was the inclusion of employees from other functions into the teams. The added cross-functionality was described to lead to quicker feedback between functions. IF9 stated, *"[manufacturing engineering] is now in our agile teams. Now they are very close if you want to ask anything."* To help each other within teams was not about doing someone else's work but rather to help and support. IF9 stated, *"The strength has always been that we can help each other, even if that doesn't mean that I draw someone else's parts."*

Decisions that were made independently by the teams were described to increase their responsiveness to change. This fell under the sub-theme of self-organizing teams. The consensus throughout all interviews was that decision-making is quite broad. However, it was limited by other stakeholders. As said by IF8 *"You can decide. However, that is based on: Is it possible to manufacture? Is it possible to pack? What does the attribute [leaders] say?"* due to a large number of dependencies present in the manufacture of the car, all stakeholder must be involved in critical decisions.

However, the developers were given the autonomy to freely take decisions, without always needing to escalate issues to higher levels of authority. This was supported by IF12, *"If I experience an obstacle, I can handle it. I don't go to my boss; I don't go to a project leader. I call the person, and we figure it out."* Engineers were allowed to act on their own and make decisions informally. In some cases, informal decisions were actively encouraged as it was known to speed-up information flow but, in other cases, the informal processes were discouraged.

Not all decisions were left up to the developers. As stated by IF3, *"Market-oriented trade-offs and similar [decisions] must be taken at a higher level,"* which described a limitation to the scope of the decision that could be taken. This meant that individuals and teams could handle specific, technical issues, but when a decision affected the end customer or another stakeholder, the question tended to be escalated. The large scale and complexity of the organization were described to affect self-organization negatively. IF5 stated that self-organization due to the large size and interdependencies in the organization. The intricate net of dependencies hindered effective self-organization.

However, two of the new work-processes that were introduced with the implementation of SAFe were described to decrease their agility. Although they were aimed at increasing self-organization and agility, the results were described by the informants to have an opposite effect. First, an event where the teams were gathered for one week to plan their future work for the next twelve weeks and to level their workload. It was stated that this was in some ways an unnecessary and wasteful procedure as Efficient Structures was a mature organization with thorough knowledge of their capabilities. The set plans hindered agility as when unforeseen and unplanned tasks needed to be included; there was little room for it. It was described that previously; they had been more adept at responding to change.

Concerning the practical work-process, a sub-theme of ownership was found. This includes both the perceived and practical ownership of tasks and articles. However, it is worth noting that ownership did entail not only formal ownership but also a personal attachment to the respective area of responsibility. IF1 best exemplified this: *"Some developers here have their articles and hold onto them with an iron grip. They have no trust for each other."* At Efficient Structures, ownership had a dual effect on agility. Ownership enabled specialization within certain areas and in doing so, increased agility. IF2 stated, *"The developers are responsible for a certain, specific area. Within these areas, they are the owners of their articles. Should an issue arise within an area, we would know which person to talk to and who has extra responsibility."* Ownership made it easier for engineers to know whom to talk to regarding specificities. However, the strict ownership led to limited diffusion of knowledge at Efficient Structures.

The implementation of SAFe led to changes in how they manage ownership. For example, as IF9 stated, *"previously, you were assigned a few articles that you were responsible for ... You were responsible for your articles and needed to ensure that they fulfilled their demands for each release."* This was further expanded upon by IF2, *"If everyone [would work on all articles] you would lose the aspect that you really feel a responsibility for certain articles."*

These changes led to initial decreases of agility but were described to have a positive long-term effect.

5.2.6 Collaboration

The theme of collaboration was one of the most thoroughly discussed and cited areas throughout the interviews. It was described as the key to achieving agility at Efficient Structures. Collaboration primarily took the form of cross-functional work and collaboration with stakeholders. A reason that this was so central for Efficient Structures was the significance given to the preparation meetings.

At the meetings, almost all stakeholders were present to discuss all matters of tasks regarding the specific areas. The meetings were described as the single activity which positively affected agility the most. IF9 stated, *"you really get a good dialogue around all involved that I know other groups don't have. It has worked very well for Efficient Structures."* The meetings allowed for instant feedback and discussions, through synchronous communication. IF3 stated, *"Everyone has the same attitude. If we have a small issue, we solve it right away. Then it's not a problem anymore."*

Furthermore, the general attitude at Efficient Structures seemed to understand the necessity of compromise, and do not see other stakeholders as competitors, but rather collaborators. As IF8 stated, *"I don't get 100% of what I want. I need to compromise towards [the other stakeholders]."*

Another area was the difference in collaboration between internal and external stakeholders. The consensus seemed to be that having in-house processes and working with internal stakeholder, responsiveness to change is significantly improved. As IF3 stated, *"Some departments place all their work with their suppliers, and then it instantly becomes much slower… More difficult to get answers to questions and so on."* Furthermore, IF2 said, *"thanks to the fact that we have this contact with our suppliers and factories, we feel that we are quite prepared to answer to changes."* Many informants described their internal supplier of VCBC as a key enabler of agility through close collaboration and partnership. IF9 stated, *"[VCBC] will help us with simulations of die-forming…They support us very much with this."*

In the interviews, it was found that the informants do not consider the end customer as a stakeholder. It was due to informants feeling too distant from the customers and that they do not have any interactions with them. Informants instead described the stakeholders as internal customers. This sub-theme encompassed the demands and requirement that teams work towards and those that set these. IF2 stated that the end customer is quite far away from the direct development. Instead, the emphasis was placed on internal Volvo customers, such as manufacturing, safety, and design. This meant that, rather than getting direct demands from the end customer, requirements are translated through attribute-leaders, a role which *"takes care of the whole picture and demands, as [the teams] cannot derive a certain demand to a certain article."* The distance between engineer and customer was described to lead to difficulty in responding to change in customer demands.

Throughout the interviews of how the department collaborated with stakeholders, software usage was a significant aspect. Primary of these software enablers was the prevalence of digital communication, e.g., the ability to send drawings and work-in-progress models between stakeholders without any hindrance. As IF3 stated, *"the communication [of drawings to VCBC] happens instantly. They can see and judge directly; 'we think this will work.' They run simulations that they return with."* This allowed for high-speed feedback loops, that were applicable on both small- and large-scale changes.

Another aspect of the virtual development cycles was that the design software allowed engineers to detect problems at a much earlier stage, than previously when physical models instead were praxis. IF3 stated, *"Earlier when we built test cars and pre-series cars, moments of 'Oh! This doesn't work' occurred. That things didn't fit together, or the operators could not assemble the car in a reasonable way."* This ability allowed for a much more iterative checking of the car, as IF3 said *"we [can] put together a car at a certain point in time, which builds on a design that everyone will design towards, and that everyone has seen works together. Then we have much more control over the whole process."* These processes were described to enhance Efficient Structures' agility.

However, there were some issues connected to virtual tools, as mentioned by IF11, virtual tools can be improved to facilitate continuous integration. As the models are worked on locally, iterative work is hindered due to the dependencies. Altering one model can affect a multitude of other parts, and in this be a very time-consuming process.

5.3 Summary of Results

The results and answers to RQ1 and RQ2 have been summarized below.

Agility was defined and operationalized as *responsiveness to change.* Processes and activities can be described in relation to their responsiveness to change. The concept was used to research factors affecting agility at Efficient Structures.

Communication affected agility by facilitating the flow of information. Communication was intertwined with the rest of the themes. The closer and more synchronous the communication was, the more it improved the responsiveness to change – face-to-face communication was responsive while communication with external suppliers was slow. Visual communication, i.e., presenting physical or virtual models of car body components, was described as a critical enabler for conveying information and receiving feedback. Similarly, informal communication networks increased agility.

Iterative and Incremental Development affected agility by facilitating a way for Efficient Structures to deliver their work, receive feedback and improve their designs. It was easier to respond to feedback from stakeholders that was received early in a process. Small and fast feedback loops were superior to slow and information-intensive ones. Inherent traits to physical products, such as cooling times of die tools, slowed down the iteration cycles. Virtual tools were imperative for iterative development within Efficient Structures.

Complex Hardware Development and inherent traits of automotive development affected responsiveness to change negatively. Agility is highly dependent on the phases within product development – within early concept phases, responding to change in stakeholder demands is simpler. In later phases, some changes are impossible to implement due to, e.g., inter-organizational dependencies and long lead times for casting die tools. Strict time plans are enacted within car projects with little room for dynamic planning, again, due to dependencies. Closeness to suppliers decreased the negative effect of the dependencies. Complexity was decreased within the development through platform and modularization technologies.

Knowledge and Learning improved problem-solving skills, and subsequently agility. Engineers with thorough knowledge were described to respond to issues faster than inexperienced ones. Broad competence, i.e., knowledge of other areas outside of one's expertise, further enhanced performance. Increase of knowledge occurred primarily through informal and interpersonal learning, and formal daily meetings. Learning is a way to improve development processes.

Teamwork was described as a means of enhancing agility by improving problem-solving and feedback loops. Cross-functional teams allow for earlier and more continuous feedback. Self-organization in the form of decision-making at a team level and informal networks improved agility but was otherwise described to be difficult to enact in an interdependent and complex development. Interdependencies limited the extent of a team's or individual's decision-making. Planning and documentation also hindered. A sense of ownership regarding work affected agility both positively and negatively.

Collaboration was a key driver for agility within Efficient Structures through the meetings with stakeholders and partners, where all were gathered and discussed the ongoing development. Work with and toward internal suppliers was described to increase agility. Development where external suppliers were involved, was less responsive. No direct collaboration was held with end-customer, which resulted in less knowledge of their demands. Contrastingly, if the customer is described as the internal stakeholders, then there was close and direct collaboration. Virtual tools enabled the sending of drawings and work-in-progress models between stakeholders without any hindrance, increased collaboration, and removed the physical barriers of automotive development.

6 Discussion

The results presented in the previous section will be discussed with the aim of answering RQ3 – 'How do the results relate to a general agile hardware development context?' The discussions follow primarily an approach where the focus lies in the problematization of existing theory and critical reflections of the empirical data, from which conclusions are drawn to develop new theory. Our interpretations of the empirical data influenced the results and the following discussions. Interpretative discussions and concluding from a specific context were considered strengths, by Dubois and Gadde (2002), for developing new theory.

6.1 Communication

The informants emphasized communication as a significant factor for affecting agility, and for enabling other processes. The difficulty of automotive development was described to lie in aligning and integrating all of the development activities to make a car. Communication was imperative in tying all operations together. In existing theory on agile development, the topic of communication was not as prevalent. Only one of the agile principles (2001b) stated that communication should occur face-to-face. Williams (2012) instead used the term synchronous communication, to encompass all forms of direct communication. Not much else has been described explicitly about communication. Williams (2012) stated that the importance of communication described within the principles was not emphasized enough. The significance of communication found in the results versus existing theory was in some ways contrasting.

Face-to-face communication was both in existing theory and the case study described as the best way to communicate. Face-to-face communication led to a high degree of responsiveness to change at Efficient Structures, but the further one came from the source, e.g., trying to communicate to a supplier's supplier, the less responsive to change the process became. This extended to suppliers as well. Similar to the result, Kim and Chai (2017) found that sharing of information and supplier involvement were vital for agility. Moe et al. (2009) and Rigby et al. (2016a) described the importance of closeness in relation to having teams co-located in order to increase agility. The complexity of automotive body development led to the communication frequently being far from the decision-points and signified a great need for close communication for achieving agility within agile hardware development. Informal communication paths that skip hierarchical and organizational barriers should, therefore, be encouraged.

The possibility to visually show the progress of the development with the use of CAD was a strength at Efficient Structures. The physicality of the components enabled the developers to efficiently discuss and present their work-in-progress in all stages of development. To gather around a physical or virtual component was described to improve the quality of communication and responsiveness to change. IF9 stated, *"You can touch and move things, test things, and so on. You get a whole other understanding when you see the things. It speeds up communication."* This aspect was not discussed in existing theory to the same extent, as literature focused on agile software development. Rigby et al. (2016a), e.g., described prototyping as a way to communicate progress to the customers. The visual aspect of communication within hardware development offers, therefore, a unique advantage in increasing agility.

6.2 Iterative and Incremental Development

The results indicated that iterative and incremental development was beneficial for agility but was challenging in some respects at Efficient Structures. As hardware development generally is much slower (Cooper & Sommer, 2018), rapid incremental cycles were difficult to achieve. IF8 stated, *"Die-casting feedback arrives, and then after five weeks all the status reports return"* This is due to the inherent attributes of physical components which, e.g., require weeks of cooling. Williams (2012) discussed the importance of short iterations within agile development; therefore, with feedback arriving several weeks after delivery, a rapid response was significantly hindered. The physical traits of hardware development limit the use of short iterations in later phases of projects and prove a stark contrast to the properties of software development.

Schuh et al. (2017b) stated that close integration with the customer is beneficial for iterative work within hardware development. In this respect, Efficient Structures held a significant advantage because of their close collaboration with partners and stakeholders. Many informants spoke positively of the rapid iterative development loops they had with their stakeholders. These loops allowed experimentation and enabled rapid knowledge sharing. It allowed Efficient Structures quickly to deliver their work, receive feedback and improve their designs. These benefits were also described by Highsmith & Cockburn (2001) and Rigby et al. (2016a). Most importantly, informal networks allow work outside of major releases, which significantly shortened the feedback cycles and improved their agility.

A vital factor enabling rapid iterative development was the utilization of software applications. As IF9 described, the use of software increased both the speed and complexity of informal iterative work and was absent in existing literature concerning agile development. However, this most likely stems from the fact that the literature was focused on software development, which inherently relates it to utilizing software in the development. Concludingly, it is crucial for organizations to identify which processes can be iterated, to what degree they can be iterated and formalized, and provide the tools that enable iterative development.

6.3 Complex Hardware Development

The most prominent factor affecting agility negatively was the complexity of development. Efficient Structures can be defined as very large-scale agile (>10 teams) (Dingsøyr et al., 2014). Therefore, the implementation of any framework will be problematic, and the coordination of the teams is very complicated (Dikert et al., 2016). The product is complex and relies on a vast number of stakeholders working together. The complexity not only decreases responsiveness to change but forces a significantly more structured organizational approach.

An example of counter-acting the complexity was through the use of platforms and modularization technologies. They increased agility and flexibility in several ways, as described by Magnusson and Pache (2014). When developing hardware, it is crucial to consider what technologies can be used to support one's development processes and increase agility.

Another significant obstacle for agile development methods within Efficient structures is the heavy reliance on planning. With tools requiring several weeks to be developed and ready for the start of production, the departments and suppliers must have a detailed plan. Strict time plans decrease agility, according to Highsmith and Cockburn (2001). Planning should be dynamic in accordance with the changing demands of the customers – otherwise, it can result in a weaker value proposition (Highsmith & Cockburn, 2001). As evidenced by Efficient Structures, complexity and physical constraints effectively force the organization to plan, and dynamic prioritization is only possible in a minor capacity. While Highsmith and Cockburn (2001) were considering a software development context, the same frameworks that are used in software are being implemented within hardware development. Therefore, agility within hardware development is constrained by the physical aspects that are not prevalent in software development.

However, planning can provide benefits within hardware development, according to Laanti (2016). The first of these is the use of a steady cadence, which the organization can sync to, and is crucial for successful implementation of agile in a hardware context. The other was to have synchronization points, where the whole product is integrated, and all the stakeholders meet (Laanti, 2016). At Efficient Structures, they followed similar processes – employing a cadence and synchronization – which were described to enhance responsiveness. While existing literature on agile hardware development is lacking, the case and Laanti (2016) state those two key factors for achieving agility in that context.

Agility at Efficient Structures was greatly affected by product maturity. In early phases, responding to change was easy, while in later stages, changing the product was impossible due to, e.g., organizational dependencies, suppliers, and long lead times for casting die tools. Schuh et al. (2017b) came to the same conclusion – agility is a dynamic element. Perhaps even more so in hardware development.

6.4 Knowledge and Learning

Within existing theory (e.g., Nerur & Balijepally, 2007) and found in our results, knowledge seemed to be a vital part for agility. For the engineers, being experienced enabled faster problem-solving. Broad competence, similarly, enabled faster problem-solving. At Efficient Structures, it was stated that many employees were knowledgeable and quick to act which led to the organization being described as agile. Therefore, building up a competent workforce appears to be essential for agility.

An often-repeated aspect at Efficient Structures was the prevalence of informal networks, where solutions, issues, or other knowledge were discussed. Individuals could directly contact co-workers they knew could help them, and this speeded up problem-solving and increased learning. In this knowledge transfer, Lawson et al. (2009) found that informal learning was superior to formal learning, as was presented in the results of this study. It means that informal networks are central to increasing communication and personal learning. This is a significant factor for a complex, large-scale organization with many informal networks that should be encouraged, e.g., through formal activities as they were described to create informal ones (Lawson et al., 2009).

Ambler (2009) described the need for using measurements of how teams work in order to improve their performance, which is a formalized process. Another formal activity is the daily stand-ups which encourage learning and collaboration. They were described as an improvement with the implementation of SAFe. Not only are these types of activities creating new forums for formal contacts, but also the potential for informal network growth. This seems to be a major benefit – shifting the burden of knowledge management from large, overhead systems, to individuals, allowing both high speed and quality transfer. However, this informality means that should an individual leave the organization, networks may potentially be broken, or knowledge permanently lost. Therefore, a way to, e.g., map the informal networks and identify where knowledge resides could be a necessity – especially in such a complex product development organization.

The informants described a lack of satisfactory knowledge management systems. When compared to the knowledge spread through informal networks, within Efficient Structures, knowledge management systems seemed more important in spreading systematic feedback, especially between stakeholders that share few connections. A long-time quality issue was easily solved when feedback was finally received and acted upon, exemplified IF8. This type of double-loop learning improves problem-solving and overall responsiveness (Nerur & Balijepally, 2007). It is therefore essential to use feedback as a means of learning.

With information and knowledge increasing agility, its utilization and management should be prioritized. It was also described that learning was a way to improve internal processes and efficiency, in line with Ambler (2009). This means that activities that enable both informal and formal knowledge transfers need to be developed and encouraged. However, it is also vital to reflect on whether the work environment, organization, and product are suitable for certain types of knowledge management.

6.5 Teamwork

Teamwork was one of the most widely discussed areas within existing literature and held a central part in most commercial agile frameworks that have been developed (e.g., Highsmith & Cockburn, 2001; Dybå & Dingsøyr, 2008). However, this stood in contrast to the findings at Efficient Structures, where teamwork was described to play a lesser role in increasing responsiveness to change compared to the other themes. A reason for this could be that previously, they did not work in teams as described by, e.g., Moe et al. (2009). However, after the implementation of SAFe, a stronger emphasis was put on teamwork. Although there has been a shift towards teamwork, the benefits have not yet been realized. This can be explained through teams requiring at least six months before they can become high-performing (Wheelan, 2009). While teamwork has not yet been realized, it was described as a means of enhancing agility by improving problem-solving capabilities and feedback loops.

The area of self-organization proved to be one of the most contentious areas. Hoda et al. (2010) placed self-organizing teams as a central aspect of agile development and effective teamwork. Within Efficient Structures, self-organizing teams were minimally implemented, and often considered non-viable. The reasoning was based on the large number of dependencies present in the development, reducing the possibility for teams to make decisions. Moe et al. (2009) and

Dikert et al. (2016) stated that organizational control issues increase alongside an increase in the number of self-organizing teams. Although self-organizing teams have been proven to increase agility (Hoda et al., 2010), their role within complex hardware development needs to be further explored.

Ambler (2009) stated that in a large-scale organization, self-organizing teams need boundaries or frameworks in which they can operate to not clash with other dependencies. Should a framework for self-organizing teams efficiently be determined, the responsiveness to change might be significantly improved at Efficient Structures. Already teams were given certain leeway in decision-making. The question is to what degree these freedoms can be increased before they affect overall agility negatively. Under this purview, initially limiting self-organization might be the pertinent path – then slowly expanding it alongside a clear framework to determine the proper level of decision-making within self-organizing teams.

However, the subject of self-organization can be expanded to encompass collaboration and informal processes. A parallel can be drawn to Takeuchi & Nonaka (1986) who described autonomy in terms of teams being allowed to decide how a task will be approached and whose expertise is needed. Throughout Efficient Structures, individuals were encouraged to independently seek out other developers or stakeholders that might hold information, going through informal networks rather than through official channels. In other cases, engineers were forced to documentation and work through formal processes which were deemed to be slow and unnecessary. While it was somewhat approached by Takeuchi & Nonaka (1986), informal processes as a form of self-organization were lacking in the existing theory, as far as could be seen. In hardware development, teams should – similar to Efficient Structures – be encouraged to use informal processes to increase agility but, as mentioned, need to be balanced by a framework.

Cross-functionality within teams was described to increase agility at Efficient Structures. Similarly, within the existing theory, having a set of broad expertise within a team is vital for agile development, and teams should have all the competencies required to solve their tasks (e.g., Hoda et al., 2010; Ambler, 2009). Within the context of a complex organization, having all types of competencies within the teams could be overwhelming. Therefore, it is important to understand the limitations of achieving cross-functionality and to set up teams in a suitable way, as it affects responsiveness to change positively. For example, is it viable to embed a 'Vibrations and Acoustics' engineer in a team, for a minor task? Should this be done, self-organization might be more viable due to increased knowledge, but it might also lead to over-dimensioned and inefficient teams.

Regarding overlapping skills within the teams – which Nerur and Balijepally (2007) stated increased agility – the topic of ownership was discussed. Ownership affected agility both negatively and positively. It led to specialization within an area but also led to a low degree of overlapping skills among team members. Teams and organization need to evaluate how a sense of ownership can be utilized and adapted to their specific processes in order to decrease the adverse effects. Takeuchi and Nonaka (1986) described that such factors need to be shared on a team level.

6.6 Collaboration

The results and within existing theory showed that collaboration increases responsiveness to change. However, there are some differences in how collaboration was approached, and which aspects were most significant. First of these is the role of the customer. Hoda et al. (2011), Highsmith and Cockburn (2001), and the agile manifesto (Beck et al., 2001a) all stress the necessity of a close collaboration with the customer. Throughout existing theory, the customer is generally seen as the end-customer (e.g., Beck et al., 2001a), which ultimately will purchase and use the product. There are some wider views, such as Highsmith & Cockburn's (2001) expanding the definition of the customer to encompass internal and external stakeholders.

However, a general theme was the direct contact between the development teams and the end-customer. Within Efficient Structures, the end-customer was a very distant stakeholder and instead they viewed their customers as those that will directly receive their work, i.e., internal customers. The end-customer was only translated to the developers through demands and requests, which passes many levels of hierarchy and bureaucracy, hindering effective responsiveness to outside change. However, solely moving the developers closer to the end customer is not a directly viable option either. The complexity of the product and the distance from the customer lower the responsiveness to changing customer demands and is a fact to consider for agile hardware development.

In the interview data, the preparation meeting was described as one of the most vital aspects of the communication and collaboration themes. From the results, the meeting seems to be one of the largest contributors to responsiveness to change. Stakeholders and developers interacted on a direct basis, utilizing software to show and discuss problematic areas. Furthermore, these meetings utilized visual models of the parts that are developed, allowing for a tactile and dynamic method of communication. Co-locating a team or meetings with stakeholders was described as a key factor for improving the efficiency of teams (Moe et al., 2009; Rigby et al., 2016a). In order to increase agility, organizations can facilitate similar meetings where all stakeholders meet face-to-face regularly.

When discussing collaboration with stakeholders, several informants mentioned a divide between internal and external stakeholders. This divide was considered quite stark – external stakeholder collaboration was limited while the close collaboration with internal stakeholders improved Efficient Structures agility. This is similar to the discussion in chapter 6.1. Hardware development is reliant on supplier involvement. Therefore, it is central to consider the role of supplier involvement in relation to its effect on agility.

Throughout the area of collaboration, a comparison can be made between Efficient Structures and Rigby et al. (2016a) who presented the right conditions for agile development. When analyzing Volvo Cars, and Efficient Structures on a large scale, the only condition that seemed to be met was that of an *unstable market.* Experimentation is costly, and close customer collaboration is near impossible. However, when looking on a team level within efficient structures, all these conditions are met. The teams work very closely with their customers and can experiment within specified limits. With the unstable market, the dependencies ensure that the demands placed on the development teams always change, and in this regard, promotes an agile development methodology.

The advancement of CAD and CAE software allows for both increased collaboration with the customer, and an increase in experimentation (Rauh, 2003). The informants confirmed Becker et al.'s (2005) view that through the effective use of software and virtual development tools provide improved problem-solving capabilities. Software provided almost instant communication and feedback loops, as massive amounts of data could be sent to a supplier for confirmation, and rapidly be returned with feedback. Through this iterative process, developers seemed to not only benefit in their work, but it also provided insight for the stakeholders into each other's work. Again, virtual tools seem to provide a basis for agile processes within hardware development

6.7 Key Aspects for Agility

A number of key aspects for agility within agile hardware development were identified within the different themes. This study and the aspects provide a basis for agile hardware development. These have been grouped according to relevant themes, as shown in Table 4.

Communication between stakeholders, teams, and others should be fast and direct, and is paramount within a complex organization. **Face-to-face** communication increases agility the most, while the further one goes from the decision-point, the less responsive the process becomes. A strength within hardware development is the possibility to visually show progress by, e.g., presenting physical or virtual models of car body components, in all stages of development. **Visual communication** is an easy and understandable way of conveying information and receiving quick feedback.

Iterative and Incremental Development enables the developers to deliver their work, receive feedback from stakeholders, and act upon said feedback. Short iterations should be strived for to allow faster updates of stakeholder needs and for responding to smaller changes in demands is easier. Within complex hardware development, iterations are costlier and slower than within software development. Iteration cycles can provide a set cadence for organizations which is beneficial for agility. However, the possibility for iterative development is limited by **physical constraints**, especially in later stages. Within the development of physical products, the different phases of product development and product maturity lead to **changing degrees of agility**. In early concept phases, responding to change in stakeholder demands is simpler. In later phases, some changes are impossible to implement.

The physical constraints and the multitude of stakeholders, dependencies, and teams within a large-scale hardware development organization results in inherently reduced responsiveness to change. Decreased **complexity** should therefore be aimed for. Several technologies can be used within hardware development to decrease the complexity and increase the agility, e.g., **platform and modularization**. **Virtual tools** are also exceptional enablers of agility within the context. They remove many of the physical limitations and provide a way of working that is similar to software development, and that would be impossible otherwise.

The physical aspects of the development and supplier involvement forces strict **time plans**, which lead to less dynamic planning. Although strict planning has a negative effect on agility, they provide clear schedules and milestones.

Within agile hardware development, these primarily happen on a team level. **Continuous improvement and learning** can occur through formal and informal activities and will lead to more knowledgeable workers and efficient processes. A higher degree of **knowledge and competence** within developers improves problem-solving skills, and, consequently, agility. Specific expertise within an area combined with general knowledge of processes is preferable.

A central aspect of agile development is **teamwork**. Teams are more efficient and work together to increase problem-solving capabilities and learning. Teamwork requires collaboration skills, support, leadership, and training. Allowing teams to **self-organize** leads to faster and more efficient decision-making. However, there are many limitations to the degree of autonomy due to interdependencies and complexity. The degree of autonomy should be decided by a framework that is adapted to the context of the organization. The framework should allow for meaningful decisions while setting clear boundaries of their extent. Broader competence and early feedback on work can be achieved by assembling teams with members from a variety of functions. Within large organizations with a myriad of functions, assembling **cross-functional** teams can be difficult. A balance between efficiency and team width needs to be determined.

A critical factor within agile hardware development is close **collaboration**. Meetings where stakeholders gather face-to-face and regularly to discuss ongoing work are vital for agility. It enables coordination between stakeholders. Work with and toward internal stakeholders is often more responsive, while toward external stakeholders, e.g., suppliers, processes are less responsive. Therefore, a consideration of the **role of the supplier** and their effect on agility is needed. Collaboration with the **end-customers** is challenging as developers work a considerable distance from them. Responding to and communicating changes are, therefore, difficult. However, the customer can instead be seen as the internal customer who receives the work, but with the risk of losing customer-orientation, which should be the goal of agile hardware development – delivering products the customer wants, by always being able to respond to changes in customer demands.

Informal networks are a way for employees to organize and work outside the formal structures of an organization. These networks were an underlying aspect throughout the themes. Communication and collaboration occur between coworkers that know each other. Working through informal networks is faster, more direct, and lead to increased responsiveness to change. Within a large organization, there are many possibilities to build up these networks, and it is something that is to be encouraged.

Themes	*Communication*	*Iterative and Incremental Development*	*Complex Hardware Development*	*Knowledge and Learning*	*Teamwork*	*Collaboration*
Key Aspects	Visually Conveying Information	Phase-Dependent Agility	Use of Technologies that support Agility	Knowledgeable Workers	Self-Organization within a framework	Stakeholder Roles
	Face-to-Face	Physical Constraints	Heavy Reliance on Time-Plans	Continuous Improvement	Cross-functional Teams	Distance from End-Customer
	Informal Networks					

Table 4: Themes and their underlying key aspects of agility within hardware development

7 Conclusions

This study aimed to identify key aspects for agility within agile hardware development. Agility was defined and operationalized as responsiveness to change and was used to guide and translate the results and discussions to a general context. The empirical findings resulted in six themes – Communication, Knowledge and Learning, Iterative and Incremental Development, Teamwork, Complex Hardware Development and Collaboration – that affect the agility at Efficient Structures, with each theme consisting of two underlying aspects, and informal networks affecting all themes. Many of the aspects concerned the organic and humanistic side of agile development within a rigid and mechanistic traditional organization, which, ultimately, can be reduced to a question of fixed versus flexible. Striking a balance between fixed and flexible is vital for agile hardware development. A notable example is within self-organizing teams where they should have the freedom to make meaningful decisions within set boundaries.

The presence of informal networks in all themes seemed to be crucial for responsiveness to change within hardware development. This aligns with the humanistic side of agile development where, ultimately, the people in the organization create its agility.

7.1 Future Research

Throughout the research, several areas of interest were identified, but were determined to fall outside the scope of this study. These areas could be useful to further problematize agile hardware development, agile development in general, or exploring a basis for agile hardware development.

Strong Theoretical Focus on Commercial Frameworks – During the exploration of existing agile development literature, the heavy reliance on the agile manifesto (Beck, et al., 2001) as a foundational piece of literature presented several issues, such as lacking a clear definition of agility, a poor distinction between theory and practice, and unclear theoretical founding. The researchers of this study posit that it would be interesting to further explore what effect this has on existing agile literature, and how the lack of a unified definition of agility affects both commercial frameworks and literature.

Interface between Management Literature and Agile Development Literature – As mentioned above, much of the literature relies on the agile manifesto. In this vein, the agile manifesto is often used as short-hand for aspects found in management theories, such as change management or organizational theory. However, due to the unclear theoretical grounding of the agile manifesto, the validity of this short-hand is questionable. Therefore, it is recommended that the validity of the agile manifesto, in terms of management theory, is explored to a greater degree, and this is instead used as the basis for future agile exploration.

Agile development of Integrated Systems – The scope of this study was strictly limited to hardware development, which was perceived to lie the furthest from software development. However, throughout the research process, issues regarding how the agility of integrated hardware and software development is affected by factors of strict hardware or software development. The researchers posit that this interface might present unique factors, that are not present within strict hardware or software development.

Informal Networks in an Agile context – During the empirical data collection, informants placed a large emphasis on informal networks, and the perception that this improved agility was widespread. However, within agile development literature, informal networks and communication is largely absent, and commercial frameworks such as SAFe do not broach the subject. Therefore, an exploration of the role of informal networks within agile development is recommended, and potentially an exploration of how its role differs between agile hardware and software development.

7.2 Managerial Implications

Following this study, it is recommended that the management of Efficient Structures considers agility as a philosophy rather than a set of practices. Utilizing the study's operationalization of agility as responsiveness to change, a clearer overarching philosophy can be developed within the firm. This philosophy can both be used to educate staff and build engagement. In the long run, it is a worthwhile goal to strive for. By approaching it as a philosophy, it does not strictly subscribe to any singular framework. Instead different facets might be incorporated where applicable.

Furthermore, in deciding to transition to agile product development, a closer inspection of existing frameworks is recommended. The direct application of software-based frameworks in complex hardware environments might prove difficult. Instead, by considering the abovementioned factors, a better evaluation of existing frameworks is possible. For example, frameworks which rely on rapid iterations and feedback might be less appropriate, as these are difficult to achieve within hardware development.

Finally, it is recommended that management considers the key aspects in decision-making. Due to the inherent physical restrictions of hardware development, factors such as complexity and phase-dependent agility are difficult to avoid. Therefore, any decision that affects an agile hardware development organization need to consider them. While managers with relevant experience might already be aware of these challenges, converting the tacit knowledge to explicit knowledge is useful in a wider decision-making context.

References

Abrahamsson, P., Conboy, K. & Wang, X., 2009. 'Lots done, more to do': the current state of agile systems development research. *European Journal of Information Systems,* Volume 18, pp. 281-284.

Alvesson, M. & Kärreman, D., 2007. Constructing Mystery: Empirical Matters in Theory Development. *Academy of Management Review,* 32(4), pp. 1265-1281.

Ambler, S. W., 2009. The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments. *IBM Rational Software.*

Becker, M. C., Salvatore, P. & Zirpoli, F., 2005. The impact of virtual simulation tools on problem-solving and new product development organization. *Research Policy,* Volume 34, pp. 1305-1321.

Beck, K. et al., 2001. *Manifesto for Agile Software Development.* [Online]
Available at: https://agilemanifesto.org/
[Accessed 12 February 2019].

Beck, K. et al., 2001. *Principles behind the Agile Manifesto.* [Online]
Available at: https://agilemanifesto.org/principles.html
[Accessed 23 April 2019].

Boehm, B. & Turner, R., 2005. Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE software,* 22(5), pp. 30-39.

Braun, V. & Clarke, V., 2005. Using thematic analysis in psychology.. *Qualitative Research in Psychology, 3(2), 77-101,* 3(2), pp. 77-101.

Bryman, A. & Bell, E., 2011. *Business Research Methods.* 3rd ed. Oxford: Oxford University Press.

Burawoy, M., 1998. The Extended Case Method. *Sociological Theory,* 16(1), pp. 4-33.

Conboy, K., 2009. Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research,* 20(3), pp. 329-354.

Conboy, K. & Fitzgerald, B., 2004. Toward a Conceptual Framework of Agile Methods: A Study of Agility in Different Disciplines. *Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research (WISER),* pp. 37-44.

Cooper, R. G., 2016. Agile–Stage-Gate Hybrids: The Next Stage for Product Development. *Research Technology Management,* 59(1), pp. 21-29.

Cooper, R. G. & Sommer, A. F., 2016. The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity. *Journal of Product Innovation Management,* 33(5), pp. 513-526.

Cooper, R. G. & Sommer, A. F., 2018. Agile–Stage-Gate for Manufacturers. *Research-Technology Management,* 61(2), pp. 17-26.

Diener, E. & Crandall, R., 1978. *Ethics in social and behavioral research. : U Chicago Press..* Oxford, England: U Chicago Press.

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

Dingsøyr, T., Fægri, T. E. & Itkonen, J., 2014. What is Large in Large-Scale? A Taxonomy of Scale for Agile Software Development. *Product-Focused Software Process Improvement,* Volume 8892, pp. 273-276.

Dingsøyr, T., Nerur, S., Balijepally, V. & Moe, N. B., 2012. A decade of agile methodologies: Towards explaining agile software development. *The Journal of Systems and Software,* Volume 85, pp. 1213-1221.

Dove, R. & Labarge, R., 2014. *Fundamentals of Agile Systems Engineering – Part 1 and Part 2.* Las Vegas, Nevada, INCOSE International Symposium.

Dubois, A. & Gadde, L.-E., 2002. Systematic combining: An abductive approach to case research. *Journal of Business Research,* 55(7), p. 553– 560.

Dubois, A. & Gadde, L.-E., 2013. "Systematic combining"—A decade later. *Journal of Business Research,* Volume 67, pp. 1277-1284.

Dybå, T. & Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology,* Volume 50, pp. 833-859.

Gren, L. & Lenberg, P., 2019. Agility is responsiveness to change: An essential definition. *IEEE Software,* Volume In review.

Gren, L., Torkar, R. & Feldt, R., 2017. Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. *The Journal of Systems and Software,* Volume 124, pp. 104-119.

Highsmith, J. & Cockburn, A., 2001. Agile Software Development: The Business of Innovation. *Computer ,* 34(9), pp. 120-122.

Hobbs, B. & Petit, Y., 2017. Agile Methods on Large Projects in Large Organizations. *Project Management Journal,* 48(3), pp. 3-19.

Hoda, R., Noble, J. & Marshall, S., 2010. Organizing self-organizing teams. *ACM/IEEE 32nd International Conference on Software Engineering,* pp. 285-294.

Hoda, R., Noble, J. & Marshall, S., 2011. The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology,* Volume 53, pp. 521-534.

Hohl, P. et al., 2018. Back to the future: origins and directions of the "Agile Manifesto" – views of the originators. *Journal of Software Engineering Research and Development,* 6(15), pp. 1-27.

Kim, M. & Chai, S., 2017. The impact of supplier innovativeness, information sharing and strategic sourcing on improving supply chain agility: Global supply chain perspective. *International Journal of Production Economics,* Volume 187, pp. 42-52.

Kuusinen, K., Gregory, P., Sharp, H. & Barroca, L., 2016. Strategies for doing Agile in a non-Agile Environment. *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.*

Laanti, M., 2016. Piloting Lean-Agile Hardware Development. *XP '16 Workshops,* pp. 1-6.

Laanti, M., Similä, J. & Abrahamsson, P., 2013. *Definitions of Agile Software Development and Agility.* Berlin Heidelberg, Springer-Verlag, p. 247–258.

Lawson, B., Petersen, K. J., Cousins, P. D. & Handfield, R. B., 2009. Knowledge Sharing in Interorganizational Product Development Teams: The Effect of Formal and Informal Socialization Mechanisms. *The Journal of Product Innovation Management,* Volume 26, pp. 156-172.

Magnusson, M. & Pasche, M., 2014. A Contingency-Based Approach to the Use of Product Platforms and Modules in New Product Development. *Journal of Product Innovation Management,* 31(3), pp. 434-450.

Moe, N. B., Dingsøyr, T. & Dybå, T., 2009. Overcoming Barriers to Self-Management in Software Teams. *IEEE software,* 26(6), pp. 20-26.

Nerur, S. & Balijepally, V., 2007. Theoretical Reflections on AGILE DEVELOPMENT METHODOLOGIES. *COMMUNICATIONS OF THE ACM,* 50(3), pp. 79-83.

Rauh, J., 2003. Virtual Development of Ride and Handling Characteristics for Advanced Passenger Cars. *Vehicle System Dynamics,* 40(3), pp. 135-155.

Rigby, D. K., Sutherland, J. & Noble, A., 2018. Agile at Scale. *Harvard Business Review,* Volume May-June, pp. 88-96.

Rigby, D. K., Sutherland, J. & Takeuchi, H., 2016. Embracing Agile. *Harvard Business Review,* 94(5), pp. 40-50.

Rigby, D. K., Sutherland, J. & Takeuchi, H., 2016. The Secret History of Agile Innovation. *Harvard Business Review Digital Articles,* pp. 2-5.

Robertson, D. & Ulrich, K., 1998. Planning for Product Platforms. *Sloan Management Review,* 4(4), p. 19.31.

Sarker, S., Munson, C. L., Sarker, S. & Chakraborty, S., 2009. Assessing the relative contribution of the facets of agility to distributed systems development success: an Analytic Hierarchy Process approach. *European Journal of Information Systems,* 18(4), pp. 285-299.

Scaled Agile, 2018. *About SAFe.* [Online]
Available at: https://www.scaledagileframework.com/about/
[Accessed 20 April 2019].

Schuh, G., Dölle, C., Kantelberg, J. & Menges, A., 2018. Identification of Agile Mechanisms of Action As Basis for Agile Product Development. *Procedia CIRP, vol. 70, pp. 19–24,* Volume 70, p. 19–24.

Schuh, G., Gartzen, T., Soucy-Bouchard, S. & Basse, F., 2017. Enabling agility in product development through an adaptive engineering change management. *Procedia CIRP,* Volume 63, pp. 342-347.

Schuh, G., Salmen, M., Kuhlmann, T. & Wiesea, J., 2017. Highly Iterative Product Development Within The Tool And Die Making Industry. *Procedia CIRP,* Volume 61, pp. 576-581.

Shenton, A. K., 2004. Strategies for ensuring trustworthiness in qualitative research projects. *Education for information,* 22(2), pp. 64-75.

Smith, J. A., 2015. *Qualitative psychology : a practical guide to research methods.* Third ed. London: SAGE.

Takeuchi, H. & Nonaka, I., 1986. The New New Product Development Game. *Harvard Business Review,* 64(1), pp. 137-146.

Van Maanen, J., 1979. The Fact of Fiction in Organizational Ethnography'. *Administrative Science Quarterly,* 24(4), pp. 539-550.

Wheelan, S., 2009. Group Size, Group Development, and Group Productivity. *Small Group Research,* 40(2), pp. 247-262.

Williams, L., 2012. What Agile Teams Think of Agile Principles. *Communications of the ACM,* 55(4), pp. 71-76.

Volvo Cars, 2013. *Ny skalbar produktarkitektur låter Volvo Car Group rulla snabbare mot en krockfri framtid.* [Online]
Available at: https://www.media.volvocars.com/se/sv-se/media/pressreleases/136653/ny-skalbar-produktarkitektur-lter-volvo-car-group-rulla-snabbare-mot-en-krockfri-framtid
[Accessed 4 May 2019].

Volvo Cars, 2019. *Detta är Volvo.* [Online]
Available at: https://www.volvocars.com/se/om-volvo/foretaget/detta-ar-volvo
[Accessed 08 May 2019].

Volvo Cars, 2019. *Heritage | Volvo Cars.* [Online]
Available at: https://www.volvocars.com/intl/cars/heritage
[Accessed 08 May 2019].

Yin, R., 1994. *Case Study Research.* London: SAGE.

Appendix A

A.1: Interview Guide for IF1

Note: These have been translated from Swedish

Short description of the master's thesis

Discuss ethics, i.e. tell interviewee that their personal information will be confidential

Is it okay if we record the interview?

Can you describe how you work in practice?

Who are your stakeholders?

How are product/customer requirements communicated?

Formally?

Informally?

How do you respond to change in requirements?

What factor is important for your team's responsiveness to change?

What factor hinders your team's responsiveness to change?

(Optional) What is your opinion on Agile and the change-process?

(Optional) What is the goal of Agile in your opinion?

(Optional) How have you had to adapt your way of working to the new agile practices?

Would it be possible to contact you again after the interview for additional information?

Anything you would like to add?

A.2: Interview Guide for IF2

Note: These have been translated from Swedish

Short description of the master's thesis

Discuss ethics, i.e. tell interviewee that their personal information will be confidential

Is it okay if we record the interview?

Can you describe how you work in practice?

Who are your stakeholders?

How are product/customer requirements communicated?

Formally?

Informally?

How do you respond to change in requirements?

What factor is important for your team's responsiveness to change?

What factor hinders your team's responsiveness to change?

(Optional) What is your opinion on Agile and the change-process?

(Optional) What is the goal of Agile in your opinion?

(Optional) How have you had to adapt your way of working to the new agile practices?

Would it be possible to contact you again after the interview for additional information?

Anything you would like to add?

A.3: Interview Guide for IF3

Note: These have been translated from Swedish

Short description of the master's thesis

Discuss ethics, i.e. tell interviewee that their personal information will be confidential

Ask if we may record the interview

Tell us about yourself, and how the development of cars has changed throughout the years?

What do you work with today?

How do you work with changes in demands from different stakeholders?

What affects your responsiveness to change?

How do you test and present changes?

How does software (CAD/CAE/Visualization etc.) affect your work?

How do you work with….

- Teams?
- Iterative and Incremental Work?
- Collaboration?
- Continuous Learning and Improvement?
- Communication?

Is there anything you feel we have missed?

A.4: Interview Guide for IF4

Note: These have been translated from Swedish

Short description of the master's thesis

Discuss ethics, i.e. tell interviewee that their personal information will be confidential

Tell us about yourself?

When we have been studying agility and "Responsiveness to Change", we identified continuous learning and Improvement as a potential factor for this. How do you work with this, and do you feel that it is helping you?

When you communicate information to your coworkers, how do you do it? Is it through formal processes? Informal? And how/what do you experience the effect of this?

Do you have any reflections on the spreading of competence within your teams?

We have talked to people who consider the ”T-shape" to be very useful within hardware, while others have mentioned that they would rather have "specialized-competens". What do you think?

What opportunities/hinders are there to create a broad and overlapping competens within teams?

A.5: Interview Guide for IF5

Note: These have been translated from Swedish

Short description of the master’s thesis

Discuss ethics, i.e. tell interviewee that their personal information will be confidential

How do you respond to change in requirements?

What factor is important for your team’s responsiveness to change?

What factors hinders your team’s responsiveness to change?

Quick fire

- Team-based
    - In what ways do you self-organize?
        - How does it affect RTC?
        - Have you seen any hinders to this?
    - In what ways do you work cross-functionally?
- Communication and collaboration
    - Can you describe how you communicate?
        - Within teams
        - Towards stakeholders
        - How do you collaborate with stakeholders?
- Iterative and Incremental Development
    - Do you work iterative?
    - Short, time-boxed sprints?
- Continuous learning and improvement
    - In what ways do teams improve their ways of working?
    - How do you learn/improve competence?

A6: Interview Guide for IF7

Note: These have been translated from Swedish

What opportunities/hinders have you seen for agility within hardware development?

Quick fire

- Team-based
    - In what ways do you self-organize?
        - How does it affect RTC?
        - Have you seen any hinders to this?
    - In what ways do you work cross-functionally?
- Communication and collaboration
    - Can you describe how you communicate?
        - Within teams
        - Towards stakeholders
        - How do you collaborate with stakeholders?
- Iterative and Incremental Development
    - Do you work iterative?
    - Short, time-boxed sprints?
- Continuous learning and improvement
    - In what ways do teams improve their ways of working?
    - How do you learn/improve competence?

A7: Interview Guide for IF8

Note: These have been translated from Swedish

Short description of the master's thesis

Discuss ethics, i.e. tell interviewee that their personal information will be confidential

Is it okay if we record the interview?

Background

Can you describe how you work in practice?

Who are your stakeholders?

How are product/customer requirements communicated?

Formally?

Informally?

How do you respond to change in requirements?

What factor is important for your team's responsiveness to change?

What factors hinders your team's responsiveness to change?

Quick fire questions -

- Team-based
  - In what ways do you self-organize?
  - In what ways do you work cross-functionally?
- Communication and collaboration
  - Can you describe how you communicate?
    - Within teams
    - Towards stakeholders
    - How do you collaborate with stakeholders?
- Iterative and Incremental Development
  - Do you work iterative?
  - Short, time-boxed sprints?
- Continuous learning and improvement
  - In what ways do teams improve their ways of working?
  - How do you learn/improve competence?

Is there anything you would like to add?

A8: Interview Guide for IF9

Note: These have been translated from Swedish

Is it okay if we record the interview?

Can you describe how you work in practice?

Who are your stakeholders?

How are product/customer requirements communicated?

Formally?

Informally?

How do you respond to change in requirements?

What factor is important for your team's responsiveness to change?

What factor hinders your team's responsiveness to change?

Quick fire questions -

- Team-based
  - In what ways do you self-organize?
  - In what ways do you work cross-functionally?
- Communication and collaboration
  - Can you describe how you communicate?
    - Within teams
    - Towards stakeholders
    - How do you collaborate with stakeholders?
- Iterative and Incremental Development
  - Do you work iterative?
  - Short, time-boxed sprints?
- Continuous learning and improvement
  - In what ways do teams improve their ways of working?
  - How do you learn/improve competence?

Would it be possible to contact you again after the interview for additional information?

Anything you would like to add?

A9: Interview Guide for IF11

Note: These have been translated from Swedish

Short description of the master's thesis

Discuss ethics, i.e. tell interviewee that their personal information will be confidential

Is it okay if we record the interview?

Can you describe how you work in practice?

Who are your stakeholders?

How are product/customer requirements communicated?

  Formally?

  Informally?

How do you respond to change in requirements?

What factor is important for your team's responsiveness to change?

What factor hinders your team's responsiveness to change?

(Optional) What is your opinion on Agile and the change-process?

(Optional) What is the goal of Agile in your opinion?

(Optional) How have you had to adapt your way of working to the new agile practices?

Would it be possible to contact you again after the interview for additional information?

Anything you would like to add?

A10: Interview Guide for IF12

Note: These have been translated from Swedish

Tell us about yourself?

When we have been studying agility and “Responsiveness to Change”, we identified continuous learning and Improvement as a potential factor for this. How do you work with this, and do you feel that it is helping you?

When you communicate information to your coworkers, how do you do it? Is it through formal processes? Informal? And how/what do you experience the effect of this?

Do you have any reflections on the spreading of competence within your teams?

We have talked to people who consider the ”T-shape” to be very useful within hardware, while others have mentioned that they would rather have “specialized-competens”. What do you think?

What opportunities/hinders are there to create a broad and overlapping competens within teams?