DRAFT

THE CHALLENGES OF BECOMING AGILE

IMPLEMENTING AND CONDUCTING SCRUM IN INTEGRATED PRODUCT DEVELOPMENT

PhD Thesis by Nis Ovesen
Aalborg University

PHD THESIS
JUNE 2012

DRAFT

THE CHALLENGES OF BECOMING AGILE

IMPLEMENTING AND CONDUCTING SCRUM IN INTEGRATED PRODUCT DEVELOPMENT

NIS OVESEN

DEPARTMENT OF ARCHITECTURE, DESIGN & MEDIA TECHNOLOGY

DOCTORAL PROGRAMME OF PLANNING AND DEVELOPMENT
THE DOCTORAL SCHOOL OF ENGINEERING AND SCIENCE

AALBORG UNIVERSITY

The Challenges of Becoming Agile
Implementing and conducting Scrum in Integrated Product Development

Number of pages: 200
Tirage: 20

PhD Candidate:

Nis Ovesen
Industrial Design
Department of Architecture, Design & Media Technology
The Faculty of Engineering and Science
Aalborg University

Supervisors:

Kaare Eriksen
Associate Professor
Department of Architecture, Design & Media Technology
The Faculty of Engineering and Science
Aalborg University

Christian Tollestrup
Associate Professor
Department of Architecture, Design & Media Technology
The Faculty of Engineering and Science
Aalborg University

TABLE OF CONTENTS

PREFACE

"What happens if we bring methods from other industries into the traditional product development practice?"

The broad question above essentially sparked this PhD thesis three years ago, but the curiosity was soon focused on Agile Development in software industry, as it had been gaining enormous momentum during the preceding decade. At the same time – in the corners of a few Danish companies – people led by the same curiosity were slowly starting to implement Scrum in their respective product development environments. Now, three years later, Scrum is gaining foothold in an increasingly broader span of companies within this field.

This research project has benefitted and nourished from the drive and energy of these "early adopters" and in return the project will hopefully bring some clarity for other development environments that are about to engage in the process of implementing the Scrum framework. Thus, the purpose of this research project has been to identify and present *the challenges of becoming agile*.

About writing a Ph.D. and becoming a researcher

The process of writing this PhD has been a process of learning. As a designer I have been trained to come up with solutions by combining domain-specific knowledge with technical skills, creativity, and empathy. Some argues that designing is a discipline balancing between arts and craft. During the process of writing this PhD, I have deluded myself that being a researcher is a radically different discipline than the one I originally come from. At some point in the process I started to think different. The quote of Jaan Valsiner – "Research is a form of art, where the art is to learn from all the mistakes we make on the way" – started to make sense, and the idea of research, just as the design-discipline, being a combination of arts and craft, started to build on me.

About the participants in the collaborating companies

Due to the nature of the questions asked in this thesis, the interviews with company representatives have all been focusing on extracting and collecting a variety of challenges, mistakes, flaws, hurdles and shortcomings. The case descriptions and the subsequent analysis reflect that and may unintentionally provide the reader with a biased impression of how the companies perform in developing products. However, I cannot stress too clearly how much I admire the work of the employees in the involved companies. They have all chosen to embark on journeys that only a few have attempted before. Scrum in hardware and mechanical development is by nature an uphill battle, but these development environments have faced the challenges, cracked the codes, altered the processes, and eventually come out much stronger and competitive. Hats off.

ACKNOWLEDGEMENTS

I am thankful that I got the opportunity to make this PhD thesis. Writing it has been a great experience mostly with *ups* but also some and *downs*. For sure, it has been an experience that I wouldn't go without, but at times it really has been heavy going. Friends, family, and colleagues have been extremely supportive during this process and I would like the thank all of those who have made this work possible:

- For great supervision: Kaare Eriksen and Christian Tollestrup. You two guys are largely responsibility for this, you know… Thank you for a great collaboration!
- For collegial support and great advice: Louise Møller Nielsen, Søren Bolvig, Thomas Arvid Jaeger, Poul Kyvsgaard Hansen,
- For sharing your deep insights and discussing the topic with me: Bent Myllerup, Martin Thorshauge, and Jacob Steendahl Nielsen
- For your valuable time and interest: All the interviewees from the seven participating companies
- For leading me on the track of Agile Development: René Clausen Nielsen (Advice Digital), Ole Vilster (Nokia), Henrik Søgaard Pedersen (Inwido Denmark), Gitte Klitgaard Hansen (IBM)
- For an enriching research collaboration about Agile Development in educational environments: Chris Dowlen, London South Bank University
- For Proof reading: My dad (who has not complained about it even once)
- For support, entertainment and dinner: My great family and family-in-law.
- And finally, for bearing with me during the last couple of months: Lovely Mette (who has taken care of everything at home), Carla (who is now able to ride a bike by herself!), and Vigga (who was born this spring and just gave me the biggest smile when I left home this morning)

Nis Ovesen
June 2012

LIST OF CONTRIBUTING PAPERS

The published papers in the list below have each in their own way contributed to the process of writing this thesis. However, none of the papers are directly included in the content.

- Ovesen, N. and Dowlen, C. The Challenges of Becoming Agile – Experiences from New Product Development in Industry and Design Education, In *Proceedings of the 14th International Conference on Engineering and Product Design Education, E&PDE2012*, Antwerp, September 2012 (Design Society, UK)

- Ovesen, N., Eriksen, K. Tollestrup, C. Agile Attitude: Review of Agile Methods for use in Design education, In *Proceedings of the 13th International Conference on Engineering and Product Design Education, E&PDE2011*, London, September 2011 (Design Society, UK)

- Ovesen, N., Eriksen, K. Tollestrup, C. Speeding up development activities in student projects with time boxing and scrum. In *Proceedings of the 13th International Conference on Engineering and Product Design Education, E&PDE2011*, London, September 2011 (Design Society, UK)

- Ovesen, N. Changeability: Unfolding a Method for Learning and Designing in a fast paced Market, In *Proceedings of the 12th International Conference on Engineering and Product Design Education, E&PDE2010*, Trondheim, September 2010 (Design Society, UK)

- Ovesen, N. Design Processes – Learning from Associated Professions, *Design Education 2050 - ICSID Design Education Conference 2009*, Singapore, November 2011

CH.01. INTRODUCTION

Every morning in Africa, a Gazelle wakes up. It knows it must run faster than the fastest lion or it will be killed. Every morning a Lion wakes up. It knows it must outrun the slowest Gazelle or it will starve to death. It doesn't matter whether you are a Lion or a Gazelle... when the sun comes up, you'd better be running.

"In today's fast-paced, fiercely competitive world of commercial new product development, speed and flexibility are essential. Companies are increasingly realizing that the old, sequential approach to developing new products simply won't get the job done."

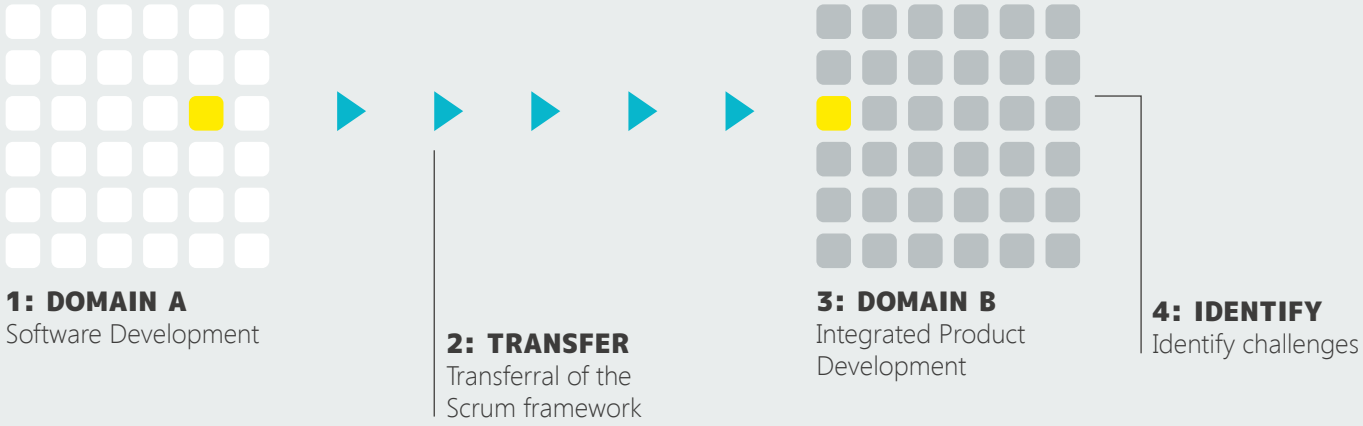
TAKEUCHI & NONAKA, 1986

Now twenty-five years old, the statement from the 1986-edition of Harvard Business Review seems more relevant than ever before. Commercial markets are moving faster and faster, and time as a strategic source of competitive advantage is receiving increasing attention (Souza et al., 2004). Furthermore – and just to make the situation even more challenging – the competitive game of product development is bullied by constant changes in customer behaviour, competitors' initiatives, breakthroughs in technology, and even by internal organisational changes (Smith, 2007). These are the present conditions for Development Teams, and at the same time also the point of departure for this research project.

1.1 MOTIVATION AND CLAIM OF THE STUDY

The motivation for this research project is twofold. Firstly, in the considerations about how to improve the conditions for product development teams, this research project turns its attention towards the software industry with curiosity and a desire to learn from it. Secondly, *the Manifesto for Agile Software Development* (Highsmith, 2001) is celebrating its ten-years' anniversary; and companies within the domain of integrated product development are starting to adopt its agile methods in the development of physical products. The two motives align quite well. A need for new development methods is now generally acknowledged, and this research project takes inspiration from other industries in its search for such methods. And companies are actually starting to do exactly the same thing by specifically adopting Agile Development methods from the software industry.

Agile Development is a general term covering a number of different, but somewhat similar, methods that have evolved within the software community as a responce to years of market uncertainty, increasing pace and competition. It is also an acknowledgement and a result of the high complexity found in many software projects. In contrast to the traditional Stage-Gate process control, which is most often used in integrated product development (Cooper, 2011), agile methods represent an iterative and incremental approach to development activities. Agile methods emphasise flexibility and openness towards changes in the product requirement specification when the surrounding market and potential users demand it. In traditional development a well-known criteria of success has often been the ability of "sticking to the plan" (Smith, 2007). Based on the motivation described above, the aim of this research project is to investigate the challenges of implementing agile methods from the domain of software in the domain of integrated product development. The agile framework called *Scrum,* developed by Ken Schwaber and Jeff Sutherland, is known to be one of the most widely used agile practices, and it is the one used by the participating companies in this project. So here Scrum is regarded as the representative agile development method. The basic manoeuvres of this research project are illustrated in figure 1.1 to the right.



1.1 The overall manoeuvre of this research project is to bring the Scrum framework from domain A (software development) to domain B (integrated product development) and identify the challenges that surface in domain B as a result of this transfer.

The claim of this dissertation is that Scrum – as a representative of agile development methods – can be successfully implemented in integrated product development environments. However, it is not the purpose of the project to measure performance improvements as a result of Scrum, but merely to identify the challenges of implementing it. The research project argues that the implementation of Scrum is often motivated by a number of fundamental challenges affecting the development organisation and the activities of the development teams. These challenges are 1) a demand for faster development, 2) difficulties in handling in-project changes, and 3) increased product complexity. The research project furthermore claims that the adoption of Scrum in integrated product development carries along a series of additional challenges.

The basis of these claims is a series of audio-documented interviews with employees at companies implementing Scrum in their product development environments. The companies are six medium and large-sized development- and production companies and a management firm with competencies in Scrum training. A number of video-documented workshops at a client of the management consultancy have furthermore contributed to the research by deepening the insights about how Scrum is implemented in the organisation in practice.

The rest of this introductory chapter consists of 1) *a positioning of the research project within the field of product development*, 2) *an introduction to the fundamental challenges mentioned above*, and 3) *a presentation of the research questions as well as the alleged gaps in the present knowledge*. Finally, the chapter will present a summary of the main contributions of the research project as well as an overview of how the rest of the dissertation is composed.

IM.01 **ŠKODA FABIA ASSEMBLY LINE, CZECH REPUBLIC**
Steady pace and getting rid of waste is the norm in Lean manufacturing. Now, similar concepts are applied to development.
Photo: Nataliya Hora

1.2 POSITIONING THE RESEARCH PROJECT

Faster, Better, Faster, More! This research project is part of the discussion of performance within the field of Product Design and Development. Traditionally, this discussion has been centred around manufacturing systems, partly caused by the arrival of concepts like Lean Manufacturing (Krafcik, 1988) – a set of tools for improving flow and reducing "waste" in manufacturing. But an increasing awareness of New Product Development activities as a significant factor in achieving strategic and financial goals has since then broadened the performance discussion to also include the development processes (Hertenstein & Platt, 2000). This research project concentrates on the integrated product development process prior to manufacturing, and it investigates how companies utilise Scrum to improve their performance.

The idea of high performance alone is not meaningful unless a set of matching performance measures is established, but in contrast to measuring performance in manufacturing, measuring the development performance is rather difficult. The general scheme in manufacturing is about repetition; streamlining the manufacturing process is relatively simple because of its small human factor. The nature of development, in contrast, is often one-off events with a large human factor (Reinertsen, 2009). A study of Hertenstein & Platt (2000) reveals that about half of their surveyed companies measure product development performance, and that these companies use a combination of financial and nonfinancial measures. It is not the purpose of this project to throw light on how companies measure development performance, nor to come up with new ways of doing it more rightfully. Performance improvement has merely been the typical point of departure, when the companies involved in this research project turned to the idea of implementing the Scrum framework.

Scrum has evolved from the field of software and systems development. Within this field, Scrum has often been used from the very first

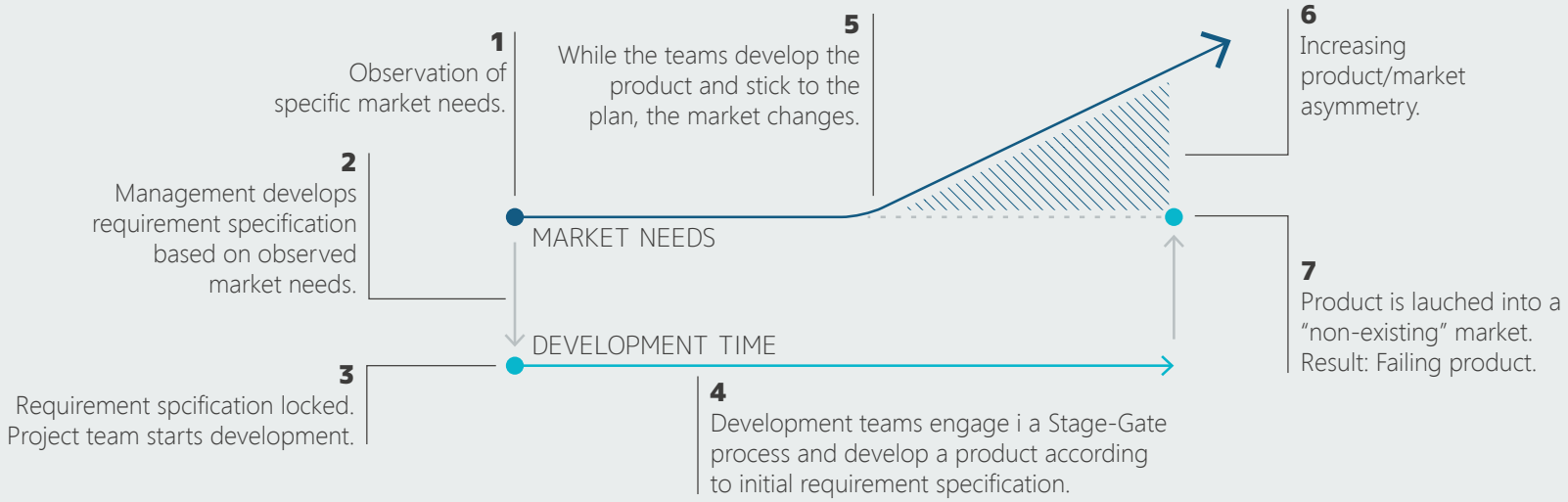
1.2 Positioning the project within Development stage in the Stage-Gate model by Cooper (2011)

stages of development and all the way throughout the implementation and maintenance phases. This project, however, is positioned within the domain of *integrated product development* with a focus on implementing the Scrum framework *deriving* from software development. Here "Integrated Product Development" (IPD) is used as a term, that – in contrast to software development – spans a wide areas of elements such as mechanical development, construction, firmware, hardware, interface design and styling, but often also software. Pure software development projects often have a rather fluid border between development, production and maintenance, but IPD has some inherent constraints that most often make a distinction between development and manufacturing necessary. The focus of this research project is therefore limited to the phases in the Stage-Gate model of Cooper (2011), prior to the ramp-up phase. This is illustrated in figure 1.2 above.

Within this framing of the project two specific areas are of interest. These are 1) the practical execution of product development tasks within cross-functional development teams using Scrum and 2) organisational issues in regard to implementing Scrum in a traditional Stage-Gate organisation.

The first area of interest concerns the applicability of Scrum in cross-functional development teams with a special interest in the work of the product designers – including both mechanical developers and industrial designers. The second area of interest concerns the organisational challenges of implementing Scrum in the integrated product development environment in regard to existing company structures, roles and responsibilities. The inclusion of organisational challenges is justified by the fact that Scrum is not only regarded as a development method, but is considered a complete process-control framework for development and thereby influences a larger part of an organisation. The empirical context of this research project is therefore development departments with cross-functional teams in larger companies implementing Scrum. All companies involved in the project are located in Denmark but operate and compete in a global market. To illustrate the general stance in the involved companies, the quotation below seems rather descriptive:

> *"(...) At our company we have also just started trying out Scrum in device development, and we have now completed a couple of Sprints [Scrum term for short iterative cycles]. We are already seeing advantages, but are also experiencing some challenges."*
> *Brian Munkholm, Radiometer*



1.3 Conditions in an unstable market

1.3 FUNDAMENTAL CHALLENGES IN PRODUCT DEVELOPMENT

As just stated, this research project centres on the cross-functional development teams and their ability to continuously improve performance and produce commercial, competitive, and viable product concepts ready for ramp-up and production. For even the best-organised and competent team this is not an easy task, and it will almost inevitably encounter some challenges during the course of any project. However, companies have always been using various kinds of management and development frameworks for steering their development activities through the otherwise chaotic surroundings. During history these development frameworks have had many different forms, but a seemingly prevailing ideal throughout the latter part of the 20th century has been heavy up-front planning followed by sticking to the plan (Smith, 2007). This model works well for product development in a stable and predictive environment, but to many companies these are not the typical conditions.

Figure 1.3 above describes the conditions for the development teams in unstable markets. In the following chapters three fundamental challenges to product development as well as corresponding approaches are presented. The fundamental challenges are *accelerating speed*, *change-ability*, and *increasing product complexity*. All three challenges are associated with the figure above, and all are regarded as reasons for implementing agile methods.

1.3.1 ACCELERATING SPEED

> *"Faster, faster, faster! We're in a world that is obsessed with speed. Time has won the race to become our most valued resource (...) Today, speed is everything."*
> *Joseph T. Vesey, 1991*

Accelerating speed is arguably an endless challenge to development teams as – due to the basic commercial and market-oriented scope of product development – they are in a ubiquitous state of competition.

The fact that the general market speed is increasing is supported by both philosophy and practice.

From a philosophical perspective, the concept of speed has been described in the Dromology, meaning *the science of speed*. The cultural theorist and philosopher Paul Virilio coined this philosophical genre, and one of its most important theories is that the speed of any given development will always attempt to rise, and that the organisation of the world changes with this acceleration of speed. According to Virilio, the logic of Speed is that it will always rise. He furthermore claims that mankind has always organised its world with only one goal: Everything must go faster and faster (Brügger, 2001).

One could ask, then, why everything is not escalating into incomprehensive chaos if the speed of any given development is always accelerating. The answer is to be found in the fact that a new, fast development only supplements the existing, rather than succeeding it. In that way old and new exist side by side. However, Dromology tells that the greatest speed or development is always prevalent; the one that is leading affects all areas. Everything that does not accelerate with it will not fully disappear, but instead become less influencial. To summarise the consequence of Virilio's theory to teams working with product development, it is argued that the teams will inevitably need to be continuously more efficient in bringing products to market, as the market will always increase its speed.

As mentioned earlier, it is not only from a philosophical perspective that speed is of fundamental concern to product development teams. The need for speeding up development activities is also documented in marketing literature. Menon et al. (2002) advocate exactly the same concern from a market-competition perspective:

> *"(...) In addition to higher profit, for example, a short new product development (NPD) cycle can afford companies significant cost reduction, greater market segment coverage, and a dominant leadership role in the marketplace."*
> *Menon et al., 2002*

Menon et al. (2002) argue that a reduced product development cycle will result in a competitive edge against competitors. This argument matches the discussion about time-to-market to which Professor of Management, Joseph T. Vesey, has been contributing. Vesey argues that traditional management practice has been based on having plenty of time, and that heroes of this practice were the ones who thoroughly weighed the options and operated within budget on low-risk projects. Due to global competition pressure is now put on managers to reduce time-to-market (Joseph T. Vesey, 1991). His argument is substantiated by a study of McKinsey & Co. that shows the relationship between time-to-market and profits.

IF YOUR COMPANY IS LATE TO MARKET BY:					
6 Mo.	**5** Mo.	**4** Mo.	**3** Mo.	**2** Mo.	**1** Mo.
Your gross profit potential is reduced by:					
-33%	-25%	-18%	-12%	-7%	-3%
Improve time-to-market by only 1 mo., profits improve:					
+11,9%	+9,3%	+7,3%	+5,7%	+4,3%	+3,1%
For revenues of $25 Million, annual gross profit increases:					
+$400K	+$350K	+$300K	+$250K	+$200K	+$150K
For revenues of $100 Million, annual gross profit increases:					
+$1600K	+$1400K	+$1200K	+$1000K	+$800K	+$600K

TABLE 1.1 TIME-TO-MARKET VERSUS PROFIT
(McKinsey & Company in Vesey, 1991)

As seen in table 1.1 above, there is a great economical gain in reducing the time-to-market and a rather great loss when being late. The fact underlines the importance of focusing on speed in development from a financial perspective. It thereby supplements the arguments of speed being a fundamental challenge that development teams are facing.

Several strategies towards increased market speed exist. Two separate strategies are A) being ahead of the market in order to buy more development time and B) being faster by shortening the development lead-time. The two strategies can be combined in order to further improve the conditions for keeping up. The strategy of being ahead of the market is accomplished by deploying forecasting techniques such as trend spotting and thorough market intelligence in the leading markets. However, this strategy also entails a risk as timing becomes of great importance. Being too early or too late is dangerous because of the consumer's "gap" in appreciation (Brannon, 2005). Furthermore, this approach – if applied without further caution – is still fragile to the second challenge of unexpected change.

The strategy of being faster by reducing development cycle time is possibly the most practiced way to overcome the market speed challenge. Several studies suggest various ways to achieve this. Lean Product Development, which already has been mentioned, comprises a series of initiatives that streamline development activities and create optimal development flow, aiming at steady tact and frequent product launches (Nielsen, 2011).

1.3.2 CHANGE-ABILITY

"It is change, continuing change, inevitable change, that is the dominant factor in society today. No sensible decision can be made any longer without taking into account not only the world as it is, but the world as it will be. This, in turn, means that our statesmen, our businessmen, our every man must take on a science fictional way of thinking."
Isaac Asimov

Isaac Asimov, an American professor and pioneer within the genre of Science Fiction, is one of many that could be cited about change as a dominating influence in today's world. Actually, the science fictional genre, which he represents, and the domain of product developers may be especially close to each other: Both are dealing with imagination and technological concepts that are more or less plausible in a future setting.

To development teams, change creates new opportunities and is often exactly what justifies the development of new products. But change is also making the job difficult: Throughout the course of de-

To the Moon!

A story about accelerating speed from one of the seven cases.

30 employees were gathered at a workshop in order to plan and coordinate a new project. Due to certain conditions in one of their markets, the team was under pressure from a tight schedule. The aim for the meeting was to come up with two plans. The first plan was a traditional plan that would allow the product to be launched 22 months from the day of the event. The second plan was to develop the same product ready to launch in just 13 months. The second plan was to use VVSM and Scrum as the primary drivers for the increased efficiency. The upper management advocated the second plan as the one to go for by pointing out that people had been to the moon and it could be done again. "You can do what you want, if you really try."

The situation was far from a typical development project, but it was a experiment was clear message from the management questioning the ordinary process.

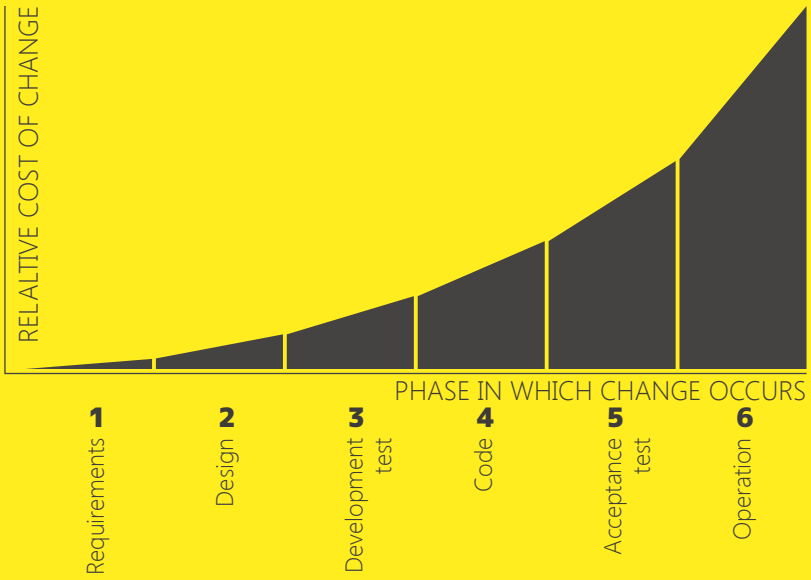
veloping a new product, the market can change, and as a result the original product specification risks becoming obsolete. A competitor presents its new product that makes it necessary to revise the plan, or an unexpected breakthrough in technology forces the development team to change horses in the otherwise thoroughly planned development project. Seen in isolation, these events are not problematic or unnatural, but it is when they are to be taken into account in a project, which is already in an advance state, that makes things difficult. Especially when it comes to the financial aspect of adapting to changing markets, technologies or competitor initiatives. Just as Time-to-market is a fundamental aspect in regard to speed, so "Cost of Change" is an essential aspect in regard to handling changing environments. According to Barry Boehm, the cost of changes in software projects climbs almost exponentially as the project proceeds. This rule can arguably also to be considered true in physical product development.

Figure 1.4 below explains why changing the course of a development project midstream is often not welcomed, and why "plan the work – work the plan" has been the predominant ideal in product development.

Returning to the quote of Asimov, the morale is about the necessity of trying to look forward into the future, as we cannot expect it to be just like the present. Changes *will* happen, and one way of coping with them is to try to predict them and subsequently plan according to the predictions. This way of thinking has fostered a whole new industry of technology forecasting, market intelligence and trend spotting, which has the purpose of establishing a starting point as solid as possible with all *predictable* changes taken into account (Brannon, 2005). However, this does not change the fact that careful and strict up-front planning and sticking-to-the-plan-strategy don't incorporate any scheme for handling the *unpredictable* changes.

It becomes clear that the ability to adapt to the unpredictable changes, whether it is in form of opportunities or threats, is crucial for being able to launch a product that is in sync with the market. This need for flexibility throughout a project and ideally up to launch is paradoxical, as non-reversible decisions need to be taken during the project (Mikkelsen & Riis, 1996; Smith, 2007).

1.4 The Cost of Change (Boehm, 1981)



Change is coming.

A story about change from a presearch interview.

"The conventional project management methods have strengths, for sure. But they also have some serious problems. Especially in our line of business, which is moving incredibly fast. Competitors are launching new products all the time. This means that when you are working on a project, you cannot just ignore all that. You have to be able to react on it, and that's virtually impossible with the conventional project management method."

The head of a development department is struggling with a rigid project management scheme that doesn't allow him to change the course of his project, when competitors launch disruptive products.

As to "Time-to-Market", what really seems to matter is the length of the period from the last point in time where changes in the specific design are implementable to the product ships. This period decides the level of sync with market (Smith, 2007). When considering the claim of increasingly faster moving markets in the previous chapter about speed, this understanding of "Time to Market" implies the importance – but also the challenges – of remaining flexible to change throughout the development process.

Just as to the challenge of speed a few strategies towards handling *change* already exist. One strategy is to accept the fact that change is inevitable and therefore try to be prepared for it. Drawing a reference to Toyota, set-based design is a technique that keeps the space for potential solutions open for as long as to "the last responsible moment" (Smith, 2007). The technique contrasts with point-based development, which concentrates on making choices, limiting flexibility towards change. Set-based design is characterised by its emphasis on constraints and by its focus on sets of solutions within the boundaries of those constraints. The quote of former baseball player Yogi Berra sums it up quite well: "When you come to a fork in the road, take it".

Some other contributions to the discussion about development process and changeability share the ideas of Agile Development from software. The two concepts *Agile Enterprise* (Lin et al., 2006) and *Flexible Development* (Smith, 2007) both claim to address the challenges of market uncertainty and change by adopting some of the specific methods known from agile software development.

1.3.3 INCREASING PRODUCT COMPLEXITY

> *"In a complicated world - you can and should control events. In a complex world you cannot control events - there are too many interacting variables. So what can you do?"*
> *Robert Paterson, 2006*

In a world of increased speed and rapid change, complexity in development projects becomes a significant issue, which both the organisation and development teams have to face (Kim and Wilemon, 2007). However, too often managers rely on leadership approaches that are based on the fundamental assumption that a certain level of predictability and order exists. This assumption encourages sim-

1.5 The concept of complexity on a scale by Snowden & Boone (2007)

SIMPLE	COMPLICATED	COMPLEX	CHAOTIC
DOMAIN OF BEST PRACTICE	DOMAIN OF EXPERTS	DOMAIN OF EMERGENCE	DOMAIN OF RAPID RESPONSE
Sense	Sense	**Probe**	**Act**
Categorise	**Analyse**	Sense	Sense
Respond	Respond	Respond	Respond

plifications that are useful in ordered circumstances, but fall short in more complex conditions (Snowden & Boone, 2007).

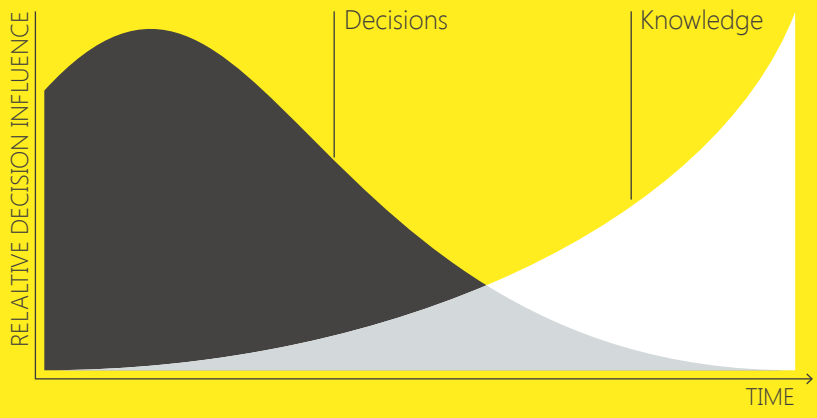
According to Kim & Wilemon (2003) complexity can severely influence product development efforts in many different areas. With a high level of complexity the risks are *reduced development speed*, *negative impact on development performance (such as high unit cost),* and *significant challenges regarding the cooperation and communication between functional groups (such as team frustration and misunderstandings)*. On the other hand issues, such as *competitive advantage* as well as *high learning and cooperative culture, are* some of the benefits if an organisation is able to tackle the complexity (Kim & Wilemon, 2003).

Many definitions of complexity have been proposed in relation to product development. Murmann (1994) and Griffin (1997) respectively define complexity as *the number of parts in a product* and *the number of product functions embodied in a product*. Tatikonda & Rosenthal (2000) broaden the definition by also including the nature of the work which the specific project implies. They argue that project complexity is defined by *the nature, quantity and magnitude of the organisational subtasks and subtask interactions required by the project.*

Referring to the previous chapter about *change*, complexity could possibly be described as *second-order change*, or as Snowden & Boone put it: The realm of the "unknown unknowns" (Snowden & Boone, 2007). They relate product development complexity to decision-making and suggest a scale that positions complexity in-between "complicated" and "chaotic" as illustrated in figure 1.5 to the left.

This understanding of complexity suggests that most New Product Development projects lie within the domain of emergence, as unknown unknowns exist. A complex context is like a rainforest, constantly in flux. Species are extinct, weather patterns change, an agricultural project reroutes a water source – and the whole of it is far more than the sum of its parts (Snowden & Boone, 2007). The Domain of Emergence thereby aggravates the paradoxical condition that knowledge is minimal in the beginning of a project – traditionally the time for big decisions as shown in figure 1.6 below.

1.6 Influence of decisions vs knowledge (Mikkelsen & Riis, 1996)



Product Complexity.

A story about product complexity and development challenges.

For a long time the software team at Company A had been struggling with the development of device drivers for one of their products. The team was scattered across three continents, and the problems were threatening the whole project. It was first when the management decided to gather the three sub-teams for two weeks in one place and conduct the troubleshooting activities as Scrum that the project started to get back on track.

Prior to the decision of gathering the team, the development work had reached a level of complexity that could simply not be handled while the team was also struggling with collaboration difficulties due to distance and communication issues.

1.4 DELIMITATION

The three challenges presented above are considered the most relevant to cover in relation to this project. However, it does not mean that other challenges do not exist. Development teams are challenged in a number of other ways than those of speed, change and complexity. Some of the companies participating in this project are faced with considerable challenges in relation to legislative issues, outsourcing issues, globalisation issues, IPR issues, communication issues, and many others. Even though those seem just as important as the ones presented here, they are not within the scope of this project and therefore not being dealt with. The project is also delimited by only considering challenges to the development teams and organisational issues that are related to the development environment. This means that other important functions in the participating companies such as production, finance, administration, marketing, and sales are not taken into account in this project.

1.5 GAPS IN THE PRESENT KNOWLEDGE

As it is shown in the chapters above, various strategies and isolated approaches towards the three challenges already exist, but knowledge about the practical implementation of agile methods as a strategy against the fundamental challenges seems to a certain extent to be missing. The application of Scrum in integrated product development – due to its immediate simplicity – offers practical guidance that other approaches arguably are missing. Nevertheless, it is clear that Scrum has not been very well described in relation to integrated product development, as it is a method that has been evolving in the field of software. A survey of the various product development environments in Denmark quickly substantiates this impression. Only a few companies consider themselves to be beyond an experiential process of testing Scrum outside the software environments. Documented experience is therefore lacking.

1.6 RESEARCH QUESTIONS

In view of the gap in present knowledge described above, this research project is led by the overall question of *what are the basic challenges of becoming agile in integrated product development environments using Scrum?* This question is further elaborated in the following sub-questions:

RQ.1
What are the primary challenges to **the product designers** when working with **Scrum** in **integrated product development** environments?

P.38
Description of the Product Design discipline

P.20
Scrum Explained

P.32
Definition of integrated product development

RQ.2
What are the primary **organisational** challenges when implementing **Scrum** in **integrated product development** environments?

P.42
Unfolding organisational aspects

1.7 Summary of General Conditions and Challenge Themes from the findings of this research project

1.7 MAIN CONTRIBUTION OF THIS RESEARCH PROJECT

By answering the research questions presented in paragraph 1.6, this research project aims at providing an improved, empirically based, and more precise understanding of the *actual* challenges of implementing and conducting Scrum in integrated product development environments. Furthermore, the project should provide indications towards viable solutions to some of the identified challenges, which can also be the groundwork for future research about how to overcome or meet the challenges presented in this thesis.

From a design perspective this thesis may offer an interesting insight in the challenges of an iterative approach in a framework that covers more than just the designer's expertise. In essence, Scrum is not new to design, but the extent of the framework may allow designers to use their expertise in a broader context with colleagues from different disciplines. Even though this project indicates reluctance from product designers to engage in development conducted through Scrum, a potential is acknowledge due to its immediate resemblance with the basic design process.

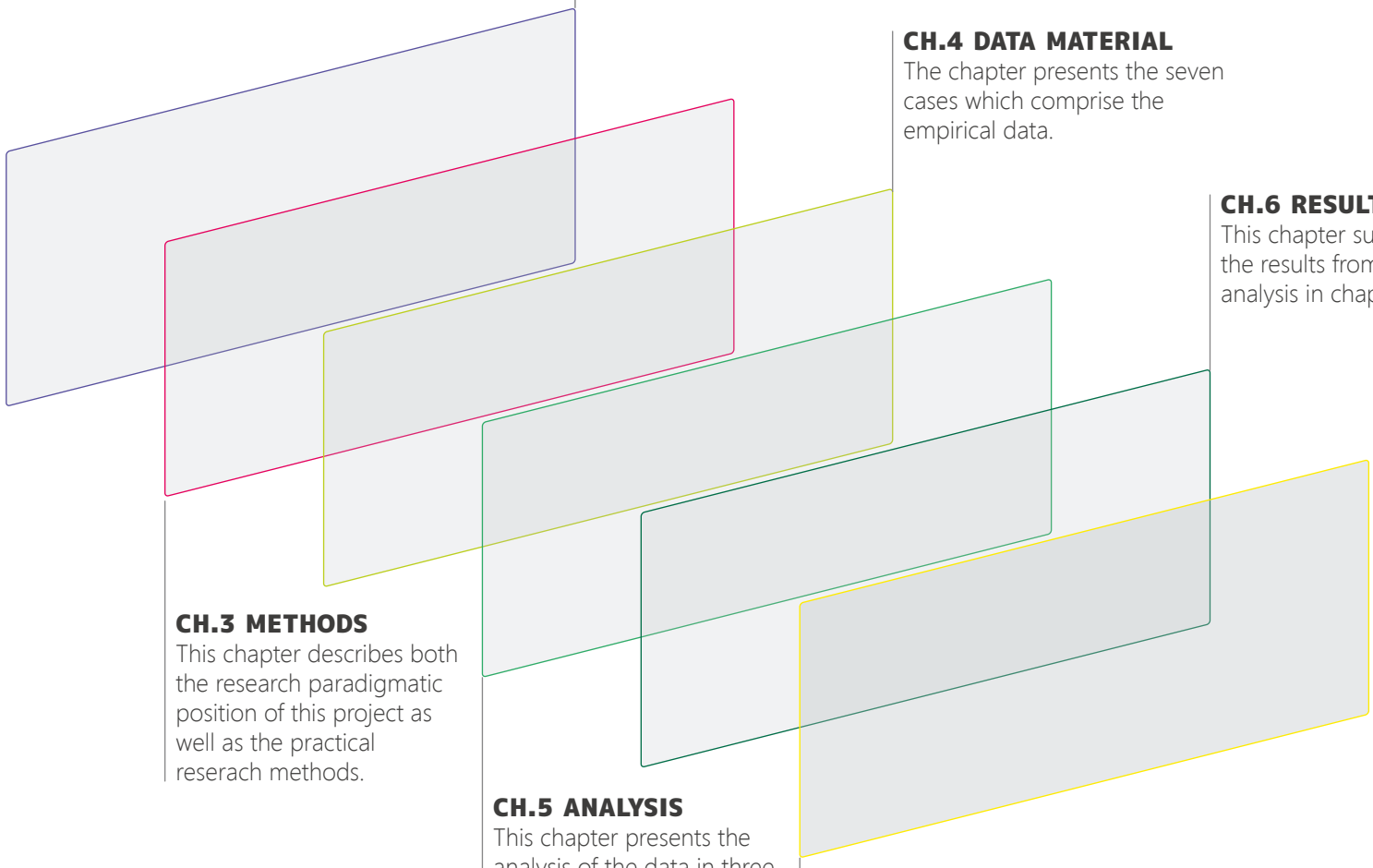
The research project is primarily of a descriptive character as it focuses on creating an overview of the challenges of becoming agile with Scrum in integrated product development environments. However, due to the scarce knowledge within this specific field, the findings of this project are considered to be of value.

More specifically, the findings of this project are divided into three levels of abstraction and of increasing generalisation: *practical challenges identified in the seven cases*, *challenge themes*, and *general conditions*. While the *practical challenges* from the first level are presented in Part 1 of the analysis (chapter 5.1), the challenge themes and general conditions are briefly summarised in figure 1.7 above and later unfolded in chapter 5.2 and 5.3 as well as in a condensed version in chapter 7.2 of the conclusion.

1.8 SUMMARY OF CHAPTER 1

The main objective of chapter 1 has been to provide an introduction to this research project by clarifying the motivation and the position of the study. The introduction has furthermore presented three fundamental challenges to product development that motivates companies to implement Scrum. This has led to a description of a gap in present knowledge as well as to the main research questions answered in this thesis. The list below briefly summarises the contents of chapter 1.

- A need for new development methods is acknowledged and this research project takes inspiration from other industries in its search for such methods.
- A few companies in Denmark are starting to experiment with the agile development framework, called *Scrum*, outside its original domain of software by implementing it in integrated product development.
- The implementation of Scrum is often motivated by the fundamental challenges of product development, namely *accelerating speed*, *change-ability*, and *increasing product complexity*.
- The project argues that present knowledge lacks information about how Scrum is implemented and conducted in integrated product development environments, which leads to the formulation of the main research questions of this thesis.



CH.2 THEORATICAL FOUNDATION
The chapter unfolds the Scrum framework, integrated integrated development process, and organisational issues.

CH.4 DATA MATERIAL
The chapter presents the seven cases which comprise the empirical data.

CH.6 RESULTS
This chapter summarises the results from the analysis in chapter 5.

CH.3 METHODS
This chapter describes both the research paradigmatic position of this project as well as the practical reserach methods.

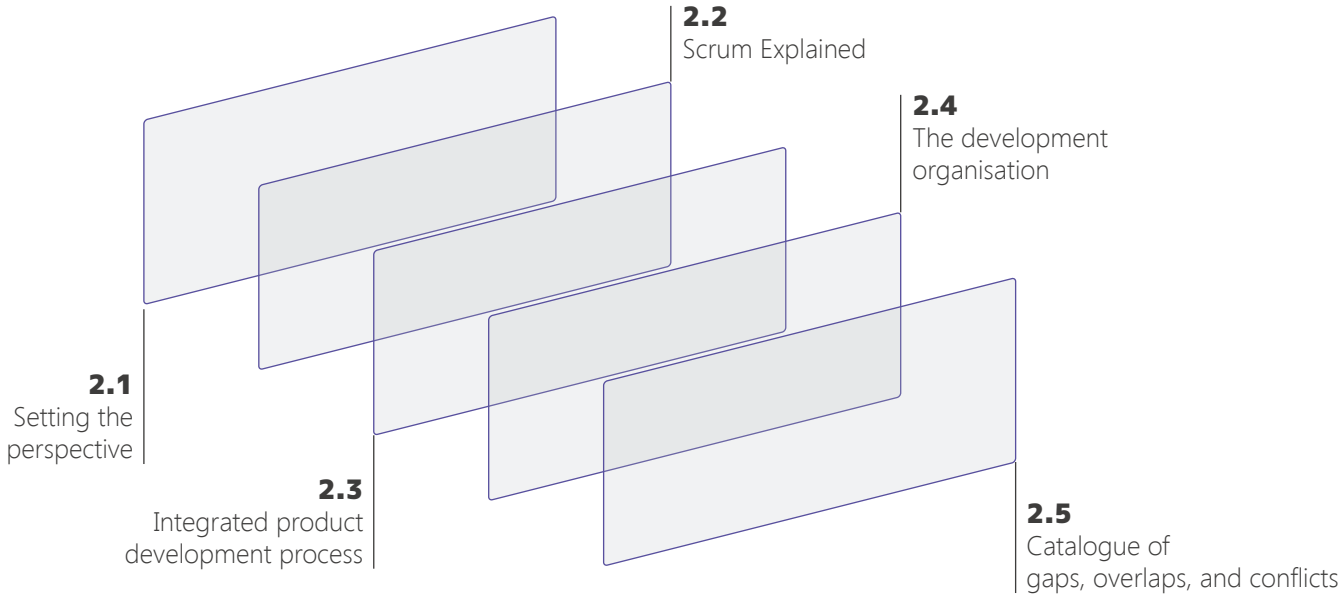
CH.5 ANALYSIS
This chapter presents the analysis of the data in three parts: Challenges, Themes, and General Conditions.

CH.7 CONCLUSION
The last chapter presents the answers to the research questions and evaluates the reliability and the methods used in the project. The chapter also include a series of alternatives and workarounds for conducting Scrum in integrated product development

1.8 OVERVIEW OF THE FOLLOWING CHAPTERS

The rest of the thesis is composed as shown in this figure.

CH.02. THEORETICAL FOUNDATION



2.1

OVERVIEW OF CHAPTER 2

Chapter 2 comprises descriptions of Scrum, integrated product development, the development organisation, and finally a presentation of a catalogue of gaps, overlaps, and conflicts

The purpose of this chapter is twofold. The first purpose is to establish a state-of-the-art theoretical foundation with the intention to position the study in relation to existing knowledge and theories. This should also allow a later clarification of how this research project adds to the state-of-the-art. The second purpose is to develop a theoretical framework for understanding and interpreting the empirical findings by establishing a theoretically based catalogue of issues, which can be expected when transferring Scrum from software development to integrated product development.

The chapter builds upon the project positioning presented in chapter 1, and starts out by clarifying the perspectives in which both the empirical scope and the theoretical framework are to be seen. After introducing the concepts of Integrated Product Development process and Organisational aspects, the project moves on to establish the theoretical frame of reference. This consists of a part about the Scrum framework followed by two bodies of relevant knowledge, which in each way relate various insights and elements to Scrum. These knowledge bodies respectively concern Scrum in relation to *Integrated Product Development* and Scrum in relation to the *development organisation*. Finally, the chapter contains a catalogue of gaps, overlaps, and conflicts between the two knowledge bodies and Scrum. The catalogue derives from the theoretical framework and is marked as small concluding statements in throughout the chapter from 2.3 and onwards. The overall structure of this chapter is illustrated in figure 2.1 above.

2.1 SETTING THE PERSPECTIVE

As it has been described in chapter 1, the empirical basis of this research project is a series of interviews and activities within the development environment of seven Danish companies working with the Scrum framework. The interest in this specific empirical scope could be motivated by a number of different research strategies, focusing on a multiplicity of themes ranging from management, technology, design for production to market alignment, organisational issues, risk management, and many others. However, though it might be tempting to further follow such themes, the primary intent in *present* research project is to retrieve data from development environments with the purpose of throwing light on the development challenges *and* organisational challenges in transferring Scrum to the domain of integrated product development.

The main perspective, in which this project approaches the empirical data and the following analysis, is delimited by two major concepts related to the research questions. These concepts, *integrated product development process* and *development organisation,* are defined in the subsequent subchapters and, later in this chapter unfolded in relation to Scrum.

2.1.1 DEFINING INTEGRATED PRODUCT DEVELOPMENT PROCESS

An important concept in this project is *the integrated product development process*, which constitutes the first body of knowledge related to the Scrum framework. The definition of integrated product development process has primarily been borrowed from Andreasen & Hein (2000), but for the purpose of this project, the concept has been broadened to include additional aspects. Andreasen & Hein (2000) describe the potential of integrating various company divisions in a coordinated manner. Barry (1993) furthermore supplements this by defining integrated product development as interplay between various functional disciplines. The two definitions supplement each other well and both of them are further elaborated later in chapter two.

For the purpose of establishing a broad theoretical foundation, the concept of integrated product development is flanked by two additional concepts: *process control* and *design process*. While integrated product development represents the interplay across divisions and functions on the operational level, process control models is considered to represent a higher level of abstraction, which in this project is defined as the macro-level. The latter concept, design process, represents the intrinsic process of creation, which is taking place on a personal level during development assignments. Due to its relation to the individual developer, the design process is denoted as the micro-level in this research project. This division, which purely serves as a basis for the structure of the knowledge and insights unfolded throughout chapter 2.3, is listed below:

- Macro-level: Process control systems
- Meso-level: Integrated product development
- Micro-level: Design process

2.1.2 DEFINING DEVELOPMENT ORGANISATION

The second knowledge body related to the Scrum framework concerns the development organisation that supports the integrated product development process. In this research project, the development organisation is defined by three interlinked elements. Firstly, the basic *organisational structures* in which the communication and activities are taking place. Secondly, the supporting management system that surrounds the third part, which is the development team itself. The organisational structures described in this theoretical foundation are primarily borrowed from Hayes et al. (1988) and Ulrich & Eppinger (2003), whereas management roles and responsibilities derive mainly from Kerzner (2009). Aspects related to the development team draw on several sources and are clarified later in chapter two. The three parts are listed below:

- Organisational Structures
- Management roles and responsibilities
- Development Team

2.2 SCRUM EXPLAINED

So far, focus has been on presenting the circumstances and insights that are necessary to understand the context and driving perspective of this project. Now the focus shifts to describing the various parts in the theoretical frame of reference. The overall chapter is organised, as described earlier in figure 2.1, starting with an overview of the Scrum process and its relation to Agile Development. After this two separate themes are presented and related to the Scrum process.

2.2.1 OVERVIEW OF THE SCRUM FRAMEWORK

The Scrum framework is continuously evolving, and throughout the recent years, the two founding authors, Ken Schwaber and Jeff Sutherland, have published a new edition of their Scrum guide on regularly. This overview, which is based on the latest Scrum guide from 2011 as well as other sources, will present the basic structure and concepts of Scrum. But before going into detail about what Scrum is, some clarification about how to talk about Scrum may be appropriate.

The word "Scrum" may refer to a process, a method, a methodology, a set of rules, a concept, and even an attitude. However, this causes quite some confusion. Is Scrum a process? Not in a technical sense of the word. *A process* is usually based on routines, repetition and series of steps taken in order to produce the same result. As it will be illustrated later in chapter 2, Scrum has some built-in mechanisms for adjusting itself, which is why the term *process* may not be the right one. Scrum is more than just a *concept*; it is an actual practice. Some people use the term "Scrum methodology," which is probably because they mix up *methodology* with *method*. Methodology means the study of methods, and that is not what Scrum is about. Scrum may more rightly be understood as a *method*, but that is not offering much guidance in how to perform the actual development activities. Scrum is basically a set of rules that control certain activities in a development team. This is why Scrum is named a framework in this research project. The term *framework* indicates that Scrum is merely a frame of certain principles, activities, and tools.

Scrum is a framework for managing product development. It is based on iterative development cycles called sprints in Scrum terminology. Sprints typically last two to four weeks and follow each other like pearls on a string. The results of the development activities in each sprint are *potentially shippable increments* of the final product. The Scrum framework is illustrated in figure 2.2 to the right.

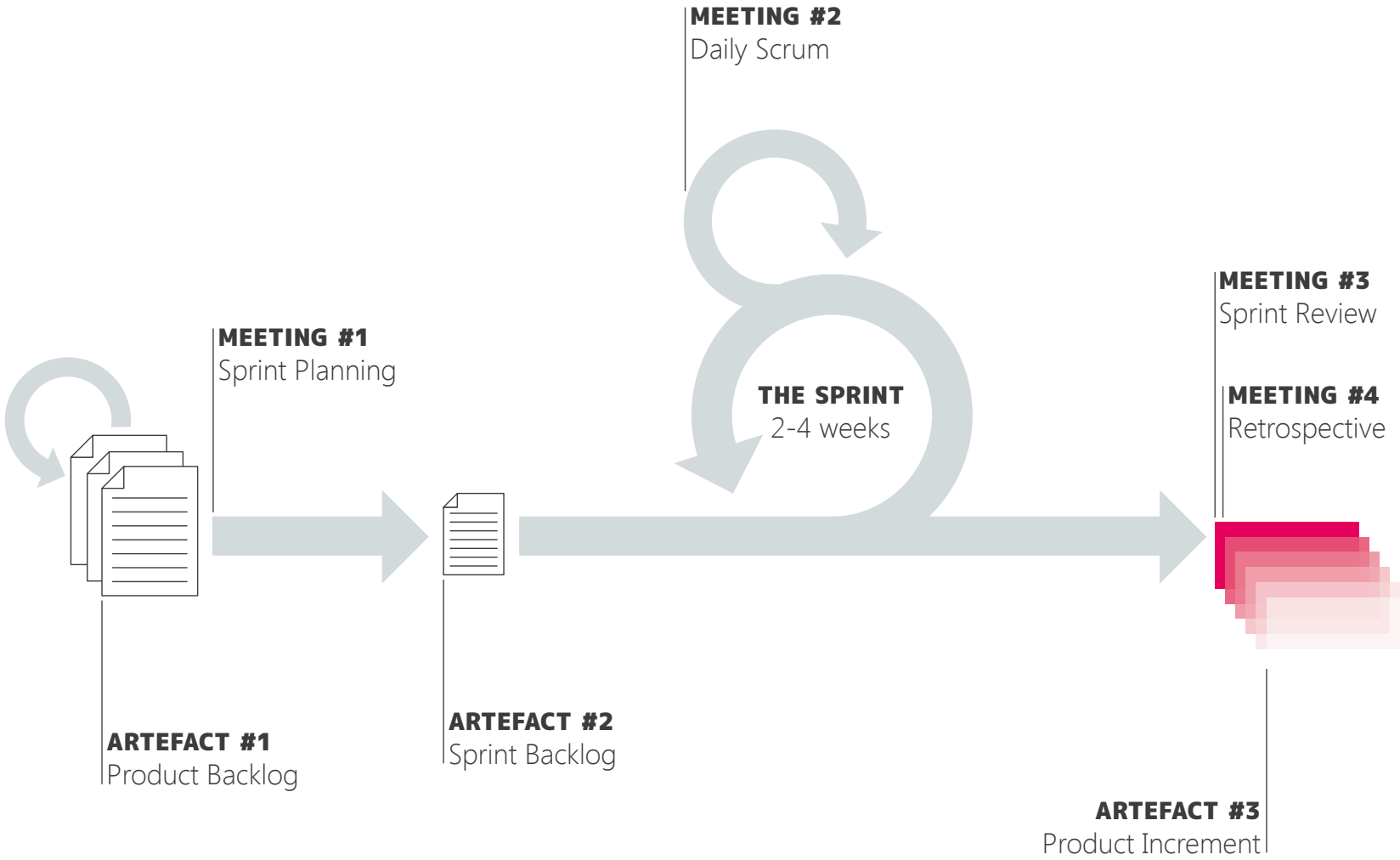
One characteristic that makes Scrum different from most other development frameworks is the locked periods or *time boxes* that decide how development tasks are divided. In traditional development it is the extent of the total workload and its logical division that defines the time-wise extent of the separate phases. In Scrum, development activities are broken down into small sub parts that can be done within a sprint. This reverse relationship between time and workload has great impact on establishing a common commitment in the development team, and at the same time it sets the scene for the formalised meetings and events in the Scrum framework.

2.2.2 THREE ROLES, FOUR MEETINGS, AND THREE ARTIFACTS

The Scrum framework consists of three types of roles and four different types of meetings. Beside these elements it also contains a number of artefacts, but depending on the choice of sources, the included artefacts vary. The purpose of this sub-chapter is to presents these elements in Scrum.

SCRUM CONTAINS THE FOLLOWING ELEMENTS:
3 ROLES The Product Owner, the Developer Team, and the Scrum Master
4 MEETINGS Sprint Planning, Daily Scrum, Sprint Review, and Retrospective
3 ARTEFACTS The Product Backlog, the Sprint Backlog, and the Increment

TABLE 2.1 ROLES, MEETINGS, AND ARTEFACTS IN SCRUM
(Schwaber & Sutherland, 2011)



2.2 THE TYPICAL REPRESENTATION OF THE SCRUM FRAMEWORK

The Scrum framework consists of three roles, four types of meetings, and at least three artefacts.

ROLE #1: THE PRODUCT OWNER

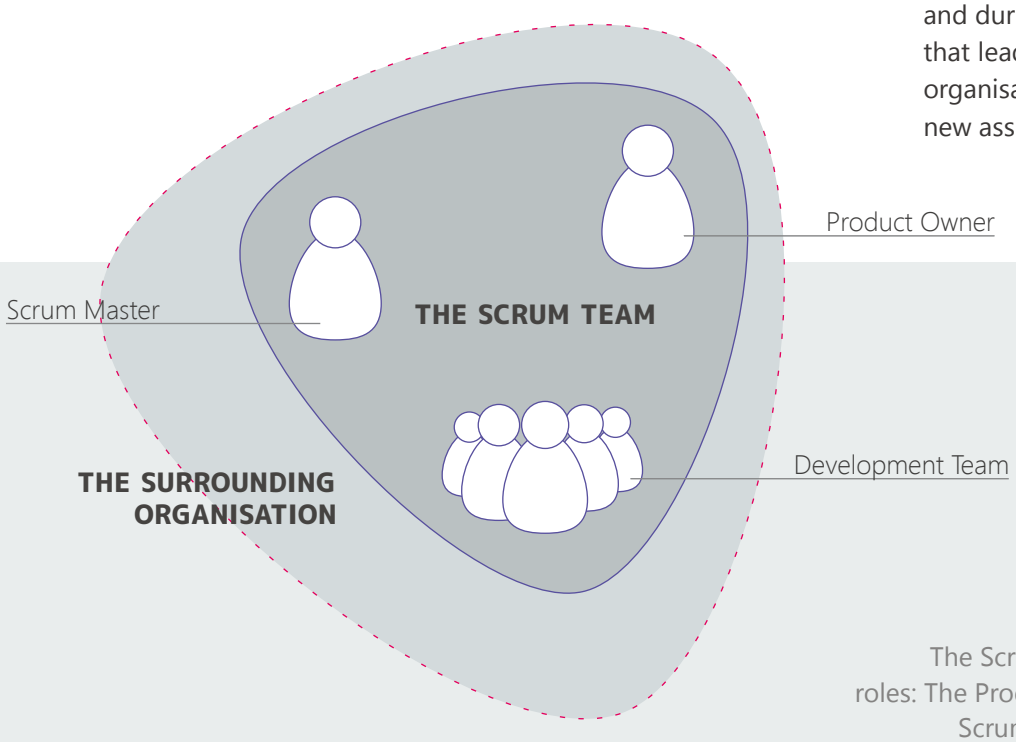
As the name indicates, the Product Owner *owns* the product in development. He or she represents the full product responsibility and is accountable for making the decisions that will drive the development in the direction which will create the most value for the customer. The Product Owner can even *be* the customer.

It is the responsibility of the Product Owner to get the most value out of the work of the development team by clearly communicating the jobs to be done in the sprints. He or she does this with the use of the Product Backlog artifact – a list of clearly prioritised tasks to be solved. Maintaining the Product Backlog is the sole responsibility of the Product Owner, but this task *can* be delegated to the development team.

ROLE #2: THE DEVELOPMENT TEAM

The Development Team consists of the persons who do the actual development work and who have the mutual responsibility of delivering the agreed-upon results at the end of each Sprint. Scrum emphasises cross-functional and self-organising teams without any formal or informal hierarchic structure. Everyone in the development team has the title of "Developer," and together the team members decide how the agreed-upon tasks are solved. Before the beginning of each Sprint, the team members collectively commit themselves to a certain workload, and it is therefore also important that tasks can be solved by more than just one certain person on the team.

According to the Scrum Guide, the Development Team size should be between three and nine persons. Most often, the Development Team is co-located within the same physical environment to ensure a clear and continuous communication amongst the team members, and during a Sprint, the Development Team is only working on tasks that lead directly to the next Sprint Goal. No one in the surrounding organisation is allowed to further load the Development Team with new assignments during the Sprint.



THE ROLES IN SCRUM

The Scrum Team-organisation contains three formal roles: The Product Owner, the Development Team, and the Scrum Master. The Scrum Master is often, but not necessarily, a part of the Development Team.

2.3

ROLE #3: THE SCRUM MASTER

The Scrum Master has many responsibilities. Often, the Scrum Master is one of the ordinary Development Team members, but this is not necessary and that arrangement can have drawbacks as well as advantages. The Scrum Master is responsible for maintaining a certain flow in the development process in accordance with the instructions of the Scrum framework. This means facilitating daily meetings with the Development Team and making sure that the team has a common understanding of the product vision, the overall goals and the development tasks in the on-going Sprint.

The Scrum Master also acts as a gatekeeper to the Development Team and as an information officer to the environment. At the same time the Scrum Master is to protect the Development Team members from all irrelevant assignments thrown upon them by their surroundings, thereby ensuring peace to work. Likewise he is to serve curious colleagues and spread the knowledge about Scrum to the surrounding organisation. Scepticism among colleagues outside the Scrum Team can be significant, so this part of the Scrum Master's role is not to be underestimated.

Just as the Scrum Master connects the Development Team to the surrounding organisation, he or she also functions as the link to the Product Owner in a similar way. It is the responsibility of the Scrum Master to facilitate the cooperation between the Development Team and the Product Owner about the Product Backlog and to secure that everyone fills his role in the process of selecting tasks for an upcoming Sprint.

MEETING #1: SPRINT PLANNING MEETING

Every Sprint begins with a Sprint Planning Meeting between the full Scrum Team, meaning the Product Owner, the Scrum Master and the Development Team. Together they plan the next sprint by deciding which tasks and activities that have to be taken into *the Sprint Backlog* – a document just like the Product Backlog, but with emphasis on the activities in a Sprint. The Sprint Planning Meeting is typically divided into two separate parts. A strategic part and a tactical part:

01. Strategic considerations: What will be delivered in the Increment resulting from the upcoming Sprint?

02. Tactical considerations: How to achieve the work needed to deliver the Increment?

An additional role:

The Surrounding Organisation

It is not a part of the official Scrum Guide, but the surrounding organisation has a rather important role in a successful implementation of Scrum. In most cases the implementation of Scrum is a strategic decision made by the management, and in fewer cases Scrum is an initiative running below the radar, started by a group of development engineers. Regardless of which way Scrum has entered an organisation, a successful implementation is dependent on a blessing – or at least acceptance – from the surrounding organisation and staff. The one single-most important task of the surrounding organisation is to accept that the Development Team is not able to take on any other tasks during the course of a Sprint. This is one of the most clearly described rules of Scrum.

In the strategic part of the meeting, the Scrum Team decides *what* has to be done in the upcoming Sprint based on a presentation from the Product Owner. The Development Team and the Product Owner subsequently negotiate which tasks to include in the Sprint backlog. The parts thoroughly discuss what each task entails.

The tactical part of the meeting is concerned with the question of *how* to fulfil the tasks. It is the Development Team alone who decides how to organise itself in order to get the job done within the next Sprint. However, it is important that each task is broken down into smaller work packages that can be concluded by one single developer in maximum one day. This is done to ensure a sufficient level of detail on what has been decided and what the Development Team has committed itself to solve before the Sprint deadline. Various tools that are not actual parts of the Scrum framework are often used to assist this process. One of the most popular ones is Planning Poker – a game for collectively estimating the extent of the respective tasks.

MEETING #2: DAILY SCRUM

If the Sprint cycle is the foundation of Scrum, then Daily Scrum is one of the supporting pillars. Daily Scrum is a 15-minute meeting held each day by the Development Team. The purpose of this meeting is to synchronise the work and adjust the chosen approach to reaching the Sprint goals. It is the Scrum Master's responsibility to facilitate the meeting and make sure that every Development Team member makes himself heard. In that respect each Development Team member has to answer the following three questions at the Daily Scrum meeting:

01. What has been accomplished since the last meeting?
02. What will be done before the next meeting?
03. What obstacles are in the way?

The Daily Scrum meeting is typically placed in a fixed timeslot and is thereby meant as a permanent and regular event occurring on daily basis.

MEETING #3: SPRINT REVIEW

The Sprint Review is an informal meeting that concludes the just completed Sprint. In contrast to both the Sprint Planning meeting and the Daily Scrum, the Sprint Review is open to a broader audience than just the Scrum Team. The aim is to inspect the work done in the Sprint in a constructive way, and to the Development Team members it is a chance to present what they have been working on for the last weeks. The discussions and considerations at the Sprint Review typically function as a basis for the Product Owner's input at the next Sprint Planning meeting for upcoming Sprint.

	1 SPRINT PLANNING	2 DAILY SCRUM	3 SPRINT REVIEW	4 RETROSPECTIVE
CHARACTERISTICS	Forward-looking	Corrective	Controlling	Retrospective
MAIN FOCUS	Breakdown and estimation	Burn-down of tasks	The Product	The Process
PURPOSE	Planning	Team communication	Showing work done	Improving the process
PARTICIPANTS	Scrum Team	Dev. Team & Scrum Master	Dev. Team & stakeholders	Dev. Team & Scrum Master
INPUT	Product Backlog Items	Development status	Work done in the Sprint	Process-related problems
OUTPUT	Sprint Backlog	Team synchronisation	Input to Product Backlog	Plan for process improvement

TABLE 2.2 OVERVIEW THE MEETINGS IN SCRUM
(Schwaber & Sutherland, 2011)

According to earlier versions of the official Scrum Guide, it has traditionally been of great importance that the results of each sprint could be characterised as a potentially shippable product increment, meaning an independently working part of the final product or a working prototype. The level of its refinement should be equal to or very close to the standard of the final product. In the latest Scrum Guide a greater emphasis has been put on the "Definition of Done". *Done* represents a shared understanding within the Scrum Team about exactly what qualifies a certain task as finished. This is probably because of the difficulties in always reaching a potentially shippable increment of the final product after each Sprint, and "definition of done" is a more pragmatic way of still being able to maintain strict deadlines and clear evaluation measures for the tasks.

MEETING #4: SPRINT RETROSPECTIVE

Just as the Sprint Review is an in inspection of the *product* of the development, so the Retrospective is an inspection of the *process*. The Sprint Retrospective follows immediately after the Sprint Review and has the purpose of identifying the part of the Scrum process that can be refined in order to improve the efficiency of the team. It is the Scrum Team members' opportunity to bring forward collaboration issues, process suggestions or issues related to technical equipment or work environment.

The result of the meeting is a plan for implementing the improvements that the Scrum Team has agreed upon. In that way, the Sprint Retrospective becomes a critical part of the continuous improvement strategy called "inspect and adapt", which is part of the Scrum framework.

ARTEFACT #1: PRODUCT BACKLOG

The Product Backlog is a list of "work to-be-done". It consists of all the "features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases" (Schwaber & Sutherland, 2011). As mentioned earlier, the Product Backlog is maintained by the Product Owner and is a dynamic document that evolves as the product development progresses.

The Product Backlog may change throughout the product development lifecycle and therefore also functions as a document gathering material for later versions or releases of the same product. Changing marked conditions, new business requirements or emerging technologies can potentially result in changes in the Product Backlog if they are relevant and have influence on the future value of the product in development.

The Product Backlog is listed in various ways. Depending on development strategy, the list can, for instance, be sorted by value, risk, priority, or necessity. It is common for all the ways of sorting the Product Backlog that top-ordered items are the most detailed ones. Items further down the list are typically more coarse-grained. This represents the distinction in their immediate importance to the Development Team. The Development Team and the Product Owner continuously "groom" the coarse-grained Product Backlog Items as the development progresses.

ARTEFACT #2: SPRINT BACKLOG

The Sprint Backlog is a tool for the Development Team for making visible the work that the team identifies as necessary in order to meet the Sprint Goal. Just as the Product Backlog, the Sprint Backlog is a dynamic document that develops throughout the Sprint. It has just enough detail for the Development Team to establish an overview of the remaining work and remaining time to do the work. In this way, the Sprint Backlog assists the Development Team in planning the ongoing Sprint and makes the work of the Development Team transparent to the Product Owner.

ARTEFACT #3: PRODUCT INCREMENT

As mentioned earlier, the Product Increment is the resulting work of a Sprint. It thereby represents a number of Product Backlog items that have been chosen for development in a Sprint and hereafter removed from the Product Backlog. Scrum underlines that the increment must be in a useable condition after it has been developed during a Sprint. It is the Scrum Team's definition of "Done" that decides to what extent this means *shippable*. It is up to the Product Owner to decide whether or not the Product Increment is to be used "as is," as a part of the final product, changed in a later sprint, or not used at all.

ARTEFACT #4: BURNDOWN CHARTS

The Burndown Chart in general represents the work left versus time left and is often used in relation to the Product Backlog or the Sprint Backlog. Scrum Teams often choose to keep one for both of them.

ARTEFACT #5: SCRUM BOARD

The Scrum Board can have many different forms and is not an official part of the Scrum framework. However, the Scrum Board helps the Development Team to fulfil the overall Scrum goal of maintaining a clear and visual communication between the team members. By using post-it notes and large writing on a whiteboard or the like, the Scrum Board assists the Sprint Backlog as an easily accessible, and to everyone visible, graphic representation of the tasks to be done, tasks in progress, and tasks already done in a Sprint. In some versions, the Scrum Board also includes the Burndown Chart as well as an overview of the available hourly resources.

2.2.3 THE STORY ABOUT SCRUM

After a presentation of Scrum and its elements in the previous chapter a general understanding of this framework for managing product development should be established. The purpose of this sub-chapter is to give a supplementary and short historical introduction to the Scrum framework and thereby relate it to its historical roots in both software development and product development.

Ken Schwaber and Jeff Sutherland are together considered the founding fathers of the Scrum framework. They represented the idea of Scrum in 2001 when 17 software pioneers collectively authored The Agile Manifesto for Software Development (Highsmith, 2001), and they founded the Scrum Alliance as a non-profit organisation and knowledge resource in 2004. Later they started to periodically publish revised editions of The Scrum Guide. However, it is neither Schwaber nor Sutherland who was the first to use the term Scrum in relation to product development activities. In 1986 – fifteen years before the first appearance of the Agile Manifesto – Takeuchi and Nonaka in Harvard Business Review presented a development process, based on a series of studies in large Japanese companies within the automotive- and information technology industries (Takeuchi & Nonaka, 1986). They compared this development process with the *scrummage event* in Rugby, where all players on a team gather to plan the next few seconds after the ball comes into play. The process, which Takeuchi and Nonaka described, made up with the Waterfall model (Royce, 1970) that had been the predominant development standard in more than two decades within both software and systems development but also in physical product development. They described the method as a holistic method with some characteristic elements such as self-organising project teams, less top-down management, focus on learning and other aspects.

At the same time, the critique of the Waterfall model began to appear from various sides as a result of a large number of failed and expensive development projects. Barry W. Boehm – a leading software engineer within the field of systems development – at the same time as Takeuchi and Nonaka proposed an iterative software development model called the Spiral Model (Boehm, 1988). In a short monologue in one of his famous articles he expressed perhaps the most clear-cut critique of the Waterfall model: "The waterfall model is dead." "No, it isn't, but it should be" (Boehm, 1988).

The critique of the Waterfall model was primarily centred on its insufficient handling of the increasing complexity that made one project fail after another in the seventies and eighties. A couple of the software pioneers who later became co-authors of the Agile Manifesto claim as followings:

"Traditional approaches assumed that if we just tried hard enough, we could anticipate the complete set of requirements early and reduce cost by eliminating change. Today, eliminating change early means being unresponsive to business conditions—in other words, business failure. (…) This approach assumes that variations are the result of errors. Today, while process problems certainly cause some errors, external environmental changes cause critical variations. Because we cannot eliminate these changes, driving down the cost of responding to them is the only viable strategy. Rather than eliminating rework, the new strategy is to reduce its cost."
Highsmith & Cockburn, 2001

Truex et al. take an extreme stand and state that methods like the Waterfall Model are "merely unattainable ideals and hypothetical "straw men" that provide normative guidance to utopian development situations" (Truex et al., 2000). Larman (2007) equally argues that Software Development is equivalent to New Product Development in respect to unpredictability and therefore does not fit the logics and rigidity of stage-wise development models.

In the beginning of the nineties Jeff Sutherland and Ken Schwaber worked and experimented with various ways of increasing efficiency in the respective software development departments of two different companies. Independently, they started getting experience with some of the initiatives that both Boehm and Takeuchi & Nonaka and others had promoted years before. At a conference in 1995 the two of them collaboratively presented their shared experiences as what they called the Scrum Method. During the following years they worked together on refining Scrum.

The two following chapters, 2.2.4 and 2.2.5, are largely based on the peer-reviewed paper *Agile Attitude: Review of Agile Methods for use in Design Education* published in the Proceedings from the 13th International Conference on Engineering and Product Design Education by Design Society in 2011.

2.2.4 RISE OF AGILE SOFTWARE DEVELOPMENT

Agile is an attribute often associated with animals like the big cats. Alert and responsive – quick and well coordinated in movement. Nevertheless, these are also the features that can be associated with agile methods in a process management perspective. The Scrum framework is considered one of the most influential and popular agile methods, so in the following sub chapter an overview of Agile Development and its principles are made in order to position Scrum in relation to the main development trends of the last ten years.

Agile Development, as a term, was coined early in 2001 during a two-day meeting between seventeen people gathering at Snowbird Ski Resort in the Wasatch Mountains of Utah (Highsmith, 2001). The people gathered here were representatives from various surfacing disciplines in software development trying to establish a common ground and explicate a united stance in the worldwide software development community. Ken Schwaber and Jeff Sutherland, the founding fathers of the Scrum framework were two of the people gathered. The outcome of the summit in this extraordinary place was The Manifesto for Agile Software Development, which, after it's authoring, has had a vastly influential role in the software development industry throughout the following ten years. In a sense Agile Development is a response to the traditional – and ultimately failing – software development methods that have been dominating the 20th century. Table 2.3 on the following page shows the highly influential value set of The Agile Manifesto from 2001. The four statements clearly make up with the command-and-control development processes in traditional development.

The "invention" of the manifesto can be seen as the latest culmination of a decade-long discussion about approaches to software development. As far back as in the 1960s, programming techniques were beginning to lag behind the developments in software with respect to both size and complexity. Software projects were increasing dramatically in man-years, which started to challenge the organisational perspectives. An approach of deploying more programmers into the projects became known as the "Million-monkey approach,"

TABLE 2.3: THE VALUE SET OF AGILE DEVELOPMENT

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.

(Highsmith, 2001)

TABLE 2.4: 12 PRINCIPLES BEHIND THE AGILE MANIFESTO
(Highsmith, 2001)

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

but without better structure on projects, the only results were failing and too late delivered software. These growth challenges became known as "The Software Crisis" (Vliet, 2007).

According to the authors, the Agile Manifesto is based on 12 principles about useful software development practices, which are presented in table 2.4 to the left. Because of their attention on actual development practices, these principles may give an even clearer picture of what Agile Development is about. Comparing the principles to the values and practices of Scrum, it is obvious, that Scrum fits very well into this definition of Agile Development.

2.2.5 SCRUM IN RELATION TO THE THREE FUNDAMENTAL CHALLENGES

The project in hand promotes Agile Development, and more specifically the Scrum framework, as a possible way of handling the three fundamental challenges of product development set in the introductory chapter. Now that Scrum has been briefly presented with both its various elements, historical origins and relation to Agile Development, this sub chapter establishes the arguments for promoting Scrum as a potential solution to the fundamental challenges based on the principles of Agile Development and the description of Scrum.

To order to recapitulate: the three fundamental challenges in traditional product development presented in the introductory chapter were *accelerating speed*, *change-ability*, *increasing product complexity*.

SCRUM AS AN ANSWER TO SPEED

One of the primary characteristics of Scrum is the steady rhythm of its continuing cyclic sprints. Just like in Lean Manufacturing, the strict tact and frequent inspection creates the necessary sense of urgency needed for getting the job done. Sutherland (2007) directly states that Scrum was design to increase speed and that "Scrum is an agile software development process designed to add energy, focus, clarity, and transparency to project teams developing software systems." In this respect, Scrum has a series of aspects, that stress development efficiency in order to meet a need for increased market speed:

- ***Timeboxing*** ensures continuous and steady progress and avoidance of "gold plating" the product in development by over-performing.
- ***Time and resource estimation*** on every small task puts emphasis on getting the job done and clearly identifies small manageable tasks.
- ***Focus on one product at a time*** enables the team to maintain a high level of concentration and commitment and a minimal level of disturbance.

SCRUM AS AN ANSWER TO CHANGE

Embracing change is one of the virtues of an agile attitude, and the Agile Manifesto even promotes the phrasing "harness change for the customer's competitive advantage" as one of its 12 principles. Scrum emphasises an *inspect-and-adapt strategy* by formalising continuous revision of the product backlog as presented in chapter 2.2.2. Scrum assists development teams in handling change through following aspects:

- ***Dynamic Product Backlog*** means no heavy up-front planning and no "sticking to the plan"
- ***Formalised meetings*** ensure continuous execution of inspect-and-adapt activities
- ***Incremental deliveries*** in small and independent work packages entail greater product flexibility

SCRUM AS AN ANSWER TO COMPLEXITY

"Complex problems are those that behave unpredictably. Not only are these problems unpredictable, but even the ways in which they will prove unpredictable are impossible to predict" (Schwaber, 2004). According to Schwaber (2004), Scrum is based on *empirical process control*, which is suitable, when projects are too complicated or complex

to be handled by a *defined process control* (Schwaber 2004). Three elements are needed to uphold an implementation of empirical process control. These are *visibility, inspection* and *adaption*. Beside the three elements, Scrum also puts emphasis on not over-shooting the development efforts compared to product requirements.

- ***Visibility*** through frequent and close team communication and physically visible backlogs.
- ***Frequent Inspection*** during formalised daily meetings.
- ***Formalised adaption*** to new conditions on regular basis in regards to both product and process.

As shown above, Scrum arguably handles the three fundamental challenges – at least theoretically. For the sake of establishing an argumentation to promote Scrum as a potential solution to the fundamental challenges, this is sufficient. It could be said, then, that it is the purpose of this thesis to *empirically test* Scrum in relation to the fundamental challenges within physical product development.

2.2.6 MANAGEMENT CHALLENGES IN IMPLEMENTING AGILE IN SOFTWARE DEVELOPMENT

The overall purpose of chapter 2.2 is to explain and present all the aspects of Scrum that are of interest to this research project. In that sense, the chapter still operates within the field of software development, as it is the original domain of Schwaber and Sutherland's Scrum framework. The present sub-chapter stays within that domain in an attempt to establish an overview of all the known challenges that must be faced when migrating from traditional development to Agile development *within* the software domain. The chapter presents the studies of Boehm and Turner (2005) about the management challenges of implementing agile processes in traditional development organisations. It is worth underlining that these challenges are identified in a management perspective, which is not the actual scope of this thesis. However, they do concern some related issues of integrated product development and development organisation, which are sought to be clarified in the present project.

The studies of Boehm and Turner (2005) are based on *barriers to agile processes* identified in a series of annual workshops. From these workshops, the challenges have been sorted into three categories: *development process conflicts*, *business conflicts* and finally *people conflicts*, which are presented in table 2.5 to the right. The studies relate to agile methods in general, but Scrum is exemplified as a representative method in this work of Boehm & Turner (2005).

TABLE 2.5: BARRIERS TO AGILE PROCESSES
Software development in large organisations
(Boehm & Turner, 2005)

DEVELOPMENT CONFLICTS

VARIABILITY
Managing variability in subsystems and teams has proven difficult. If both agile and traditional teams are working together on project, they risk obstructing each other's work without some means of coordination. Large organisations must pay attention to synchronising teams.

DIFFERENT LIFE CYCLES
Working with different life cycles is also difficult. Agile processes focus on immediately delivering functionality, while traditional methods focus on optimising development over a longer period. The traditional longer life cycles require adjustments to the agile processes.

LEGACY SYSTEMS
Applying agile processes to legacy systems, whether within maintenance or as new development, raises numerous issues.

REQUIREMENTS
Differences between how agile and traditional approaches perform the requirements process can also cause problems. Agile requirements tend to be primarily functional and reasonably informal.

BUSINESS CONFLICTS

HUMAN RESOURCES
Organisations must learn to accommodate human-resource issues such as timekeeping, position descriptions, team-oriented versus individual rewards, and required skills. Agile development team members often cross the boundaries between standard development position descriptions and might require significantly more skills and experience to adequately perform.

PROGRESS MEASUREMENT
Traditional contracts, milestones, and progress measurement techniques might be inadequate to support agile processes' rapid pace.

PROCESS STANDARD RATINGS
One area of conflict for mature organisations will be in how agile processes will affect their ratings with respect to CMMI, ISO, or other process standards.

PEOPLE CONFLICTS

MANAGEMENT ATTITUDES
Migrating from traditional to agile management attitudes can be difficult. Large-scale management processes such as earned value and statistical process control evolved from a manufacturing paradigm and tend to cast employees as interchangeable parts.

LOGISTICAL ISSUES
Some logistical issues directly affect people in agile environments. Agile teams must nearly always be colocated. The typical agile workspace requires pair-programming stations, walls for status charts and assignments, a layout that allows team members to easily converse to share information, and sufficient equipment to support continuous integration and regression testing.

HANDLING SUCCESSFUL PILOTS
The negative impacts of how organisations handle the success of pilot projects are often overlooked in reporting outcomes. "Fire or promote the manager and/or split up the team." This destroys team relationships, it dilutes the knowledge gained and lessons learned, and it sends the message that trying new things might be hazardous to your career.

CHANGE MANAGEMENT
Change management experts often describe the organisational antibodies that begin to gather as soon as something new appears in the existing culture. Concerns of inadequacy or obsolescence surface, jealousy about assignments and business accoutrements is aroused, and defense mechanisms rapidly deploy. This can result in several destructive behaviors.

2.3 INTEGRATED PRODUCT DEVELOPMENT PROCESS

This chapter establishes an overview of the present knowledge and insights about integrated product development (IPD) in relation to Scrum. It is the first of two bodies of relevant knowledge within the theoretical framework. The purpose of this chapter is to build a basis for understanding the empirical data in relation to the IPD process and to identify a set of theoretically based challenges when working with Scrum in this context.

2.3.1 WHAT IS AN INTEGRATED PRODUCT DEVELOPMENT PROCESS?

Asking the question *"What is integrated product development process?"* naturally leads to a series of other questions. One of them – and perhaps the most urgent one – is *"What is a process?"* Ulrich & Eppinger (2003) describe the most basic process from a systemic point of view as "a sequence of steps that transforms a set of inputs into a set of outputs." The figure 2.4 below shows this in its simplest form.

A *product development process* can therefore be described as a transformation of a set activities employed by an organisation in order to generate an output in the form of products, usually with commercialisation and profit as the main objective.

2.4 **THE SIMPLEST FORM OF A PROCESS**
Input - transformation - output
(based on Ulrich & Eppinger, 2003)

The next question surfacing would be *"what does 'integrated' mean?"* According to an encyclopedic look-up, the verb *to integrate* means that someone or something is included in something else and thereby fusing into a whole. The term has been known in relation to product development for the last thirty years, describing a holistic approach to development in contrast to a purely sequential process (Andreasen & Hein, 2000).

According to Andreasen & Hein (2000), integrated product development (IPD) is an idealised model for product development which is integrated in terms of 1) *creation of market*, 2) *product* and 3) *production*. This means that product development activities should be integrated with the other basic activities in the organisation. With *business* as their common objective, the three elements are more specifically responsible for 1) recognition and creation of a market and establishment of sales outlets, 2) creation of a product that satisfies this market and is produced by the third element, 3) the production system (Andreasen & Hein, 2000). Figure 2.5 to the right illustrates these simultaneously developing elements of a business in contrast to a sequential-development model.

Proposed by Barry (1993), another, but similar, definition of Integrated Product Development emphasises the integrating aspects *within* the development environment:

> *"Integrated Product Development is a philosophy that systematically employs a teaming of functional disciplines to integrate and concurrently apply all necessary processes to produce an effective and efficient product that satisfies customer's needs. Product, in this sense, is not only what is delivered to your customer (e.g., hardware, software, services, and documents), but also processes (e.g., design, manufacturing, test, and logistics) which make the product possible."*
> *Barry, 1993*

The two definitions respectively describe Integrated Product Development as a process in which activities takes place in a parallel and coordinated manner. While Andreasen & Hein (2000) focus on

a close interplay between traditionally separate company divisions, Barry (1993) seems to emphasise close integration between functional disciplines within the actual development environment.

PRACTICES SIMILAR TO IPD

Even though Integrated Product Development is a well-established concept – and has been so for several decades – a multiplicity of practices and lines of thought have evolved simultaneously. According to Naveh (2005) IPD is implemented in 64% (Gerwin and Barrowman, 2002) of all New Product Development projects in various forms such as Concurrent Engineering, Design for Manufacturing, Early Manufacturing Involvement and Time-based Competition. All these initiatives have integrative characteristics and emphasise elements such as cross-functional teams, close coordination, and early involvement of all stakeholders associated with the new product (Adler, 1995; Koufteros et al., 2002).

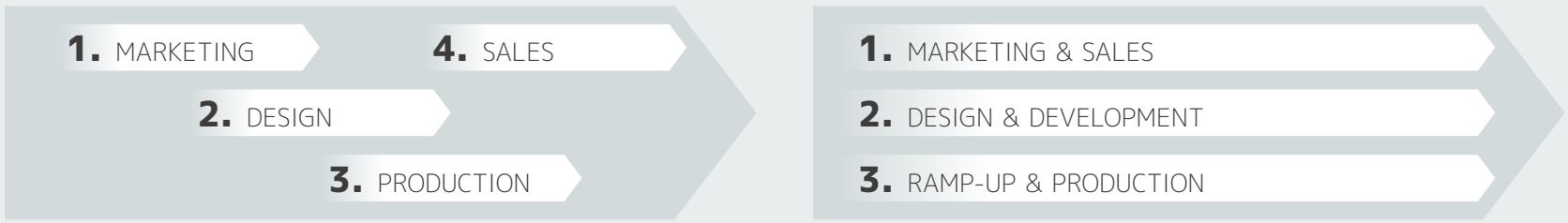
2.3.2 MACRO-LEVEL: TWO TYPES OF PROCESS CONTROL MODELS

As the definitions in the previous chapter state, the IPD process promotes a close interplay between both different functional disciplines and different company divisions. The chapter also stresses the fact that over 60% of NPD Projects utilise IPD in various forms. The objective of this chapter is to present the most common models on the macro level for *controlling* the integrated development process.

According to Ogunnaike & Ray (1994), two approaches to process control exist: The defined process control model and the empirical process control model. The basic differences between their models, as well as their application areas, are outlined in the following quotation:

> *"It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood. When the process is too complicated for the defined approach, the empirical approach is the appropriate choice."*
>
> *Ogunnaike & Ray, 1994*

The Scrum framework falls into the category of empirically based process control models, as it is characterised by its focus on visibility, inspection and adaption in complicated or even complex development projects. This means that Scrum not necessarily ensures that a project will progress exactly as expected, nor bring the results exactly as expected. On the other hand, it is argued by Schwaber (2004) that Scrum controls the development process to guide work towards the most valuable outcome possible. The frequent inspection of the de-



UNFORTUNATE SEQUENTIAL PROCESS
Poor interplay between the divisions.

INTEGRATED DEVELOPMENT PROCESS
Strong interplay between divisions conducting activities in parallel.

2.5 Integrated- vs. sequential development model (Andreasen & Hein, 2000)

velopment process allows the team to identify potential problems and react accordingly based on their continuous observations.

The author of the book "The Principles of Product Development Flow," Don Reinertsen in his answer to the blogger, Alan Shalloway, describes three distinct dimensions that define the characteristics of process control models. Despite their origin in an informal discussion on the web, the three dimensions listed below seem to form a useful description that gives some nuance to the otherwise relatively simple distinction between defined and empirical processes.

01. Degree of process definition

02. Randomness of its output

03. Amount of feedback that the process uses

(Reinertsen, 2009)

Evaluating Scrum against these three dimensions, it can be argued that

A. the development process facilitated by the Scrum framework is to some extent defined by its strict and continuous rhythm, and

B. it is possible to define its output in clearly separated end-to-end work packages, and

C. it utilises a great amount of feedback.

If Scrum with its adaptability and flexibility is a representative of the empirical approach, then what characterises a defined process control model? As it is stated in the quote from Ogunnaike & Ray (1994), the defined process is preferred when the underlying mechanics by which a process operates are reasonably well understood or known. This gains resonance with certain types of well-established and plan-driven methods that represent the traditional requirement/design/build paradigm known from the mainline engineering field (Boehm & Turner, 2004). In opposition to agile principles such as *welcoming change even late in development*, the main success criterion in plan-driven models is *following a plan*. As Smith (2007) argues, these models promote a plan-your-work, work-your-plan approach.

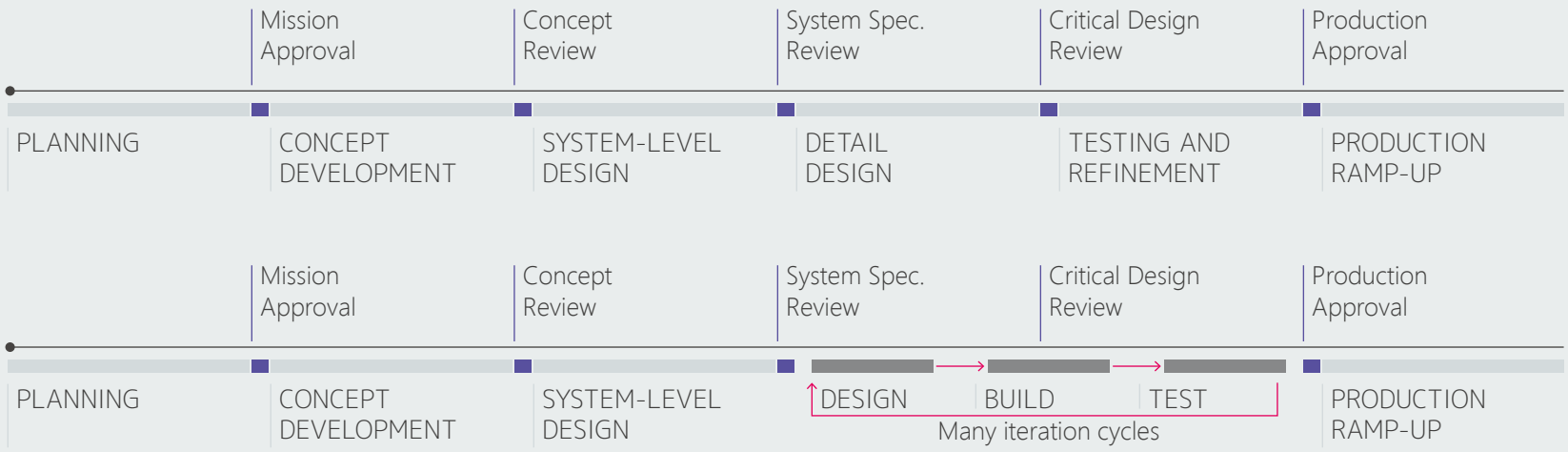
Perhaps the most widely known example of such a development process is the Stage-Gate model developed by Robert G. Cooper. In his book, "Winning at New Products" first published in 1988, Cooper describes an Idea-to-Launch system based on a phased development model. More specifically, the Stage-Gate model is defined by Cooper as:

> *"(...) a conceptual and operational map for moving new-product projects from idea to launch and beyond – a blueprint for managing the new-product development process to improve effectiveness and efficiency. Stage-Gate is a system or process not unlike a playbook for a North American football team: It maps out what needs to be done, play by play, huddle by huddle – as well as how to do it – in order to win the game."*
>
> *Cooper, 2011*

In contrast to the empirical approach, this definition stresses the parallel to a playbook or a map that can assist the organisation to win the new product development game. The development process is, so to speak, predictable and therefore possible to define rather specifically. The Stage-Gate model is illustrated in Figure 2.6 to the right and is chosen as on of the main reference points to evaluate Scrum against due to its vast popularity in NPD companies. According to Griffin (1997), 69% of the best NPD firms use Multi-functional stage gate approaches.

Many versions of the stage-gate models exist (Phillips et al., 1999) and as it can be seen by the empirical data of chapter 4, all seven case companies deploy various forms of this type of process control models. Ulrich & Eppinger (2003) describe a similarly phased development model with six phases from Planning to Production Ramp-Up. Furthermore they describe a slightly different version called the "Spiral Product Development Process." According to Ulrich & Eppinger

2.6 **THE IDEA-TO-LAUNCH STAGE-GATE PROCES MODEL**
(Cooper, 2011:104)



2.7 **GENERIC AND SPIRAL PRODUCT DEVELOPMENT PROCESS MODELS**
(Ulrich & Eppinger, 2003)

(2003), this spiral process takes advantage of the fast prototyping cycles commonly used in development of electronics products. Recalling the three dimensions from the discussion between Shalloway and Reinertsen, the process thereby becomes more flexible and responsive by actively formalising a feedback loop in the process. The Spiral Product Development Process, which actually has a great resemblance to the *original Waterfall model* presented in 1970 by Royce (1970), as well as the *The Spiral Model of Software Development and Enhancement* later published by Boehm (1988), is presented in figure 2.7 above.

In general, these phased models, which are broadly deployed in the product development industry, seem to contrast to the Scrum framework in many aspects. Some of these aspects are unfolded in the subsequent chapters.

2.3.3 PROCESS INPUT AND OUTPUT

Recalling the basic concept of a process, it features a set of inputs and a set of outputs. The purpose of this chapter is to describe the characteristics of the inputs and outputs of the two contrasting process control models presented earlier. Here the two models are represented by the Scrum framework and the Stage-Gate model.

As it has been presented earlier in the review of Scrum, the input of the development process is a set of dynamic product features, conceptually described in a Product Backlog (Schwaber & Sutherland, 2011). The Product Backlog is allowed to change over time as the project matures, and the Product Backlog items that are not about to be developed in the upcoming Sprint are coarse-grained.

In the Stage-Gate model presented by Cooper, the development phase, which is taking place in Stage 3, is initiated after a significant lock-down of opportunities in the preceding Gate 3. At Gate 3, "the funnel is led into a tunnel," the product definition is committed to by the management, a project plan for the rest of the process is reviewed and approved, and the project team and funds are formally committed (Cooper, 2011).

As it has already been indicated earlier, the output of a development process controlled by the Scrum framework is not necessarily the one expected. Instead, it is continuously manoeuvred in the direction that gives the most value to the customer (Schwaber, 2004). The output is ideally released to the customer as working functionality after each Sprint and as early in the process as possible.

As it is presented in this chapter, the inputs and outputs are significantly different in the two extremes of process control models. This indicates that certain issues could be expected when implementing Scrum in traditional stage-gate development environments.

Similarly, the Stage-Gate process contains an output after each Gate. The output, which is determined by the criteria set in the specific gate, can be *Go*, *Kill*, *Hold* or *Recycle*. However, the upper management decides the outputs of each gate meeting, and a consequently the final product is first launched to market when it has been successfully approved with *Go* throughout all gates (Cooper, 2011). In contrast to the output of the empirically based Scrum framework, the success of a Stage-Gate project is a close coherence between the project definition approved in Gate 3 and the resulting product, which is presented at Gate 5 prior to the Launch stage.

2.3.4 BASIC ELEMENTS OF THE STAGE-GATE DEVELOPMENT PROCESS

A well-defined development process can be useful for several reasons. The list of reasons below is inspired by Ulrich & Eppinger (2003):

- Quality assurance
- Coordination
- Planning
- Improvement

In this chapter, the generic development process represented by Stage-Gate is reviewed and unfolded according to the listed elements above.

QUALITY ASSURANCE

Essentially, the gate meetings in the Stage-Gate process act as the quality control checkpoints, where two fundamental questions are asked (Cooper, 2011):

01. Are you doing this project right?
02. And are you doing the right project?

As Blessing (1993) argues, the quality of the product strongly depends of the quality of the process. The Stage-Gate process provides a clearly

defined process, which according to Phillips et al. (1999) is an important part of this. Furthermore, Quality Management System standards such as the ISO 9000 family of standards fit well with the rigid and plan-driven process models, such as Stage-Gate (McMichael & Lombardi, 2007).

Scrum, as a contrast, has been criticised for its lack of critical design reviews like the gates in the Stage-Gate process (Boehm & Turner, 2005). However, as described earlier, Scrum promotes a significantly shorter development cycle, which ends with a Sprint Review and a Retrospective meeting. These meetings, even though they are not necessarily including the management, seem to act as a sort of much more frequent review.

COORDINATION

A clearly articulated development process acts as a master plan, which informs the team members when their contribution will be needed and with whom they will need to exchange information and materials (Ulrich & Eppinger, 2003). In the last of the seven cases presented in chapter 4, the Value Stream Mapping method (Rother & Shook, 2003) was used to coordinate the interdependencies between sub development teams, acting as a practical coordination level between the upper level Stage-Gate process and Scrum practice on daily basis.

Beside informal communication and formal meetings, one of the most important coordination mechanisms in typical Stage-Gate projects is the project schedule, usually in the form of Gantt charts or the like (Ulrich & Eppinger, 2003).

The Scrum framework promotes close coordination through frequent meetings. The Daily Scrum meeting coordinates the present development tasks in the development team, and Sprint Planning meetings as well as Sprint reviews ensure the coordination to the Product Owner, and eventually the customer (Schwaber & Sutherland, 2011).

PLANNING

As it has been mentioned earlier, most phased processes like Stage-Gate encourage heavy up-front planning followed by sticking to the plan (Smith, 2007). The planning activities take place in Stage 2 prior to the actual development phase and after the Discovery phase and the scoping efforts in Stage 1. The emphasis on up-front planning is clearly illustrated by the quote below:

> *"Securing sharp, early, stable and fact-based product definition before Development begins is one of the strongest drivers of cycle-time reduction and new-product success."*
> *Cooper, 2011*

Focus is on stable and early product definitions. The output of the planning phase is a clear definition of the product, which include the following key elements:

- Target market
- Product concept
- Positioning and benefits to be delivered
- Value proposition
- Attributes, features, requirements
- High-level specs

(Cooper, 2011)

Even though the list may seem relatively close to what could be expected in an agile project plan, there are some radical differences in how these documents are perceived:

> *"The Product Backlog is never complete, and the Product Backlog used in the project plan is merely an initial estimate of the requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves."*
> *Schwaber, 2004*

IMPROVEMENT

According to Ulrich & Eppinger (2003) a careful documentation of a development process often helps to identify places for improvement. The Stage-Gate model is often said to be heavy on documentation (McMichael & Lombardi, 2007; Smith, 2007) and as such it should have a predisposition to improvement. However, formalised learning from past or on-going projects does not seem to be implemented in most development projects. A survey shows that 80% of research and development projects are not reviewed after completions and the remaining 20% lack formal review guidelines. Furthermore, most projects that are prematurely terminated never undergo a retrospective analysis on their cause of failure (Zedtwitz, 2002). Even though no formal review or learning process is part of the typical Stage-Gate, Ulrich & Eppinger (2003) describes a set of corrective actions that can be taken, if undesirable deviations from the project plan are happening. These actions include:

- Changing the timing or frequency of meetings
- Changing the project staff
- Locating the team together physically
- Soliciting more time and effort from the team
- Focusing more effort on the critical tasks
- Engaging outside resources
- Changing the project scope or schedule

(Ulrich & Eppinger, 2003)

Once again, Scrum seems to contrast the formal Stage-Gate process as it includes frequent process inspection in the Retrospective meeting at the end of each Sprint. However, as it has been mentioned earlier, a great number of variations of the Stage-Gate process exist and there is no formal hindrance that prevents development teams from deploying inspection and improvement routines.

When comparing Stage-Gate to the Scrum framework, it seems to lack a clearly defined retrospective analysis.

2.3.5 MICRO-LEVEL: THE DESIGN PROCESS

Some of the most important elements of the integrated product development process and the prevailing process control models have been unfolded in the preceding chapters. The purpose of this chapter is to zoom in to the *micro-level* and more specifically focus on the *design process* of the product designer in order to clarify the resemblance to certain aspects of Scrum.

DEFINING THE PRODUCT DESIGNER

Even though this project focuses broadly on implementing Scrum in integrated product development environments, it has a special interest in *some* of the many disciplines involved. The group earning this special interest includes *mechanical developers* and *industrial designers*. While the mechanical developers are typically integrated as a part of the cross-functional Development Team, the industrial designers are often loosely attached to the Development Teams (Ulrich & Eppinger, 2003). The special interest in these groups is motivated by the fact that their competences are related to the far end of the span of integrated product development disciplines opposite software developers. While software development – the origin of Agile Development and Scrum – can be said to concentrate on the non-physical elements of the products, the mechanical developers focus on the physical elements of products. According to Ulrich & Eppinger (2003) the industrial designer generally focuses on aspects such as *utility*, *appearance*, *ease of maintenance*, *low costs,* and *communication (communicating the corporate design philosophy through visual qualities of the product)*. It may be argued that mechanical designers and industrial designers have a certain overlap in competences, which is why, to some extent, they are considered a homogeneous group and, in this research project, collectively denoted as *Product Designers*.

THE GUIDELINE APPROACH

- **Goal:** Externalised rationality
- **Principles:** Possible to formulate them as generic design principles
- **Focus:** The Process
- **The good designer:** The ability to follow prescribed actions.
- **Extreme:** The designer becomes merely an operator
- **Measures:** The design becomes a map, which can always be compared and tested against reality

THE AESTHETIC APPROACH

- **Goal:** hidden rationality
- **Principles:** The designer react according to the specific situation
- **Focus:** The Product
- **The good designer:** Guided by aesthetics and understanding of quality
- **Extreme:** The designer becomes an arbiter of taste
- **Measures:** no right or wrong solution, only good or bad designs. No guarantee of a good product

TABLE 2.6: GUIDELINE APPROACH AND AESTHETIC APPROACH
The dichotomy in approaches to design process after Stolterman (1994)

THE DESIGN PROCESS

While it may be difficult to conclusively define the exact design process of Product designers, Stolterman (1994) suggests a dichotomy between two fundamentally different approaches to the design process. He divides the design processes into two separate approaches: the *guideline approach* and the *aesthetic approach*. Even though this segregation is made in relation to design teaching, it may very well describe the span in design practice, too. The characteristics of the two approaches are presented in table 2.6 above.

According to Stolterman the guideline approach seeks to externalise the rationality in the design process by formulating generic principles that ensure transparency and a satisfactory end result. In the most extreme case the design process is considered independent of the abilities of the designer except for the ability of following instructions.

In contrast to the guideline approach, the aesthetic approach is solely dependent on the designer's ability to judge quality. The designer is guided by his or her own hidden rationality and aesthetic beliefs, which cannot be externalised or formulated into generic principles. This also implies that no guarantees can be given of a good (Stolterman, 1994).

The differences between the two approaches may very well illustrate an existing split in the design community, but this categorising is also helpful in bringing further clarity as to what types of design activities are of interest in this research project. The integrated product development environments that are in focus in this project's empirical foundation are tending towards an explicit and rational approach to design tasks, which is also often considered to be compatible with the relatively dominant engineering environment.

ITERATIONS AND RESEMBLANCE TO SCRUM

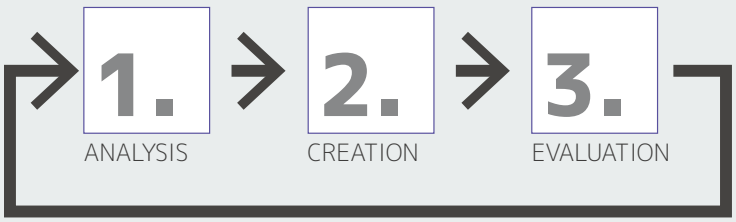
A few general principles in the field of design are commonly accepted. These are respectively the iteration-based process cycle and the concepts of divergent and convergent phases. As described by Stanton (1998), the basic concept of an iterative design process is as a cyclic process between *analysis*, *creation* and *evaluation*, which is depicted in figure 2.8 on the next page.

The model describes how the process of designing undergoes an evaluation of an outcome followed by a subsequent phase of analysing the status that, again, leads to new actions in order to reach the design objectives. This model can be further elaborated by relating it to the concepts of divergent and convergent thinking (Chamorro-Premuzic & Reichenbacher, 2008; Striim, 2001). While the divergent thinking process focuses on creating solutions, the convergent thinking process evaluates and synthesises the established solutions, consequently reaching a higher level of realisation.

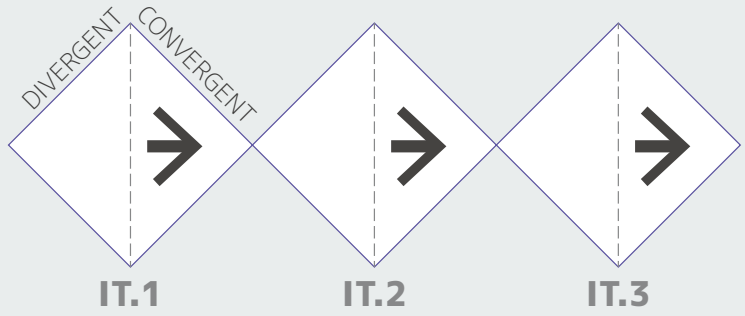
Stanton's model of the cyclic design process in many ways resembles the model for reflection put forward by Argyris & Schön (1978), which describes the process *reflection in action*. However, an additional level of reflection is presented by Argyris & Schön, which is called *reflection on action*. While reflection *in* action may resemble the design process presented by Stanton, *on-action reflection* questions the perception of the problem itself and the methods used for addressing that problem.

Zooming out to the macro-level, these iterative design process models seem familiar to the Scrum framework model itself. Whereas the Sprint, the Daily Scrum meetings and the Sprint Review are equivalents to the *in-action reflections* in the model of Argyris & Schön, the Retrospective in Scrum resembles the *on-action reflection* in its way of evaluating the approach of Development Team.

On basis of the close resemblance between the two models, it can be argued that Product Designers would consider the iterative process like that of Scrum to be natural to their normal creative practice. However, a counterargument could be that the processes are taking place on two different levels and with different degrees of awareness. Furthermore, some researchers argue against the well-defined and iterative model as the right representation of the design process. Bryan Lawson (2005) describes the design process *"as a negotiation between problem and solution through the three activities of analysis, synthesis and evaluation."* To some extent these activities are similar to the ones in Stanton's model. The main difference is found in the relationship between these activities. Lawson criticises the rigid and cyclic models by arguing that the design process *"rather resembles one of those chaotic party games where the players dash from one room of the house to another simply in order to discover where they must go next"* (Lawson, 2005)



2.8 **THE ITERATIVE DESIGN PROCESS**
Including the phases of analysis, creation and evaluation (Stanton, 1998)



2.9 **DIVERGENT AND CONVERGENT PHASES**
Two ways of thinking (after Striim, 2001)

Perhaps Lawson describes the chaotic process of designing more honestly, whereas other models attempt to establish a certain order in the construction of the process. Anyhow, this project merely attempts to establish a certain overview of different lines of thought in this regard, and some readings of the design process seem closely related to the model of the Scrum framework.

2.3.6 NEW MOVEMENTS IN INTEGRATED PRODUCT DEVELOPMENT

Before shifting the focus to the second body of knowledge, which describes the development organisation, the first body of knowledge is concluded by presenting three recent trends that relate Agile Development and Scrum to integrated product development process. These three trends, which are *Agile in Stage-Gate*, *Lean Product Development* and *Flexible Product Development*, are briefly presented in the sub chapters below.

AGILE IN STAGE-GATE

The Stage-Gate model has been known and widely used for more than two decades since the first edition of Cooper's "Winning at New Products" was published in the 1980'ies. As Cooper writes in the fourth and latest edition of this book, the phased model has become an evergreen that have evolved with the industry. This is also why the 2011-version addresses the last 10 years' industry trends by including elements of adaptable, flexible and agile processes. The quote below illustrates the softening of the otherwise rather predefined and strict practice, which has been the well-known trademark of Stage-Gate.

> *"The notion of a rigid, lockstep process is dead. Today's fast-paced Stage-Gate is flexible, as opposed to a rigid 'book of rules and procedures' to be religiously followed. No activity or deliverable is mandatory. Stage-Gate is a guide that suggests best practices, recommended activities, and likely deliverables."*
> *(Cooper, 2011)*

The Scrum framework is expected to match the process of the product designer due to its immediate resemblance with the basic design process.

Similar to the earlier mentioned spiral product development process, Cooper suggests several feedback loops from Stage 2 through to Stage 4 and thereby allowing changes to the project definition relatively late in the process. However, he also notes that it should be clearly described in the project definition which parts are fluid and which parts are fixed.

LEAN PRODUCT DEVELOPMENT

Lean product development is a term inspired by the lean manufacturing paradigm, which is essentially about getting rid of all the waste that don't add value to the customer. The strong value-to-customer focus is compatible with the set of values in Agile Development, but otherwise are the conditions in manufacturing radically different from the ones in development. Whereas manufacturing revolves around predictable and repeatable tasks, homogenous task durations and homogeneous delay costs, development is the exact opposite. In development, high variability, non-repetitive and non-homogeneous tasks are the norm.

In line with these facts, Reinertsen (2009) describes how well-defined phase-gate processes fail. Whereas mutual exclusive phases may appear quite sensible, what really happens is that 95% of all developers quietly starts developing before all requirements are known.

In his 2009-publication, Reinertsen describes eight major these, which characterises what he calls Flow-based Product Development or simply Lean Development. These are:

- Economically based decision making
- Economics of queues
- Managing variability
- Batch size reduction
- Work-in-progress constraints
- Cadence, Synchronisation and flow control

- Fast feedback
- Decentralised control

(Reinertsen, 2009)

Several of the themes above resemble the ones from Agile and Scrum. In Scrum, Work-in-progress is constrained by extreme focus on only a few tasks at a time; cadence and synchronisation is maintained by short and continuous Sprint cycles; Fast feedback is ensured through emphasis on prototyping and reviews; Decentralised control is obtained through self-organising and appropriately empowered Scrum Development Teams.

FLEXIBLE PRODUCT DEVELOPMENT

Preston G. Smith is a representative for the last of the trends briefly summarised here. In his book *Flexible Product Development – Building Agility for Changing Markets*, Smith discusses how flexibility can be implemented in traditional product development in order to gain a competitive advantage in global markets (Smith, 2007). The principles of Flexible Product Development are inspired by elements such as Extreme Programing from software development, Toyota's Set-based Design (Sobek et al. 1999) and delayed decision-making. Preston emphasises the ability to accommodate change and refers directly to the Agile Manifesto of Software Development (Highsmith, 2001) in his arguments for Flexible Product Development.

2.4 THE DEVELOPMENT ORGANISATION

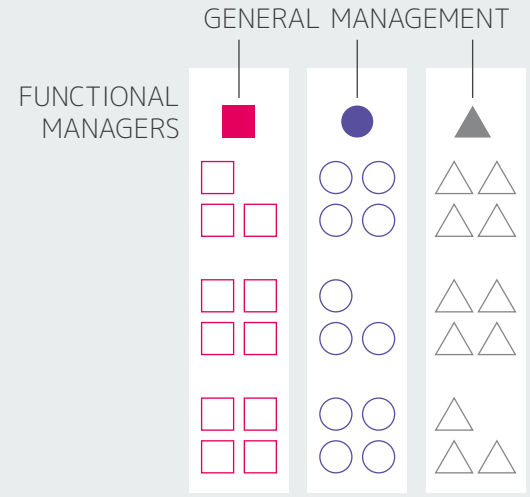
This chapter establishes an overview of relevant knowledge and insights about the organisation supporting the integrated product development activities. The first body of knowledge concerned the actual development process and its control mechanisms. This second knowledge body presents the most common organisational structures, roles and distribution of responsibility and lastly some insights about various team setups.

2.4.1 ORGANISATIONAL STRUCTURES

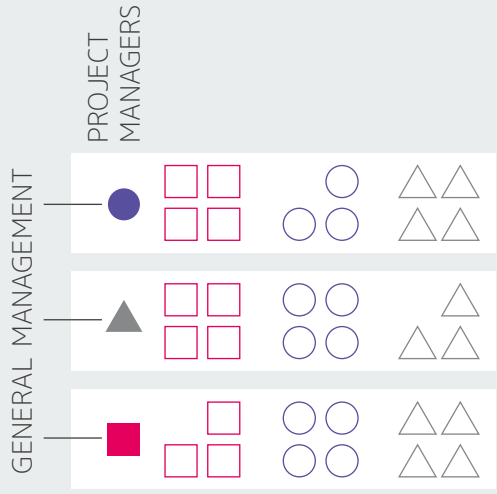
The development process itself is merely an intellectual construction if it is not put to life in an effectively structured organisational setup. In a development context an organisational setup is established by links between individual developers. According to Ulrich & Eppinger (2003) these links may be aligned with professional functions, projects or a combination of both.

In its purest form, a functionally based organisational structure is dividing individuals into functional units such as marketing, manufacturing, stress analysis, mechanical development or similar functionally defined units. In a strictly functional setup, the individuals of one unit would report to the same manager, who would evaluate their respective efforts and set their salaries. In contrast to the functional setup, developers related to specific projects constitute a project organisation. A project team would include individuals from several different functional disciplines, and they would report to a project manager (Ulrich & Eppinger, 2003). The two setups are shown in the upper part of figure 2.10 to the right.

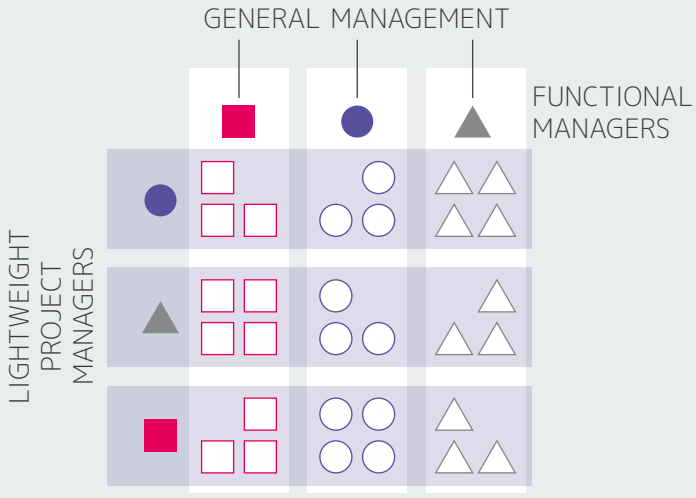
In the investigated cases described in chapter 4, the typical setup seems to be neither a functional nor a project-based setup. Most of the cases consist of various combinations of the two. Organisational structures combining the functional setup with the project-based setup are known as *matrix organisations*. In such organisations each developer has relations to two different supervising managers – a functional line manager and a project manager. According to Hayes



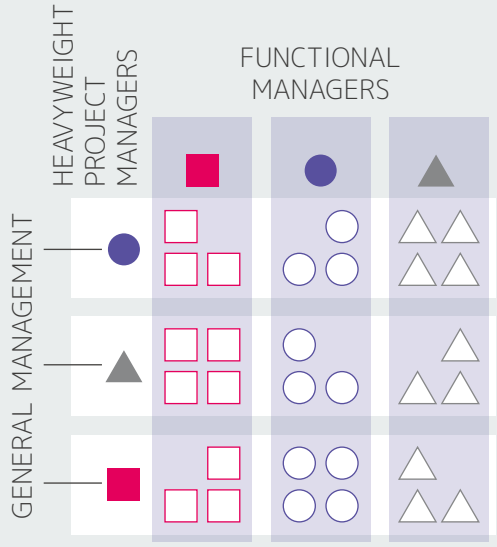
FUNCTIONAL ORGANISATION



PROJECT ORGANISATION



LIGHTWEIGHT PROJECT MATRIX ORGANISATION



HEAVYWEIGHT PROJECT MATRIX ORGANISATION

2.10

VARIOUS ORGANISATIONAL STRUCTURES OF DEVELOPMENT ENVIRONMENTS

Functional Organisation, Project Organisation, Lightweight Project Matrix organisation, and Heavyweight Project Matrix Organisation (Ulrich & Eppinger, 2003; Hayes et al., 1988)

et al. (1988), two variants of the matrix organisation exist. These are *the heavyweight* and *the lightweight project organisation*. The lightweight project organisation contains strong links between the developers and their functional manager, while the heavyweight project organisation has stronger links between the developers and their respective project managers.

The two variants are a result of the fact that most development is carried out in projects and that development organisations put different emphasis on how leadership and empowerment is distributed. According to Andreasen & Hein (2000), a project of such an organisation has certain characteristic properties:

- It runs for a definite period of time
- It has its own staffing resources – a project group
- It is executed within well-defined resource limits
- It spans the entire organisation
- It is interdisciplinary
- It is characterised by development – the output is innovative products
- It is critical to the survival of the company and therefore controlled by top management.

(Andreasen & Hein, 2000)

The Scrum framework calls for a largely project-based structure, as it emphasises the importance of committed and cross-functional teams. However, this does not seem to conflict with most organisation setups, as projects are arguably the most common product development structure. It should to be noted, though, that developers in project organisations are often assigned to multiple projects, which is conflicting with the single-focus policy of Scrum.

Both Scrum and Stage-Gate are compatible with a project-focused organisational structure, however, single-project focus in Scrum conflicts the tendency for multiple projects in many matrix organisations. Results in lack of needed focus.

2.4.2 ROLES AND RESPONSIBILITIES

A part of the organisational structure is the management system of roles and responsibilities for managing and coordinating the use of resources such as money, manpower, equipment, facilities, materials, and technologies. The purpose of this chapter is to look into the most common roles of such management systems surrounding the development team.

As it has been noted in the previous chapter, matrix organisations are hybrids of functional units and project-based teams. This dual-relationship setup requires two managers to safeguard both functions and projects, and this is the well-known interface between line managers and project managers. Both are directly related to the developers. The management roles and their respective responsibilities are described in the following paragraphs.

THE PROJECT MANAGER

The main responsibility of the project manager is to coordinate and integrate activities across multiple functional lines. The integrated activities include development of project plans, execution of project plans and making changes to the project plan. Though the project manager may have a great responsibility, the assignments may not be accompanied with a similar amount of authority. This forces the project manager to negotiate with both upper-level management and functional-line management in order to control the resources needed to fulfil his tasks (Kerzner, 2009). In a project-based culture as the one in a matrix organisation, the project manager must therefore balance between several frontiers in order to maintain healthy relationships:

- Within the project team
- Between the project team and the functional organisations
- Between the project team and the senior management
- Between the project team and the customer's organisation, whether internal or external

(Kerzner, 2009)

It is not without reason that the project manager's role is sometimes called *interface management*. Because of these interfaces to many of the organisation's departments, the project manager will s have an extensive insight to the operation of the company.

THE FUNCTIONAL LINE MANAGER

According to Kerzner (2009) the functional managers are responsible for three different elements. These are:

- Responsibility to define how and where the tasks will be done
- Responsibility to provide sufficient resources to accomplish the objective within the project's constraints
- Responsibility for deliverables

(Kerzner, 2009)

This means that the collaboration between the project manager and the functional manager is carried out in a way that allows the project manager to identify requirements for a given project, and so that it becomes the responsibility of the functional manager to identify the technical criteria. As noted earlier, the various types of matrix organisations are weighting the relationship between the two managers and the team differently. In the table 2.7 below, the typical reporting relationships are illustrated.

THE GENERAL MANAGERS

Depending on the size of the organisation, the management is represented by upper- or lower-level managers in projects. This executive person is expected to take part in the following aspects of the project:

- Project planning and objective-setting
- Conflict resolution
- Priority-setting
- As project sponsor

(Kerzner, 2009)

As a project sponsor, the manager ensures that project information is reaching the executive level in the customer's organisation and that the customer's money is being spent wisely. During the execution of the project, the manager – or project sponsor – provides assistance to the project manager and the development team, if needed, and otherwise keeps a certain distance. According to Kerzner (2009), the project sponsor must maintain an open-door policy and stay ready to assist if needed. It should be noted that this executive role could be in a position, where he or she is supporting multiple projects, which is why detailed project knowledge is often reserved for the project manager and the functional manager.

TYPE OF PROJECT MANAGMENT	TYPE OF MATRIX STRUCTURE	PM NEGOTIATES FOR	EMPLOYEES TAKE TECHNICAL DIRECTION FROM	PM RECEIVES FUNCTIONAL PROGRESS FROM	EMPLOYEE PERFORMANCE EVALUATIONS MADE BY
Lightweight	Weak	Deliverables	LMs	Primarily LMs	LMs only with no input from PM
Heavyweight	Stong	People who report informally to PM but formally to LMs	PMs and LMs	Assigned employees who report to LMs	LMs with input from PM
Tiger Teams	Very strong	People who report entirely to PM full-time for duration of project	PM only	Assigned employees who now report directly to PM	PM only

Legend: Project Manager (PM) / Line Manager (LM)

TABLE 2.7: REPORTING RELATIONSHIPS
Kerzner (2009)

2.4.3 SCRUM ROLES VERSUS MANAGEMENT ROLES

Though the management roles just described may seem to have certain similarities with the roles of Scrum, there are still some significant differences that have the potential of hampering the implementation of Scrum in a matrix project organisation as the ones described earlier. One of the basic differences is the focus on products versus projects. While the most significantly empowered role in Scrum, the Product Owner, is similar to the project sponsor in some aspects in regard to the customer relationship, the Product Owner, furthermore, has an extensive insight and influence on the prioritisation of the development tasks and a close relationship to the team. In a way, the Product Owner seems to share the level of empowerment of the executive manager in a traditional project organisation and the detail orientation similar to the project manager.

The Scrum Master primarily relates to the Development Team as a process facilitator and thereby resembles the project manager in his concern for the progress in the development assignments. However, according to Schwaber & Sutherland (2011) the Scrum Manager has no further authority than this facilitator's role and thereby acts as a "servant-leader" to the Development Team. Finally, the Development Team of Scrum to a larger extent is authorised to organise its development efforts itself without the influence from either the Scrum Master or the Product Owner.

According to Myllerup (2012) it can be argued that the traditional management structure and the Scrum framework represent two different paradigms in development. Rudman (2010) shares this claim and argues that Scrum in "traditional organisations" is a clash of paradigms, where the main difference is the distribution of responsibility. While project management in traditional matrix-organised development environments is the responsibility of the project manager, in Scrum it is a shared responsibility among the Development Team members. It could be argued that the two management systems to some extent have significant asymmetries in regard to their distribution of roles and responsibilities.

Asymmetry in roles and responsibilities between the two systems

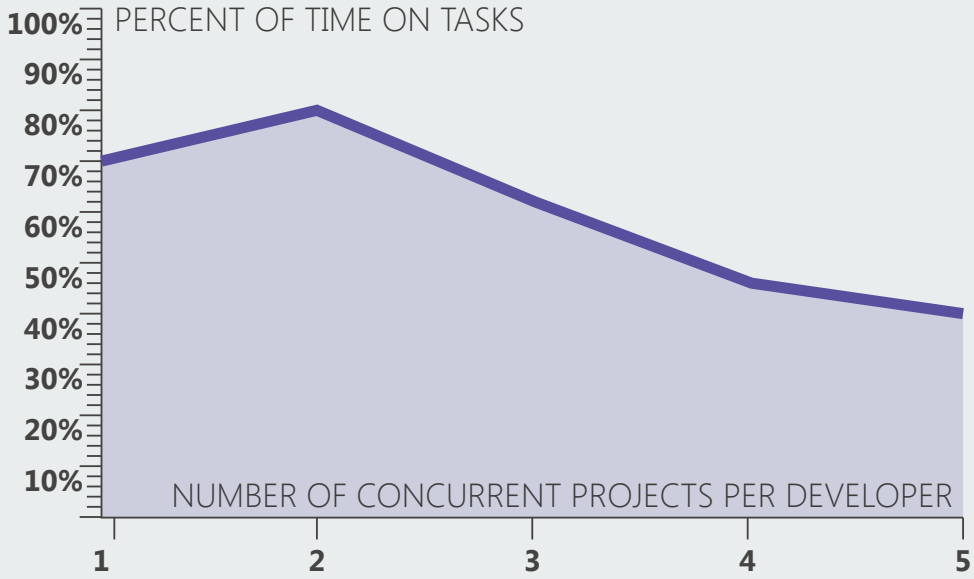
2.4.4 DEVELOPMENT TEAMS

According to several sources (Andreasen & Hein, 2000; Ulrich & Eppinger, 2003; Kerzner, 2009 and others) the cross-functional development team is a typical construction in integrated product development. It brings the necessary interdisciplinary span of competences to the integrated development project. The Scrum framework also promotes cross-functional teams as the optimal development conditions. However, the advantages of cross-functional teams do not come without certain drawbacks. While functional diversity in cross-functional teams of NPD projects according to Keller (2001) has an indirect *positive* effect on technical quality, schedule performance, and budget performance, it has an indirect *negative* effect on team cohesiveness in cross-functional teams due to increased job stress. Based on his study of 93 research- and new product development groups, Keller furthermore argues that this negative effect *"results from stressful relationships among project group members who do not share the same education, training, functional goals, and cultural norms, but who have to work together, often under speed-to-market pressure. "*

Nevertheless it seems as if the cross-functional development team has gained a firm rooting in development environments across industries. In his *Investigation of Factors Contributing to the Success of Cross-Functional Teams*, McDonough III (2000) shows how more than 50% of the companies adopting cross-functional teams are motivated by improving *time-to-market*, whereas only just above 30% are motivated by increased quality of the outcome or increased customer satisfaction. However, the performance of teams has been widely discussed and it has become clear that certain factors are of significant influence on the teams' performance. Two significant factors: *internal team communication* and *project focus* are presented below.

Losada & Heaphy (2004) emphasise the importance of positivity and connectivity in the establishment of high performance teams. They argue that a high ratio between positive and negative verbal expressions as well as a healthy relationship between inquiry and advocacy

Diversity in cross-functional teams may result in decreased team cohesiveness



2.11 **RELATIONSHIP BETWEEN TIME SPENT ON VALUE-ADDING TASKS AND NUMBER OF PROJECTS**
(Wheelwright & Clark, 1992)

in the Development Team has a significant effect on a team's performance. This could arguably be a risk in extremely cross-functional teams as, according to the findings of Keller (2001), they could be subject to stressful conditions. Another significant factor that has the potential of reducing the team performance is lack of focus. Studies by both Wheelwright & Clark (1992) and Smith & Reinertsen (1998) show that multitasking between more than one or two projects will significantly reduce the efficiency of a developer. This is in line with the practice of Scrum. In matrix organisations there is no formal upper limit for the number of projects in which a developer can participate. Figure 2.11 above illustrates the relationship between time spent on value adding-tasks and the number of development projects for a developer.

Finally, A factor that complicates the development in teams just as much is the trend towards increasingly decentralised R&D operations to local markets and centres of excellence (Gassmann & Zedtwitz, 2003). But also locally conducted projects may, to some extent, be dispersed with decreased proximity between Development Team members (Hoegl et al., 2007). The proximity of team members, however, has potentially important implications for the collaborative work of teams. According to research by Hoegl & Proserpio (2004), increasing decentralisation has a negative effect on *communication*, *coordination*, *mutual support*, *Development effort* and *team cohesion*. While dispersed teams are generally avoided in the case companies presented in chapter 4, some, though, are struggling with the problem of balancing team co-location with distribution in functional departments. As Scrum is promoting close collaboration and co-located teams, dispersed teams could – at least from a theoretical point of view – provide a significant challenge.

Tendency to dispersed teams in R&D impedes the close communication, which is part of Scrum.

2.6 CATALOGUE OF GAPS, OVERLAPS AND CONFLICTS

Due to the research questions, the purpose of this chapter has also been to establish a theoretically based catalogue of how Scrum relates to integrated product development and its surrounding development organisation. The catalogue, which comprises *gaps*, *overlaps* and *conflicts* between the Scrum framework and respectively the integrated product development process and the development organisation is presented below. The catalogue serves a guide to the development of an interview guide as is also used for evaluation purposes in the conclusion.

Gaps, overlaps and conflicts in Integrated Product Development Process

- Implementation of Scrum in Stage-Gate gives rise to a conflict between the project inputs of dynamic concepts versus precise project definition
- The output of a Scrum process follows value-to-customer and contrasts the predefined output from the project definition in Stage-Gate
- Stage-Gate lacks a clearly defined retrospective analysis
- The Scrum framework is expected to match the process of the product designer due to its immediate resemblance with the basic design process.

Gaps, overlaps and conflicts in the development organisation

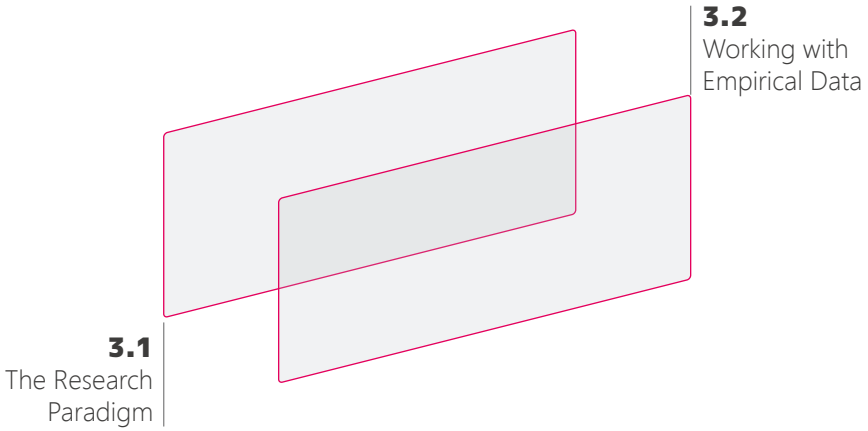
- Traditional project-organisation would be expected to fit the Scrum framework
- Single-project focus in Scrum conflicts multi-projects focus in many matrix organisations, which results in lack of focus
- Asymmetry in roles and responsibilities between the two systems is expected
- Diversity in cross-functional teams may result in decreased team cohesiveness
- Tendency to dispersed teams in R&D impedes the close communication of Scrum.

2.7 SUMMARY OF CHAPTER 2

The objectives of chapter 2 have been to establish a theoretical foundation and to develop an understanding of the empirical findings. This has been done through an *overview of the Scrum framework* and through the unfolding of two separate bodies of knowledge, namely *the integrated product development process* and *the supporting development organisation*. The list below very briefly summarises the insights and claims of chapter 2.

- Scrum is a representative development framework related to Agile Development
- Scrum originates from software development.
- The Scrum framework consists of:
  - Three roles: The Product Owner, the Development Team, and the Scrum Master
  - Four events: Sprint Planning, Daily Scrum, Sprint Review, and Retrospective
  - Three artefacts: The Product Backlog, the Sprint Backlog and the Increment
- Integrated product development is the coordinated interplay across various company divisions and across functional development disciplines
- Two extremes of process control models exist to control the product development process from a macro-level.
- Design process on a micro-level is argued to resemble the Scrum framework with its in-action and on-action activities.
- A development organisation consists of:
  - An organisational structure
  - A management system with certain roles and responsibilities
  - Development Teams
- A theoretically based catalogue of gaps, overlaps, and conflicts between the Scrum framework and respectively the integrated product development process and the development organisation is finally presented in chapter 2.6.

CH.03. RESEARCH METHODOLOGY



3.1 OVERVIEW OF CHAPTER 3

Chapter 3 comprises two parts: The Research Paradigm and Working with Empirical Data.

The purpose of this chapter is to ensure the methodological transparency of the research project by presenting its underlying research paradigmatic stance and the chosen methods for collecting and analysing the research material. The chapter is divided into two separate parts. The first part, chapter 3.1, unfolds the scientific theoretical aspects of the project such as the chosen research strategy and some of the core methodological concepts. It also discusses the importance of consistency between the interrelating building blocks of which the research project is composed. The second part, chapter 3.2, presents the practical aspects and methods used in regard to planning and undertaking the data collection as well as the subsequent analysis and interpretation of it.

3.1 THE RESEARCH PARADIGM

When elaborating the methodology of practicing research, it seems appropriate to begin by asking: *What is research?* A multitude of definitions trying to answer exactly that question exist side by side. A few of them are presented below.

> *"Research is about asking and beginning to answer questions, seeking knowledge and understanding of the world and its processes, and testing assumptions and beliefs"*
> *Wisker, 2008*

As the quote indicates, Wisker (2008) is interpreting research as a somewhat explorative practice with the aim of *understanding* the world around us. Coombes (2001) is slightly more goal-oriented, and almost political, in her definition of carrying out research:

> *"Research is a tool for getting you from point A to point B. You wish to prove an idea – research it. You wish to disprove an idea – research it. You think that fact ABC is incorrect – research it, or that fact ABC is correct – research it. Research is simply a method for investigating and collection information."*
> *Coombes, 2001*

While Wisker's definition captivates the explorative and adventurous nature of research, Coombes' definition seems ignited by a definite purpose that could be even political beyond the desire of *merely understanding*. Perhaps less value-laden is Jensen's (2004) definition of *science* – the result of undertaking research activities:

> *"Science is common sense in combination with systematics"*
> *Jensen, 2004*

However, whether a researcher subscribes to one or the other definition of research, the activity of undertaking research is always influenced by an underlying paradigmatic stance. Both Hindu cosmology and myths from North Eastern American woodland tribes depict the world supported by respectively enormous elephants and giant divine turtles as metaphors for this. Both symbolise ways of understanding the world in a way similar to how various scientific theoretical paradigms, figuratively speaking, support different understandings of the world and criteria of truth.

Two contrasting scientific theoretic paradigms as well as the paradigmatic stance of this project are presented in the following chapters.

3.1.1 THE POSITIVIST AND THE CONSTRUCTIVIST PARADIGMS

According to Bassey (1990) a research paradigm refers to a broad framework of perception, understanding, and belief within which theories and practices operate. It is a network of coherent ideas about the nature of the world and the functions of a researcher (Bassey, 1990). Two contrasting major paradigms exist in the philosophy of science: The traditional positivistic approach originating from the natural sciences and the constructivist approach, which has many different roots, but in general established as critical response to the positivistic tradition.

POSITIVISM: EXPLAINING THE WORLD THROUGH ABSOLUTE KNOWLEDGE

The positivistic philosophy established by Auguste Comte (1798 - 1858) and others reacted against religious dogmas and metaphysical speculation in the 19th century and reintroduced the importance of science solely based on observable data (Kvale & Brinkmann, 2009). During its development, various branches of positivism have been influencing the scientific scene. In the beginning of the 20th century logical positivists adhered to the concept of verification, as the basic premise of scientific knowledge. According to this branch of positivism, true knowledge had to be empirically verified through direct observations.

As a reaction to the search for absolutely *true knowledge*, Karl Popper broke with the positivistic tradition when he established *critical rationalism* – the perhaps most commonly accepted version of the empiricism. Popper replaced the search for absolute knowledge with an aspiration for *justifiable claims* with his introduction of the shift from *verification* to *falsification* (Kvale & Brinkmann, 2009). The famous black swan-example clearly represents Popper's concept of falsification: The hypothesis "all swans are white" may appear justifiable, but has to be rejected when a black swan is observed in Australia.

CONSTRUCTIVISM: UNDERSTANDING THE WORLD THROUGH INTERPRETATION

Contrasting the positivistic and critical rational approaches, constructivism perceives knowledge as social constructions and claims that observations are subjective and therefore cannot represent an absolute truth. In that sense social constructivism is deconstructing the traditional ideal of science and at the same time constructs a new understanding of how to perceive science and the act of research. Social constructivists are searching for relations and certainty, knowing that they will never obtain any absolute knowledge or certainty (Wenneberg, 2002).

Returning to the basic assumption in the constructivist paradigm, that the condition of knowledge is merely social construction, meaning is reached through subjective interpretations of observations. This calls for two other scientific theoretical concepts, namely phenomenology and hermeneutics, which respectively concern the phenomena observed and the interpretation of the phenomena.

Phenomenology was founded by Edmund Husserl as a branch of existentialism around 1900 and was later expanded by Martin Heidegger to also include the human "lebenswelt" as its object (Kvale & Brinkmann 2009). In this respect phenomenology assists in enabling an abstraction of the observed with a purpose of subsequently building theories that push the existing "knowledge" a step further.

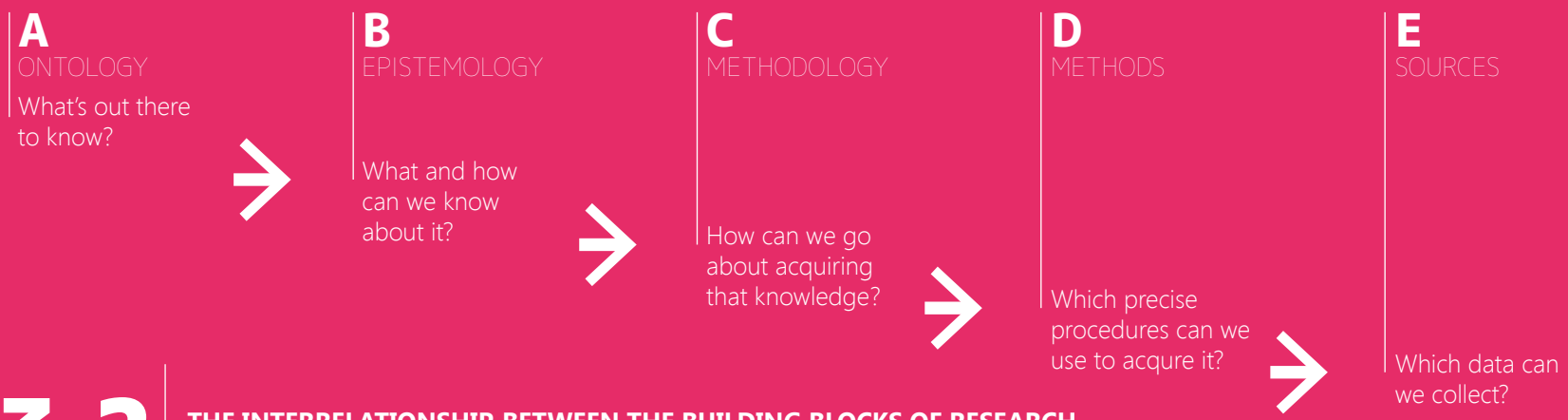
Hans-Georg Gadamer has an influential role on hermeneutics, which concerns the interpretation of text, discourse and action. According to Gadamer, we are all speaking creatures to whom the language is reality (Bernstein, 1983). Gadamer emphasises the importance and awareness of pre-understanding in interpretation. Hermeneutics thereby assists qualitative researchers in being aware of contextual aspects in and after for instance an interview situation and thereby objectify themselves from the observed.

THE POSITION OF THIS PROJECT

There is a clear distinction between the positivist and the constructivist approach. The positivist approach is purely empirical and quantitatively rooted, whereas the constructivist approach is primarily qualitative. They differ in their views of the basic purpose of scientific practice, as explaining motivates the positivist paradigm and understanding motivates the constructivist paradigm. Constructivists argue that objective science and search for absolute knowledge are erred.

The two are fundamentally different in their ontological views. Positivists think of reality as neutral and objective, consisting of isolated entities and observable data, whereas the perception of the world is regarded as a subjective construction in the constructivist paradigm.

The empirical scope of the present thesis is integrated product development environments in largely engineering-based companies. Because of the dominance of engineering, the typical scientific theoretic approach within this domain would originate from natural sciences – and therefore often be rather positivistic. However, design engineering is a cross field and utilises both positivistic and constructivist approaches in its scientific practice.



3.2 **THE INTERRELATIONSHIP BETWEEN THE BUILDING BLOCKS OF RESEARCH**
Relating ontology, epistemology, methodology, methods, and sources (Hay, 2002).

This project is focusing of identifying the challenges of becoming agile *experienced* by employees and the project therefore deals with the interpretation of subjective views on reality. The project is therefore taking on a *constructivist approach* in the process of producing *meaningful knowledge* from the empirical observations.

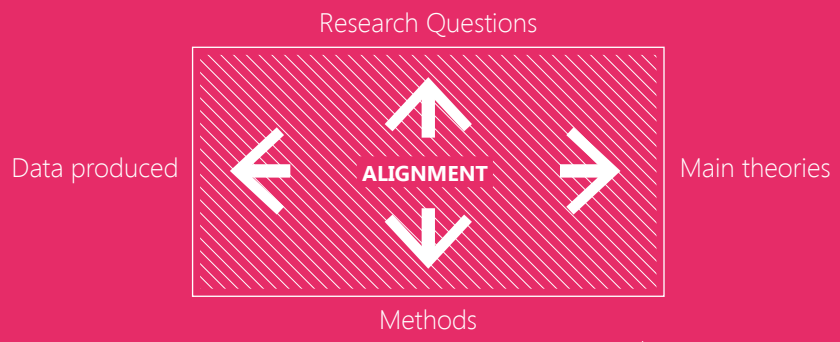
3.1.2 INTERRELATIONSHIP AND CONSISTENCY IN RESEARCH

As this thesis subscribes to the constructivist approach for producing meaningful knowledge, it also subscribes to its ontological assumptions about reality as social construction. This ontological stance is logically followed by epistemological and methodological positions, and even the practical methods and sources chosen should consistently reflect this interrelationship. Hay (2002) illustrates this interrelationship within a scientific paradigm in figure 3.2 to the left.

While ontological assumptions concern what we believe constitutes social reality, an epistemology consists of ideas about what can count as knowledge (Blaikie, 2000). In that respect the constructivist epistemology argues that also knowledge is socially constructed (Wenneberg, 2002). But constructivist epistemology is difficult to label, as multiple positions exist within the constructivist paradigm. Heylighen (1993) argues that social constructivism "sees consensus between different subjects as the ultimate criterion to judge knowledge. 'Truth' or 'reality' will be accorded only to those constructions on which most people of a social group agree."

Again, as a logical consequence of the ontological and epistemological positions in the constructivist paradigm, the relating methodology primarily concerns qualitative methods, which are appropriate when studying values, actions and other relations that are not meaningfully quantified. This research project utilises qualitative and semi-structured interviews (Kvale & Brinkmann, 2009) as the primary research method for obtaining its data.

A similar and perhaps more practically oriented way of understanding the interrelating elements in research is securing an alignment between question, method, main theories and data (Tollestrup et al., 2011). Tollestrup et al. (2011) argue that research questions should be aligned with the chosen methods, just as the data produced should be aligned with the main existing theories. This alignment is visually represented in figure 3.3 below.



ALIGNMENT IN RESEARCH PRACTICE **3.3**
Aligning RQs, Methods, Main theories, and Data produces (Tollestrup et al., 2011)

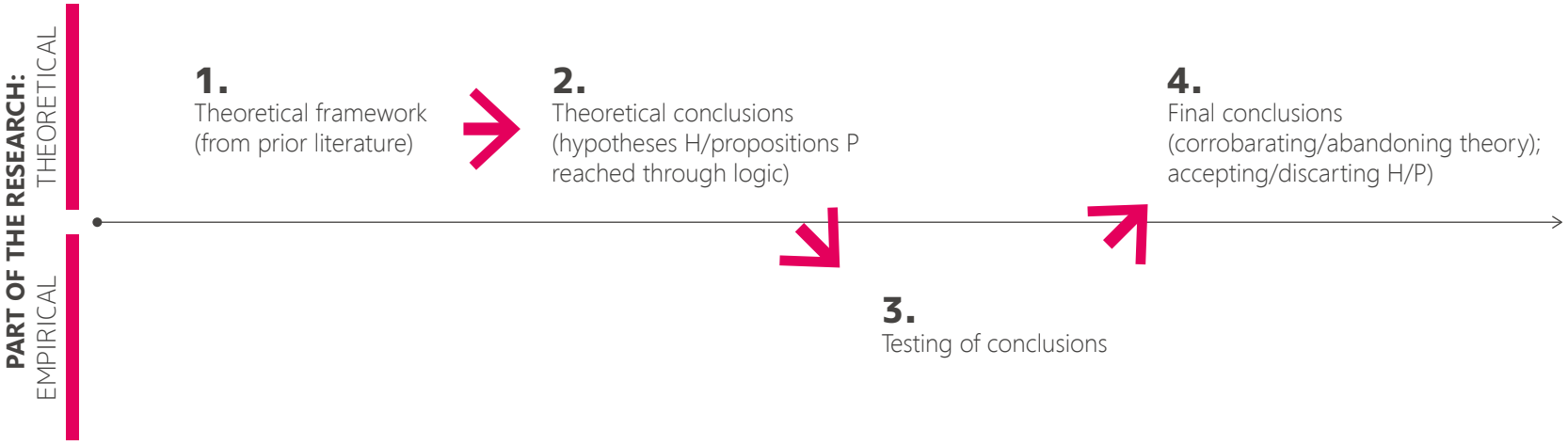
To the present research project, the alignment diagram above serves as a precautionary measure for ensuring consistency between its respective building blocks, and it has been regularly revisited throughout the course of the project.

3.1.3 THE CHOSEN RESEARCH STRATEGY

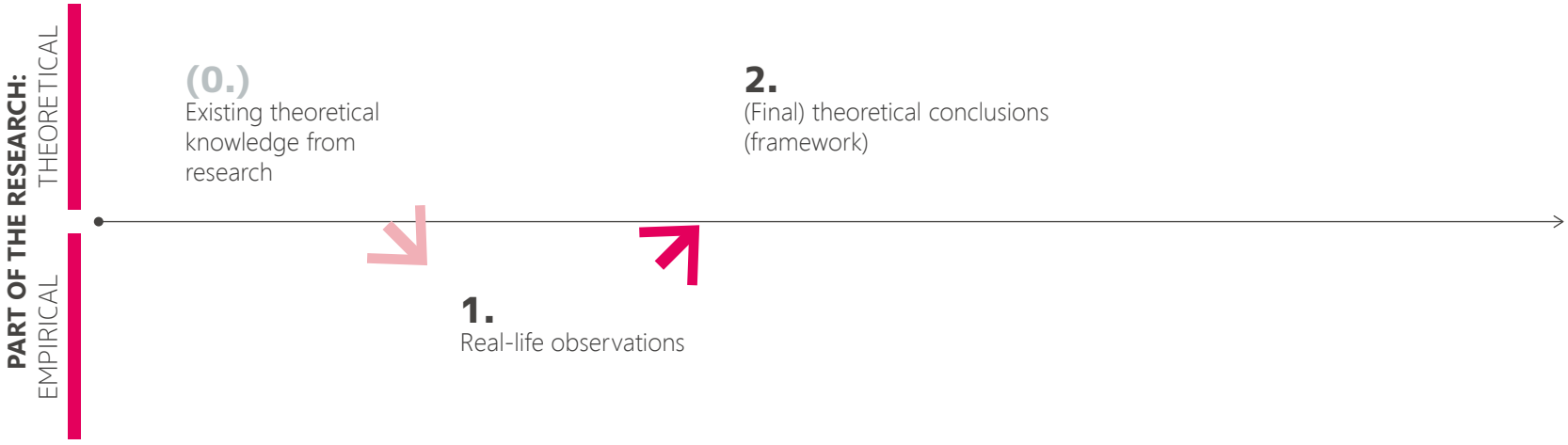
Before turning to the practical aspects of undertaking the research activities, this chapter presents the basic research strategy that the project has been pursuing.

The inductive and *the deductive* research approaches have traditionally been predominant in research, and today most research is still practiced through one or the other. In nature the two approaches differ in their opposing starting points in respectively theory or empirical data. While the purely inductive approach induces generalisations from real-life observations, the deductive approach attempts to verify (or falsify from a critical rational point of view) a constructed theory or hypothesis through observations in order to accomplish a higher level of insight (Kovács & Spens, 2005). Figure 3.4 below and figure 3.5 to the right illustrates the two approaches.

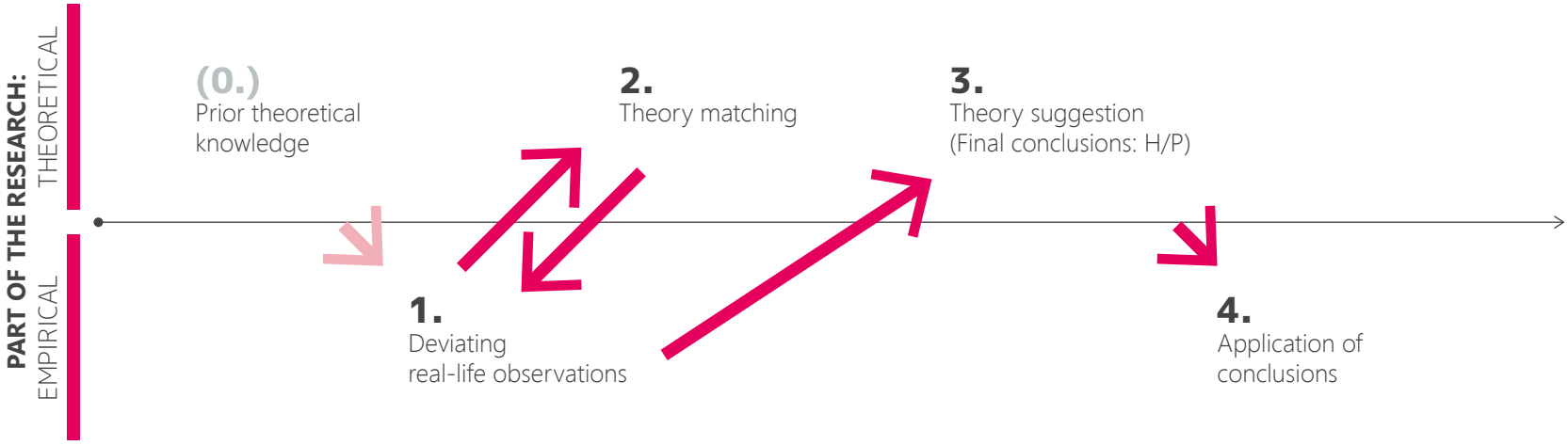
Obviously, the two mentioned approaches both have their strengths. Nevertheless, according to several sources (Kirkeby in Andersen, 1994; Taylor et al., 2002) most great advances in science have followed a third approach. This approach is called abduction and is often seen as a combination of the inductive and the deductive approach, which is illustrated in figure 3.6 to the right. The Abductive process is most often credited Charles Sanders Peirce and is described as a kind of logical inference, partly initiated by "qualified guessing". The Abductive approach is often depicted as a cyclic process between inductive and deductive activities, which is also the case in this project: The initial research activities were purely inductive and led to the first sketch of a hypothesis about the potential of Agile Development methods in physical product development. This gave rise to a series of more focused interviews with individuals working in Scrum various development organisations. Finally the series of case studies were carried out on the basis of a refined hypothesis.



3.4 **PURELY DEDUCTIVE RESEARCH PROCESS**
(Kovács & Spens, 2005).



3.5 **PURELY INDUCTIVE RESEARCH PROCESS**
(Kovács & Spens, 2005).



3.6 **PURELY ABDUCTIVE RESEARCH PROCESS**
(Kovács & Spens, 2005).

3.2 WORKING WITH EMPIRICAL DATA

This chapter unfolds the practical methodological aspects of undertaking the research activities and is structured in two subsequent steps:

01. Data collection
02. Data Analysis

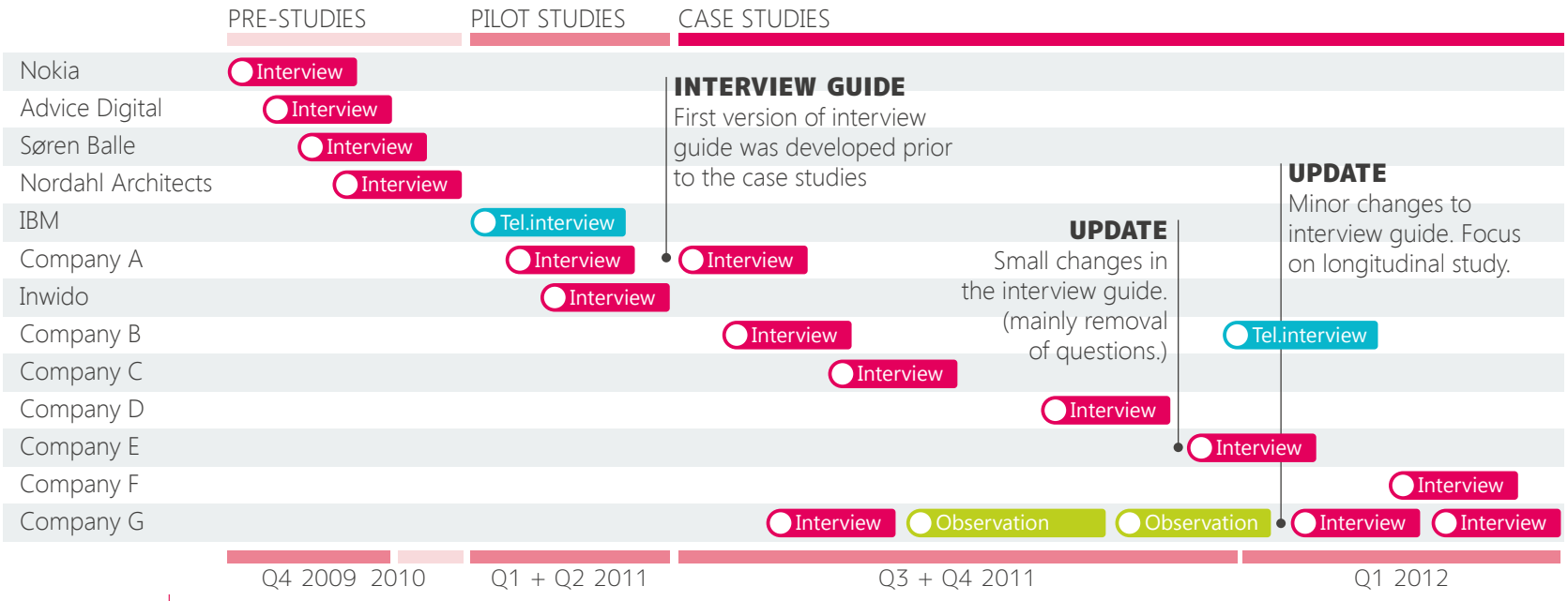
3.2.1 DATA COLLECTION

Naturally, the data collection process is closely depending on which type of data is desired. In the case of this project, the desired data are primarily of qualitative character as its object is the challenges associated with implementing Scrum in integrated product development environments *experienced* by the Scrum participants. Some of the most used qualitative methods for data collection are *participant observation*, *interviews,* and *focus groups*, and in this project the two first mentioned are used.

THE PROCESS OF PREPARING AND COLLECTING DATA – AN OVERVIEW

Before going into further description of the actual methods used for collecting the data, an overview of the data collection activities carried out in this project seems appropriate. This is found in figure 3.7 below.

This project has truly been initiated in an inductive manner. The first "pre-study interviews" were conducted with no emphasis on Scrum or even Agile Development, but merely based on the motivation of identifying potential research topics by broadly interviewing company representatives about their design processes. From this series of interviews the first clues in regard to Agile Development began to surface and slowly established the direction of the following research activities.



3.7 **OVERVIEW OF DATA COLLECTION ACTIVITIES**

The interviews gave inspiration to a subsequent literature review and the establishing of the first version of a hypothesis: *Product designers can benefit from agile software methods*. The hypothesis was tested in three pilot interviews, which also had the purpose of preparing for the first iteration of an interview guide. The interviews were conducted with employees in three different companies, IBM, TC Electronic, and Inwido, and the companies had been found through various networking activities.

After the pilot interviews, a series of cases were planned with separate companies. The main criterion as to the choice of companies was the presence of experience with the Scrum framework. The companies were primarily identified by posting a question about Scrum in integrated product development to the "Scrum Denmark forum" at LinkedIn.com. The post initiated a lively discussion about the topic and resulted in a number of arrangements for the subsequent case studies.

After the development of the first iteration of the interview guide, a series of case studies were carried out. The following chapter presents the chosen data collection methods that have only been briefly mentioned in the previous chapters.

CASE STUDIES WITH SEMI-STRUCTURED INTERVIEWS

The overall method used for collecting the data was the Case Study method. According to Yin (2003) "a case study is an empirical inquiry that a) investigates a contemporary phenomenon within its real-life context, especially when b) the boundaries between phenomenon and context are not clearly evident." Yin (2003) continues: "In other words, you would use the case study method because you deliberately wanted to cover contextual conditions". As mentioned, the cases were primarily identified through the LinkedIn group, but the initial pre-study interviews also assisted in pointing out some of the case companies. It was ensured that each of the case companies were developing products that required a broad range of disciplines and that their respective development environments were conducting Scrum to a certain extent.

The practical data collection in each case was mainly conducted through *semi-structured interviews*. According to Kvale & Brinkmann (2009), an interview-based study comprises seven phases: *identifying themes, design, interview, transcription, analysis, verification,* and *reporting*. The themes in the interviews were primarily based on

A story about the value of knowledge

R&D environments are often secretive and protective about their work. This was clearly felt when I contacted a company secretary and asked her to forward my application for using their development department as a case for my work. Less than an hour later I was contacted by a representative from the upper management from an airport in USA. He questioned me for 40 minutes about the research project and finished the conversation by stating that he had to think about whether or not they could let me in. A couple of weeks later I got a rejection. After having continued the correspondence and changed the planned research activities, I was allowed inside to conduct two interviews a couple of month later.

The story illustrates how highly some companies value their hard work with implementing Scrum in an integrated product development environment. To the company, the experiences with Scrum and the adjustments they had done to the framework was a strategic part of their competitive edge. Despite their interest and sympathy with my project, they simply just had to be careful.

the catalogue of expected challenges found in theoretical foundation, but experiences from the pre-study interviews influenced the themes. In the design phase, the themes were unfolded and restructured into an interview guide, as a preparation for the interviews. As an aid to the interview guide, a set of A3-posters was developed illustrating visual representations of some of the questions. This was done in order for the interviewees to be able to express themselves by sketching on the paper.

Before the actual interviews, the interviewees had to be selected. Selection criteria were defined as the following:

- Experience with participation in projects conducted through Scrum
- Interviewees should represent different Scrum roles

In practice, it ended up with being the company contacts who decided who should participate in the interviews. Thereby it was primarily the consideration for the development environment that dictated who was going to take part in the interviews.

RECORDING THE INTERVIEWS

The third phase according to Kvale & Brinkmann (2009) is the actual interview. All interviews were recorded on audio and video. The video camera only registered the table top with the purpose of registering the interviewees' sketches and gestures. The use of recordings carries along both advantages and disadvantages (Trost & Jeremiassen, 2010). One advantage is the possibility to listen to the tone of voice *after* the interview. A disadvantage could be the risk of hesitant interviewees, who found it uncomfortable. In practice, all of the interviewees accepted the recording equipment instantly. An advantage, not to be forgotten, was the fact that the recordings made it possible to also learn from and evaluate the process by listening to oneself.

WORKSHOP OBSERVATIONS

One case stood out from the others by also including a series of workshop observations. While all the other cases were limited in time to the actual interviews, this special case represented a rather longitudinal study of the progress of implementing Scrum in a large organisation. The workshops were organised and facilitated by a management consultancy, and the observations were carried out at the client of the consultancy without the observer being an active part of the workshop programme.

3.2.2 DATA ANALYSIS

The analysis of the collected data was carried out in several steps, which are described below.

TRANSCRIPTION

The interviews from the seven cases resulted in 20 hours of recorded dialogue and the same amount of video recordings. The first step of the analysis was therefore to review the material. Each interview was initially examined and simultaneously transcribed in very low detail, in order to establish an overview of the themes. A more thorough transcription was then carried out. However, as the focus in this research project has not been on conversation analysis, the transcription was still carried out rather roughly.

The "coarseness" and style of transcription is always a subjective judgement as there is not one truly objective way to transform the interview from oral to written form. According to Kvale & Brinkmann (2009), the question should rather be "what transcription type is useful to my research objectives?" For the purpose of this project, the transcription was merely a means to establish a thorough overview of the positions and attitudes of the interviewees in regard to the experiences with Scrum.

The result of the transcription and the review of the data was a series of uniformly structured case descriptions supported by quotations from interviewees, which can be found in chapter 4.

ANALYSIS

The analysis was conducted in three steps. The first part focused on establishing a comparative overview and on converging the challenges identified through the case descriptions into a set of themes. The themes were formed on basis of the comparison of challenges between multiple cases. In the second part of the analysis, the identified challenges were unfolded through a structured scheme. The issues brought up in this part of the analysis were compared to the theoretically based challenges from chapter 2; and the issues in each theme were condensed (Kvale & Brinkmann, 2009) into a list of aspects related to respectively *the product designer* and *the development organisation*.

The last part of the analysis was carried out by further condensing and synthesising the meanings of the challenge-themes into a set of general conditions. The purpose of this was to reach a higher level of abstraction and thereby establishing a scale of generalisation consisting of *identified challenges, themes,* and *general conditions*.

3.3 SUMMARY OF CHAPTER 3

The objectives of chapter 3 have been to ensure the methodological transparency of the research project by presenting its underlying paradigmatic stance and the chosen methods for collecting and analysing the research material. The list below summarises the contents of chapter 3.

- Two research paradigmatic stances – the positivist and the constructivist paradigms – are briefly unfolded.
- The project is taking on a constructivist approach in the process of producing meaningful knowledge from the empirical observations.
- The project subscribes to the ontological assumptions about reality as social construction.
- From this ontological stance the epistemological and methodological positions logically follow.
- The interrelation between research question, method, main theories, and data is emphasised.
- The chosen research strategy implies both inductive and deductive activities and thereby resembles an abductive approach.
- The practical data collection is carried out as case studies, primarily conducted through semi-structured interviews, but also observation.
- The recorded interviews are briefly transcribed prior to the analysis.
- The analysis comprises three parts with their respective focus: identification of challenges, themes, and general conditions.

CH.04. THE RESEARCH MATERIAL

Everything is data...

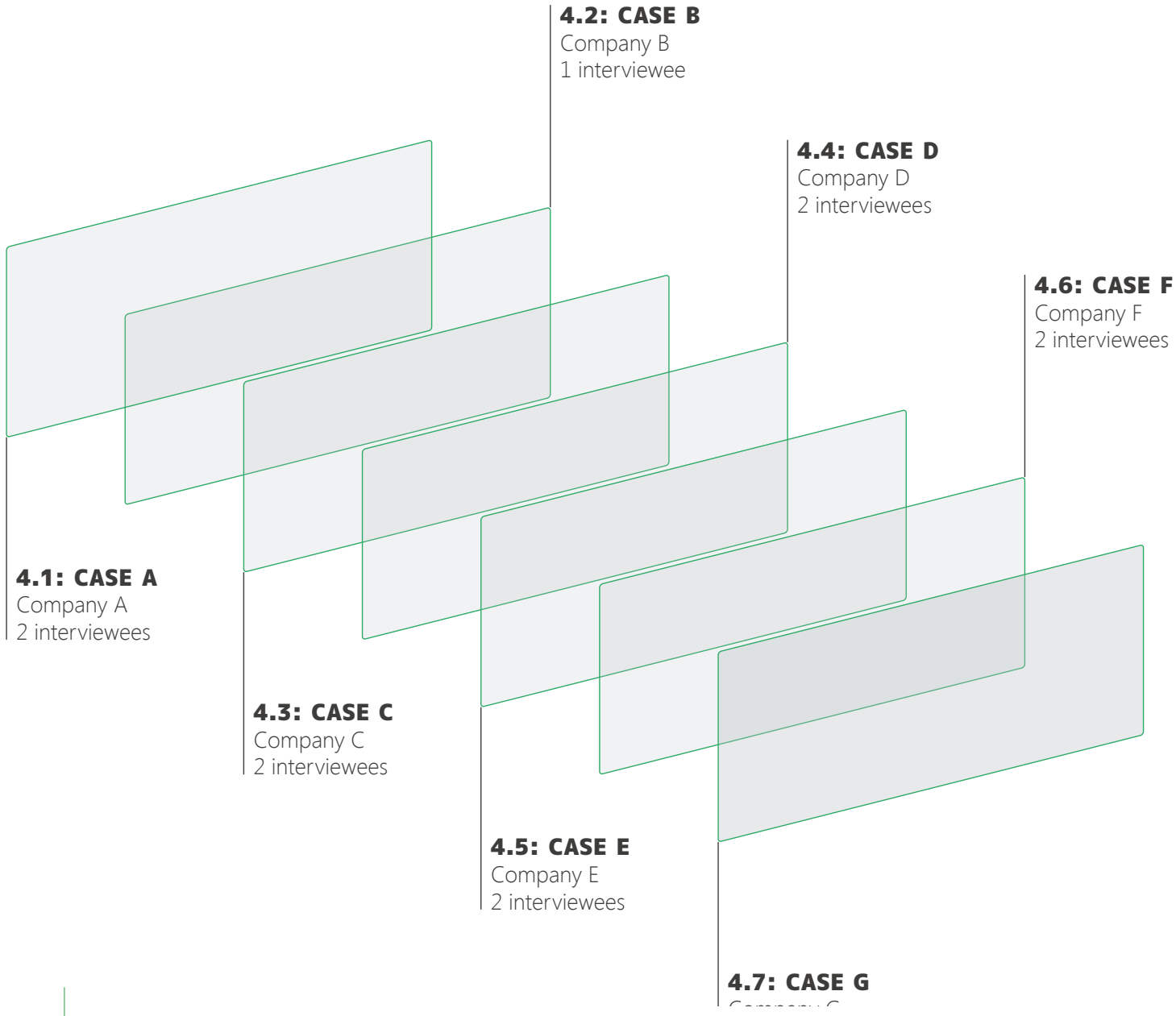
IM.02 **INTERVIEW SITUATION**

An interviewee talks about his experiences with Scrum while scribbling on the A3-posters that was developed as visual representations of some of the questions from the interview guide.

The purpose of this chapter is to present the empirical data collected through interviews and observations from seven organisations. The data from each organisation comprises a separate case, and all case descriptions follow the same format. The structure of this format is a division of the case material in the following sections:

- Basic case data
- Organisational structure
- Scrum in the development process
- Motivation and transition to Scrum
- Scrum framework #1: Product Vision and Product Backlog efforts
- Scrum framework #2: Sprint Cycle
- Design and styling efforts
- Main Challenges experienced by interviewees

While the first six cases are purely based on interviews, the seventh case also includes observations. However, the structure of the case format is the same for all seven cases. An overview of the chapter is seen in figure 4.1 to the right.



4.1

OVERVIEW OF CHAPTER 4

Chapter 4 comprises 7 cases all based on interviews. The seventh cases also contain observations.

BASIC CASE DATA
COMPANY A

CASE **A**

THE COMPANY

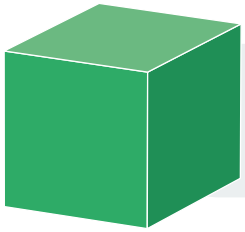
OWNERSHIP:
UNDISCLOSED

LOCATION:
DENMARK

185

IN R&D
40

PRODUCTS & MARKETS



Digital audio products for guitar and bass professionals as well as recording and broadcasting environments.

Enthusiast and professional musicians worldwide. Independent distributors in Europe, America and Asian

PRODUCTS & MARKETS

4 YEARS

Relatively long experience compared to the other cases

LENGTH OF SCRUM EXPERIENCE

Scrum is implemented in all development activities

IMPACT IN ORGANISATION

3-week Sprints running in sync between all Scrum teams

SPRINT LENGTH IN WEEKS

INTERVIEWEES

NO.1

Name Interviewee A1
Position SW Development Manager
Education Electronics engineer

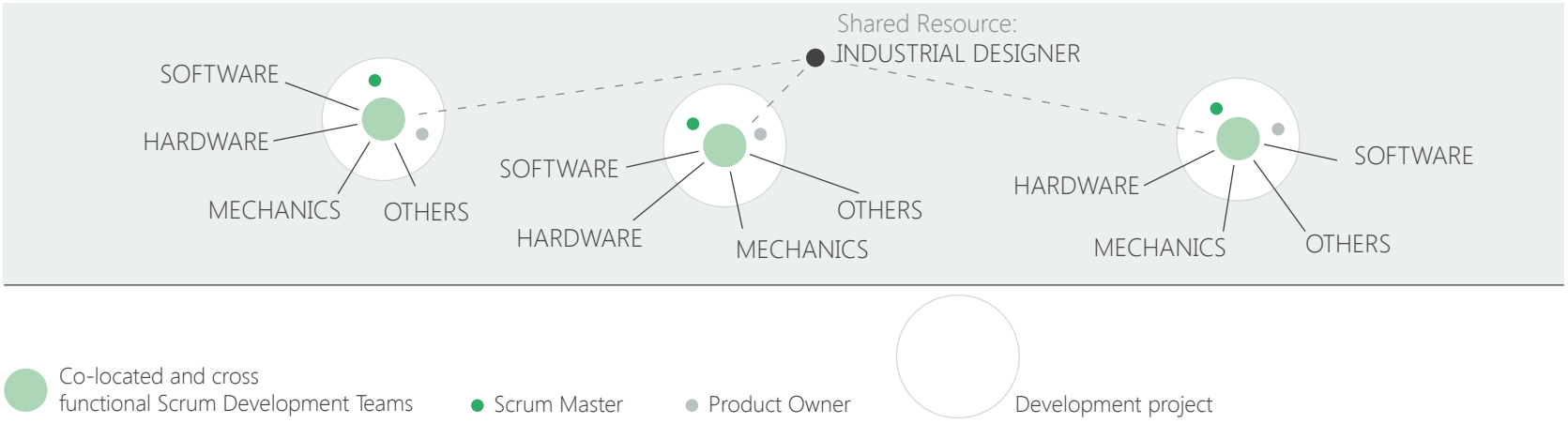
Role in Relation to Scrum
Mixed role. Formally product owner in Pro division, but also Scrum coach in relation to the two other divisions.

NO.2

Name Interviewee A2
Position Senior Product dev. engineer
Education Electronics engineer

Role in Relation to Scrum
Developer and Task Force Manager in a larger project. Works with electronic and mechanical development

4.2: BASIC CASE DATA



4.1.1 ORGANISATIONAL STRUCTURE

The development department in Company A is organised in three business areas: the Guitar & Bass division, the Instrument Amplification division, and the Pro Division. Each division has its own cross-functional Scrum team consisting of software engineers, electronics engineers, mechanical engineers and a number of people with other competencies. Some of the team members are "shared resources", meaning that these specific employees are related to more than one team. In 2008, the development department underwent a restructuring process from a traditional matrix organisation with each employee mainly affiliated to a group of similar professional competence and at the same time engaged in multiple projects of cross-functional character. The new organisation was implemented in order to meet some of the practices in the Scrum framework.

The development organisation carries out the development of software, electronics and mechanics in-house, and outsources the development of wooden components for amplifier racks and the like. Company A has a part of the production in Denmark, but mainly abroad in Thailand, and is now starting up in China as well.

4.1.2 SCRUM IN THE DEVELOPMENT PROCESS

Company A uses a Stage-Gate process model as the overall management tool for development activities. The Stage-Gate model has been implemented before the introduction of Scrum and has continued to be the principal guiding framework. The Scrum activities of the development organisation are fitted into this, primarily in the third stage called *Development*, but according to both interviewees, efforts are made to push Scrum activities into the preceding concept development stage, called *Specification*. However, Interviewee A2 argues that this is only partly possible, due to the unstructured nature of idea generation and concept development:

> *"It [Scrum] is not that structured in the concept development phase. It is a part of it. A lot of different things happen here – It can be everything from chatter at the coffee machine and unstructured stuff. It is difficult – it is the classic problem: You cannot just sit down from eight to four and get a good idea. You simply cannot do that."*

4.1.3 MOTIVATION AND TRANSITION TO SCRUM

Scrum is first tentatively implemented in three teams in the software development department on a request from the director of development. The experiment exists below radar for a while, but when a market critical project in another division is having serious problems and is about to fail, the head of software development suggests using Scrum to straighten up the project. The operation is a success and an eye-opener to the management. As the global crisis surfaces in 2008, the CEO decides to broadly implement Scrum with cross-functional teams throughout the development organisation.

Interviewee A1 recalls the situation before implementing the Scrum framework in the development organisation: *"We had some fine project models with Stage-Gate, which had been eveloped throughout the years. The actual challenge was that the employees were engaged in several projects at a time, they met only once a week on each project. We had some line managers – including me – with insufficient contact to the projects. (…) We always had to argue about the priorities of the various tasks, and some felt that others were stealing resources. (…) The individual employee was in a constant conflict about who to follow – the line manager or the project manager."*

Interviewee A2 emphasises the importance of the firm decision made by the top management about broadly implementing Scrum into the organisation as a whole: *"At some point, this [Scrum] was sort of forced upon us. Fortunately, it was a decision made by the top management, meaning that the broad organisation was being well prepared for it, and this was something that we were going to implement. After all, it is problematic in larger companies that you do not necessarily have the ability to communicate upwards. Luckily, we have had this opportunity, and we are now divided into market divisions instead of by professional competencies"*

Motivations:

- Failing projects due to communication issues
- Global crisis in 2008 calls for efficiency in development organisation
- Resource management is complicated by matrix organisation

Initiative:

- Top-down (CEO decides to implement scrum throughout the development organisation)

4.1.4 SCRUM FRAMEWORK #1: PRODUCT VISION AND PRODUCT BACKLOGS EFFORTS

In the specification phase Company A uses the Backlogs to roughly describe the product. They distinguish between a number of different levels of detail in the product backlog, and in the specification phase, focus on creating user stories as *epics* – rough and general descriptions of activities related to the product development, which can be contained in very few sentences.

The next step is the development of block diagrams, which are then developed into descriptions in the backlog. An important part of this process is the estimation of needed resources for each of the identified backlog items. The estimation is made collectively by the use of Planning Poker.

> *Interviewee A1: "We challenge each other in estimating needed resources regardless of how detailed or roughly described the backlog items are described"*

The estimation is quantified into story points, which is a measure combining size and complexity of a certain task. With an overall estimate in the form of *story points* and in existing experiences about the team's velocity, it is possible to make rough calculations on timeframe and budget.

Progressing into the project, the budget and timeframe estimation is monitored and if necessary adjusted by the use of two different tools, a version of the project management triangle and a burn down chart

In general there is a pragmatic attitude towards project planning and the backlog is seen as a dynamic document, which is able to change throughout the project – at least from the perspective of Interviewee A1. This is obvious in the following quote:

Interviewee A1: "To me, Scrum is not about the conditions being so complex that we are not able to say anything. We have to make some estimates. Later, if we realise that our estimates are failing, then we have to make new best guesses."

Interviewee A2, who, it must remembered, is part of another Scrum Team than Interviewee A1, has somewhat different experiences with the product backlog:

Interviewee A2: "When we start a project, the product backlog is very rough and describing the specification in epics. At that stage it is just a proposal to build on. After this, there is no link between the backlog and the final product. In a way, the backlog is then put behind us. (...) The reason why we are able to do this is that the projects to some extend are similar to each other."

This quote indicates that the backlog is not actively being groomed, used, or in other ways maintained throughout the project, but that the team relies on its knowledge and experiences from other projects.

When asked how they maintain the flexibility towards changes, Interviewee A2 mentions that they ought to be better at revisiting the Product Backlog, but also argues that it is fortunate that many of their products have a significant amount of software elements, which make late changes easier to implement.

Interviewee A2: "We have a great flexibility in the way we make features available to the users, and that is one thing that helps us. We try to use that to a maximum extent, that we have the possibility to change things at a late stage."

4.1.5 SCRUM FRAMEWORK #2: SPRINT CYCLE

The Scrum teams at Company A go through the Sprints in three-week cycles and the three divisions are conducting the cycles in sync. The sprints are, to a large extent, complying with the guidelines in the Scrum Guide by Schwaber and Sutherland.

The Scrum Teams split the Sprint Planning meeting into two parts, just as the Scrum Guide describes it. However, in the tactical and latter part of the Sprint Planning meeting, the team splits into two separate groups – respectively hardware and software – in order to develop the detailed plans of the development activities in the coming Sprint. This is a deviation from the actual Scrum practice.

The rest of the three-week cycle they proceed with Daily Scrum meetings across all disciplines in the Development Team. One or two of the Daily Scrum meetings during the course of a Sprint are extended a bit in order for the team to take an extra look at the development status compared to the Sprint Backlog.

Due to the high level of product integration and the dependency on sub-suppliers and development partners, the Sprint backlog is affected by the larger milestones. The teams use various versions of Gantt charts in order to manage deadlines and deliverables with external partners. This is not a celebrated tool in the Scrum framework; on the contrary it represents the traditional development practice, which at least Interviewee A1 dissociate himself from:

Interviewee A1 acknowledges the paradoxical aspects of using Gantt charts as a tool in projects driven by the Scrum framework: *A Scrum coach that uses Microsoft Project. Embarrassing! [Laughing] But I do it in a different way."*

Interviewee A2 also reflects on this issue and states that – *"there is a need for externally related coordination beyond the Scrum framework."*

After going through a Sprint, the separate team enters the Sprint Review and Retrospective meetings. According to Interviewee A2, the Sprint Review meeting has a subordinate influence compared to the Retrospective. This is because of the absence of actual prototypes. Interviewee A1 mentions that the result of the development efforts is most often *"pieces of paper rather than working functionality"* and admits that this is not in accordance with the Agile Manifesto.

As mentioned, the Retrospective is relatively important and used actively according to Interviewee A2:

> *"We are not that focused on making an open review, in fact we only do that occasionally. (…) We did it more often in the beginning, but instead we focus on the Retrospective, where we continuously attempt to identify better ways for doing stuff. What went ok and what do we need to change? – we actually use quite some time on this."*

4.1.6 DESIGN AND STYLING EFFORTS

Company A is marketing its products to musicians. Thus design and styling *is* an important part of the product development efforts. However, the mechanical design and product styling is determined relatively early in the process – often partly described in the product vision. It is not an integrated and dynamic part of the whole development process.

According to Interviewee A2, the first one or two Sprints are often concentrating on the conceptual design of the product, both in regard to specification of hardware, software, and the mechanical construction. Design-wise, a result of this process could be a 2D drawing of the product front, colour selection, or interface layout around Sprint 3.

One of the shared resources is an in-house industrial designer, who is involved in the styling and visual identity of the products in all three business divisions. It is the responsibilities of the industrial designer to ensure that a coherent visual product expression is included in the business proposals, which is prior to both the specification and the development phase.

Interviewee A1 emphasises the importance of product expressions, regardless of product category: *"We always have an expression [in the product vision], and especially in our Guitar & Bass- or Instrument Amp divisions – in the consumer products. (…) In my division it is more about 19-inch racks, you know [laughing]. But it still has an expression."*

Both interviewees mention user involvement as an issue in the development process. Interviewee A1 argues that Company A is a special case, as most of its employees are amateurs and enthusiastic musicians themselves.

> *Interviewee A1: "There's a lot about usability and the users' needs. This is perhaps an area in which we are unconsciously competent, because largely all employees are customers or users in one way or another."*

He continues saying that they are being more focused on including users in the development process and that they have also been so recently in the development of a series of lower end guitar equipment. It is not mentioned to what extent this effort has been part of the iterative process of the Sprints.

4.1.7 MAIN CHALLENGES EXPERIENCED BY INTERVIEWEES

When asked what they consider the main challenges of using Scrum in their product development, the interviewees mention a number of issues. The issues mentioned are described below in order of immediate emphasis.

POTENTIALLY SHIPPABLE INCREMENTS AT EACH SPRINT

Making independent and potentially shippable product increments in each Sprint according to both interviewees is the most significant challenge. They argue that the physical dimensions and strong integration of the often complex products makes it almost impossible to reach the desired level of maturation through one Sprint. At the same time, it is difficult to consider a single feature or functionality without considering the context which it is part of.

Interviewee A1 describes the problem through the analogy of a car: *"We cannot build a car, by developing the motor in Sprint 1 and the steering mechanism in Sprint 2. And in Sprint 5 we make the brake. That we cannot test. We need to develop all the parts, but in the sim-*

plest version possible. This means that if we concentrate on the motor in one Sprint, we also include the simplest version of a brake. Then we develop the actual brake later"

> *Interviewee A2: "When making a product including hardware and mechanics, it is clearly conflicting with the idea of having something that has been developed to a stage where you can send it to the market. It often takes several Sprints before it becomes even tangible. (…) The mantra saying that everything needs to be ready – It typically requires too much integration."*

BREAKING TASKS INTO SMALLER ITEMS IN THE SPRINT BACKLOGS

Both interviewees mention the process of breaking development tasks into smaller items for the Sprint Backlog as being a challenge. But, possibly due to their different roles in the Scrum framework, they put a different emphasis on this matter. While Interviewee A1 is a *Product Owner* and primarily has a background within software environments, Interviewee A2 is a *Developer* and works with practical issues in regard to hardware and mechanics. However, both interviewees agree that this aspect of Scrum is a challenge.

Interviewee A1 reflects about the challenge and how to handle it: *"I have one view on this, but I am not sure that a hardware engineer would have the same [laughing]. But what I am currently thinking about – and it is not perfect yet – is to use the block diagram as a driver in this, breaking the product into smaller elements, whether it is software or hardware. Then start by developing the most critical or important parts that cannot be simulated."*

> *Interviewee A2: "It is a challenge to split up the work in a way that makes sense, so that the individual parts become developed to a sufficient level for you to proceed to something else. This is a challenge."*

MAINTAINING MOTIVATION IN LARGELY CROSS-FUNCTIONAL TEAMS

Scrum promotes cross-functional projects. In software development this typically means gathering a number of different competencies within software engineering and peripheral competencies such as UX experts and testers. Even though everyone on such a team has different professional profiles, they still have software as a common ground. In a team integrating software, firmware, hardware and mechanics, people are radically different and do not necessarily speak a common language. According to Interviewee A1 is this a problem in regard to team commitment in largely cross-functional teams.

> *Interviewee A1: "At the Daily Scrum meeting it can be a challenge that someone doesn't understand why he have to listen to what everyone else in the team is doing, when they are so specialised in different areas as they are. (…) Cross-functional projects – when we include hardware, we have even more cross-functional projects. There are some issues there."*

BALANCING SHORT TERM- AND LONG TERM DEVELOPMENT ROAD MAP

While development in short sprint cycles enables the developers to always focus on the immediate and most important tasks at a given moment, a common understanding about the long term development plans is vital to successful product development with Scrum. However, it can be a challenge to balance the level of detail in which the future Sprints are described and to accept the uncertainty of the long-term plan as it is only vaguely described and only represents "the best guess".

Interviewee A1 emphasises the importance of a product vision and argues that its absence in the Scrum framework is one of the main weak spots. In the quote below, he tells about the difficulties his team had when they had to start up a new project without a formal product vision:

"We need to have an idea about what is going to happen six moths into the future. Only an idea – we must not by any means go into details on it, but we need to be able to communicate it in order for people to make the right decisions today. We need to envision the road, and this is why the vision is enormously important. In my opinion it is a big mistake that the product vision is not an artefact in the Scrum Guide. (...) We had huge problems in the beginning of this project. The first thing we discussed was the choice of DCP for the new platform, and I couldn't get an answer from anyone. At some point one of the hardware engineers asked, 'What kind of products are we making? We have been asked to make a new platform, but no one has said anything about the products'. The business managers then got busy developing a vision for the products, and the decisions got much easier after this."

BASIC CASE DATA

COMPANY B

CASE **B**

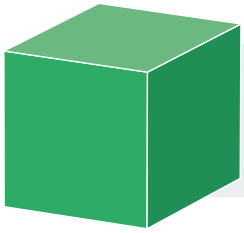
THE COMPANY

OWNERSHIP:
PUBLIC COMPANY
PRIMARILY OWNED BY

LOCATION:
DENMARK

1100 IN DENMARK

PRODUCTS & MARKETS



Components within the mobile hydraulic business.

OEM clients in the markets of agriculture, construction and material handling.

PRODUCTS & MARKETS

One group has just started testing Scrum in a project

LENGTH OF SCRUM EXPERIENCE

Scrum is conduted under the radar and to the management's concern.

IMPACT IN ORGANISATION

2-week Sprints have so far been used in the test project

SPRINT LENGTH IN WEEKS

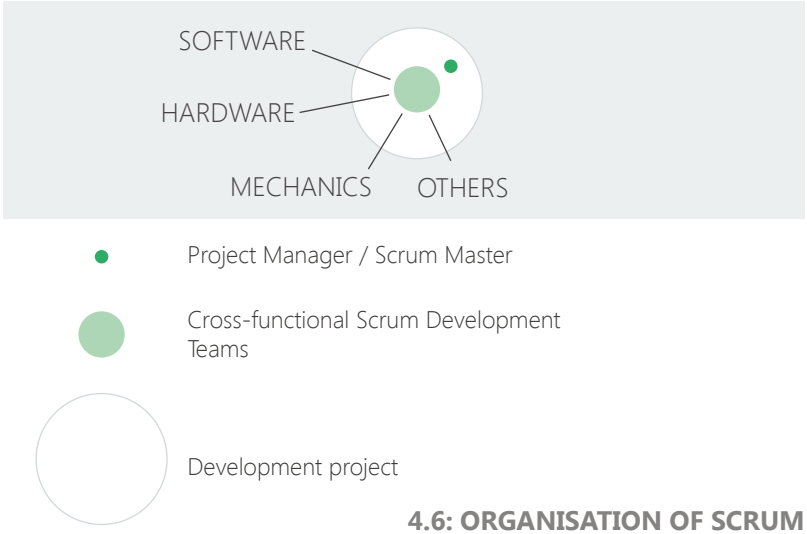
INTERVIEWEES

NO.1

Name Interviewee B1
Position Project Coordinator
Education Software engineer

Role in Relation to Scrum
Scrum Coach for the Project manager, but no actual Scrum role.

4.5: BASIC CASE DATA



4.6: ORGANISATION OF SCRUM

4.2.1 ORGANISATIONAL STRUCTURE

Company B is divided into three divisions: *Work Function*, *Propel* and *Controls*. Together, these three divisions form the portfolio of products marketed to major OEM customers within primarily agriculture and construction industries. This case concentrates on one single Scrum team in the Controls division.

The company is highly committed to its traditions of having development activities and production activities closely together. The operation of the production is having the highest priority, which means that development activities are conducted whenever there's time for it. This, of cause, influences the intensity and lifespan of the projects running:

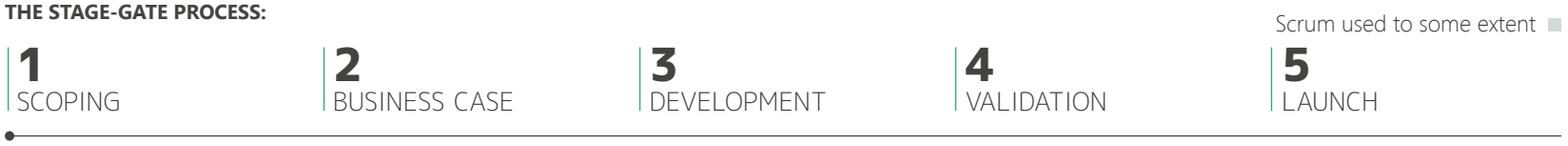
> *Interviewee B1: "To us development activities are always secondary in comparison to maintaining the operation of the production. We shall never avoid situations, where a developer is pulled from a team because he needs to solve problems in the production. (...) I would like to change this to a situation, where some focus on fire fighting and some focus their efforts on development."*

The development department is organised as a traditional matrix with team leaders managing the employees with specific professional competencies such as electronics or mechanics, and project managers conducting the development projects with a number of associated employees.

4.2.2 SCRUM IN THE DEVELOPMENT PROCESS

Company B is using its own version of the Stage-Gate process model with a number of stages that the development teams are bound to follow. The company is rich in traditions, and the implementation of Scrum has been met with concerns, especially in higher-level management. Scrum is therefore implemented in only one single project, which is currently conducted as a pilot project. Because of the often very long life span of projects in the company, Scrum has been implemented midstream in an on-going project.

According to Interviewee B1, Scrum is implemented in the product development phase and concerns development and maturation activities. Concept development activities are not yet conducted with Scrum, but the interviewee doesn't see any impediments for doing so. Because of the relatively short experience with Scrum the focus, at the present state, is on the development activities.



4.7: SCRUM IN THE DEVELOPMENT PROCESS

4.2.3 MOTIVATION AND TRANSITION TO SCRUM

Scrum is implemented in an on-going project as an experiment. The project manager was introduced to Scrum elsewhere and found it interesting in relation to the project. The initiative causes concern in higher-level management: *"Are we now throwing away all our common virtues and best practices?"* However, the experiment is started cautiously with one single team.

The main and overall motivation for implementing Scrum is to improve development efficiency, and this is further specified in two separate motives. The first of these motives is related to and increased focus on resource management and development time:

> *Interviewee B1: "In principle, it doesn't matter whether a project requires 400 or 4000 hours [human resources spent in a project are not considered in the business case]. The money is in the final product. But especially delivery time is interesting – we cannot have the customer waiting several years for something he has been promised."*

The second motive for implementing Scrum is related to the character of the projects:

> *Interviewee B1: "... The complexity of the projects has now become so critical that we cannot predict them. Scrum is pointless if you have something predictable."*

This second motive corresponds very well with one of the three fundamental challenges described earlier in this thesis.

Motivations:

- Project coordinator: Prove that Scrum works
- Project manager: Efficiency improvement
- Clarity on the consumption of resources
- Shorter development time, due to customer satisfaction
- High product complexity and lack of predictability

Initiative:

- Bottom-up (Project manger's initiative – higher level management concerned)

4.2.4 SCRUM FRAMEWORK #1: PRODUCT VISION AND PRODUCT BACKLOGS EFFORTS

The Scrum-driven project at Company B also includes a Product Vision. Even though it is not an official part of the Scrum framework according to Schwaber & Sutherland (2011), it is an additional artefact often implemented in Scrum-projects. The Product vision in the present project includes a business case, market information, expected revenue and information about the basic purpose of the product in development. As the Scrum framework is introduced midstream in a project, the information in the product vision is inherited from the original business case and project specification.

The level of detail in the Product Vision is high. It includes all technical specifications; but according to Interviewee B1, the largest task has been to rearrange and modify the development efforts originally organised in a Gantt chart. While the project specification has originally been developed with a phased and platform-oriented waterfall model in mind, the Product Backlog in Scrum prescribes a division of the development efforts into functional sub-parts.

4.8: FROM PLATFORMNING TO FUNCTIONAL FEATURES

> *Interviewee B1: "It is no problem to keep a task estimated to 200 hours in a Gantt chart, but you cannot do that in Scrum. It would exceed one Sprint, and we don't want that. The largest task has therefore been to change the focus on platforms to a new focus on functions"*

The actual breakdown of the specification to smaller tasks has been done in two iterations. The project manager and the team leaders representing the respective professional competencies have carried out the first iteration of the backlog grooming. The employees in the actual development team are doing the further detailed grooming of the backlog. At Company B, the final tasks carried out in the Sprints are not allowed to exceed an estimation of 25 hours.

The detailed grooming of the Product Backlog is only carried out one or two Sprints ahead. According to Interviewee B1, a highly detailed project plan in a Gantt chart that reaches several months ahead in time, only means a false sense of security.

> *Interviewee B1: "When you have a Gantt chart for a project running over two years, how can you be sure that a certain task will take place exactly the 5th of February the next year? Many things can happen in-between. Excuse me, but that level of detail is just stupid."*

4.2.5 SCRUM FRAMEWORK #2: SPRINT CYCLE

The development team conducts the development in two-week Sprints, and the Sprint backlog includes tasks related to hardware, software, and mechanics. According to Interviewee B1, it is important to develop all those aspects at the same time in each Sprint. In the planning of a Sprint, the external dependencies are taken into account and fitted into the plan. The development work progresses with Daily Scrum meetings, and follow-ups on sub deliveries from external parties are counted-in and estimated as any other tasks. However, the team cannot be held accountable for delays at subcontractors.

The result of a Sprint is reviewed at Sprint Review meetings, but the Retrospective meeting has not yet been implemented into the Scrum practice at the time of the interview. Instead of focusing on actual prototypes at the Sprint Review meeting, the team focuses on reaching their *Definition of Done* for all the accomplished tasks.

> *Interviewee B1: "The 'Definition of Done' has become the centre of attention, and we are starting to discuss how we can define it. And when is something delivered? We see a deliverable as, for instance, a document, a business case, or a prototype. It is not necessarily working software."*

A great effort is made to develop a thorough project specification in the period preceding the actual development activities. The contractual work with customers is carefully made, which means that change in the project specification, or, in this case, the Product Backlog rarely happens.

> *Interviewee B1: "The contract is an enormous thing here. It can easily take just as much time to renegotiate it as it takes to finish the product."*

Despite the contract, the team *has* made revisions of the Product Backlog after each of the already completed Sprints. The changes made to the backlog have had the character of internal changes such as change in sub-suppliers and the like, and has not been initiated as a result of changes coming from the customer.

According to Interviewee B1, the underlying Stage-Gate process model in general hinders flexibility and changes in the Product backlog, because the customer requirements and the business case are locked at an early stage. However, internal changes occur, and when asked about the risk of lost work as a result of this, Interviewee B1 mentions that *"the focus we have gained from gathering the team has resulted in an enormous improvement in efficiency."*

As a closing remark about the development flexibility in the Sprint

cycle, Interviewee B1 recalls an earlier Scrum project from another job, where contracts were made per Sprint cycle:

> *Interviewee B1: "After Sprint 1 we signed the contract for Sprint 3 and decided the content of Sprint 2 – and that was the way it worked. The customer was very happy with this in the end. The product had changed a lot from what he originally imagined, but in return he had got all the most highly prioritised features as they had gradually changed.*
>
> *I have great experience with this model, but it only works if you are close to the customer. Here we are not that close to the customer."*

4.2.6 DESIGN AND STYLING EFFORTS

In general the products developed at Company B are of a highly technical character. Styling and design efforts are given very low priority, as it is the technical features and quality of the products that count. Most often the products are included as technical necessities in larger systems like construction system. However, products like valves are also operated manually, so *some* user requirements are described, and *some* tests are made. According to Interviewee B1, the customers, who are often large and specialised companies, define these requirements:

Interviewee B1: *"our customers are very skilled in living up to the user expectations, and the user expectations also change rather slowly here [in this industry]. When the customer pulls the lever, the crane should move. It is conceptually simple."*

4.2.7 MAIN CHALLENGES EXPERIENCED BY INTERVIEWEES

When asked what he considers the main challenges of using Scrum in the product development, Interviewee B1 mentions a number of issues. These issues are described below.

BREAKDOWN OF TASKS (CHANGE IN ATTITUDE)

Breaking down a traditional project specification into work packages that can be fitted into the duration of a single Sprint, and also the breakdown of these packages into tasks of maximum 25 hours of work. It requires a radically different way of thinking the composition of the product – from building products in platforms to building individual functions across those platforms. According to Interviewee B1 it requires a change in attitude:

> *Interviewee B1: "Software developers are used to iterative development. Typically hardware developers and mechanical developers are not used to that – they often build up the products in layers from the bottom and up. The biggest challenge that I have noticed is definitely the breakdown of tasks to deliverables that can be fitted into two- or four-week Sprints. It is a change in attitude rather than a technical challenge."*

ESTABLISHING FOCUS AND TRANQUILLITY AROUND DEVELOPMENT

Due to the strong tradition of everyone helping out in the production and the fact that product development activities are considered second priority, the necessary focus and commitment to development is often not present.

> *Interviewee B1: "I would like to change this to a situation where some focus on fire fighting and some focus their efforts on development. We may not be able to protect everyone, but at least tip the balance, so that we can minimise the fire fighting activities for some of our resources."*

MAINTAINING THE DISCIPLINE

Discipline is also mentioned as one of the challenges that complicate the implementation of the Scrum framework into a traditional development environment.

> *Interviewee B1: "Discipline is definitely a challenge. (...) You start out by saying that we are going to do this, but eventually we fall back into our habits. (...) It is the human aspect in it. It's the habits you need to change. It takes time."*

SUPPORT FROM STAKEHOLDERS (HIGHER LEVEL MANAGEMENT IS CONCERN – LOSS OF VIRTUES)

The last of the mentioned challenges is the lack of support from stakeholders such as the higher-level management. It can be difficult for a development team to accept large changes forced upon them by their management; but it can be just as difficult to initiate a process of change if the management shows reluctance.

> *Interviewee B1: "I have not yet experienced the team being the problem. It is typically the management or the organisation that has to be convinced. Support from the stakeholders – that is often where the problem lies."*

BASIC CASE DATA

COMPANY C

CASE **C**

THE COMPANY

OWNERSHIP:
UNDISCLOSED

LOCATION:
DENMARK

150 IN DENMARK

PRODUCTS & MARKETS

Computer-based audiological equipment

The equipment is used by professional audiologists on a worldwide market

PRODUCTS & MARKETS

10 MONTHS

The Scrum teams have been running for almost a year.

LENGTH OF SCRUM EXPERIENCE

Scrum has shown its benefits, and the physical organsation has been change to fit Scrum

IMPACT IN ORGANISATION

The teams conduct monthly Sprint Reviews

SPRINT LENGTH IN WEEKS

INTERVIEWEES

NO.1

Name Interviewee C1
Position Firmware Developer
Education Software engineer

Role in Relation to Scrum
Scrum Master in the hardware/firmware Scrum development team.

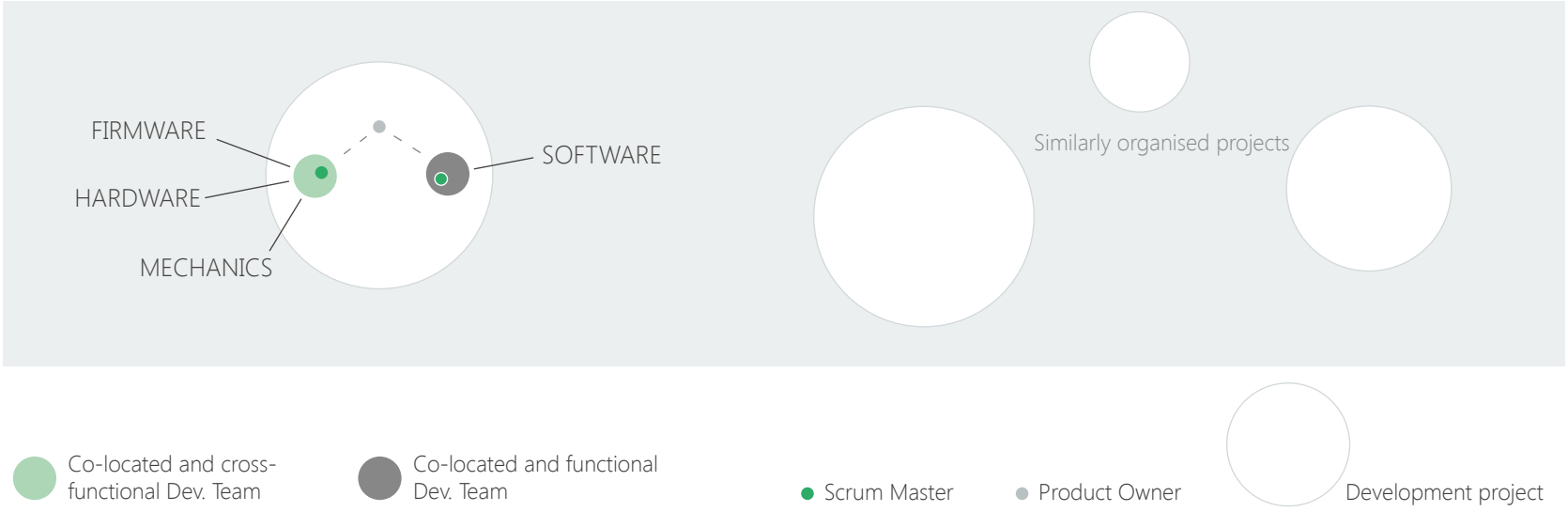
NO.2

Name Interviewee C2
Position Hardware Developer
Education Unknown

Role in Relation to Scrum
Developer in the hardware/firmware Scrum Development Team with focus on hardware.

4.9: BASIC CASE DATA

4.10: ORGANISATION OF SCRUM



4.3.1 ORGANISATIONAL STRUCTURE

The development department at Company C is organised in "super teams" carrying out the development activities with the use of Scrum. Earlier, when the Scrum framework was being implemented, the Scrum development teams were dispersed throughout the organisation as the developers were physically placed in groups by professional competence. A reorganisation of the development department within the last year has gathered the developers in cross-functional super teams working on the same project. The current development project is carried out with one Product Owner and two Development Teams – each with a separate Scrum Master. One Development Team is focusing on software development and the other on hardware, firmware and mechanical development. All developers are physically located within the same open-plan office.

Other Scrum teams are working on other projects, but are physically located in other areas of the organisation. It is primarily development activities that are carried out in the Danish department of Company C, but also product calibration, testing, and the last part of assembly is conducted at the site. Manufacturing and the major part of assembly take place abroad.

4.3.2 SCRUM IN THE DEVELOPMENT PROCESS

Company C uses a traditional phased process model as the overall management tool for development activities. However, the model is not being used on daily basis as the Scrum framework has taken over as the primary means for coordination within the development team. The dissociation from the model is apparent in the following quote from Interviewee C1:

> *Interviewee C1: "We had a development model – and we still have it running in the background. It is the classic one with definition phases and milestones and gates. Nevertheless, we were busy running around supporting old product all the time, even though it was never really pointed out to anyone."*

Scrum is not utilised in all types of activities in the phased model, but centres around actual development activities and production maturing. Activities preceding the actual development phase are not executed with Scrum, and most often market-related employees as, for instance, business managers carry out the early activities.

4.3.3 MOTIVATION AND TRANSITION TO SCRUM

At first Scrum was tentatively implemented in the software development department as a pilot project. The organisation was struggling with delayed software projects and lacked efficient handling of the feature request in the later stages of those projects. According to Interviewee C2, part of the reason for implementing Scrum in the software department was the need for transparency – both in regard to how resources were spent and to how much time the developers actually had to solve development tasks.

Scrum was subsequently implemented across all disciplines in order to broaden the improvements experienced in the software department. This was a management decision and it was met with mixed feelings:

> *Interviewee C1: "It was forced upon us from the management. (...) Why is it we do this and for what benefit? We still have people in the organisation holding on to the attitude 'I know what I have to do'."*

Interviewee C2: *"Scrum was presented to us as a way to break down tasks and a procedure where you could take over task from each other; but we had difficulties with this. To me this required a change in mentality in order to be able to see how we could benefit from this."*

The quotations above show the initial reluctance towards implementing Scrum, but this attitude seems to have changed, as the interviewees are able to list numerous benefits resulting from the use of Scrum:

> *Interviewee C1: "We are clearly having fewer digressions both in and outside the project. You are being held accountable for your work on daily basis. (...) It helps maintain one focus, and your employees will be much more committed."*

> *Interviewee C2: "I also think it is easier to handle the shorter time frames. It is a bit clearer, and you'll get less frustrated about all the things you know we have to solve, but yet don't have specific solutions for."*

Motivations:

- Software projects were too long and missing important features
- Anarchy in software department (no clear priorities)
- Resource management needed

Initiative

- Top-down (Management decides to implement scrum as a pilot in SW. Later broad implementation in super teams)

4.3.4 SCRUM FRAMEWORK #1: PRODUCT VISION AND PRODUCT BACKLOGS EFFORTS

The team is not actively using a product vision as a leading star, as it is sometimes seen in other contexts. The interviewees are not fully explicit about this, but according to Interviewee C2, the product vision is at least not precisely described in the project:

> *Interviewee C2: "The Product vision only tells that we are developing some sort of device"*

Interviewee C1 further argues that the Product Vision is present, but that it is rather down-to-earth, as it describes the product features, which do not change that much. Based on the conversation about the Product vision, it is unclear whether or not it is actively used or possibly mixed together with the Product Backlog, as some of the statements could indicate.

As for the Product backlog, the interviewees present a perhaps atypical way of handling it. As Interviewee C1 rightly points out, the formal Product Owner according to the Scrum framework should maintain the Product backlog. In this Scrum team, the Product backlog is handled differently:

> *Interviewee C1: "(...) It has not been the case here. It has been the team itself that has taken the responsibility for it [the Product Backlog]. He [the Product Owner] has given some inputs to the way we are heading and to when in the process we need to have a milestone. (...) We have almost fallen back to the old Waterfall model, as we are breaking-up everything from A to Z into stories in the beginning, rather than only looking a bit forward and planning this in detail. It [the Product backlog] now exists as a relatively untrimmed and weak part. When planning a Sprint, we only partly take the Product backlog into consideration, as we are primarily looking at the actual Sprint goal."*

It is clear that the team is utilising the Product backlog differently from what the Scrum Guide prescribes, and the interviewees – both being part of the Development Team – are aware of it. However, the Product Backlog still includes a thorough description of the product, and the interviewees indicate that it is sorted in a certain hierarchy of epics and stories. The awareness of the special use of a Product Backlog as well as the existence of a Product backlog hierarchy indicates that the team has a rather deep understanding of their own way of using Scrum and the limitation of this procedure.

Interviewee C2 also comments the arguably missing dynamics in the interplay between the Product backlog and the Sprint backlog:

> *Interviewee C2: "The Sprint backlog is almost a given thing from a hardware perspective as it depends very mush on the phases we know we have to go through, the prototypes we have to make, the first revisions, and so on. In that way, that items we bring into the Sprint backlog is a natural consequence of what we should do at a given time. Therefore we seldom change the priorities and the order in which we carry out the work. You simply have to do it in a certain order."*

The quote from Interviewee C2 describes some limitations of working with Scrum in relation to physical products. Firstly, most activities need to be performed in a certain order, as it is basically the only way it makes sense. Secondly, the dominant Stage-Gate process model, which is in fact implemented in all cases in this thesis, dictates a certain sequence of the development tasks.

4.3.5 SCRUM FRAMEWORK #2: SPRINT CYCLE

At Company C the Sprint cycles are conducted in synchronisation with the calendar months, culminating in the Sprint Review and Retrospective meetings on the last Thursday in each month. This particular rhythm is a legacy from before Scrum was implemented and allows for the stakeholders to always know when the Scrum team presents its work.

The first day or two of the Sprint is used on planning the Sprint. The Product Owner presents some inputs that help the Development Team to establish an overall Sprint goal, which has the purpose of establishing a certain focus throughout the Sprint. Together, the Scrum team select the primary stories for the Sprint Backlog. After these primary stories are agreed on, the team splits up into smaller competence-specific groups, such as hardware, firmware and mechanical developers. In each of the small groups they further break down the stories to tasks and finally meet with the full development team to

sort out and coordinate internal dependencies. At this point external dependencies, such as deliveries to or from the Industrial Designer and others who are loosely attached to the development team, are also considered. According to Interviewee C1, this process goes fairly smoothly because of the close physical organisation of the team.

> *Interviewee C1: "If we didn't sit this closely together, we would probably be more aware that we are interdependent."*

After the initial planning phase in the Sprint, the development routine runs almost continuously throughout the Sprint. Scrum meetings are held on daily basis and the work is only interrupted by possible production shutdowns that require action from the engineers in the development team.

As shown in figure 4.10 presented earlier, the hardware and software teams – led by the same Product Owner – are only synchronising the work in an informal manner, which according to Interviewee C1 is possible as the teams are physically situated closely to each other.

As mentioned the Sprint Review and Retrospective are scheduled for the last Thursday in each month. Stakeholders such as Business managers, Product Owners, and CEOs often show up to this "Show & Tell" event. The results of a Sprint can be of very different character:

> *Interviewee C1: "As for the mechanical developers, they may pull up a model from the box. (…) It can also be a document or simply sketches from a whiteboard. It doesn't need to by something physical."*

> *Interviewee C2: "It just needs to illustrate the work and communicate what you have done and maybe the output from it."*

The result of the Sprint is presented and discussed in the group present. The Development Team and the Product Owner use a Done-terminology to evaluate the maturity of the work, but have a slightly different way to do this:

> *Interviewee C1: "Our project leader [Product Owner] uses the concept of 'Done'. He uses 'Done-Done' to ensure that nothing needs to be revisited. We [the Development Team] don't use that. We just use 'Done', which means that we have closed the story, but we may possibly return to it and open it again"*

> *Interviewee C2: "The verification of what we have made is not necessarily total. It can be very difficult to make a Done-done. (…) There is a great chance that you have to get all the way back to the root and change something that you thought was closed."*

> *Interviewee C1: "It is working prototypes"*

4.3.6 DESIGN AND STYLING EFFORTS

The word *Design* has many meanings – also at Company C. Each professional genre has its own understanding of the word. In this subchapter design is used in a few different ways.

An external Industrial Designer works with the development team on the development of the product. This person is primarily linked to the sub-group that carries out the mechanical development and delivers the visual expression of the product as well as the mechanical concept. The mechanical development sub-team is the least active in regard to the Scrum activities as they only seldom supplement the backlog with stories or tasks. This also means that the Industrial Designer has a role, which is not very visible for the rest of the Development Team. Other aspects of design and styling are the usability issues and graphical user interface. This is managed by the Software team and is not directly involving the hardware/firmware/mechanics team.

Even though the Industrial Designer is only loosely attached to the development team, the interviewees argue that the trend goes towards more product iterations and an increased focus on the mechanical design in general. This, however, is not without difficulties in the market, in which Company C operates:

Interviewee C2: "The market is very slow and conservative in our opinion."

Interviewee C1: "The market is driven by standard tests, and they don't just change over night – there's 10 years between [changes in standards]."

Another place, where the term, design, is also relevant, is in relation with the group of hardware developers. Even though this does not represent the perspective in which design is focused on in this thesis, it is brought forward as it may raise some interesting issues in regard to physical development and Scrum.

Interviewee C2: "In regard to hardware, the design activity lies early in the development phase, and you don't just go back and change a design – not fundamentally. You don't just change horses midstream. Instead of working in small circles [refers to illustration on poster 3], we try to create a big circle around it all to make some outer boundaries – like a platform"

Interviewer: "Do you then prepare the platform for a second generation of the product?"

Interviewee C1: "Not really. We choose a broad enough platform to be used for other products as well, but we don't want to design for tomorrow, as we also know that what happens tomorrow is different from what is needed today."

This dialogue opens up the paradox of unpredictable change and shows that the development team is aware of uncertainties that are deeply rooted in development of products for moving markets.

4.3.7 MAIN CHALLENGES EXPERIENCED BY INTERVIEWEES

When asked what they consider the main challenges of using Scrum in the product development, the interviewees mention a number of issues. These issues are described below.

SELLING A PRODUCT THAT ISN'T THERE

According to Interviewee C1, the biggest challenge is to *sell* an item that does not have any substance:

Interviewee C1: "It is difficult to sell a document. It is not that sexy. It is also difficult to sell a measuring of sound in a box. Those people [the stakeholders] may not be that familiar with the technical issues. They would rather see something that 'does' something."

Interviewee C1 raises the issue about non-physical deliveries at the Sprint review meetings, which may be difficult to communicate.

BREAKING DOWN TASKS

The team has experienced challenges in breaking down the tasks of 100 hours to smaller tasks of maximum 12 hours.

ANTIBODIES IN THE ORGANISATION AND SHARED UNDERSTANDING

According to Interviewee C1, the organisation still struggles with employees who stick to their old approach and attitude towards development tasks and to their personal professional domains. Interviewee C2 elaborates on this:

Interviewee C2: "In the beginning it was fundamentally about establishing a common understanding for why we were doing this. Scrum can be interpreted in many different ways."

STICKING TO SCRUM

It can be a challenge to preserve Scrum as the primary development framework in the development environment. Interviewee C2 argues that there is still a risk of Scrum being dropped:

> *Interviewee C2: "I think it is a challenge to stick to Scrum. It was forced upon us, and now we do it in some sort of watered-down way, which is ours. We do it because we think it works a bit, but if the people who, in the first place, forced us to it lose interest, then I think it will be difficult to preserve it – and to improve it. If we don't always correct ourselves and make corrective actions; then, what is our benefit when we are done?"*

> *Interviewee C1: "Hopefully there are some customers in the end of each sprint who will make it all worth it. I don't think we'll let go of our daily sprint meetings, either."*

DISTURBANCES

Just as it is seen in other cases, the disturbance related to the production, such as failure in maintaining and servicing the operation, is a challenge that prevents efficient development. This has been a large problem earlier and still is to a certain extent, but according to Interviewee C2 the establishment of a Technical Investigation Group (TIG) has reduced the problem. TIG handles most of the production issues and functions as a hotline in other matters.

BASIC CASE DATA

COMPANY D

CASE **D**

THE COMPANY

OWNERSHIP:
PART OF GLOBAL ORGANISATION

LOCATION:
DENMARK

1000 IN DENMARK

PRODUCTS & MARKETS

Emergency power units and software applications for design and management of server parks and data centres.

Products are sold to a broad span of markets, such as energy infrastructure, Industry and data centres.

PRODUCTS & MARKETS

3 YEARS

Relatively long experience compared to the other cases

LENGTH OF SCRUM EXPERIENCE

HIGH

Scrum is part of the development culture in the local division, but is below radar in the global concern

IMPACT IN ORGANISATION

4

4 weeks are the typical Sprint lenght. However, Firmware is using a 2-week cycle.

SPRINT LENGTH IN WEEKS

INTERVIEWEES

NO.1

Name Interviewee D1
Position Team Leader
Education Electronics engineer

Role in Relation to Scrum
Scrum master, Firmware (+ Sharing hardware)

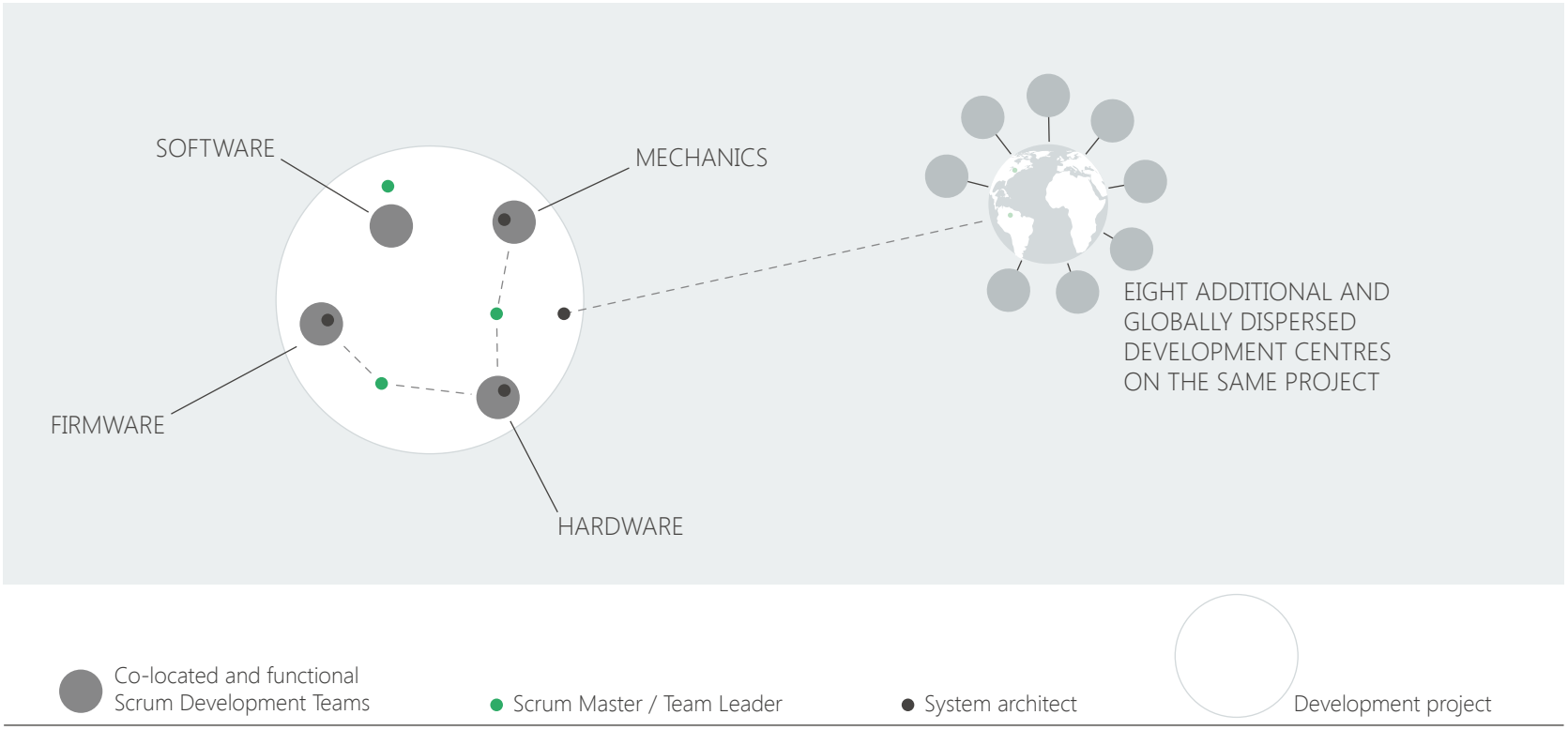
NO.2

Name Interviewee D2
Position Team Leader
Education Electronics engineer

Role in Relation to Scrum
Scrum Master, Mechanics (+ Sharing hardware)

4.12: BASIC CASE DATA

4.13: ORGANISATION OF SCRUM



4.4.1 ORGANISATIONAL STRUCTURE

Company D is partly responsible for the development of the APC-branded Uninterruptible Power Supply products, which spans from solutions used in large data centres to private home systems. The Development department in Kolding is organised as a project organisation with several parallel projects in development, and this case concerns a single one of these projects. This specific development project employs a large span of developers with competences including software, firmware, hardware and mechanics on full time. In addition to the project team a number of employees with other competencies are loosely associated on-site and nine collaborating development centres are scattered all over the world at other Company D locations.

Locally, the development environment of this current project is organised with software development activities in a separate department and the firmware-, hardware-, and mechanics development activities brought together in another department. The Software department had been using Scrum for some time before the other department decided to implement the Scrum practice three years ago. This cases focuses on the Scrum efforts of the cross-functional department, which is divided into three functional development teams led by two Scrum Masters and three system architects.

4.4.2 SCRUM IN THE DEVELOPMENT PROCESS

Company D is employing 130.000 people worldwide and the large company has its own version of the Stage-Gate process model called PMP. All development projects are bound to use this process mod-

el at top level. Nevertheless the present development project uses Scrum as a supplementary process control framework. Interviewee D1 comments on the relationship between the two *process* models:

> *Interviewee D1: "We are a gigantic organisation, and on paper we are a waterfall organisation using a PMP process, which is our own Stage-Gate model. But under the radar we do whatever we can to be agile in this team. (...) They [Stage-Gate and Scrum] don't fit together at all. Scrum is running out of sync with our Stage-Gate."*

Interviewee D1 furthermore argues that the Stage-Gate model on the daily basis is rather distant, due to the absence of the Program Manager.

> *Interviewee D1: "The gates are not that important during the development process. The only gate that is enormously important is the last one. That is when we release to production. That gate is important for everyone in our teams.*

When asked in which phases of the product development Scrum is utilised, the two interviewees discuss whether or not it is used for concept development:

> *Interviewee D2: "I don't think will ever use Scrum for the early market research, but we COULD perhaps use Scrum in the concept development."*

> *Interviewee D1: "I think we use other processes for that"*

> *Interviewer: "Is it because you think that the concept development process is less controllable?"*

> *Interviewee D1: "Yes, and less formal, I think. Scrum is a very strict process and that is not what we need for concept development. There you need a rather loose framework, and then it is about timing. It is about maintaining the cadence in both concept development and Scrum. I think that is what results in a robustness in the team – You don't have time to let down your guards and become unfocused."*

4.4.3 MOTIVATION AND TRANSITION TO SCRUM

The transition to Scrum has proceeded in several steps during a period of a couple of years. The Software development department used Scrum with great success even before Company D acquired the company. The new management demanded a better accuracy and performance in the estimation and execution of development projects, and at the same time the Firmware department was struggling with maintaining an efficient development flow. Both things led to the adoption of Scrum, which has been successfully improving the performance of the Software department. The Hardware and Mechanics departments later went along.

4.14: SCRUM IN THE DEVELOPMENT PROCESS

Interviewee D1: "We turned our coding practises upside down. Our processes are very different from earlier projects, and hardware and mechanics just went along with what the firmware team wanted. It was initiated by a desire for being more predictable."

He adds that the introduction of Scrum was not only a desire from the development team, but also a necessity, because of a demand from the new management:

Interviewee D1: "Our new vice president demanded that our development projects hit within a 15% margin of their development plans. This forced us to find another way than the traditional waterfall planning in order to make sure that we delivered the products on time."

Motivations:

- Software team had been using Scrum with succes
- Firmware needed better ways for planning and communicating

Initiative:

- Bottom-Up (an urge for change)+ top-down (Demanding better accuracy in project time estimation and execution)

4.4.4 SCRUM FRAMEWORK #1: PRODUCT VISION AND PRODUCT BACKLOGS EFFORTS

Even though the department at Company D has been conducting Scrum for three years, there are still some fundamental obstacles in the organisational legacy that hinder the Scrum process in running without problems. One of the major aspects is the Product Backlog effort, which is evidently imposed directly on the development team and the Team Leaders, instead of an actual Product Owner or the Line Manager.

Interviewee D1: "The difficult part is when we define the product in the Product Backlog. (…) The biggest challenge is that we [the two Scrum Masters], together with our system architects, are the only interface to marketing, and we receive all inputs, have all the battles, and evidently decide what can actually be made. (…) There have been a lot of battles, and they have taken a huge amount of time – at that point it would have been nice if an actual Product Owner would have made those decisions."

Interviewee D1 continues to argue that because of the size of the organisation and the recent merger, the management and empowerment structures are too complicated and dispersed. This has resulted in a highly difficult project start-up with no clear Product Vision. Eventually, the Scrum Masters and system architects came up with a detailed Product Backlog, which included not less than 1800 requirements about *what* the product had to include and even more requirements about *how* to include it. A software tool manages all requirements due to the vast amount and the high project complexity. The transition from using a traditional Scrum board to using various software tools has taken place along with the development teams getting used to Scrum.

Interviewee D1: "We don't use the Scrum board that much any longer. It was extremely important to us in the beginning. We had to to learn the basics of Scrum, so for the first one-and-a-half year we dedicatedly used the board and Post-its. Now it is all managed by our software tool."

However, the Scrum boards still play a certain role, as the Product Backlog is still visible on it. The Development Teams and the Scrum Masters gradually detail the backlog items so that instantly pending items have a high level of detail.

Interviewee D1: "The goals immediately ahead of us have a high level of detail and the ones lying further ahead are still only described on an overall level. We try to adjust it as we progress through the project. The goals close to us are pretty clear to everyone, and then we have some holistic issues, which force us to

take action now, so that we can make something happen later in the project."

It is also in the planning phase that time estimates are put on the various tasks. At Company D single tasks are thoroughly estimated, but according to Interviewee D1, it can be difficult to estimate the hours exactly in development environments:

Interviewee D1: "Some of the concept development takes place within the project, as it sometimes happens that you are having a challenge, which just cannot be solved the way you expected. Then it is back to the drawing board. We have a lot of examples of tasks that have been estimated with extreme inaccuracy, because you are actually working on concept level and really don't have a clue about how long it will take. We can easily see it on the count afterwards – it is the stuff that we don't know about that makes our overall calculations go wrong."

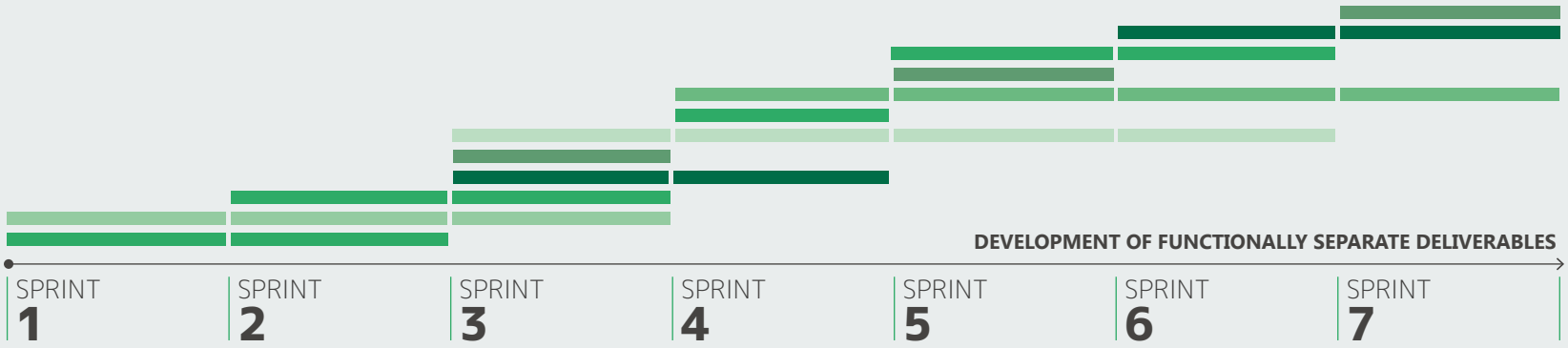
4.4.5 SCRUM FRAMEWORK #2: SPRINT CYCLE

All three Development Teams conduct the Sprints in close synchronisation. The duration of the Sprints is four weeks, except for the Firmware team, which has split-up the cycle into two separate two-week Sprints due to the special character of their tasks. Interviewee D1 describes the details in the quotation below:

Interviewee D1: "Firmware runs two-week Sprint sessions and Hardware and Mechanics runs four-week Sprints. However, the Firmware team only have Sprint Reviews and Retrospectives every fourth week. It's just like a small stopover, where the team replans the tasks. The reason why it has become like this is that the tasks of this team are highly dynamic, and it is a very big team. They are eleven developers. It was simply too may small tasks to plan for a four-week period."

Every fourth week one day is used for Sprint Review, Retrospective and Sprint Planning. The Sprint Planning is, to a great extent, managed by the Development Teams themselves with some initial inputs from the Team Leaders.

Interviewee D1: "What we put into the Sprint from the Product Backlog is defined by what we need to be able to do at a certain time, and the teams are extremely involved in deciding this. (...) We [the Team Leaders] inform our system architects [one in each team] about the top priorities for the nearby future, and then the planning happens autonomously in the team with a focus on planning tasks and issues in detail."



4.15: FUNCTIONALLY SEPARATE DELIVERABLES DEVELOPED THROUGH SEVERAL SPRINTS

During the Sprint all teams have a Daily Scrum of 15 minutes with each developer presenting his work according to the three Scrum questions in 90 seconds. As the three Development Teams share only two Scrum Masters, the Firmware team conducts the Daily Scrum at 8:30 in the morning with Scrum Master 1, Hardware at 8:45 with both Scrum Master 1 and 2, and Mechanics at 9:00 with Scrum Master 2. This arrangement allows communication about important cross-team development issues, as both Scrum Masters are present at the daily Hardware meeting.

The Sprint goes on for four weeks with only the Firmware team making the short stopover to re-plan their tasks. After the four weeks, one day is again used for reviewing and planning. The Sprint Review is open for everyone in the organisation, and all Development Teams present their recent work to each other. The deliverables from the just finished Sprint can be presented in many different forms, but they are almost always evaluated against the pass/fail criteria set by the teams. The deliverables are most often results of several Sprints and rarely developed during only the last Sprint. This is shown in Figure 4.15 to the left. The Retrospective is taking place in each team immediately after the Sprint Review.

When asked about his understanding of the product development process, Interviewee D1 describes how the Mechanics team is bound to long development iteration cycles and therefore have difficulties in carrying out *true Scrum*:

> *Interviewee D1: "[Pointing at the hierarchically organised product model on the paper sheet] This is a really good picture of how it works in the hardware-related development environment. You need to have some kind of corner stone to build upon, Sprint after Sprint, and we cannot finish one thing in one Sprint. It is just not possible in mechanics as the iteration cycle, with production included, may last from four to six months before you have the physical package you once started to develop."*

4.4.6 DESIGN AND STYLING EFFORTS

The UPS products from Company D are highly complex and technical products primarily meant for energy infrastructure, Industry, and data centres. They are often used in technical installations, and the visual styling may not be one of the primary selling points in this category of products. Rack standards and legislative requirements are furthermore limiting the design space. However, both styling and the notion of holistic design are present in this development project.

When asked how the developers manage to maintain an overview of the full project and thereby being able to ensure a holistic product design in the end, Interviewee D2 mentions the system architects as key players:

> *Interviewer: "How do you maintain the overall concept if you always focus on a small part here and a small part there? Where does the holistic view fit in?"*
>
> *Interviewee D2: "We have this definition from our marketing about what parameters to live up to. It's about cost and size, power range and so forth. (...) This is broken down to what we can actually make. The system architects and the two of us [the Scrum Masters] have then broken it further down to the 1800+ requirements, and this is actually what is ensuring the holistic version. On daily basis it is the system architects [across the three teams], who need to draw the parallels and maintain the overview."*

It seems that the lack of an actual Product Owner forces the system architects and the Scrum Masters to take over the management of the product overview.

In regard to the styling of the product, it was conducted as a subproject in the Mechanics team with an external company developing the visual concept of the product. However, the concept was later overruled from the top management as part of a process that was to ensure a corporate identity across all the business sectors in the

company. During the project some parts of the industrial design efforts have been drawn into the Sprint cycles as the quotations below indicate:

> *Interviewee D1: "Not all the industrial design has been developed using Scrum. It has been conducted as a sub-project in the project, managed by Henrik. However, the resulting tasks have then been included in the Sprint Backlog"*

> *Interviewee D2: "I have had one guy making the front door of the casing and so on, and I have been using Scrum with him. Everything else has not been included in the Scrum activities."*

The quotes above indicate that the design efforts have not been handled in *one* specific way in the current project. While an external company outside the Scrum team has been in charge of the general visually guiding concept, internal developers have conducted some of the underlying design activities. This may be an intentional strategy, chosen in order to ensure a visually holistic design.

4.4.7 MAIN CHALLENGES EXPERIENCED BY INTERVIEWEES

When asked what they consider the main challenges of using Scrum in their product development, the interviewees mention a number of issues. The issues mentioned are described below.

PRODUCT OWNER EMPOWERMENT

As it has already been mentioned earlier in this case description, the Scrum team has no actual Product Owner. This may not be a problem in all cases, but in this development project at Company D the lack of a Product Owner has resulted in a conflict between responsibility and empowerment of the Team Leaders. Though the two Team Leaders, who are also acting as Scrum Masters in the three development teams, are both handling staff responsibilities and project management responsibilities, they are not formally allowed to make decisions on the product level. Their role towards the development teams is of a coaching character. However, as the Product Owner role is not a defined role in a traditional organisational structure, there is no one close to the development teams or the Team Leaders with actual product responsibility and empowerment to set priorities.

LARGE AND DISPERSED ORGANISATION

The challenge of running Scrum in a large and dispersed organisation is closely related to, but not quite the same as the challenge above. The interviewees agree about the difficulties of carrying out the basic communication necessary for the development to progress without problems. The recent merger between Company D and another company has resulted in a rather winding line of authority and command. The interviewees talk about this in the quotes below:

> *Interviewee D1: "Our organisation doesn't have a Chief Engineer as the one we know from Toyota or others of that calibre. One person who has a full overview of the entire process from the customer to supply chain to production and at the same time knows the organisation as his own pocket. No one does that, partly because of the merger. We hardly know who to go to when we need to get answers to some question."*

> *Interviewee D2: "Our Product Line Manager, who owns the product when it goes to market and who is talking to marketing and customers, needs to be located close to the Scrum team in order to have the essential and daily communication [with the team] while managing this Product Backlog. Right now he sits in India and is extremely difficult to reach."*

> *Interviewer: "Is he some kind of Portfolio Manager?"*

> *Interviewee D1: "No, not close to. He only has the responsibility for this product. Above him there is a Line manager for General Purpose and over him is the Portfolio Manager. So, as you see, this part of the organisation is enormously heavy. This means*

that simple tasks, such as getting a technical answer, which a firmware- or mechanical developer can actually use, is extremely difficult."

One could rightfully argue that this is a common challenge in most traditional organisations. However, as one of the key values of Scrum is fast and precise communication with minimal hierarchical or physical impediments, a large and dispersed organisation is a clear challenge and hindrance to efficiently conducted Scrum.

DISTANT CUSTOMERS

In true Scrum the Product Owner *is* or *represents* the customer and pleads the customer's cause in order to maximise the value of the product and ensure a great business case. This is in accordance with one of the four key values in the Agile Manifesto for Software Development, as expressed in the phrasing *"Customer collaboration over contract negotiation"*. In the Company D case the close customer-relationship has proved difficult due to a too large distance between the end customers and developers. Interviewee D1 gives an example:

Interviewee D1: "What I really think goes wrong, is that we are not standing out there, close to customer. We are not the ones getting the direct feedback. It means that in those few situations where stuff is actually presented to the customer or someone close to the customer, then things have changed. Things that have not been predicted by anyone in the organisation, who is actually working on the project."

The example describes a rather distant or perhaps even non-existent relationship between the actual customers and the developers of the product. It is clear that this could be caused by the lack of a clear line of sight between the two peers, which might be represented by a strong Product Owner. Yet, it could also be a symptom of a strong tradition of keeping the development activities close to the chest, due to competition and secrecy directions. However, becoming agile – as the interviewees argue is important – also requires a relatively close relationship to customers.

ESTIMATING CONCEPT DEVELOPMENT ACTIVITIES

Another challenge mentioned by the interviewees is the challenge of estimating concept development activities. As it has been mentioned earlier in this case description, some concept development activities take place within the development phase after the actual concept development phase has been concluded. This is due to the nature of development activities, which are often carried out in unchartered waters – typically with a majority of *unknown unknowns*. Recapitulating an excerpt from a quote earlier in this case description, the time estimation of development tasks can be an uphill battle: *"We have a lot of examples of tasks that have been estimated with extreme inaccuracy, because you are actually working on concept level and really don't have a clue about how long it will take"*.

BREAKING DOWN TASKS

It may be difficult to correctly estimate the duration of tasks; but it is equally difficult to break down the epics of the Product Backlog into those small manageable tasks. Interviewee D1 argues that it is difficult to break down the product to small and independent packages of development activities. When asked about his attitude to the development of a physical product, Interviewee D1 relates the difficulties of breaking down tasks to the complexity of the product:

Interviewee D1: "(...) We are subject to these conditions [referring to a hierarchical perception of product development as 'building from the ground and up']. We try to do it differently [referring to the development of separate and independent parts], but just as we reach a certain degree of complexity, we fall back into the first model again."

Interviewee D1 continues to substantiate his point:

Interviewee D1: "Breaking down ideas is what Scrum tries to do methodologically. The idea you come up with has to be broken down as far as possible, in order for you to cope with it and estimate it correctly. That's where Scrum is great: you force people

to break down the tasks as far as possible. It has been extremely challenging to do that."

Interviewee D2: "In the Hardware team we break tasks down to one-hour work packages. In the Mechanics team they would rather have a large pool of 254 hours and then put in as many tasks as they find fitting, but that doesn't work as we cannot measure hours spent on documentation, concept development, and so forth."

The quotes above indicate that it is difficult to break down the development activities into small independent tasks, because the product complexity – and probably the close integration between sub-parts – hampers the process. The quotes also reveal that the Mechanics team has bigger difficulties in breaking down tasks than the hardware team, which may substantiate the idea that Scrum becomes increasingly difficult in accordance with the level of physicality.

DIFFICULT TO CHANGE HORSES IN THE MIDDLE OF THE STREAM

The core of Agile Development is the ability to dynamically react to any changes cast upon the product development, but this is exactly what challenges the development teams at Company D. When changes happen, it very often entails large consequences:

Interviewee D1: "It takes a long time to develop such a complicated product as this one. Therefore they have to be highly visionary when developing the Product Backlog and continuously believe that what we do is the right things. However, it IS a challenge when changes occur. Changes carry along large consequences for the Mechanics team, a little less for Hardware and the least for Firmware."

Interviewer: "I guess there is a danger in safeguarding yourself too broadly. To build on a platform that allows changes in the design is obviously a fine solution, but it can easily become inexpedient if you never use it, or perhaps, if you always have to think three or four product generations ahead of you…?"

Interviewee D1: "Exactly; and we also experience that it always costs quite a lot of money – and we never have too much of that. (…) So the first thing opted-out when we have to prioritise our list is flexibility."

It is clear that there are large consequences when changing horses midstream – especially to the Mechanics team where the iteration cycle is typically four to six months. This again underpins the idea that the physical aspects of a product are the most challenging when conducting Scrum in an integrated product development project.

LARGE AND EXTREMELY CROSS-FUNCTIONAL TEAMS

Lastly, the interviewees mentioned the team-related issues. Team size and necessary span of competences are great challenges. Because of the necessary integration of a vast amount of disciplines, the team size will eventually rise. The interviewees recall how this suddenly became a problem:

Interviewee D1: "Suddenly we were 32 men in a room meant for 20. An open office is a challenge, but also a necessity. We thought about breaking down the team to smaller teams, but it would destroy the commitment and sense of ownership, so we never really succeeded in finding a solution to how to reduce the teams. (…) It is a challenge when you become more than nine persons. It is evident that it becomes a problem to the independent developer to hold all that information, but what can we do about it?"

As Interviewee D1 mentions it is destructive to the commitment and ownership to split up a team that is well up in the project. However,

this is not the only team-related challenge to overcome in such integrated product development projects. Interviewee D2 mentions the difficulty of conducting true Scrum with the Mechanics team:

> *Interviewee D2: "I have six developers on the Mechanics team, all with very different competencies, so this is not true Scrum. I cannot just put anyone onto a certain task, and that's a challenge. One person does all the plastics and another does this and that. This means that some individuals are sometimes overloaded while others are not. But luckily some individuals have been here for 25 years, and that helps a little. (…) In true Scrum every developer should be able to solve most of the tasks. That is rarely true for what we do here, as we are a bunch of specialists. It doesn't matter in which direction I look in our team of about 30 developers; everyone is sitting with some sort of special skill.*

Large-sized and cross-functional teams seem to be a necessity at Company D; and this condition is seemingly preventing them from conducting *true Scrum*. These conditions arguably do not apply to only this company, as the high integration of several disciplines is the norm; in fact, it has been mentioned in several of the cases in this project.

BASIC CASE DATA

COMPANY E

CASE **E**

THE COMPANY

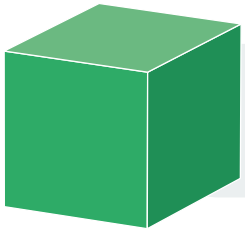
OWNERSHIP:
UNDISCLOSED

LOCATION:
DENMARK

250

IN R&D
30

PRODUCTS & MARKETS



IT solutions as integrated systems and standalone solutions. Products are sold as own brand and as OEM products.

B2B within various sectors such as DTP- and building industry and national defence. Trend towards per-industy custom solutions.

PRODUCTS & MARKETS

1,5 YEARS

Average experience compared to other cases

LENGTH OF SCRUM EXPERIENCE

The development environment is conducting "Scrum-like activities", but does not complly fully to Scrum

IMPACT IN ORGANISATION

3-week Sprints running in sync between all Scrum teams

SPRINT LENGTH IN WEEKS

INTERVIEWEES

NO.1

Name Interviewee E1
Position Project Manager
Education Electronics engineer

Role in Relation to Scrum
No formal Scrum role, but similar to Product Owner

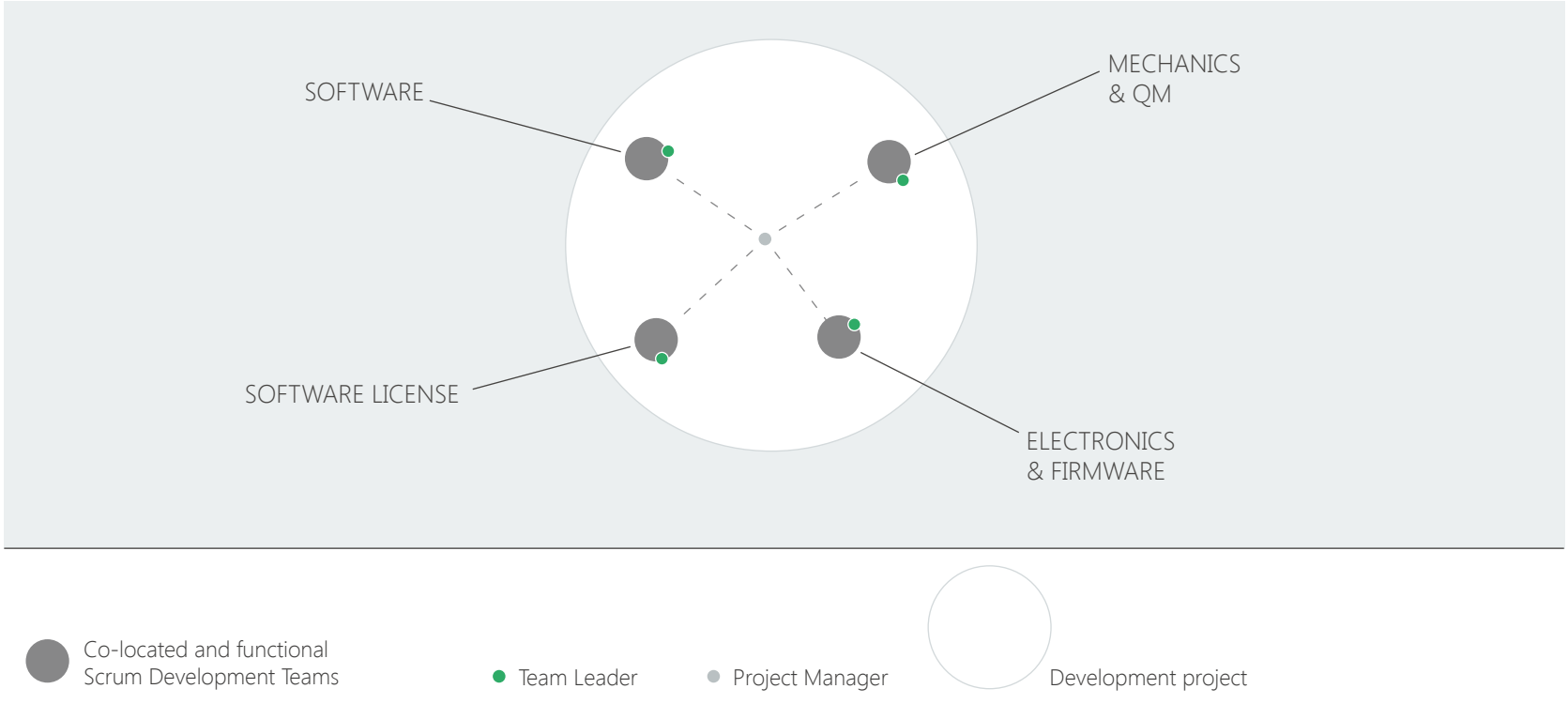
NO.2

Name Interviewee E2
Position Project Director
Education Electro Mechanical engineer + Exec.MBA

Role in Relation to Scrum
No specific Scrum role. Project director (portfolio management) Line manager (resource management within Quality & Mechanics)

4.16: BASIC CASE DATA

4.17: ORGANISATION OF SCRUM



4.5.1 ORGANISATIONAL STRUCTURE

The development department at Company E is one of four main silos in the organisation: Development, Production, Administration & Finance, and Sales. The development department is strictly organised as a matrix with several sub-departments, each with a specific competence area. These are Electronics & Firmware, Software, Software License, and Quality & Mechanics as it is also shown in figure 4.17 above. Until just a few years ago, the development department was not organised in this way. The company underwent a restructuring process from loosely organised development activities dispersed throughout the development team to a project based development structure with an overlaying Stage-Gate process model. This new structure included project managers who were responsible for the progress in the initiated development projects, and line managers who were responsible for the management of the sub-departments with their respective competence areas. Scrum was implemented as a result of a large Lean project in the Production silo, which started a discussion about how to streamline the Research & Development activities in a similar way. So the development department has been conducting its development activities through a "Scrum-like" framework for the last one-and-a-half years. It now co-exists with the matrix structure mentioned earlier, and is applied on development projects consisting of multiple development teams, divided by competence. All in all, throughout the recent years the development department has gone through a large restructuring process that covers both a reorganisation of the development environment and implementation of new methods.

The development department carries out the development of all software, firmware, electronics, and mechanics for the large-format scanning products *in-house*, and it collaborates with the production in both Denmark and Malaysia.

4.5.2 SCRUM IN THE DEVELOPMENT PROCESS

As briefly mentioned, at Company E they use a Stage-Gate process model as its overall management tool. The Stage-Gate model has been implemented six years ago and before the introduction of Scrum, and has continued to be the principal guiding framework. The Scrum activities of the development organisation are fitted into some of the stages in the model; primarily in the development stage, but also during Ramp-up and later in the stabilisation phase after the first customer shipment.

Interviewee E1 does not see any problems in using Scrum together with the Stage-Gate model:

> *Interviewee E1: "I think that Scrum fits very well into the Stage-Gate model. It does not need to be one or the other. You just alter your Scrum to fit into your Stage-Gate. (...) It may be possible to use Scrum throughout the course of a full project, but it will definitely demand something new from your steering committee, and then I guess all projects have to use Scrum."*

When asked how Scrum could be used for concept development before the development phase, both interviewees were confident that it is possible. Interviewee E2 argues that Scrum may be extended to this part of the process in larger projects. Interviewee E1 agrees that Scrum, to some extent, has already been used for some of their concept development efforts.

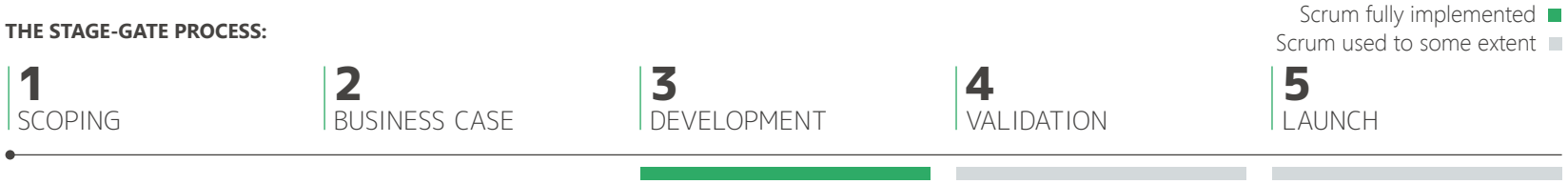
Interviewee E1 continues by mentioning what he sees as the important aspects of running Scrum:

> *Interviewee E1: "There are some things I would like to hold on to in the Scrum model. There might occur many different problems – you can have difficulties defining the tasks and what you are specifically going to do in a certain Sprint, but I would like to hold on to the fact that you have tree weeks. You stick to a tact and all teams follow this tact. If you have something to demonstrate at the end of a Sprint, then it has to be demonstrated. You have to do the Retrospective and give everyone a chance to say if something doesn't work. It can be some personal or human aspect, or just if the meetings are conducted in a wrong way."*

4.5.3 MOTIVATION AND TRANSITION TO SCRUM

The overall motivation for implementing the Scrum-like elements has been a wish from the top management to streamline the development activities in a way similar to what they had previously used with Lean principles in the production.

> *Interviewee E2: "We conducted a very big Lean project in our production that initiated a discussion in the management group about how this could be expanded. If we look just a couple of years back, some people started to talk about Lean in R&D. No one really knew how to translate this concept of highly tact-based reporting and measurability into something that is carried out years. We made our own model and used Visual Management as vehicle. (...) It was totally home-spun."*



4.18: SCRUM IN THE DEVELOPMENT PROCESS

As a run-up to implementing the Scrum framework in the development environment, the organisation started to experiment with Visual Management, which contains some of the qualities of Lean. Later they changed to Scrum, as some procedures in Visual Management didn't work that well.

Interviewee E1 also mentions the use of time as a motivation for finding ways to be more efficient:

> *Interviewee E1: "If you look at our expenses during a project, you actually don't have that great external expenses. However, time is the crucial part. We don't sell that many scanners and therefore have a relatively large margin. This also means that we have to hit the market on-time in order not to lose money"*

Interviewee E1 continues to argue that Scrum also allows the teams to plan only a few weeks ahead, instead of planning several months ahead, as suggested in other development models: *"It was exactly these problems Scrum was trying to solve: You boil down your detailed planning and maintain a certain pace in the development activities"*

Motivations:

- Planning several moths ahead is obsolete as you will evidently have to re-plan later
- Shorter planning is easier to cope with
- Reducing the overall development timeframe
- A desire in the top management to adapt the Lean mindset to R&D efforts after success in production.
- Failing projects due to communication issues

Initiative:

- Top-down (Implementation of Scrum in R&D is a Lean development initiative inspired by a large Lean project in the Production silo)

4.5.4 SCRUM FRAMEWORK #1: PRODUCT VISION AND PRODUCT BACKLOGS EFFORTS

According to Interviewee E1, the development teams are still in the process of implementing Scrum, which is also evident, due to the absence of most of the otherwise common Scrum terms. Instead of *Daily Scrum*, *Scrum Master*, *Product Owner*, and *Sprint* they purposely use terms as Morning meetings, Team leader, Project Manager and development cycle. Interviewee E1 explains why:

> *Interviewee E1: "I don't feel we are at the point were we could call it 'Scrum Master'. To do that we have to be more Scrum-like than what we are now."*

The fact that the development teams at Company E are still in the process of implementing Scrum is furthermore seen in the way that the Scrum guidelines are interpreted, and in the absence of certain Scrum artefacts. However, the interpretation of – and commitment to – the Scrum framework vary among the development teams working together on the same project. While the Software teams are relatively strict in their use of Scrum, the Electronics and Mechanics teams are having difficulties in adhering fully to the Scrum guidelines. Two of the aspects in which these teams are only loosely following Scrum are in the Product backlog and Sprint planning efforts; they do have Product backlogs and often plan the following Sprint on a less detailed level. As the only one of the teams, the Mechanics team only conducts "morning meetings" twice a week instead of daily meetings.

Despite large variations in how the teams at Company E are using Scrum – even in one single project – the project operates with a sort of Product Vision, which is developed in the Investigation phase. The Product Vision sets the outer boundaries of the project and may include some clarifying aspects in regard to chosen technologies and design related aspects. As just mentioned the teams are handling the Product Backlog step in different ways, but all teams are roughly estimating expected resources in collaboration with the Project Manager at the beginning of the project.

When asked how the teams break down the tasks to fit the relatively short three-week sprints, the two interviewees mention the difficulties related to the Mechanics team:

> *Interviewee E1: "It has been pretty difficult to run Scrum in the Mechanics team, but I think it has something to do with the culture in the mechanics department in the company. I cannot see through how much of it is culturally determined, or whether that is actually possible."*
>
> *Interviewer: "I guess it also has something to do with how you break things down as you said earlier. They may just work in a different way?"*
>
> *Interviewee E1: "Very much. That's also why it is difficult to say what you want to do in a certain Sprint, because there are just some things that are closely related."*
>
> *Interviewee E2: "I see the mechanical part as some kind of an amoeba that develop itself in the three-dimensional plan. Your concept and its elements become more and more mature. You know what you have to do - you may look at the transmission part and do a first shot on this. Then it gets better and better and along the way other elements surface. It ultimately ends with refined and three-dimensional models. It's like an amoeba. It is not possible to tie up any ends."*

4.5.5 SCRUM FRAMEWORK #2: SPRINT CYCLE

The development teams at Company E run three-week Sprints and all teams that work on the same project perform synchronised Sprints. All Sprints start with a combined Retrospective and Sprint Planning meeting every third Thursday. Because of the overall Scrum team size, the meeting often takes place over two days and is split into the competence-specific departments. However, all planning activities take place in a certain project room, in which all the Scrum boards belonging to the development teams are visible to everyone. This room plays a central role in the cross-team communication, which is otherwise not formally facilitated. In this project room Post-it notes of a certain colour signal the interdependencies that exist between the teams.

As mentioned earlier most of the teams have daily morning meetings – another word for Daily Scrum meetings. The mechanics team is the only one restricting itself to only two weekly meetings as they consider more meetings too time consuming.

The three-weeks Sprint ends with a Review for each development team, which is broadly announced in the company.

> *Interviewee E1: "Everyone gets an invitation down in the canteen – also stakeholders such as marketing and even finance and service."*
>
> *Interviewer: "Does that mean that the Software team gets a chance to see what the Mechanics team has been doing?"*
>
> *Interviewee E1: "Well, we haven't done this to the same extent with the Mechanics team. We don't invite everyone. The more people, the more chatter. Everyone has an opinion of the mechanical solutions. When people see visuals, there are always negative opinions, and we do not want that noise."*

The interviewees keep going back to a discussion about the mechanics team and its special conditions. Interviewee E2 argues that the big challenge is to synchronise the output of a Sprint:

> *Interviewee E2: " (...) You take some procedures and routines [from software] and try to make them fit. It has been very difficult to find out what kind of deliverables we should provide. The method [in mechanical development] is not that you concentrate*

on one corner of the product and tie up its ends, and then concentrate on another corner afterwards. You work on something and move on to something else that affects everything, and then you are back again. I don't think we have succeeded in demonstrating anything that has been truly 'Done'."

Interviewee E1: "My point is just that even though you cannot define exactly what you have done, you have to make a Review of it. You might just have to explain it or put a headline to it. Because the few times we have not made the review, it has hit us twice as hard later."

Interviewee E1 recalls how these challenges in mechanical development were present in a recent project:

Interviewee E1: "I think we used six or eight months on getting the concept right and get all the ends tied up, and it was done through that [the Sprint cycle]."

Interviewee E2: "And then the rest is just documentation. This is perhaps what makes it different – you work for an incredibly long time where everything is dynamic, and then suddenly it is locked, and all the drawings rattle out very quickly. To some extent this is the nature of mechanics."

4.5.6 DESIGN AND STYLING EFFORTS

To a large extent it is the Mechanics team that independently creates the design of the Company's products. Discussions about design efforts are very common in this team, which also affect the bi-weekly morning meetings. Interviewee E1 talks about the on-going design-related discussions in this team:

Interviewee E1: "(…) But we have had difficulties getting the Mechanics team to have the morning meetings. They say it's a waste of time. The reason why their morning meetings take that much time is that they have long discussions about design elements. They simply cannot stop discussing details, so I guess there's a need for considering this."

It is a clear fact that the design efforts in general are conducted in various less rigid ways than other development activities in the project. Both interviewees express this:

Interviewee E1: "Our mechanical developers are very innovative and perhaps less structured. If they were to sit down and write yellow Post-its [for a Product backlog], they wouldn't produce any. But if I asked them what tasks they were about to do on the product, they would just start reeling off a bunch of things. So please, write it down! It is exactly what I ask you to write [on the Post-its]. It was at this point I felt it was something in their culture that conflicted with the way they work."

Interviewer: "Perhaps they work in a more holistic manner?"

Interviewee E1: "Yes, I definitely think so."

Interviewee E2: "If I see it from the perspective of the mechanical developers, I think you should be careful not only to judge it as a cultural problem, because there ARE elements in the way you work with mechanics that make it illogical to break things down to very small parts. Delivery time is one thing; external dependencies is another thing that can be very dominant and very unpredictable – certainly compared to hardware; and software almost hasn't any ."

As it is expressed in the quotes above, the Mechanics team certainly seems to have difficulties in carrying out the design efforts in combination with the Scrum activities. However, as it has also been indicated in earlier quotes, they are still able to keep a balance in the development throughout the course of the project by staying dynamic and by not tying everything up in the beginning.

4.5.7 MAIN CHALLENGES EXPERIENCED BY INTERVIEWEES

When asked what they consider the main challenges of using Scrum in their product development, the interviewees mention a number of issues. The issues mentioned are described below.

A SPECIAL CULTURE IN MECHANICAL DEVELOPMENT

Throughout the interviews of this case the aspect of a certain culture in the Mechanics team has been mentioned several times. It is clear that both interviewees find that this team to some extent is particular in regard to how the Scrum framework is received and implemented. Interviewee E1 argued that the mechanical developers could be characterised as innovative rather than structured. Naturally, a negative effect of this special culture is the team's hesitation towards fully adopting the Scrum practises, which they almost consider a waste of time. This alleged special culture may not be a challenge in itself, but it certainly is an underlying condition that is part of the reason for the other challenges mentioned in the following.

CREATING SEPARATE DELIVERABLES IN EACH SPRINT AND BREAKING DOWN TASKS

Creating separate deliverables as the Sprint result is one of the challenges that the Mechanics team is facing – just as it is a challenge to break down the task in the beginning of a Sprint. Interviewee E2 argues that some of the earlier mentioned cultural differences between the various disciplines lie in the way developers carry out the development activities. While software developers are able to split up a project in features and easily work with separate deliverables, the mechanical developers work horizontally and iteratively across all aspects of the product in development, which makes it difficult to adhere to the Scrum practices. Interviewee E2 recalls the challenge of defining tasks in the Mechanics department for Scrum projects:

> *Interviewee E2: "It is very important how you define your tasks in order to be able to remove them from the board after three weeks. Quite often we have ended up defining tasks on a too high level, which has resulted in Post-it notes staying on the Scrum board for too long – and so we have missed the point."*

RESOURCE ESTIMATION FOR DEVELOPMENT TASKS

Another challenge mentioned by both interviewees is time and resource estimation. In the two quotes below, Interviewee E1 expresses the frustration about estimating tasks:

> *Interviewee E1: "A difficult part is to work with time boxes. You will never get to the lowermost items in your backlog. I see it work in Software, but when you get to mechanics and electronics, there's just things that you cannot leave out. You HAVE to get through all the things, and that's the difficult part of working time boxed. If you stumble upon some problem that you just cannot solve, well, then it just takes more time."*

> *Interviewee E1: "I think the world is changing so quickly that we can only plan one Sprint ahead, and we always just finish 50 or 60 % of what we planned. But still, the product gets finished in time. How does that come about?"*

It is clear that the teams – and the Mechanics team in particular – are having difficulties in estimating time for development tasks. This may be a real challenge or just a symptom of too little emphasis on estimation. Whatever is the case, the quotes below indicate the general attitude towards this aspect.

> *Interviewee E1: "At the Sprint meetings we actually don't use hour estimates. You sort of lean back – each person has a rather good understanding of how much time the tasks will take and whether or not it looks reasonable."*

> *Interviewee E2: "It doesn't make sense to break it down into hours. Days, at best."*

Interviewee E1: "It is more like some kind of estimate of what we will achieve in the next three weeks."

Interviewee E2: "We simply cannot use it for anything."

KEEPING ALL OPTIONS OPEN

The challenge of keeping all option open is closely related to one of the fundamental differences between traditional development and Agile Development. The Agile Manifesto for Software Development preaches embrace change, but this can be a challenge:

Interviewee E2: "It has been a challenge that nothing was locked. Even though you showed off your 3D model, it would still be subject for discussion. In reality it was really difficult to tie up anything. We haven't succeeded in that part."

Interviewee E1: "No, and I don't know if it is possible. But maybe we don't have to. We don't need to implement Scrum too rigidly, because I'm not sure it is possible at all. You need to be aware of that."

SUPPLIER DELIVERY TIME AND OTHER EXTERNAL DEPENDENCIES

The interviewees argue that, compared to other disciplines the Mechanics team has certain domain-specific challenges that have something to do with the physical character of the resulting work. While Software and Firmware teams don't have the dominating physical constraints, the Mechanics team has to consider delivery times from sub-suppliers and other external dependencies. These aspects require long-term planning as delivery times for certain parts are often more than half a year. This clearly creates a fundamental paradox in regard to conducting true Scrum in conditions like those. This paradox might be more theoretical than practical, but nevertheless is it one of the challenges mentioned by the interviewees, and it certainly necessitates a supplementary planning system. In this case a special Microsoft Project plan keeps track of the aspects that stretch further into the future than just the next few Sprints.

POLARISATION BETWEEN TEAMS

The special culture in the Mechanics team has been mentioned quite a few times and often in relation to some of the challenges, but the fact that the nature of the teams differs creates yet another problem in itself. Due to the difficulties of adhering to the Scrum practices, the Mechanics team often ends up as the black sheep of the family, which results in a polarisation of the whole project team.

Interviewee E2: "When it really works well here and not that well there, it might create some kind of polarisation. There's the team that knows how to do it, and then there's the team that doesn't. I think we need to learn how to handle this in a way. We need to get an understanding of the basic differences. Why we act in this way and you act in that way – as part of the same overall team. That's my impression."

Interviewee E2 argues that a growing polarisation creates a potential risk for losing the team spirit in the overall project team, and this may ultimately result in an inefficient development environment and an unproductive internal competition.

BASIC CASE DATA

COMPANY F

CASE **F**

THE COMPANY

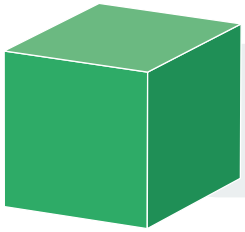
OWNERSHIP:
PART OF A DANISH CORPORATION

LOCATION:
DENMARK

1100 IN DENMARK

IN R&D IN DK **130**

PRODUCTS & MARKETS



Advanced acute care solutions that simplify and automate acute care testing.

Healthcare domain and medical labs

PRODUCTS & MARKETS

1,5 YEARS

Average experience compared to other cases

LENGTH OF SCRUM EXPERIENCE

Scrum is implemented in large parts of the development environment.

IMPACT IN ORGANISATION

4-week Sprints with intermezzi of one week for planning and ad hoc tasks between Sprints.

SPRINT LENGTH IN WEEKS

INTERVIEWEES

NO.1

Name Interviewee F1
Position Electronic Design, R&D
Education Electronic Technician

Role in Relation to Scrum
Scrum Master

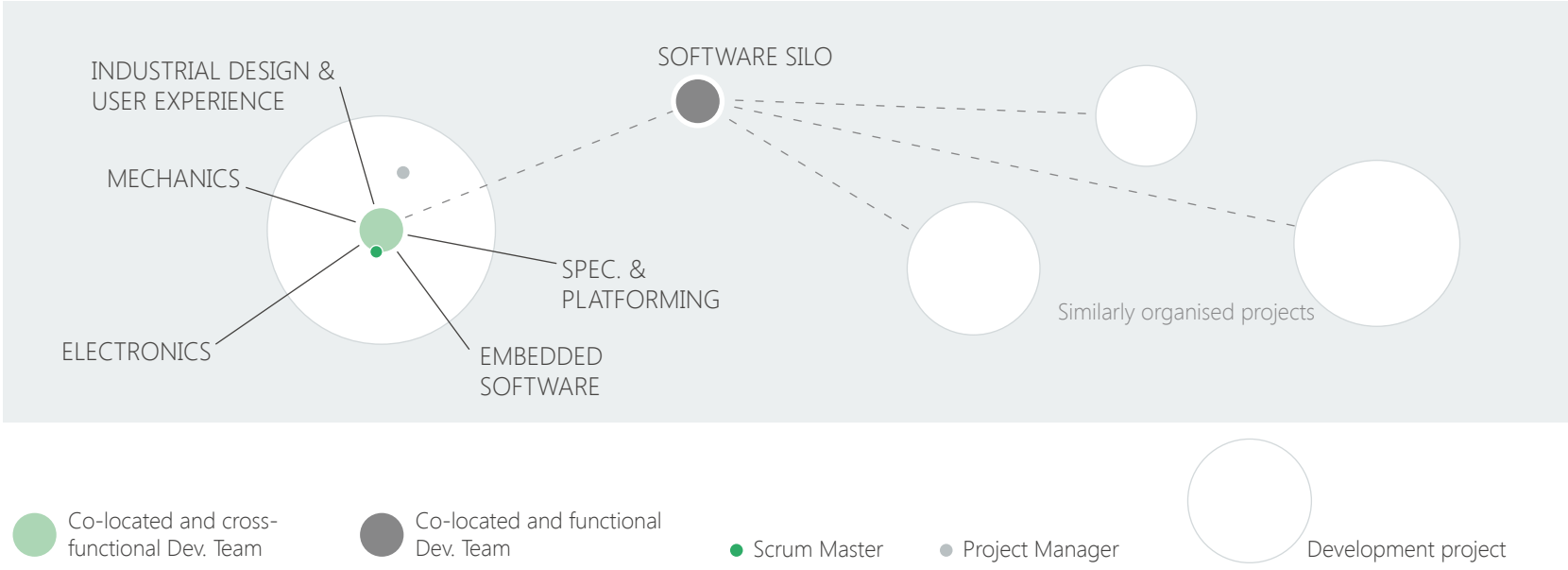
NO.2

Name Interviewee F2
Position Laboratory Technician
Education Laboratory Technician

Role in Relation to Scrum
Developer

4.19: BASIC CASE DATA

4.20: ORGANISATION OF SCRUM



4.6.1 ORGANISATIONAL STRUCTURE

The development department at Company F is organised in parallelly working development teams. The teams are cross-functional teams and work on a single project each. The team in focus in this case description consists of 10 to 12 individuals covering disciplines ranging from mechanics and industrial design to embedded software and electronics. All team members are physically co-located, and the team includes a project manager, who participates in bi-weekly Scrum meetings, and Interviewee F1, who is Scrum Master, but also carries out development- and coordination tasks as a normal team member. According to Interviewee F1, Company F has always been a strong project organisation with highly cross-functional teams, and earlier the teams also included software developers. However, two years ago all software developers were moved out of the project teams in order to establish one single and large silo for software development.

According to the two interviewees, it seems that similar competence silos are about to be realised within the disciplines of mechanics, chemistry and electronics. The software team has acted as a good example, and the management is interested in spreading the success to other disciplines and functions of the development. Despite the high level of cross-functionality in the team, outsourcing of certain development tasks to external partners is still used in parts of the development project.

4.6.2 SCRUM IN THE DEVELOPMENT PROCESS

Company F is deploying a Stage-Gate process model as overall management tool for development activities. According to Interviewee F1, development projects are typically running over a three- or four-year timespan, and some of the most complex development projects are up to 12 years on their way. The fact that Scrum at Company F is a relatively new and unsettled addition to the development toolbox is clearly seen by the following quote. Here Interviewee F1 compares the company practice with “true Scrum”:

Interviewee F1: "I don't know if I would use the word Scrum, because I actually don't think we use Scrum – Scrum is in the world of software. I think we are utilising agile methods rather than Scrum. We use the word 'Scrum Master' because it is a well-known concept, and there are things from Scrum that we are also using – such as Backlogs and so on. But I rather see our method as being agile, as we have modified it quite a bit. The two are very different – we don't have a common method in our company culture. Our software team has been conducting Scrum in the last three years with great success – it is truly by the book. We, on the other hand, have adopted it and attempted to use it with its qualities – both good and bad."

His reflections whether "Scrum" is the right word are symptomatic of a general level of awareness, and show how many considerations are put into the implementation process. It also shows that Interviewee F1 is quite knowledgeable about what Scrum *is* and *is not*, and that Scrum needs to be modified in order to work in non-software projects.

When asked where they use Scrum, Interviewee F1 points to a rather large part of the process and argues that they have been working hard on implementing Scrum relatively early in the development process:

Interviewee F1: "The really big change here is to be found in the way we develop. Earlier it was the development engineer who came up with the way forward, but now Industrial Design, User Experience, and Usability certainly have the greatest influence, and we have made a lot of efforts to get there.".

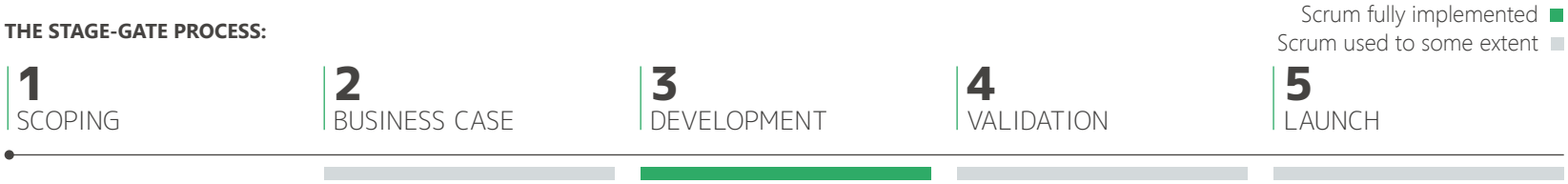
4.6.3 MOTIVATION AND TRANSITION TO SCRUM

Scrum was implemented in software development activities three years ago, and after it had proved successful in this discipline, the management decided to spread the practices to other parts of the development organisation. The management is urging for very short Sprints in order to gain a transparent and frequent feedback in the form of burn-down charts. This is, however, opposed by Interviewee F1, as it would result in a too large administrative burden to update charts and plan Sprints each week.

Both interviewees feel that Scrum has provided a useful and solid tool for managing the development activities. When asked what kind of tools they were using before, Interviewee F1 mentions large and confusing project plans:

Interviewee F1: "We used giant Gantt charts that no one bothered to look at. (...) Of course, we have had a giant Gant Chart, which just hung there on the wall and which no one could grasp. That's the worst part of it. The big picture disappears."

Later, on a small tour around in the development environment during a break from the interview, the interviewees comment on their Scrum boards on the walls, evaluating it in comparison to their old practice with Gantt charts:



4.21: SCRUM IN THE DEVELOPMENT PROCESS

Interviewee F2: "The big advantage is that you get a little less stressed. You are able to grasp all the tasks. You could easily have 30 tasks lying on your desk, and which one should I concentrate on?"

Interviewee F1: "All the tasks lying in piles on your desk are not visible to the team. In this way [on the Scrum board] your workload and tasks become apparent to the team."

The quotes above indicate that the interviewees are predominantly positive towards the implementation of Scrum even though the management forced it upon them less than two years ago.

Motivations:

- Scrum has been a success in Software and is therefore implemented in other development disciplines
- Demand for more transparency from the management

Initiative:

- Top-down

4.6.4 SCRUM FRAMEWORK #1: PRODUCT VISION AND PRODUCT BACKLOGS EFFORTS

At Company F, the teams do not have a Product Vision, but rather a five-year product plan that indicates the type of product that the company aims at developing in the near future. Interviewee F1 argues that the lack of a Product Vision is due to the fact that the company has some extremely long development cycles, lasting up to twelve years in some cases.

Interviewee F1: "If you look at our vision, some parameters have been part of it for the last ten years, and we are still working on developing them."

Instead of a Product Backlog based on a Product Vision, it is a traditional requirement specification that leads the development activities. However, according to Interviewee F1, the team is actually working without a real Product Backlog:

Interviewee F1: "It has been a long discussion with all the new guys. They say they are Product Owners, but when I ask them to give us a Product Backlog for the project we are doing, they really don't have the fundamental overview needed. And then it ends with the team framing the development tasks."

Interviewee F1 continues to argue that this part of the development process ought to be better supported by the management:

Interviewee F1: "I would like my boss [the project manager on the present development project] to have a much better overview of the activities he thinks need to be done at a certain point. But that's not how things are right now. (…) As a Scrum Master I miss some inputs for a Product Backlog. In reality our projects operate without Product Backlogs. We are creating something similar to the Product Backlog by making some broad scopes – the various 'builds' we need to develop. These are part of the Backlog, which we break down in some sub tasks that actually just become small Sprint goals – and that is a really good help."

The quote above shows that the development team and its Scrum Master are left alone with the maintenance and grooming of a sort of self-defined Product Backlog. Interviewee F1 suggests that this is a source of frustration, and also points to the fact that a Product Backlog would help the team in prioritising the list of development tasks, even though this might prove more difficult in integrated product development than in software:

Interviewee F1: "In software they have this fantastic ability to freely prioritise tasks. We [in integrated product development] just can't avoid putting wheels on the car, can't we? Then it's not a car. And this is a condition, when something is physical. (…) There is some functionality that we just need to have."

4.6.5 SCRUM FRAMEWORK #2: SPRINT CYCLE

The development teams at Company F conduct Sprints of four weeks in a five-week cycle, which leaves one week after each Sprint for Retrospective, Planning, and catching-up with other tasks. This also means that the four-week Sprint is concentrated development without other interruptions than the scheduled stand-up meetings within the development team twice a week. According to Interviewee F1 a daily meeting would be too much, which is why his team has agreed on conducting them only twice a week.

Earlier in the interview, Interviewee F1 argues that the team is working without a Product Backlog. When asked how the team copes with this in the planning of a Sprint, Interviewee F1 describes how the team uses a Sprint Goal to assist this process:

> *Interviewer: "You mentioned that you are working without an actual Product Backlog. How do you then identify tasks for the Sprints?"*
>
> *Interviewee F1: "It happens indirectly as we set a goal for each Sprint at the Sprint Planning meetings, and implicitly this is a Backlog. In reality it is the 'Definition of Sprint', in the same way as 'Definition of Task'. We agree upon an expected output of the Sprint and then break that down into tasks for the Sprint. (...) We try to do this, but I honestly miss the Vision and a true Backlog to work with."*

As it is clarified in this quote, the team is handling the breakdown to Sprint tasks in its own way despite the lack of a Product Backlog. During the Sprint Planning, the team also estimates each task, as it is the practice of Scrum. A part of this work is also to carefully develop a Definition of Done for each task. All this information about the tasks is printed on separate task cards and clearly communicated on the Scrum board. The planning of tasks also includes a time estimate for solving them, and this part of the process has proved highly difficult to the team, as it requires quite a lot administrative work to obtain an overview of how resources are spent, or whether or not certain developers are overloaded or idle in the Sprint.

During the four weeks of each Sprint, the team meets in front of the Scrum board twice a week. However, according to Interviewee F1 the procedures and roles are still not fully embedded in the habits of the team.

> *Interviewee F1: "I am NOT the one taking the decisions, and that has been very difficult to understand for some in the group. It is your Scrum board and your plan. I am just the facilitator and have the responsibility to facilitate and ask questions. That is what I attend to, but some see it [the Scrum Master role] as a project management role, and it is not. We need to remember that – we already have a Project Manager."*

The statement shows how deeply the traditional roles and responsibility structures are rooted in everyone. The responsibility of the Scrum Master is often mixed up with that of the Project Manager. As to the developer role defined by the Scrum Guide, the optimal conditions are to have a close and highly committed group of individuals throughout the project. However, this has proved to be difficult at Company F. This seems to be a result of two things, namely 1) the team changes throughout the project, and 2) the complexity of the project necessitates the inclusion of experts rather than generalists:

> *Interviewee F1: "In the software silo there is a critical mass of developers with the same competences. In our project team we can easily be just one electro technician, one chemical engineer and one mechanical engineer. They just really can't share other tasks than getting the coffee."*

Especially the dominant cross-functionality that exists in the development team makes some of the basic Scrum practices difficult, such as collaborative time estimation and mutual criticism in the reviews.

At Company F the results of a Sprint output after four weeks of concentrated development can be of almost any character. Interviewee F1 argues that the outcome can be almost anything as long as the task is thoroughly described.

> *Interviewee F1: "I strongly feel that filling out the 'Definition of Done' on our task cards is the most important thing, as it is the only aspect we have a chance to gauge. It is important that there is a clear consensus in the group before a task is marked as finished. (…) There are quite a few members who are very good at describing what they are going to do, but I always ask them to also note the aspects which are NOT part of the task."*

As the team is rather cross-functional, matching the expectations for each task is an important part of the communication and collaboration across disciplines. This is also why it may seem unexpected that the team has actively chosen to leave out the Sprint Review process in the end of each Sprint:

> *Interviewee F1: "It doesn't make sense to present something just because the Sprint is completed (…) We do it once in a while – just not as part of the agile process."*

Instead of adhering to the Scrum practice of conducting Sprint Review meetings after each Sprint, the development team uses other occasions for presenting their development, such as at gate meetings in the overlying Stage-Gate process, or when one of their predetermined builds is ready for presentation.

4.6.6 DESIGN AND STYLING EFFORTS

At Company F industrial design seems more commonly used in the development environment. Disciplines such as Industrial Design, Usability and User Experience are included in the concept development by the Scrum team from the early phases and are gradually taken away from the team as the project progresses. Nevertheless, the integration of these disciplines is not without its difficulties:

> *Interviewee F1: "We have this group which is simply just beyond any pedagogical reach [pointing at Industrial Design and Usability]. There's no planning, things turn up from day to day, and they expect that everything can be done by tomorrow. "*

Interviewee F1 also mentions that the individuals in these disciplines have had some difficulties in contributing to the Sprints, but emphasises that they have been part of the team.

> *Interviewee F1: "Their expectations are not aligned with the practical implementation of things. You can do many things in form and colours on paper or cardboard – it is something we have been discussing with them. You have to be much more clear when presenting your needs. (…) They are part of the Sprint but it is not without problems."*

It is evident that the team considers the integration of the ID, Usability and UX competences important. Nevertheless, it seems that there is a radical difference in how these individuals and the rest of the development team carry out product development. It has already been mentioned earlier in the case description that increasing emphasis is put on ID-, UX-, and Usability efforts in the early concept development phase. This indicates that these disciplines are to be a permanent part of the product development efforts at Company F.

4.6.7 MAIN CHALLENGES EXPERIENCED BY INTERVIEWEES

When asked what they consider the main challenges of using Scrum in their product development, the interviewees mention a number of issues. The issues mentioned are described below.

BALANCING CRITICAL MASS OF COMPETENCE WITH CROSS-FUNCTIONALITY

One of the challenges frequently mentioned by the interviewees is the organisational change from cross-functional teams to the establishment of several competence silos. As mentioned earlier the soft-

ware competences are gathered in a separate department and act as sub-contractors in the projects, and according to the interviewees more silos are on the way. The tendency reflects a paradox between the need for a certain critical mass of homogeneous competence in a Scrum development team on one hand and a need for close interaction between multiple competences on the other. At Company F it is clear that the high integration of a large amount of specialist competences is complicating this situation even further than in most other organisations using Scrum. Recalling a quote above, Interviewee F1 ironically suggests that making coffee is the only skill, which everyone on the team shares with the others. The quote seems just as true as it is ironic.

Yet another complicating aspect related to the composition of development teams is the fact that the needed competences in the development team changes over time.

SYNCHRONISATION AND DEPENDENCIES ACROSS TEAMS

Another complex challenge closely related to that of team composition is the interdependency *between* teams working together on the same project. According to Interviewee F1, some of the projects at Company F require close coordination between several teams. This, however, is a significant challenge as the sub-contracting teams may have their own priorities that not necessarily fit well into the plans of the other team.

> *Interviewee F1: "If we are just focusing on our own little area of competence, we shall deliver in time for sure. We just don't deliver it in time to those who need it, and that's a really big challenge."*

He continues to argue that the internal interdependencies are part of the conditions when conducting Scrum in an integrated product development project:

> *Interviewee F1: "Those interdependencies just ARE part of this, and we don't have a working product until we have been through this process. The parallelism is a bit more present in software as here they are able to work on a small module without being dependent on others."*

BALANCING TASK ESTIMATES WITH RESOURCES

The process of balancing the tasks with the available resources in an upcoming Sprint has proved to be one of the most significant challenges, according to Interviewee F1. The challenge actually consists of two parts: Firstly, estimating the extent of the tasks, and secondly, adjusting the amount of tasks to the available resources. The latter part is a considerable concern of Interviewee F1, who once *has* tried to take the responsibility of calculating the exact workload for each person in his team in order to ensure that no one had been overloaded with tasks. However, this took up all his time, and therefore he stopped doing it.

> *Interviewee F1: "... it is really difficult to tell if there's a too big workload on one individual. And we don't want to have a large bureaucratic burden of counting and calculating workload. Instead we have agreed that as an individual developer, one has to come forward and put down one's foot if one feels one get overloaded."*

PRIORITISING TASKS

Another challenge related to tasks is the work of prioritising them. In opposition to the conditions in software development, Interviewee F1 argues that task prioritising is more complex in physical product development. The quote below substantiates this argument:

Interviewee F1: "In software they have this fantastic ability to freely prioritise tasks. We [in integrated product development] just can't avoid putting wheels on the car, can we? Then it's not a car. And this is a condition, when something is physical. (...) There is some functionality that we just need to have."

CO-EXISTENCE BETWEEN SCRUM ROLES AND TRADITIONAL ROLES

The Company F case reveals several undesirable results of merging Scrum roles and traditional management roles. One is the fusion of the Project Manager-role and the Product Owner-role. According to both interviewees the Project Managers do not take on the ownership that is necessary to lead the team and promote a Product Backlog with clear priorities. This may be explained by the fact that Company F is not conducting Scrum by the book, but is rather developing its own version that suits to the organisation.

Another conflict of roles concerns the Scrum Master. That is: When he is considered a sort of project manager by the Development Team, which he is not. Interviewee F1 also mentions that the team members are hesitant in taking on the mutual responsibility of maintaining the Scrum board and maintaining the status of tasks during Sprints.

DIFFERENCES BETWEEN THE MINDSETS OF DESIGNER AND ENGINEERS

Finally, Interviewee F1 alleges that the collaboration with certain creative individuals with competencies in Industrial Design or Usability has proved difficult. This, like some of the statements made in the case description, may reflect a personal preference; but it could also reflect a real conflict or gap between two ways of understanding and undertaking development activities.

BASIC CASE DATA

CLIENT OF COMPANY G

CASE **G**

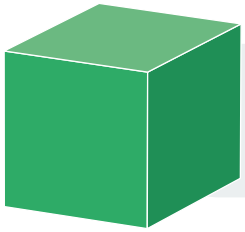
THE COMPANY

CLIENT OWNERSHIP:
PRIVATELY OWNED

LOCATION:
DENMARK

300 IN R&D IN DK

PRODUCTS & MARKETS



Supporting Building component for several types of housing

The client sell its product to privates and professionals on a global market

PRODUCTS & MARKETS

8 MONTHS

Less than average experience compared to the other cases

LENGTH OF SCRUM EXPERIENCE

Parts of Scrum have been implemented, but large obstacles exist due to the organisational legacy

IMPACT IN ORGANISATION

Teams choose different Sprint lengths.

SPRINT LENGTH IN WEEKS

INTERVIEWEES

NO.1

Name Interviewee G1
Position Management Consultant
Education Global Business Development Engineer

Role in Relation to Scrum
Consultant and Scrum coach

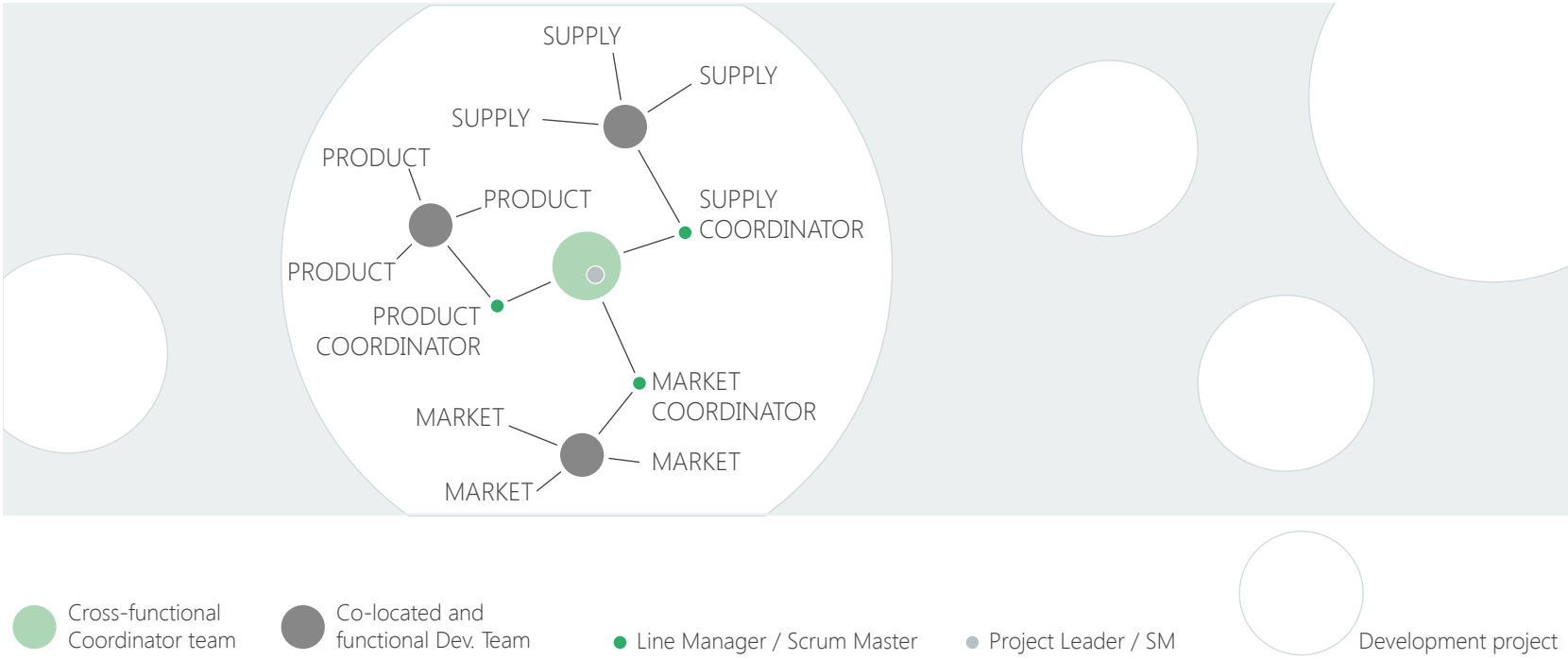
NO.2

Name Interviewee G2
Position Owner of Howbiz Management Consulting
Education Ph.D. Operation Management

Role in Relation to Scrum
Consultant and Scrum Coach

4.22: BASIC CASE DATA

4.23: ORGANISATION OF SCRUM



4.7.1 ORGANISATIONAL STRUCTURE

The development organisation at the Company G-client is roughly divided into two business areas – the main product division and an accessories division. The accessories division has approximately 130 developers and the main product division counts a bit more. The organisation is structured around projects that hold from as little as five or seven employees up to 30 or more in some projects. The company has its Danish employees scattered around at several geographically dispersed locations, primarily divided by functional disciplines. This means that Supply is primarily located in one place, product development at another location, and market related activities in a third place. The company is a large concern and its organisational structure is rather complex as it carries a lot of legacy and therefore is not easily grasped. However, each project most often includes an officially assigned project manager, a product coordinator, a supply coordinator, and a market coordinator. The combination of these four roles corresponds to the four areas of the development organisation.

Company G has been hired as an external partner to facilitate an extensive process of change in the company with focus on improving *lead time*, *hit rate*, and *intensity*. The implementation of Scrum is one of many enterprises that Company G has launched in order for the company to reach those goals.

4.7.2 SCRUM IN THE DEVELOPMENT PROCESS

The development organisation followings the company's own Stage-Gate process model, which to a large extent resembles the original Cooper model. The model has been part of the formalised development process for long, and it still is the most influential process control mechanism in the company as every project is built around its stages and gates with upper-level management taking go-/kill-/hold- and recycle decisions. Company G has chosen to supplement the combination of Stage-Gate and Scrum with an intermediate method that has the purpose of closing the otherwise rather large gab between the two. While the Stage-Gate model is covering the

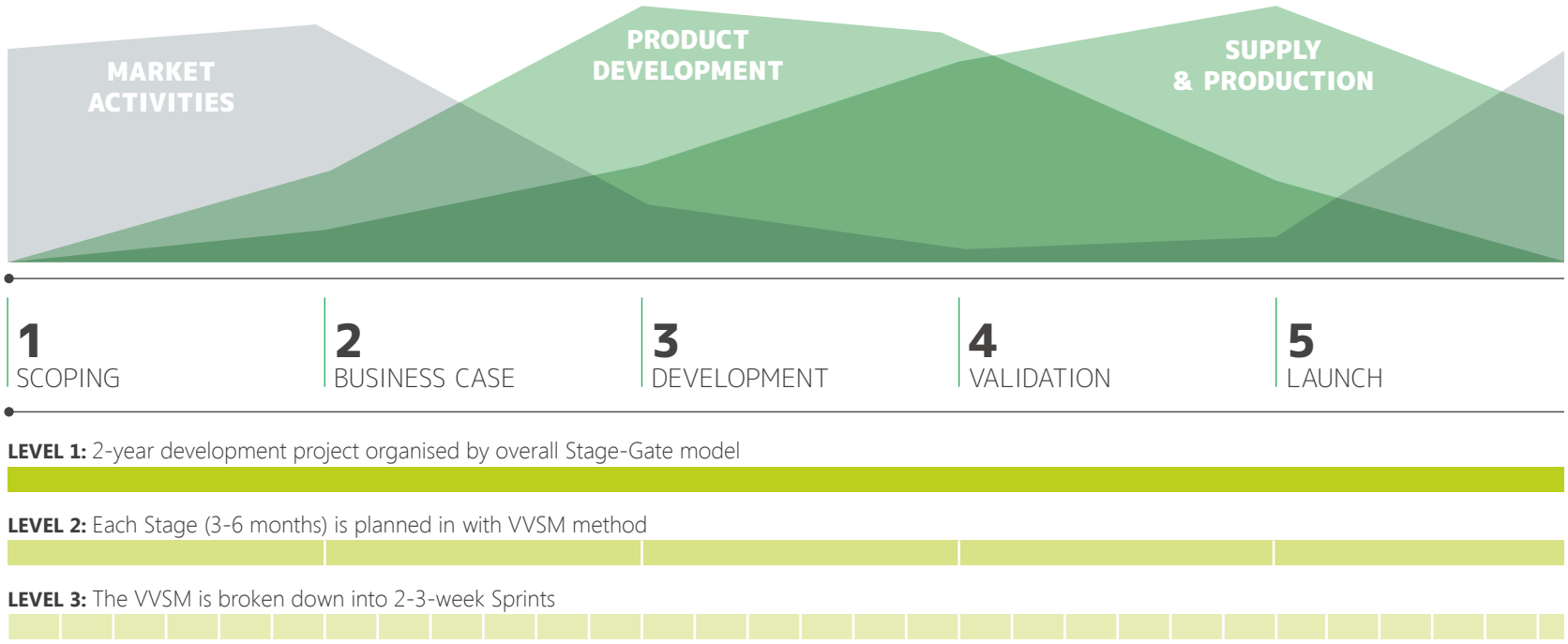
entire development process, this additional method, Vertical Value Stream Mapping (VVSM), is used to coordinate the activities of the separate stages. The VVSM method helps the development teams in identifying the major tasks of the project in the actual stage and these are subsequently the basis for the Scrum activities. In this way the three levels are complementing each other in an increasingly detailed process of breaking down the overall development efforts.

Scrum is thoroughly implemented in the project management, product and supply areas, whereas it has not yet been fully implemented in the market area. The intensity of Scrum varies throughout the development process in the separate areas, as it is shown in figure 4.24 below.

4.7.3 MOTIVATION AND TRANSITION TO SCRUM

The implementation of Scrum has been driven by a motivation for improving the lead time of the many development projects, and as it has been mentioned, Scrum is only one of many initiatives for reaching this goal. Interviewee G2, who has developed the change process for his client, recalls one of the initial meetings with the organisation concerning this process.

> *Interviewee G2: "The management had a strategic goal of improving the lead time from idea to launch, so we invited 25 leaders and engineers to a workshop with the purpose of figuring out what prevented the organisation in reaching its goals. One of the things that repeatedly came up was difficulty with planning."*



4.24: VARYING INTENSITY IN THE THREE DIVISIONS
Market, Product, and Supply divisions have different curves for intensity on a project, whish is also reflected in their Scrum actitivies. The figure also shows the relationship between Stage-Gate, VVSM and Scrum

IM.03 **VERTICAL VALUE STREAM MAPPING**
A team is gathered for a VVSM event with the purpose of planning the next stage of a development project
Photo: Own photo

Scrum has since the initial meetings with the management become a formal part of the development process. Each development project has to include the Scrum framework at some point in the process, and each project team has the opportunity to get assistance from Company G in order to start this process.

Motivations:

- Improve lead time
- Improve hit rate
- Improve project intensity

Initiative:

- Top-down

4.7.4 SCRUM FRAMEWORK #1: PRODUCT VISION AND PRODUCT BACKLOGS EFFORTS

As mentioned earlier, the company uses its own phased process control model, and all projects start out with a project contract. In the initial project phases the team qualifies the project and slowly builds up a product specification. According to Interviewee G1, the product specification then forms the basis for a Product backlog:

> *Interviewee G1: "Typically, a project manager or a coordinator takes the initiative to establish the Product Backlog. It is initially made in general terms and functions, and perhaps comprises 20 different elements in varying detail. This document subsequently becomes the basis of a project-breakdown workshop for the De-*

velopment Team, which results in the actual backlog. If the breakdown gets too detailed, some tasks are transferred to the Sprint Backlog,"

Due to the size and complexity of the organisational structure, it is difficult to find the right Product Owners for the development projects. The project manager often takes the role of the Scrum Master, whereas no one really takes on the Product Owner role. This has led to an increased focus on product specification. As it is argued by the interviewees, this, to some extent, can make up for the lack of a Product Owner.

According to Interviewee G1, the Product Backlog is relatively static as the team focuses on describing the backlog items on a generic level. When the team needs new tasks for the next Sprint, it only takes a couple of hours to further detail the backlog items into actual tasks.

At the Sprint Planning meeting the Product Backlog is broken down to smaller tasks of maximum eight hours. A single task can take more than eight hours if several developers share it – just as long as it can be carried out in one workday. The time estimation is typically carried out collectively with the use of Planning Poker, but according to Interviewee G2 it requires some training in order to estimate precisely.

Interviewee G2: "It may even take years. Time estimation is a totally new culture, and they [the developers] have never been used to that. We use Scrum in our own office, and we also face difficulties when we estimate time on tasks and activities."

4.7.5 SCRUM FRAMEWORK #2: SPRINT CYCLE

The Sprint Planning meeting is conducted in the team and due to the fact that most of the developers are working on several projects at a time, the amount of available resources is initially discussed. The individuals in the Development Team mention the hours that are available to them. This is used to estimate the extent of the Sprint Backlog. The Sprint Backlog is not described in features, but to a great extent in activities. According to Interviewee G1, this may just be a question of words. Even though the Scrum framework focuses on building functional features, it can be discussed whether it is the process or the product of the process that the team has to break down.

After the development of the Sprint Backlog, the work is initiated. The teams are in general trying to plan one or two full days each week where the whole team is working on the specific project, but the close contact and communication is typically hampered, as the team members are often spread out in several physical locations across Denmark. In practice this means that the team members are available on phone or Skype if activities need to be clarified across the team.

Daily Scrum meetings are held with varied frequency, depending on the specific project. Some teams have a weekly Scrum meeting, whereas others have it on daily basis or twice a week. Interviewee G1 argues that it has to be the teams themselves that drive the implementation of Scrum, so Scrum meetings are not a mandatory daily event; but the teams are allowed to find their own rhythms. This is also the case when deciding the Sprint lengths. Some Teams use two-week Sprints while others use three-weeks Sprints.

The number and frequency of the various meetings is continuously evaluated as the implementation of Scrum is still in its early stage. Some team members do not regard meetings as "real work," which is part of the reason for this seemingly loose implementation. Traditions in the company are strong, and pushing the implementation may just provoke a resistance in the development organisation. According to Interviewee G2, the implementation of Scrum might take five years, as it is not only a question about learning Scrum, but also a process of maturing the organisation.

4.7.6 DESIGN AND STYLING EFFORTS

Design and styling of the products is carried out according to a design guide developed on corporate level. This does not leave much

freedom to the Development Team in regard to those aspects. However, it is the responsibility of the mechanical developers in the Development Teams to follow the guidelines while also meeting the mechanical requirements. One of the few situations in which the Development Team is allowed to deviate from the design guide is when it conflicts with functionality or manufacturability. In such cases it is still representatives from the market division that have to weigh the arguments.

4.7.7 MAIN CHALLENGES EXPERIENCED BY INTERVIEWEES

When asked what they consider the main challenges of using Scrum in their client's product development, the interviewees mention a number of issues. The issues mentioned are described below.

SCRUM IN DISPERSED TEAMS

One of the most distinct challenges of implementing Scrum in the organisation of Company G's client is the condition of dispersed teams. This hampers a close collaboration and the important communication within the team. Interviewee G2 describes this challenge in the following quote.

> *Interviewee G2 "They are placed all over the country, and, believe me, sitting in a building across the street can be a long distance. Originally we had a vision about the developers working closely together in project rooms, but this is just not the way it works. They are divided into their respective areas of disciplines and scattered across the contry."*

As the quote indicates, Company G initially urged the development teams to sit together, but in practice this is not possible with the present organisational structure of the company.

TIME ESTIMATION ON CONCEPT DEVELOPMENT TASKS

The developers are seemingly not having difficulties in breaking down their development projects into small manageable tasks. However, when it comes to estimation of time needed to solve the tasks, the problems start to appear. During lunch break at one of the workshop observations a mechanical developer tells about his difficulties in finding a good solution to a certain problem. After coming up with more than 20 solutions and prototypes of the specific part of a product, he seems rather disillusioned, as none of the solutions have withstood a specific test. He cannot see how he could ever be able to estimate the hours and resources spent on that specific task in advance. His reluctance as to practicing Scrum is easily understood.

The statement from the mechanical developer is supported by Interviewee G2, who argues that the complexity of development tasks grows with the degree of unknowns:

> *Interviewee G2: "I think we have to dig into the innovation theory, because the question is whether it is radical or incremental development. It inevitably becomes much more difficult to estimate things when you are engaged in radical innovation. I think it is important to say that the complexity increases with the degree of innovation."*

TOP-DOWN

The implementation of Scrum is a decision taken by the upper management. This causes challenges in regard to introducing and maintaining the framework in the development environment. According to Interviewee G1, one of the greatest challenges is to find the right balance with a carrot-and-stick approach.

> *Interviewee G1: "How much should I push them? I have to push them a bit in order to make results and give them a feeling of progress and make them see Scrum as a benefit. But I cannot push them too much, as it will just affect their motivation negatively, and Scrum will by my method rather than their method."*

As it has already been stated, Interviewee G2 furthermore argues that it is likely that it will take approximately five years before Scrum is part of the culture.

TOO MANY PROJECTS PR. EMPLOYEE

The Company G client is organised as a project organisation, which means that all activities are focused around projects. However, the company's product portfolio is relatively large, and many development projects are running concurrently. This means that each developer or project manager is engaged in several projects at a time. This condition is a clear contrast to the guidelines of Scrum.

> *Interviewee G2: "Some people are engaged in 10 or 15 different projects, which eventually causes problems in relation to their level of involvement in the individual projects. Imagine what would happen if they had to take part in all the Daily Scrum meetings!"*

The engagement in many projects at a time hampers the focus on *one* project and creates a challenge in relation to team commitment. As everyone is only working part-time on a specific project, it is difficult to establish a "core team" and the necessary continuity that ensures a certain development flow.

4.8 SUMMARY OF CHAPTER 4

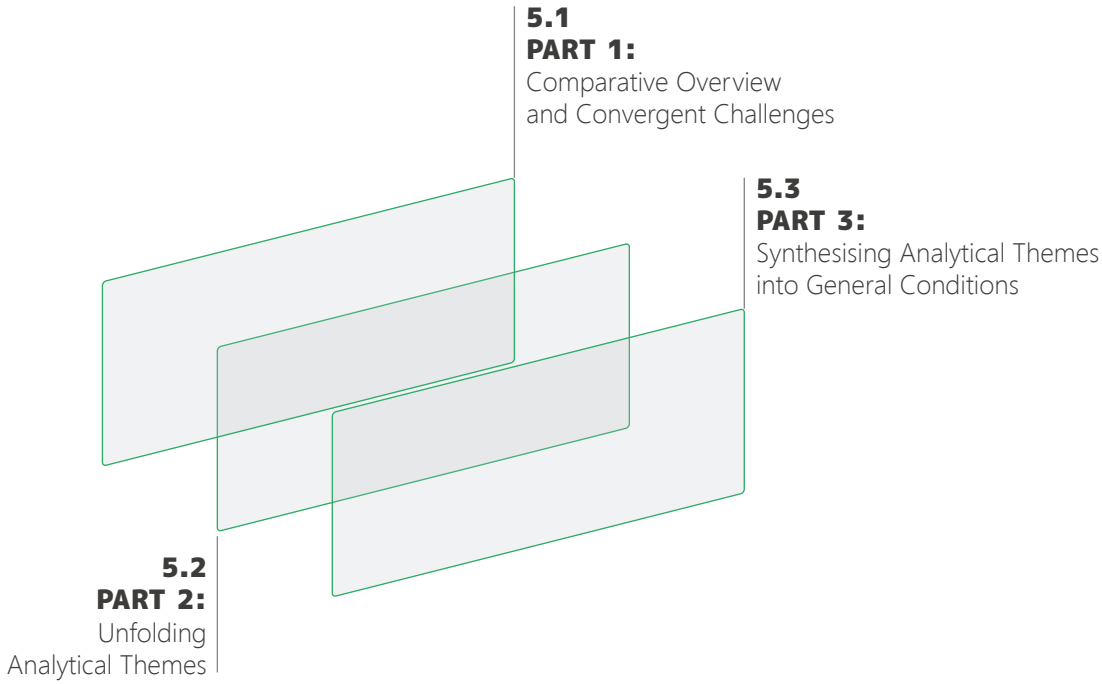
The objective of this chapter has been to present the empirical data collected through interviews and observations in seven Danish companies working with integrated product development. Each of the cases has been presented with quotations from interviewees, and all cases follow the same pattern. More specifically, the chapter includes the following items:

- A presentation of seven cases
- Overview of basic case data for each case
- Descriptions of development organisations and motivation of implementation
- Descriptions of how each company conducts Scrum in practice
- Relation to design and styling efforts
- Description of main challenges in each case

CH.05. ANALYSIS

Analysing empirical data is like putting together the pieces of a puzzle...

...The only difference is that in empirical research there is not only one "big picture." It may take different forms depending on how you turn the pieces.



5.1 OVERVIEW OF CHAPTER 5

Chapter 5 consists of three seperate parts of analysis

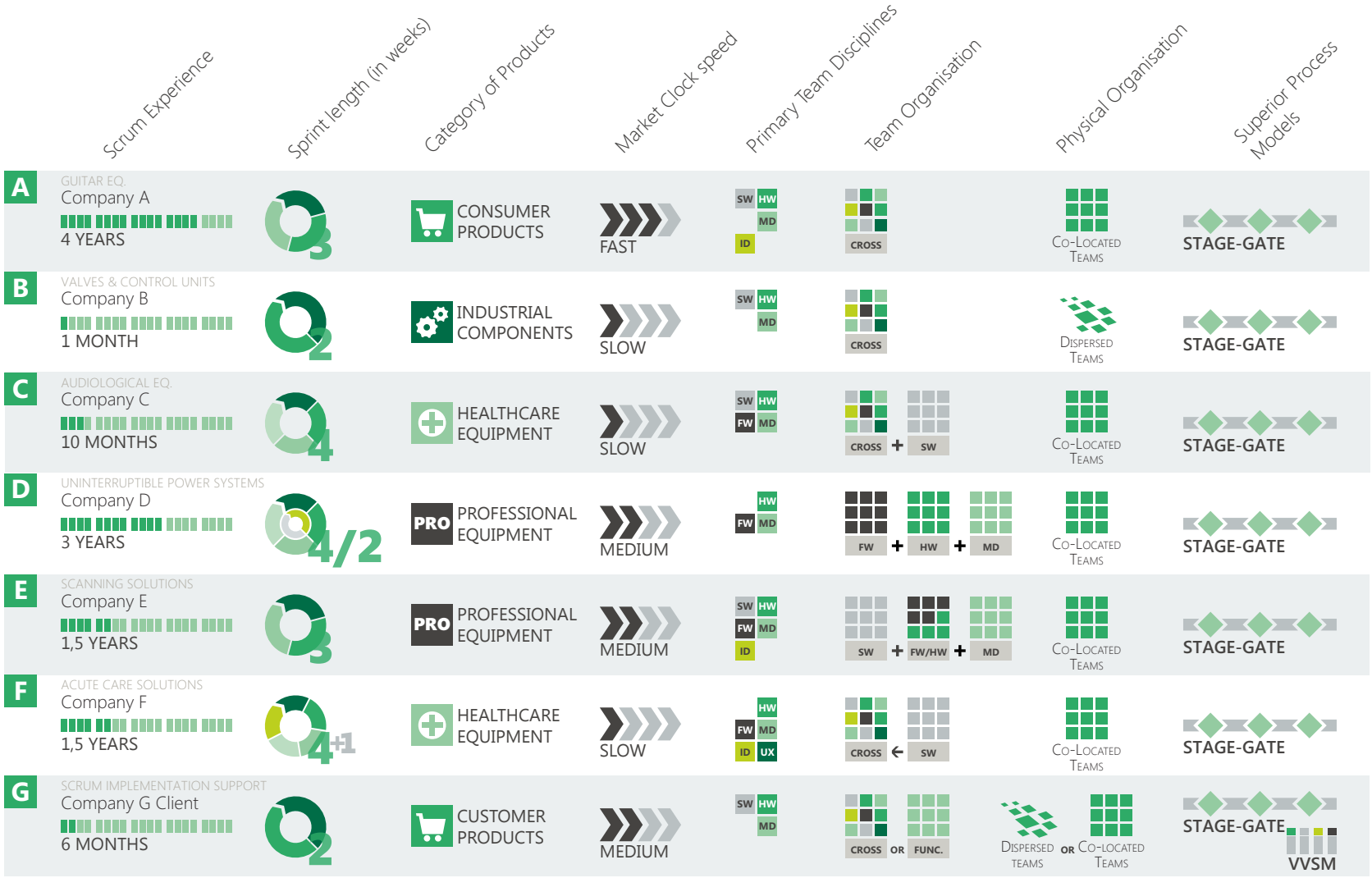
The objective of this chapter is to analyse the empirical data from the seven cases presented in chapter 4. The chapter is divided into three parts with the first part presenting a comparative overview of the cases, focusing on the basic case data, on the compliance with Scrum, and on a collection of convergent challenges identified in the data. The second part contains a series of eight thematically based analyses, which are all rooted in the collection of convergent challenges established in the first part. While the second part is unfolding the eight themes based on the *level of occurrence* of the separate challenges, the third part aims at creating cross-theme synthesis. The synthesis is achieved by cross-referring the analytical themes into a set of general conditions, which influence the implementation and conduction of Scrum in integrated product development. The results of the separate parts of the analysis are summarised in the subsequent chapter 6.

5.1 PART 1: COMPARATIVE OVERVIEW AND CONVERGENT CHALLENGES

This first part of the analysis consists of a series of comparisons between the seven cases from chapter 4. The various relationships are graphically illustrated and cover *basic case data comparison*, *compliance with the Scrum framework*, *compilation of convergent challenges as analytical themes*, and *per-case occurrence of the identified themes*.

5.1.1 OVERVIEW OF BASIC CASE DATA

The case overview in the illustration below compares Scrum experience, Sprint length, product categories, market clock speed, team compositions, physical organisation, and superior process control of the seven cases.



LEGEND: SW Software FW Firmware HW Hardware MD Mechanical Design ID Industrial Design UX User Experience

OVERVIEW OF THE BASIC CASE DATA
Comparison between the seven cases
5.2

5.1.2 CASE COMPLIANCE WITH THE SCRUM FRAMEWORK

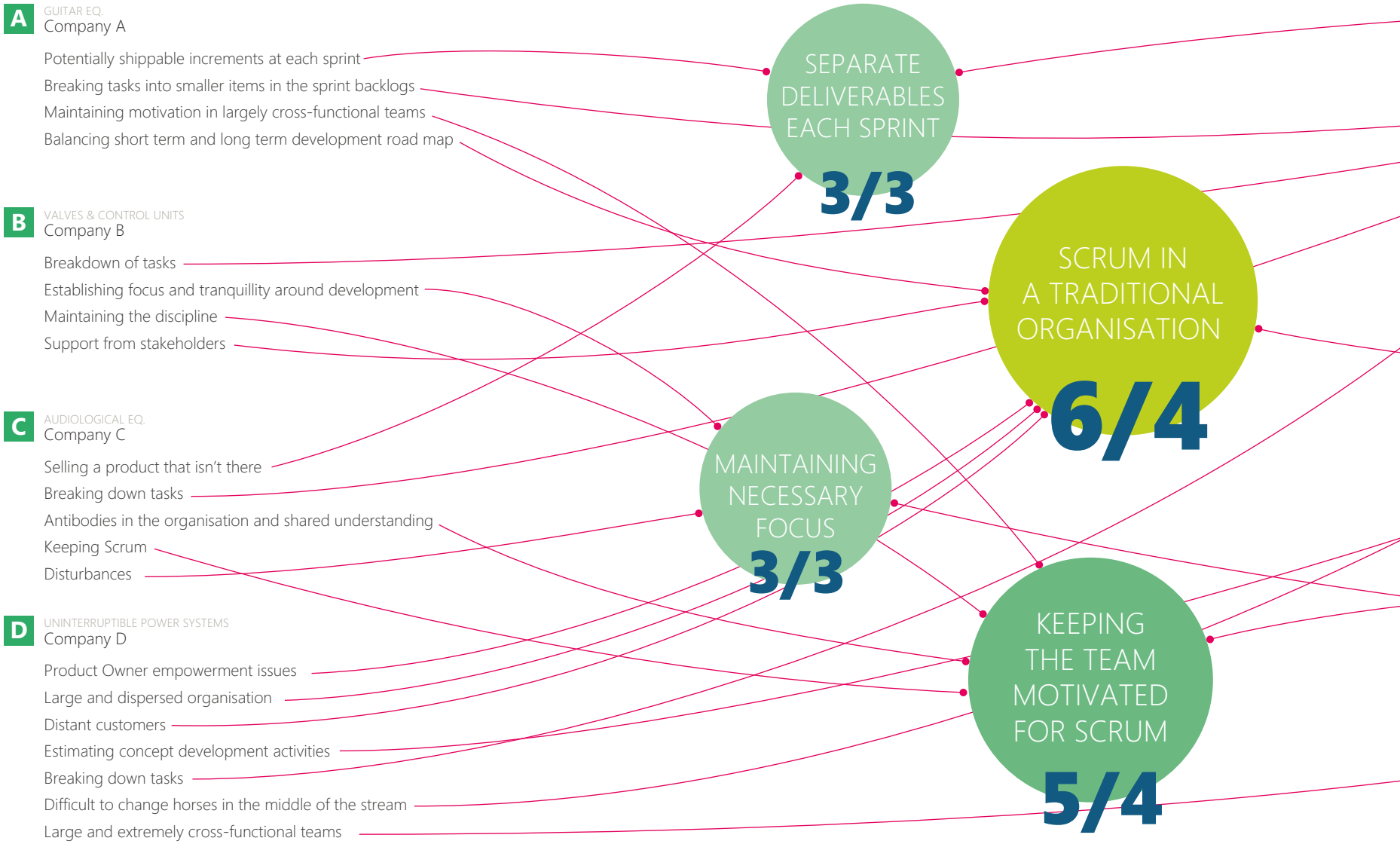
The illustration below presents how the seven separate cases comply with the roles, events and artefacts of the Scrum framework as defined by Schwaber & Sutherland (2011). An additional column with best practices has been added to the far right.

5.3 **CASE COMPLIACE WITH THE SCRUM**

Each case have its own way of interpreting and implementing Scrum

5.1.3 COMPILATION OF CONVERGENT CHALLENGES AS ANALYTICAL THEMES

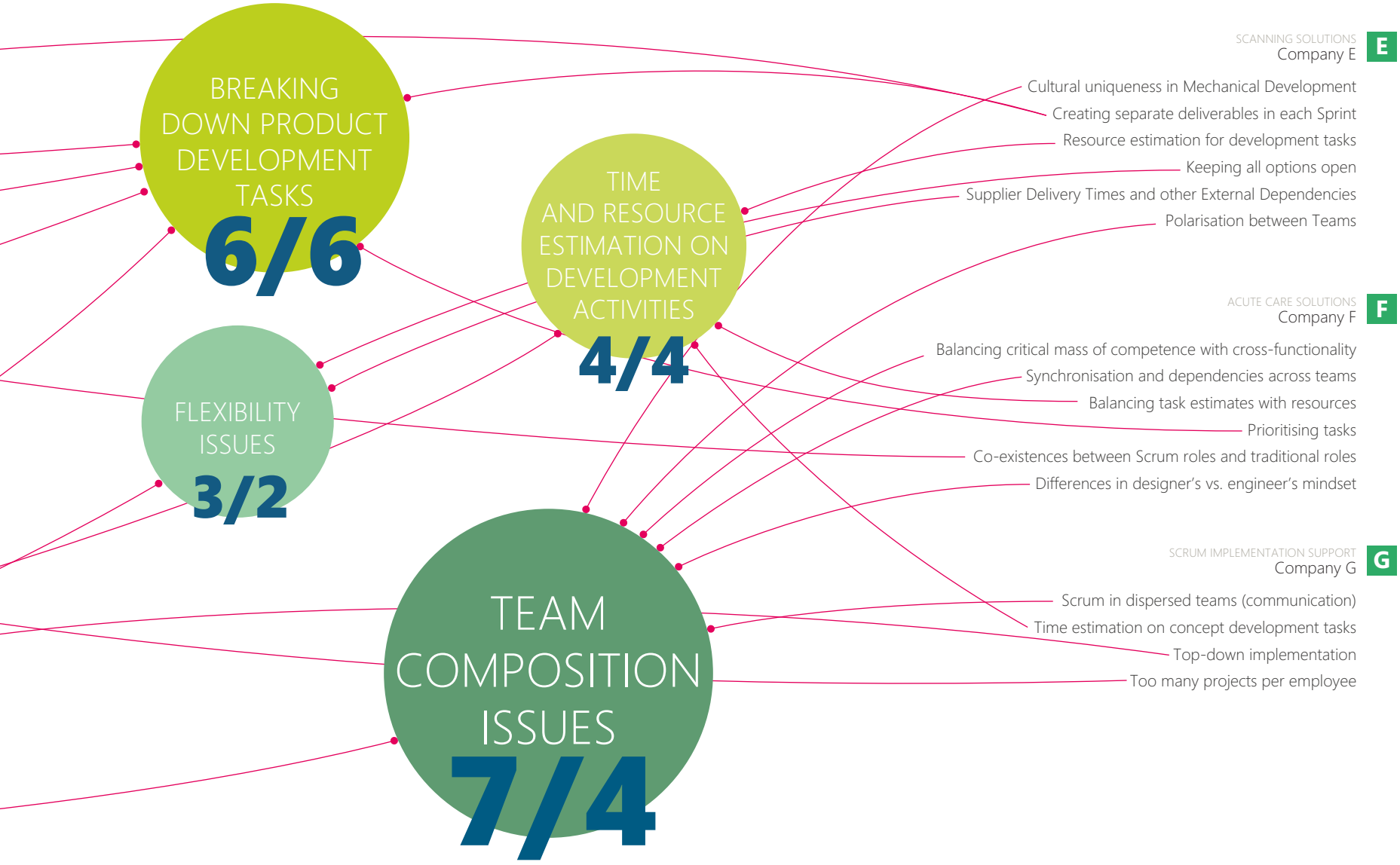
The illustration below represents the process of compiling the challenges identified in the seven cases into eight corresponding themes. The challenges from each case are listed in the left and right sides with lines connecting them to the corresponding theme. Each theme gathers challenges from two to six cases. The numbers related to each theme refer to respectively *number of identified challenges related to the theme* and *number of companies covered by the theme*. The theme "Team composition issues", for instance, is reflecting seven identified challenges from a total of four companies.



COMPILATION OF CONVERGENT CHALLENGES

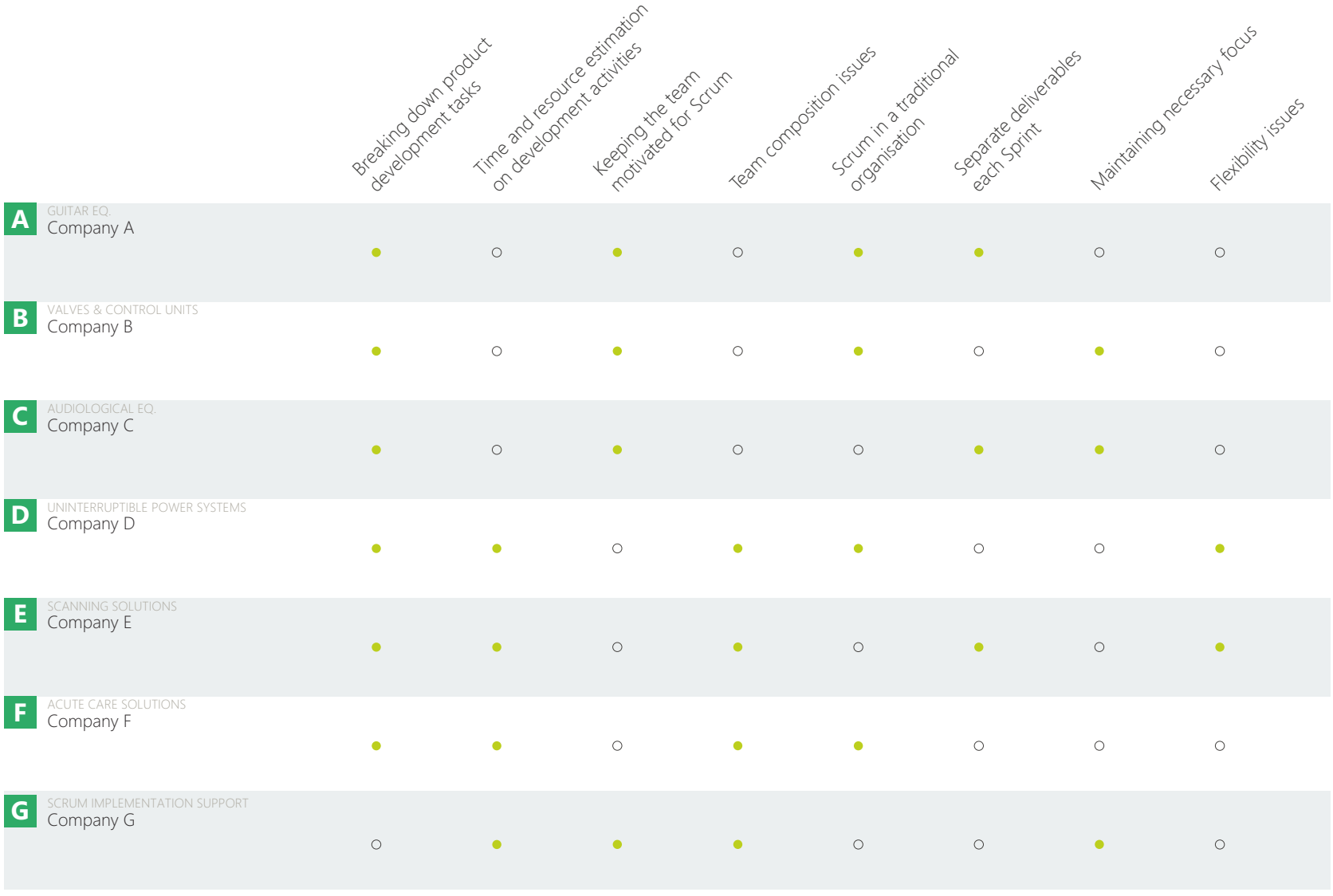
The challenges identified in the case data are gathered into 8 analytical themes

5.4



5.1.4 PER-CASE OCCURRANCE OF IDENTIFIED THEMES

The illustration below shows how the eight themes are distributed on the seven cases. The themes the left are the most consistent ones, whereas the ones to the right are less present.



PER-CASE OCCURRANCE OF IDENTIFIED THEMES
theme occurrance varies from 2 to 6 cases

5.5

5.2 ANALYSIS PART 2: UNFOLDING ANALYTICAL THEMES

This part of the analysis unfolds the eight themes identified in the first part of the analysis. All themes are analysed in the way, according to the following list of topics:

- Description
- Analysis
- Product Designer Perspective
- Organisational Perspective
- Viable Solutions
- Summary

Each thematic analysis is concluded with a summary and the overall results are presented in Chapter 6.

5.2.1 THEME 1: BREAKING DOWN PRODUCT DEVELOPMENT TASKS

DESCRIPTION

Just by looking at its massive occurrence in six of seven cases, the challenge of breaking down product development tasks seems almost inevitable. Furthermore, the essence of this challenge seems to lie extremely close to the core of the Scrum framework. A quotation from Company D describes it rather clearly. The quote both emphasises the importance of the breakdown of tasks *and* the fact that this is a challenge:

> *Interviewee D1 (Company D) "Breaking down ideas is what Scrum tries to do methodically. The idea you come up with has to be broken down as far as possible, in order for you to cope with it and estimate it correctly. That's where Scrum is great: you force people to break down the tasks as far as possible. It has been extremely challenging to do that."*

This analytical theme primarily contemplates the Sprint Backlog and the Sprint Planning event, which is taking place in the very first part of a Sprint. However, to some extent, it will also cover the grooming of the Product Backlog, which is typically done continuously as the development project goes on.

ANALYSIS

As mentioned, almost all companies recognise this challenge. The Company G case is the only one not mentioning it, which, in fact, may be an even more interesting finding. A possible reason for this will be unfolded later. The fact that the challenge of breaking down development tasks is present in six of seven cases makes it difficult to track down any distinct patterns or specific conditions that trigger this challenge to emerge. It seems to appear regardless of the level of Scrum experience, Sprint length, product category, market clock speed, or any team organisational aspect.

One aspect of the challenge is its occurrence; another aspect is how the challenge extends and peaks throughout the implementation and work of the Scrum framework. In other words: Is it a beginner's challenge that eases off as the Scrum Team becomes more experienced, or will it continue to be a challenge to break down development tasks?

According to several of the interviewees across the cases, breaking up development tasks seems to be a rather significant challenge that is more due to the nature of the development work than to the fact that many of the companies are relatively new to Scrum. Two of these interviewees represent the most experienced companies in regard to Scrum:

> *Interviewee A2 (Company A): "It is a challenge to split up the work in a way that makes sense, so that the individual parts are developed to a level sufficient for you to proceed to something else. This is a challenge."*

> *Interviewee D2 (Company D): "In the Mechanics team they would rather have a large pool of 254 hours and then put in as many*

tasks as they find appropriate, but that doesn't work as we cannot measure hours spent on documentation, concept development, and so forth."

However, another interviewee adds a nuance to that claim by arguing that the challenge of breaking down tasks is rather a question of change in attitude:

Interviewee B1 (Company B): "The greatest challenge, that I have noticed, is definitely the breakdown of tasks to deliverables that can be fitted into two or four week Sprints. It is a change in attitude, rather than a technical challenge."

This argument of relating the challenge of breaking down tasks to a certain attitude may not directly contradict the first argument, that the challenge is related to the nature of development work. As indicated by the Interviewee D2 in the previous quote, it seems that the challenge is primarily related to the mechanical development. This is confirmed by several cases. At Company E both interviewees bring forward the issue of a certain culture in the Mechanics team:

Interviewee E1 (Company E): "It has been pretty difficult to run Scrum in the Mechanics team, but I think it has something to do with the culture in the mechanics department in the company. I cannot see through how much of it is culturally determined or whether it is actually possible."

Interviewee E2 (Company E): "I see the mechanical part as some kind of an amoeba that develops in three dimensions. Your concept and its elements become more and more mature. You know what you have to do - you may look at the transmission part and do a first shot on this. Then it gets better and better and along the way other elements turn up. It ultimately ends with refined and three-dimensional models. It's like an amoeba. It is not possible to tie up any ends."

The interviewees at Company E describe how mechanical development differs radically from other development disciplines such as software and firmware development. In contrast to these disciplines, it may be argued, mechanical development has more internal dependencies. Everything is entangled, which means that it is difficult to separate one single part from the rest – the exact purpose of breaking down development tasks. Recalling the argument by Lawson (2005), it could be argued that the design processes in these companies' mechanical departments on the micro level resemble the one in his party-game-room-crashing metaphor, and that the formal task-breakdown process is simply too rigid.

PRODUCT DESIGNER PERSPECTIVE

It is clear that the challenge of breaking down development tasks is directly affecting the developers working with Scrum, and seen from the perspective of the product designer, it may seem difficult to commit to the practice of Scrum as the breakdown of task is arguably going against the nature of their way of working. This assumption is confirmed by several cases.

As earlier stated in a quote from the Company A case, the difficult part is to split up the development into small parts that make sense. This may lead to the risk of breaking down tasks into too large pieces, as mentioned by Interviewee E2 from Company E:

Interviewee E2 (Company E): "It is very important how you define your tasks in order to be able to remove them from the board after three weeks. Quite often we have ended up defining tasks on a too high level, which has resulted in Post-it notes staying on the Scrum board for too long – and so we have missed the point."

Interviewee C1 from Company C also argues that it is difficult to break down a 100-hour task into sub-tasks of less than 12 hours. However, defining tasks too broadly affects the Scrum practice negatively as it hinders a precise *measure of progress*, and, which is perhaps more important, reduces the *sense of progress*. It furthermore makes the estimation of the separate tasks more difficult.

ORGANISATIONAL PERSPECTIVE

Breaking down development tasks may not seem as a big problem in an organisational perspective, and it can easily be argued that this challenge is solely related to product development activities. However, it is a fact that the challenge is not evenly shared across the development disciplines; but it is clearly most distinct in the mechanical development environment. This could raise some concern relating the organisational aspect.

According to the Company E case, the Mechanics team has a special culture, which may result in a polarisation within the Development Team. So the Mechanics team may become the black sheep:

> *Interviewee E2 (Company E): "When it really works well here and not that well there, it might create some kind of polarisation. There's the team that knows how to do it, and then there's the team that doesn't. I think we need to learn how to handle this in a way. We need to get an understanding of the basic differences. Why we act in this way and you act in that way – as part of the same overall team. That's my impression."*

In the quote above, the Interviewee E2 takes an organisational position, when emphasising the importance of understanding the differences in how the various professional disciplines handle the work with Scrum. This is important in order to avoid a stigmatisation of the Mechanics team. However, this discussion is closely related to team composition and will be unfolded as part of the analytical theme focusing on that.

VIABLE SOLUTIONS

Though this challenge is common across almost all the cases, it does not seem to be completely disruptive. In all cases the development activities progress through the Sprint cycles in spite of the struggle with breaking down the tasks. This fact shows that the Scrum teams to some extent have found ways to overcome the challenge. And the fact that the challenge has not even been mentioned in the Company G case, indicates that they have found a viable solution.

One explanation to the unproblematic breakdown of development tasks in the Company G case may be found in the only distinctive difference between this case and all other cases: the use of an extra level of process management in-between the overall Stage-Gate model and the Scrum framework. Company G has introduced a Vertical Value Stream Mapping process, which assists the Development Team in breaking down and in coordinating the separate stages in the Stage-Gate model. The VVSM events help the Development Teams to communicate and coordinate the right sequence of the development activities across the functional units. However, even though relation between VVSM and a successful handling of this challenge may seem feasible, the empirical evidence supporting this claim is still too fragile. The fact that the interviewees at Company G have not pointed at the breakdown activity as a major challenge may also be a result of other aspects. One may be that the interviewees are a sort of third party, in the sense that they act as external consultants for their client and therefore risk missing or disregarding certain challenges. Another explanation could be that other challenges of implementing Scrum at the client overshadow the process of breaking down tasks.

SUMMARY OF THEME 1

- The challenge exists regardless of the companies' levels of experience with Scrum
- The challenge seems to be most evident in the mechanical development environment

Breaking down development tasks is a challenge to the product designer because:

- Mechanical designs and hardware-related elements consist of interrelated and integrated subparts that are not easily detached from each other.
- A unique culture or attitude may exist in the mechanical development environment, which is an obstacle in the exhaustive breakdown of tasks.

Breaking down development tasks indirectly implies the risks of becoming an organisational challenge, because:

- An unfruitful struggle with this challenge in especially mechanical environments may result in a polarisation of the broad development team and a disbelief in Scrum as a viable solution in mechanical development.

5.2.2 THEME 2: TIME- AND RESOURCE ESTIMATION OF DEVELOPMENT ACTIVITIES

DESCRIPTION

The challenge of estimating time and resources of development activities is closely related to the challenge of breaking down development tasks. The two challenges are found at the same stage in the Scrum framework, namely at the Sprint Planning meeting in the very first part of the Sprint, and both challenges concern the handling of development tasks.

The challenge is present in four out of seven cases, which makes it a relatively central aspect of the implementation of Scrum. However, the fact that three cases do not point out estimation as a challenge indicates that there might be ways to avoid it. This analytical theme covers the challenges of estimating both time *and* resources, as it is argued that the underlying aspects nourishing these two parts are basically the same.

ANALYSIS

Plenty of examples of the two challenges are given in four out of seven cases. However, just as in the first theme it is rather difficult to identify any characteristic pattern which the four companies have in common. Despite the fact that all four companies are also the ones experiencing challenges with team composition – the sixth analytical theme – no real patterns appear. Once again, the level of experience is varying from relative experts to beginners in Scrum, and no aspect of team organisation, such as physical or functional distribution, seems to present any pattern.

However, when looking into the nuances of the quotes from the various cases, they reveal certain interesting points in regard to this challenge.

Interviewees from Company E and Company F reflect upon the basic difficulty in estimating work. Both interviewees seem frustrated about the lack of precision and transparency in estimations in general:

> *Interviewee E1 (Company E): "(...) we always just finish 50 or 60 % of what we planned. But still, the product gets finished in time. How does that come about?"*

Interviewee F1 (Company F): "(...) it is really difficult to tell if there's a too big workload on one individual. And we don't want to have a large bureaucratic burden of counting and calculating workload."

Comparing the level of Scrum experience of these two companies, it is striking that both of them have only been in the process of implementing Scrum for the last one and a half year. Both quotes could therefore be reflecting a lack of experience with quantifying even simple development activities to precise estimations. This claim may be underlined by the fact that the two companies that are most experienced in Scrum in general are able to estimate tasks rather precisely. In the quote below, Interviewee D1 from Company D reflects upon the challenge of estimating tasks and reveals a rather important detail, namely that the character of the task has a vast influence on the estimate:

Interviewee D1 (Company D): "Some of the concept development takes place inside the project as it sometimes happens that you have a challenge which just cannot be solved the way you expected. Then it is back to the drawing board. We have a lot of examples of tasks that have been estimated with extreme inaccuracy, because you are actually working on concept level and really don't have a clue about how long it will take. We can easily see it by the count afterwards – it is the stuff that we don't know about that makes our overall calculations go wrong."

Interviewee D1 raises the aspect of uncertainty in relation to making estimates of tasks that have the character of concept development. It clearly shows an awareness and acceptance towards the fact that tasks with factors of great uncertainty are difficult to quantify, which is not found in the quotes before it. This aspect seems to relate nicely to the complexity theory by Snowden & Boone (2007) presented earlier en this thesis. The development activities are balancing between being complicated and being complex – In other words between the domain of experts, consisting of *known unknowns*, and the domain of emergence, consisting of *unknown unknowns*. As Interviewee D1 from Company D put it, *"Estimating the extent of something that you don't have a clue about is essentially difficult, if not impossible"*.

A last aspect of the challenge of estimating time and resources on tasks is related to the quote from Company E, revealing that the accuracy in their estimates is between 50 and 60 %. In the following quote he argues that certain constrains of physicality exist when working with physical aspects of a product:

Interviewee E1 (Company E): "You will never get to the lowermost items in your backlog. I see it work in Software, but when you get to mechanics and electronics, there's just things that you cannot leave out. You HAVE to get through all the things and that's the difficult part of working time boxed. If you stumble upon some problem that you just cannot solve, well, then it just takes more time."

Once again, a challenge is specifically related to the mechanical development environment, just as it has been the case with the breakdown-challenge unfolded throughout the first analytical theme.

PRODUCT DESIGNER PERSPECTIVE

Clearly, estimating tasks is a practical issue of conducting Scrum that primarily belongs to the developers. The quotes in the previous subchapter reveal frustrated and disillusioned developers struggling with this challenge on several levels. It is argued that the challenge includes at least three aspects, which are 1) resource and time estimation in general is difficult if the developers have no or little experience in it. This often results in estimates being too optimistic; 2) some development tasks, such as concept development activities, have a complex character, which make them extremely difficult to quantify, and 3) estimating tasks in the mechanical environment is a greater challenge as you cannot leave anything out.

ORGANISATIONAL PERSPECTIVE

From an organisational perspective, the developers' struggle with time and resource estimation primarily seems to be a problem in regard to monitoring progress of the development activities. However, in cases where the top management has introduced Scrum in the organisation, the estimation challenge, most likely, will be laid on the Development Teams with the purpose of forcing developers to become more aware about how hours and resources are spent. The Company G case is an example of how the management has initiated the implementation of Scrum in order to increase product development efficiency and to force developers to become aware that resource and time consumption is part of it.

VIABLE SOLUTIONS

As it has been mentioned in the beginning of this analytical theme, some of the involved companies are using specific methods to overcome this estimation challenge. At Company A the teams use the game of Planning Poker in the Sprint Planning meeting. Planning Poker is a simple and collective task-estimation game, which facilitates the discussion about task estimation in the team:

> *Interviewee A1: "We challenge each other in estimating needed resources regardless of how detailed or roughly the backlog items are described"*

The point of this game is to continuously become better at estimating tasks in the Development Teams, and to keep a common understanding of the project in development across the involved disciplines.

Though the game of Planning Poker may be a good solution for most estimation activities, it probably comes short when estimating the previously mentioned *complex tasks*, which often have the character of concept development. In that case a distinction between projects may be necessary. When talking about this subject in a break from the interview at Company E, Interviewee E1 revealed that some projects at Company E are marked as technology projects. Even though these projects have a lower priority on a daily basis, they are characterised by being less product-oriented and more focused on concept development and learning. The same pattern is seen at Company F. Just as the complex tasks, such as learning- and concept-focused projects, are characterised by being difficult to estimate. However, it may be a viable solution to have some sort of distinction between known tasks, which can be estimated rather precisely by experience, and tasks that are concept development-oriented and imply a great level of uncertainty.

SUMMARY OF THEME 2

- The challenge of estimating tasks exists, no matter how much experience the different companies have with Scrum.
- However, experienced companies seem to be able to carry out estimation of simple tasks and handle tasks with large factors of uncertainty (e.g. complex tasks) in a different way.

Time and resource estimation of development activities is a challenge to the product designer because:

- Developers may generally lack experience in estimating work as the level of detail required in Scrum may be unfamiliar to them.
- Concept development tasks, characterised by many unknown unknowns, are difficult – if not impossible – to estimate.
- It is argued that the challenge of estimating development tasks is greater to mechanical development environments than other environments.

Time and resource estimation of development activities is an organisational challenge because:

- Bad estimates may make it difficult for the management to correctly monitor the progress the development work.

5.2.3 THEME 3: KEEPING THE TEAM MOTIVATED FOR SCRUM

DESCRIPTION

The effort to keep the team motivated for Scrum focuses on various aspects related to the implementation of the framework. This analytical theme therefore covers aspects such as changing habits, lack of persistence, antibodies and top-down management. The theme is based on data from four of seven cases.

ANALYSIS

Just as in the previous analytical themes it is difficult to trace clear patterns binding together the four companies that have pointed at this challenge. Once again the challenge covers a broad range of Scrum experience from the company less experienced in Scrum to the most experienced one. This might indicate that a continuous effort is needed in order to stick to Scrum in the organisation. One rather vague pattern connecting the four cases is that the four development environments are all organised in cross-functional teams. A fifth case fails to match the pattern: Company F, just as the four cases, deploys cross-functional teams, but interviewees from that case have not indicated any issues related to the challenge of keeping the team motivated for Scrum. However, the pattern is supported by the findings of Keller (2001), who argues that the team cohesiveness may be affected negatively by this cross-functionality and that the result may be lack of a shared language.

At Company A, Interviewee A1 acknowledges the fact that, on the whole, cross-functional teams may influence the commitment to the Scrum framework, as developers have difficulties in speaking a common language:

> *Interviewee A1 (Company A): "At the Daily Scrum meeting it can be a challenge that some don't understand why they have to listen to what everyone else in the team is doing, when they are so specialised into different areas as they are. (…) When we include hardware, we have even more cross-functional projects. There are some issues there."*

This quotation – as well as the findings of Keller (2001) – emphasises that the deployment of largely cross-functional teams is one aspect of the challenge of keeping the team motivated for Scrum. However, the reason why motivation is a challenge may also be found in the fact that organisational change will always entail a certain amount of resistance – especially if the demand for change is a top-down initiative from the part of the management. Several quotes are supporting this argument. In the following quote Interviewee C2 from Company C indicates that the transition to Scrum has not been without certain difficulties:

> *Interviewee C2 (Company C): "I think it is a challenge to stick to Scrum. It was forced upon us, and now we do it in some sort of watered-down way, which is ours. We do it because we think it works a bit, but if the people who, in the first place, forced us to it lose interest, then I think it will be difficult to preserve it – and to improve it. If we don't always correct ourselves and make corrective actions; then, what is our benefit when we are done?"*

The discussion with the mechanical developer at the Company G client also enforces this argument about the existence of antibodies when implementing Scrum. However, in the quote from Company C, Interviewee C2 is not purely negative towards the initiative, but emphasises the importance of a supporting organisation. Equally, according to Interviewee G2 from Company G, it is the intensive teaching, coaching, and support that has made the developers finally welcome the Scrum framework.

Even though fighting antibodies in the implementation process may be a potential struggle, other challenges, just as subtle, are *habits* and *lack of discipline*. The last aspect of this challenge unfolded here is the danger of falling into old habits, which is exemplified in a quote from Company B:

Interviewee B1 (Company B): "Discipline is definitely a challenge. (…) You start out by saying that we are going to do this, but eventually we fall back into our habits. (…) It is the human aspect in it. It's the habits you need to change. It takes time."

In contrast to the other three companies involved in this challenge, the Scrum-implementation process at Company B is initiated as a bottom-up initiative. Despite this fact, according to the Interviewee B1, it is difficult to maintain the discipline needed for Scrum and keep away the old habits.

PRODUCT DESIGNER PERSPECTIVE

While the two previous challenges have been largely related to the practical aspects of adhering to the rules of Scrum, this challenge of keeping the team motivated for Scrum is perhaps more related to the human aspects of meeting change rather than the practical-technical issues in the Scrum framework.

During the interviews it became evident that several of the interviewees across the various cases have opposed the implementation of Scrum in their respective development environments, and some are still struggling to adjust themselves to it.

ORGANISATIONAL PERSPECTIVE

The challenge of keeping the team motivated for Scrum is primarily an organisational challenge. The quotes from the sub-chapters above show the importance of continuous support of – and attention to the developers who are affected by the implementation process. According to Interviewee C2 from Company C, there is a real risk of the team retreating to the old development process, if the decision makers lose their interest in Scrum, or if they do not pay attention to the implementation process.

VIABLE SOLUTIONS

Keeping the team motivated for Scrum is clearly a fundamental necessity for a successful implementation of Scrum, and it can be argued that all seven companies are still in this process. However, despite the fact that all seven companies are struggling with a variety of challenges in this regard, it is also clear that the companies that have most experience with Scrum are the ones with the deepest and smoothest integration of Scrum. Time seems to have an effect, which makes a persistent management a most important aspect in order to keep Scrum. The Company G case is perhaps a good example of this, as Company G has been asked to facilitate the implementation of Scrum over a longer period of time ensuring a continuous attention, training, and support to the development environment.

SUMMARY OF THEME 3

- The challenge of keeping up motivation exists at all levels of experience with Scrum, however –
- The most experienced companies are the ones with the deepest and smoothest integration of Scrum.

Keeping the team motivated for Scrum is a challenge to the mechanical developers because:

- Very cross-functional teams may hinder the individual developer's commitment to Scrum, due to lack of a common language.
- Habits may result in developers falling back into the old development practice

Keeping the team motivated for Scrum is an organisational challenge because:

- Continuous organisational support to the development environment is necessary in order to overcome the resistance to the implementation of a new development framework

5.2.4 THEME 4: TEAM COMPOSITION ISSUES

DESCRIPTION

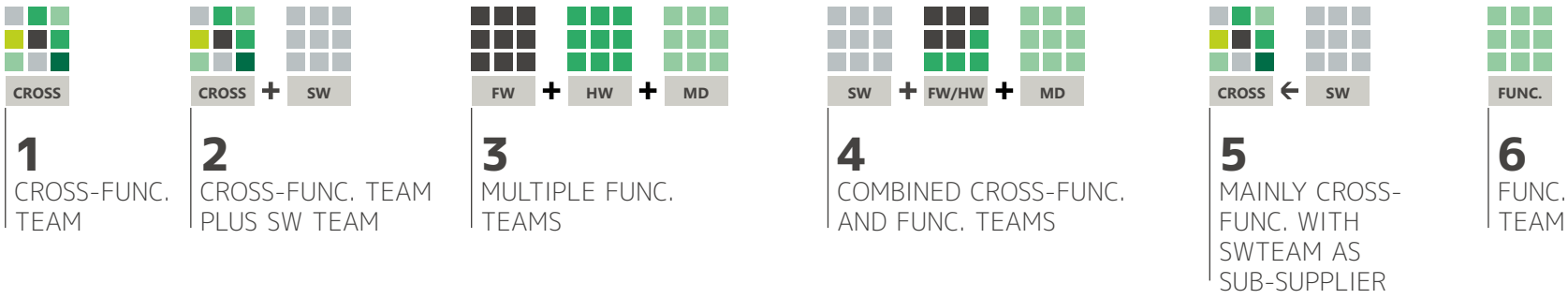
This analytical theme is based on excerpts from four of the seven cases and covers a number of issues in regard to team composition and organisational conditions affecting the Scrum development teams. These issues are concerned with the question of cross-functional- versus functional teams, team synchronisation, cultural uniqueness, and issues related to dispersed teams.

ANALYSIS

Not two organisations are identical, and team composition is highly dependent on the organisational setup in the respective companies. This may also be the reason why all seven cases in this research project are different in regard to how development teams are set up. The four companies with team composition challenges do not seem to have any other significant correlations than the fact that they all tend to deploy large teams or multiple teams working on the same project. Beside that, they all differ in regard to their level of compliance to Scrum, their experience with Scrum and their respective team compositions.

The official Scrum guide promotes cross-functional teams, and in software development, which is the original domain of the Scrum guide, cross-functionality means a mix of disciplines *within* the software domain. In integrated product development cross-functional teams entail a significantly larger variety of involved disciplines. This fundamental difference in the transition from software development to integrated product development clearly has some consequences to the teams in the investigated cases. As it has already been mentioned earlier, extremely cross-functional teams have certain communication issues, which, in some cases, lead to a drop in motivation. The fundamental difference between the two domains is that in software, team members have software development as a unifying discipline; the majority of a cross-functional software development team has to some extent overlapping competences. This is not necessarily the case in cross-functional teams in integrated product development, which may very well include software, firmware, hardware, mechanics, industrial design and more.

The Scrum Guide requests cross-functionality on one hand and a large amount of involved disciplines on the other. In most cases this would compromise the guide's recommendations in regard to team size. This dilemma has been solved in different ways as illustrated in Figure 5.6 below.



5.6 **DIFFERENT COMBINATIONS OF TEAM COMPOSITION**
The six different team compositions above are all represented in the seven cases

As it is seen in figure 5.6 on the previous page, the teams are composed in different ways in almost all seven cases. No universal solution seems to exist. Interviewee F1 from Company F reflects on the difficulty of having a cross-functional team in an integrated product development project:

> *Interviewee F1 (Company F): "In the software silo there is a critical mass of developers with the same competences. In our project team we can easily be just one electro technician, one chemical engineer and one mechanical engineer. They just really can't share tasks other than getting the coffee."*

As indicated by Interviewee F1, the cross-functional team is not able to achieve the same collaboration synthesis as is achievable in the software development silo. This might very well be the reason why the organisation is slowly starting to build up competence silos of mechanics, firmware and chemistry similar to the software development silo. In the Company D-case, which describes three synchronised and functionally divided teams, Interviewee D2 argues that even with functional teams the close collaboration may be difficult, due to the need of a broad variety of experts:

> *Interviewee D2 (Company D): "I have six developers on the Mechanics team, all with very different competencies, so this is not true Scrum. I cannot just put anyone onto a certain task, and that's a challenge. One person does all the plastics and another does this and that."*

Together, the two quotes above reflect some of the difficulties in composing teams in integrated product development. It can be argued that two different paths can be taken, when composing Scrum development teams for integrated product development of a certain complexity:

- Several parallel functional teams
- Large and extremely cross-functional teams

In this regard, the challenge is to balance cross-functionality with a certain critical mass of homogeneous competence in the development teams.

In both the Company F case and the Company D case multiple Scrum development teams are working together on the same project. However, the two companies are handling this with varying success. While the teams at Company D are systematically communicating through both Scrum Masters and System architects, Company F has not yet established a formal way of synchronising collaborating teams. At Company F, the teams are not conducting the Sprints synchronously; and the fact that the software development silo is supporting several other development projects, makes a close collaboration a big challenge:

> *Interviewee F1 (Company F): "If we just focus on our own little area of competence, we deliver in time for sure. We just don't deliver it in time to those who need it, and that's a really big challenge."*

The difference in the two cases may be explained by the difference in the experience with Scrum. While Company D has had a couple of years more to establish an efficient way to work with multiple teams, Company F may still be in the process of implementing the basic Scrum process within the separate teams.

At Company E the Scrum development teams are also divided by disciplines. In this case, a project room with a common Scrum board and visible task descriptions across all the involved Scrum development teams supports the communication and synchronisation between the teams. However, in the Company E case the division in functional teams leads to another team composition issue:

> *Interviewee E2: "When it really works well here and not that well there, it might create some kind of polarisation. There's the team that knows how to do it, and then there's the team that doesn't. I think we need to learn how to handle this in a way. We need to*

get an understanding of the basic differences. Why we act in this way and you act in that way – as part of the same overall team. That's my impression."

According to Interviewee E2, the Mechanics team has difficulties in sticking to the Scrum practice in the same way as the other teams, and this creates a polarisation in the overall development team. In the quotation of Interviewee E1, it is argued that it has something to do with a certain culture in the Mechanics team. In the same way Interviewee E2 describes mechanical development as an amoeba, which could be interpreted as if mechanical development were beyond any rational structure or logic.

To some extent, the distinctive character of the Mechanics team is also seen in the Company F case, where Interviewee F1 argues that the Industrial designers working with the development team are beyond any pedagogical reach:

Interviewee F1 (Company F): "We have this group which is simply just beyond any pedagogical reach [pointing at Industrial Design and Usability]. There's no planning, things turn up from day to day, and they expect that everything can be done by tomorrow."

In both cases, the specific groups have difficulties in adjusting themselves to the rather strict practice of Scrum, which may somehow result in a polarisation of the team.

The last team composition issue brought forward here concerns dispersed versus co-located teams. The majority of the development environments from the seven cases are co-located; but the Company G case and the Company B case include dispersed teams. It seems rather clear that dispersed teams are a challenge as Scrum is promoting a relatively close contact and daily communication within the team. The Company C case is an example of an organisation starting out with Scrum in dispersed teams. However, as it has also been described earlier, the development department was quickly reorganised and the Scrum development teams were gathered in co-located teams, which improved the situation considerably. This is in line with the insights presented by Hoegl et al. (2007) about dispersed teams in chapter 2.4.4.

Interviewee G2 from the Company G case describes the difficulties that their client faces:

Interviewee G2: "They are placed all over the country, and, believe me, sitting in a building across the street can be a long distance. Originally we had a vision about the developers working closely together in project rooms, but this is just not the way it works. They are divided into their respective areas of disciplines and scattered across the country."

PRODUCT DESIGNER PERSPECTIVE

Seen from a developer's perspective, collaborating in a cross-functional team can be just as difficult as it can be enriching. In Scrum the development team collectively commits itself to the chosen Sprint backlog, and in cross-functional teams it may be difficult to the individual developer to assess the extent of the tasks outside his or her own area of competence, just as the communication in such a team may prove difficult. The discussion about whether functional or cross-functional teams are preferred is much influenced by the organisational conditions, but it is also a matter of the type of the developers. According to Interviewee G2 from Company G, smaller development environments tend to have a preference for generalists, whereas large development organisations often have a lot of specialists. This may very well influence the way the developers prefer to work.

ORGANISATIONAL PERSPECTIVE

As it was mentioned in the beginning of this analytical theme, all seven cases have organised their respective Scrum development teams in different ways. Some are organised across multiple teams and others in just one single team. This shows that there are several strategies for conducting Scrum, and it shows that each company has to find a way to set up the Scrum development teams that is right

for that specific situation. The concept of cross-functional teams is promoted in the original Scrum guide, but in integrated product development it seems to almost always be preferred. Even if the teams are functionally divided, a certain cross-functionality seems to be inevitable. However, seen from an organisational perspective, the challenge in regard to team composition seems to be to compose the right teams based on the available resources, taking into account cultural differences and domain-related difficulties, and lastly to ensure synchronisation and close communication between teams working together on the same project.

VIABLE SOLUTIONS

Clearly there are plenty viable solutions. The challenge is due to the fact that not two organisations are alike, and that each organisation has to find its own way to fulfil the principles of Scrum when setting up development teams. Nevertheless, there are still *some* ways and experiences shown by the cases, which may generally improve the situation.

Interviewee E2 at Company E has a point when he says that it is important to be aware of certain cultural differences between the involved disciplines and their respective abilities in regard to complying with Scrum. This may be a key to prevent a polarisation in the overall team. It is equally important to avoid too large and too cross-functional teams. Balancing team size and critical mass of competence seems to be an important aspect.

5.2.5 THEME 5: SCRUM IN A TRADITIONAL

SUMMARY OF THEME 4

- The challenge exists regardless of the team size and team composition, however
- Companies with multiple, collaborating teams on the same project seem to be over-represented.

Team composition is a challenge to the product designer because:

- The amount of special expertise needed in integrated product development may require too many specialists for a certain team spirit and common commitment to be achieved.

Team composition is an organisational challenge because:

- No one-size-fits-all solution exists.
- Each organisation has to set up Scrum development teams in a way that takes into account the available human resources and competences. Functionally divided team and cross-functional teams may both be viable paths.

ORGANISATION

DESCRIPTION

Most large organisations have been developing its values and traditions over time. A certain identity may be shared by the employees and a shared history of previous successes and failures, customs and practices may be exactly what keeps the organisation together. When implementing Scrum in such organisations rich in traditions and well-rooted practices, it will inevitably cause some adjustment. The organisational legacy may be put the test, as Scrum, in many ways, is a new and disruptive element in the organisation.

This analytical theme covers a series of issues related to the implementation of Scrum in organisations with traditions and established processes and practices. The issues are identified in the descriptions of four out of seven cases and cover support from stakeholders, roles and responsibilities, the organisational setup, and line of command, as well as the radical changes in planning and handling development activities.

ANALYSIS

When implementing Scrum in an organisation with a traditional management structure it will require some adjustment of one or both. Just as traditional management, Scrum has a portfolio of formal roles that are required in order to be conducted properly. As it has been argued earlier in chapter 2.4.3, Scrum roles and traditional management roles, such as the project manager role, have a certain overlap, but they may be said to represent two different paradigms in development. While the responsibility for the execution of projects is born by the project manager in the traditional management model, project management is a shared responsibility in the self-organising Scrum development team as argued by Rudman (2010). The Scrum Master is only facilitating the process, and the Product Owner represents the customer and the business perspective.

In most of the investigated cases, the Product Owner is absent, and often the traditional management model and the Scrum framework only flank each other in the presence of one single person playing the dual role as both Project Manager and Scrum Master. However, according to the data obtained through the cases, this dual role does not seem to cause any significant problems. On the other hand the role of the Product Owner – perhaps more rightly the absence of it – seems to cause some frustrations in the development teams in at least two of the cases:

> *Interviewee D1 (Company D): "There have been a lot of battles and they have taken a huge amount of time – at that point it would have been nice if an actual Product Owner would have taken those decisions."*

> *Interviewee F1: "As a Scrum Master I miss some inputs to a Product Backlog. In reality our projects operate without Product Backlogs."*

In both cases the lack of a Product Owner results in frustrated Scrum Masters and Development Teams, and without a Product Owner taking the responsibility of managing the Product Backlog, this task trickles down to the development team. Due to the long lines of command in the surrounding organisation, the teams at Company D found it difficult to manoeuvre and take the necessary decisions in the extensive hierarchical organisation:

> *Interviewee D2 (Company D): "Our Product Line Manager, who owns the product when it goes to market and who talks to marketing and customers, needs to be located close to the Scrum team in order to have the essential and daily communication [with the team] while managing this Product Backlog. Right now he sits in India and is extremely difficult to reach."*

> *Interviewee D1 (Company D): "(...) Above him there is a Line man-*

ager for General Purpose and over him is the Portfolio Manager. So, as you see, this part of the organisation is enormously heavy"

The Scrum framework does not give any guidance to how it should be implemented in a large organisation. This is decided by the management, and to some extent also the development environment in which it is implemented. The frustration found in the quotes above is not present in the cases of Company A and Company C, as both of them are deploying Product Owners with the responsibility of grooming the Product backlogs.

Another issues that becomes clear when trying to implement Scrum in a traditional organisation is the seemingly contradictory planning paradigms, which the two systems represent. The best practice when planning a development project in a traditional organisation has typically been to plan in detail up front; but Scrum proposes a significantly shorter and continuously moving horizon for detailed planning. This has often been characterised as two conflicting extremes: Scrum versus Stage-Gate or Waterfall – or empirical- versus defined process control, as shown by Schwaber (2004). However, in all seven cases Scrum is a supplement to an existing process model in various versions of Stage-Gate. This seemingly works relatively well as none of the cases indicates something else. I some cases, it has been mentioned that Scrum is conducted "below the radar" or without any relation to the Stage-Gate model above it. In other cases, as for instance the Company G case, Scrum is intentionally used as the lowermost process framework, which is fitted into the sub-sections of the VVSM process.

Company A has not had significant problems in integrating the two process models, but Interviewee A1 still argues that Scrum requires another mindset in regards to the practical planning of the process:

Interviewee A1 (Company A): "We need to have an idea about what is going to happen six months into the future. Only an idea – we must not by any means go into detail on it, but we need to be able to communicate it in order for people to make the right decisions today."

Interviewee A2 from Company A uses the concept of "Just-in-time planning" to describe the essence of Scrum in opposition to the traditional waterfall planning model. As it is also emphasised in the quote above, Scrum requires a different attitude towards planning.

The last two issues that are part of this fifth analytical theme are respectively *support from stakeholders* and *distant customers*. Both issues have only been present in one case each and therefore do not indicate any general pattern. They are, however, both worth mentioning, as, in each their way, they reveal some interesting insights about implications that might arise under certain circumstances.

The first of the two issues concerns the organisational reluctance when Scrum is introduced through a bottom-up approach. In this case, the motivation is present in the development environment, while the top-level management is halting its support to the initiative. This specific situation is known from the Company B case and, to some extent, from Company D case as well. However, in the latter case, it has been a lack of organisational support rather than an actual concern that the Scrum process might be a disruptive phenomenon that has affected the implementation process. Both cases are examples of proactive teams in conservative or even inert organisations, and they clearly show that the implementation process *might* be driven as a bottom-up initiative; but they also show that it is an uphill-battle. Interviewee B1 from Company B, who has been implementing Scrum in other organisation before, describes this issue:

Interviewee B1 (Company B): "I have not yet experienced the team being the problem. It is typically the management or the organisation that has to be convinced. Support from the stakeholders – it is often where the problem lies."

The last issue originates from the Company D case, but it might gain some resonance most other cases as well. The Agile Manifesto for Software Development emphasises customer collaboration, but according to Interviewee D1 from Company D, this is not fulfilled in their organisation:

> *Interviewee D1 (Company D): "What I really think goes wrong, is that we are not standing out there, close to customers. We are not the ones getting the direct feedback. It means that in those few situations where stuff is actually presented to the customer or someone close to the customer, then things have changed. Things that have not been predicted by anyone in the organisation actually working on the project."*

Interviewee D1 describes a distant relationship between developers and customers, which is arguably caused by the hierarchical setup in the organisation. The pattern seems recognisable in most of the cases despite the absence of any indications of it in the interviews. However, this may be explained by the fact that the respective companies are not used to involving customers or end-users in the development process to the extent which is advocated by the manifesto. But with the lack of a Product Owner to decide priorities and set the directions, which seems to be normal, access to the actual customers becomes even more important in order to gain the necessary information from the market.

PRODUCT DESIGNER PERSPECTIVE

It has been stated earlier that implementing Scrum in a traditional organisation requires changes to customs and well-rooted practices, and it has been stated that it requires adjustments within at least one or both development models. In several cases this means conducting Scrum without a Product Owner, which results in Product Owner responsibilities trickling down to the Development Team. This seems to be one of the major issues of this analytical theme, if seen from a developer perspective. Issues like this are symptoms of a crippled Scrum framework that has only been partly implemented, that is: perhaps without considering the consequences it has to the developers.

ORGANISATIONAL PERSPECTIVE

Seen from an organisational perspective, the co-existence of two seemingly contradicting process models, namely the Stage-Gate model and the Scrum framework, could prove difficult in the sense that, in a way, they represent two different development paradigms. As it has been mentioned earlier in chapter 2, Scrum represents an empirical process, which advocates frequent inspection and adaption to continuously changing and emerging conditions around the development environment. The traditional Stage-Gate process advocates a rather defined process and considers heavy planning up front to be the best practice. Clearly, in principle the above two models are contradicting – also in several other respects than planning schemes – but they *do* exist side by side in all cases, which also indicates that the combination is possible in practice. Of course it is a matter of the actual levels of compliance to the Scrum framework in the organisations; but as it has been mentioned in the analysis above, this may depend more on a change in planning attitude than on technical problems.

VIABLE SOLUTIONS

It is difficult to point at the one best model of fitting Scrum into a traditional organisation. All the cases show that a different condition exists in each organisation. To carefully assign all the Scrum roles to the right people is clearly an important step towards avoiding an overload of responsibility on the Development Team. It is also important to make clear in which areas the two co-existing process models touch each other.

SUMMARY OF THEME 5

- This group of challenges is multi-facetted, and no clear patterns are identified between the involved cases.

Scrum in a traditional organisation is a challenge to the product designer because:

- Scrum is rarely fully unfolded, and roles, such as Product Owner, are often not assigned to anybody in the organisation.
- Adhering to two co-existing process models that represent very different paradigms and values, requires adjustment to existing practices and mind-sets.

Scrum in a traditional organisation is an organisational challenge because:

- Roles and responsibilities of traditional development practice and Scrum seem difficult to fully combine.
- The two process models require different management approaches to driving development efforts.

5.2.6 THEME 6: SEPARATE DELIVERABLES EACH SPRINT

DESCRIPTION

An important concept in the Scrum framework is the ability to continuously produce functionality that creates immediate value to the customers. The term "Potentially Shippable Product Increment" refers to this ability to push separate increments of the product to market, as they are developed in order to maintain a close adjustment to the present state of an ever-changing market. In most software development projects this practice is possible, due to the relative flexibility in developing software compared to physical product development. The present analytical theme concerns this challenge of adhering to this part of the Scrum framework, when developing product with physical components across several disciplines.

ANALYSIS

The challenge of creating separate deliverables each Sprint is perhaps the most obvious challenge, when adopting Scrum from software development to integrated product development. The physical aspect of integrated product development intuitively seems to conflict with this extreme demand for flexibility. However, it is only regarded as a challenge in three out of seven cases, which could indicate that ways of overcoming this challenge have been found in at least some of the other cases.

The basis of this analytical theme is statements from respectively Company A, Company C and Company E. Once again, it is difficult to point at any clear correlations that may indicate a general pattern. The cases differ in their level of compliance to Scrum, in experience with Scrum, and as to ways of conducting the various Scrum events. If any pattern exists, the relatively limited data sets, which exist in this research project, do not reveal it. However, it is possible that the interviewees' reflections in this regard could be biased, as this challenge is arguably the most spoken-of issue, when relating Scrum with integrated product development.

The most obvious observation mentioned by the interviewees points to the fact that physical product development is difficult to split into separate deliverables. Of course, this issue is related to the challenge of breaking down tasks. Interviewee A1 from Company A uses the analogy of a car to describe his point about this fundamental challenge:

Interviewee A1 (Company A): "We cannot build a car, by developing the motor in Sprint 1 and the steering mechanism in Sprint 2. And in Sprint 5 we make the brake. That we cannot test. We need to develop all the parts, but in the simplest version possible. This means that if we concentrate on the motor in one Sprint, we also include the simplest version of a brake. Then we develop the actual brake later"

Interviewee C1 from Company C adds a nuance to this issue in his reflections about presenting something that "isn't here". His statement is a consequence of the fact that the delivery of working prototypes or independent product increments is almost extremely difficult to achieve for each Sprint Review:

Interviewee C1 (Company C): "It is difficult to sell a document. It is not that sexy. It is also difficult to sell a measuring of sound in a box. Those people [the stakeholders] may not be that familiar with the technical issues. They would rather see something that 'does' something.

In both the Company A-case and the Company C-case, the accepted types of deliverables for each Sprint have been broadened to also include documents, analytical results and even simple learning goals instead of merely working- and potentially shippable prototypes. In fact, this has been the case in all companies in the research project. Interviewee E2 from the Company E-case points at the ability to define tasks as being the most important aspect of solving this challenge of separate deliverables each Sprint:

Interviewee E2 (Company E): "It is very important how you define your tasks in order to be able to remove them from the board after three weeks. Quite often we have ended up defining tasks on a too high level, which has resulted in Post-it notes staying on the Scrum board for too long – and so we have missed the point."

At Company F, Interviewee F1 has implemented a relatively strict procedure for describing development tasks. This procedure includes the creation of a clear "Definition of Done" for each development task, and this, to a great extent, eliminates the difficulties that might otherwise be related to reviewing development efforts that cannot immediately be prototyped or made physical. Several of the other cases – including Company C and Company A – have their own equivalent ways to evaluate the stage of completion of the individual tasks. There is a demand for potentially shippable product increments in the Scrum framework, but this is circumvented when the companies put emphasis on the Done-criteria in integrated product development. The slight change in procedures may not be in line with the Scrum framework, but it seems that all companies have the necessary pragmatic attitude to their level of compliance with Scrum. It might even be argued that strictly adhering to the principles of Scrum in regard to separate deliverables would be impossible.

PRODUCT DESIGNER PERSPECTIVE

Perhaps delivering separate product increments is merely a theoretical challenge to the Development Teams as this aspect of the Scrum framework is seemingly being circumvented in literally all cases. However, the changed practice of adding precise definitions to task completion is a considerable change to most of the development environments; and the idea of collective evaluation through the Sprint reviews, based on the Done-criteria, is new to most of the developers involved in the research interviews. Furthermore, it can be argued that even though the resulting deliverables are not necessarily representing separate functional elements of the final product, they still have to be clearly defined in advance. Defining the Done-criteria is closely related to the earlier presented challenge of breaking down development tasks; and just as in that case, this challenge of separate deliverables is primarily a problem to the mechanical developers due to substantial integration and interdependencies in the physical aspects of the products.

ORGANISATIONAL PERSPECTIVE

The challenge of separate deliverables each Sprint is primarily a technical challenge affecting the practical development effort in the development environment. It is, however, important to acknowledge that the continuous and frequent evaluation of the Sprint outputs is radically different from the practice found in a traditional Stage-Gate process. Here critical design reviews most often happen only in connection with the relatively rare gate meetings, as described in chapter 2.3.2, based on Cooper (2011)

VIABLE SOLUTIONS

Above it has already been emphasised that the pragmatic attitude towards the demand for potentially shippable product increments is dominant in the majority of the seven cases. There is not so much focus on physical- and working prototypes in the Sprint Reviews. The rules of Scrum have been relaxed, and other types of deliverables are now accepted. The focus has been turned towards the evaluation of the carefully defined Done-criteria for each separate task.

SUMMARY OF THEME 6

- The challenge of separate deliverables each Sprint is recognised in three of seven cases.
- However, the three cases do not indicate any clear pattern that might relate to this challenge.

Separate deliverables each Sprint is a challenge to the product designer because:

- The physical aspects of integrated product development complicate the separation of the product into smaller functional elements that are potentially shippable.
- The creation and the evaluation of strict Done-criteria for each small task is a radically different practice that is closely related to the challenge of breaking down tasks, and influences developers working with physical aspects of the product.

Separate deliverables each Sprint is an organisational challenge because:

- The continuous and frequent evaluation of the Sprint outputs is radically different from the practice found in a traditional Stage-Gate process in regard to design reviews.

5.2.7 THEME 7: MAINTAINING NECESSARY FOCUS

DESCRIPTION

Scrum advocates a strong focus on the development tasks to which the team has committed itself during a Sprint. In fact, it is one of the official responsibilities of the Scrum Master to act as a protective gatekeeper, bouncing off any incoming request or task from the rest of the organisation during the Sprint period. This analytical theme concerns the challenge of maintaining the necessary focus when conducting Scrum.

ANALYSIS

This part of the analysis focus is based on statements from three out of seven cases, but in contrast to most of the other challenges, there seems to be a clear pattern in the presence of this challenge. Of all the seven cases, it is the three companies with the shortest Scrum experience that are facing this challenge. The companies are Company B, Company C and the Company G client, and their experience range from only one month to approximately ten months. The fact that only these three companies have mentioned this challenge – and none of the more experienced companies – may indicate that the it will eventually disappear, or in some way become manageable.

The primary issue of this challenge is disturbances caused by "fire fighting" in relation to the operation of the production. Both Interviewee B1 from Company B and Interviewee C2 from Company C mention this as a challenge:

> *Interviewee B1 (Company B): "I would like to change this to a situation, where some focus on fire fighting and some focus their efforts on development. We may not be able to protect everyone, but at least tip the balance, so that we can minimise the fire fighting-activities for some of our resources."*

At Company C – the most experienced of the three companies – this issue has been the main reason for establishing the Technical Investigation Group. This initiative has minimised the level of disturbance from the production facilities and thereby reduced this challenge. Company D, which is one of the most experienced companies in regard to Scrum practice in this research project, is using a similar solution.

This specific challenge may indicate how the organisations prioritise respectively the operation of the production and the development efforts. In the case of Company B, the development activities are clearly secondary, and in such a case it is difficult to fully eliminate disturbances from production.

Another issue of this challenge concerns the ability to focus on only one or few projects. Interviewee G2 from Company G describes a situation, which is familiar to many of the client's employees:

> *Interviewee G2, Company G): "Some people are engaged in 10 or 15 different projects, which is eventually causing problems in relation to their level of involvement in the individual projects. Imagine what would happen, it they had to take part in all the Daily Scrum meetings!"*

Beside being far from compliant with the practice of Scrum, the participation in multiple projects, according to both Wheelwright & Clark (1992) and Smith & Reinertson (1998), is directly influencing the time spent on value adding tasks in a negative direction. This is seen in figure 2.11 on page 47. It does, however, seem to be a long way ahead – if even possible – to change the organisational setup causing this project overload of individuals at the Company G client.

PRODUCT DESIGNER PERSPECTIVE

Seen from a developer's perspective, an uninterrupted focus on the development tasks is crucial to the commitment to the project. Too many disturbances hamper the possibility to correctly estimate the available resources in a Sprint, and create frustration within the Development Team. This is clearly a challenge that influences the Development Team, but requires organisational changes in order to be defeated, just as it is the case with the issue of too many projects per employee.

ORGANISATIONAL PERSPECTIVE

As it has just been mentioned above, the challenge of maintaining necessary focus on development tasks primarily affects the developers. It requires initiatives on an organisational level to get rid of. The fact that the challenge is only present in the least experienced companies may indicate that it requires a certain amount of time for the organisation to realise that actions have to be taken in order to change the condition around the Development Teams. As it has also been mentioned in the theoretical framework, the organisation surrounding team could arguably be considered to be a fourth role in Scrum as – just as the other roles – it requires a certain training and experience to fill it in.

Most likely, it is considerably more difficult to significantly reduce the number of projects per employee, as this would entail some rather radical changes in the organisational setup.

VIABLE SOLUTIONS

Setting up walls around the Development Team in the form of formal fire fighting units seems to be an effective solution in some of the reviewed companies. This could be one of the initiatives taken in order to train the surrounding organisation to be considerate about how it treats the Development Teams, in order not to interrupt the Sprint more than necessary.

SUMMARY OF THEME 7

- The challenge of maintaining necessary focus is recognised in three out of seven cases.
- The challenge is present in cases with less than one year of Scrum experience

Maintaining necessary focus is a challenge to the product designer because:

- Disturbances from the surrounding organisation prevent the development team in focussing fully on the development tasks to which it has committed itself.
- Estimation of available resources for the Sprint is hampered by not having clear agreements about the extent of sporadic fire fighting tasks.

Maintaining necessary focus is an organisational challenge because:

- Establishing the necessary tranquillity around the development team requires changed practices as to the treatment of the team.

5.2.8 THEME #8: FLEXIBILITY ISSUES

DESCRIPTION

Within the domain of software Scrum is valued for its ability to keep the development process flexible towards the dynamics of the market, the competition, or the organisation itself. This last analytical theme concerns the ability of maintaining the same level of flexibility in spite of the transition to the domain of integrated product development.

ANALYSIS

This analysis is based on statements from only two of the seven cases. This challenge is therefore marginal in comparison to some of the earlier presented challenges. However, it does concern the fundamental value from Agile Development, namely "responding to change over following a plan", and is therefore brought forward as the last theme. The cases representing this challenge are Company D and Company E, but it is difficult to highlight any patterns from such a small extract of the total case material.

According to the interviewees at Company E, mechanical development is subject to certain domain-specific challenges concerning flexibility due to the physical character of the work. Interviewee E2 argues that working with flexibility in the product development process is difficult for the Mechanics team:

> *Interviewee E2 (Company E): "It has been a challenge that nothing was locked. Even though you showed off your 3D model, it would still be subjected to discussion. In reality it was really difficult to tie up anything. We haven't succeeded in that part."*

At Company D, they also struggle with handling changes midstream in the development process. Due to the complexity of their products, they have to carefully plan the development up front, which is eventually contradicting the practice of Scrum:

> *Interviewee D1 (Company D): "It takes a long time to develop such a complicated product as this one. Therefore they have to be highly visionary when developing the Product Backlog and continuously believe that what we do is the right things. However, it IS a challenge when changes occur. Changes carry along large consequences for the Mechanics team, a little less for Hardware and the least for Firmware."*

In both cases, it seems that the challenge of remaining flexible towards changes is most evident in the mechanical aspects of the products, and it thereby substantiates a tendency with several of the other analytical themes in regard to restraints of physicality.

PRODUCT DESIGNER PERSPECTIVE

As indicated by the quote from Company E above, maintaining a high level of flexibility in the product development may cause some headache to the developers as, due to the high integration in the mechanical design, this means keeping all options open. Juggling with several open ends is difficult and conflicts with the typical measures of progress: taking decisions and closing open ends.

ORGANISATIONAL PERSPECTIVE

Even though this challenge is seemingly affecting the development environment the most, the change in attitude from following plans to starting to embrace change is also an organisation concern. The change may force the organisation to tip the balance from adhering to the best practices of waterfall development and Stage-Gate models to actually giving room for necessary changes to occur as advocated by the Scrum framework.

VIABLE SOLUTIONS

"Designing for the battle, not the war" is one of the known phrasings from Scrum, and it means that one should not try to envision and prepare the full course of the development process, but rather concentrate on only looking a small step ahead. Looking too far into the future will only result in vague guesses. This may apply well to software, but in the case of integrated product development, where all decisions taken arguably have larger consequences for the following development activities, the phrasing may not be that suitable. While it can be argued that software development to some extent is more open and robust towards changes in, for instance, the backlog, integrated product development may require a certain amount of up-front planning that reduces the subsequent flexibility – even though it is conflicting with the values of Agile Development.

SUMMARY OF THEME 8

- The challenge of maintaining flexibility is regarded as an actual issue by only two of seven cases.
- The presence of this challenge does not reveal any clear patterns.

Maintaining flexibility is a challenge to the product designer because:

- The high integration in mechanical designs hampers the ability of taking decisions without significantly reducing flexibility.
- Changes carry along large consequences for the Mechanics team, a little less for Hardware and the least for Firmware.

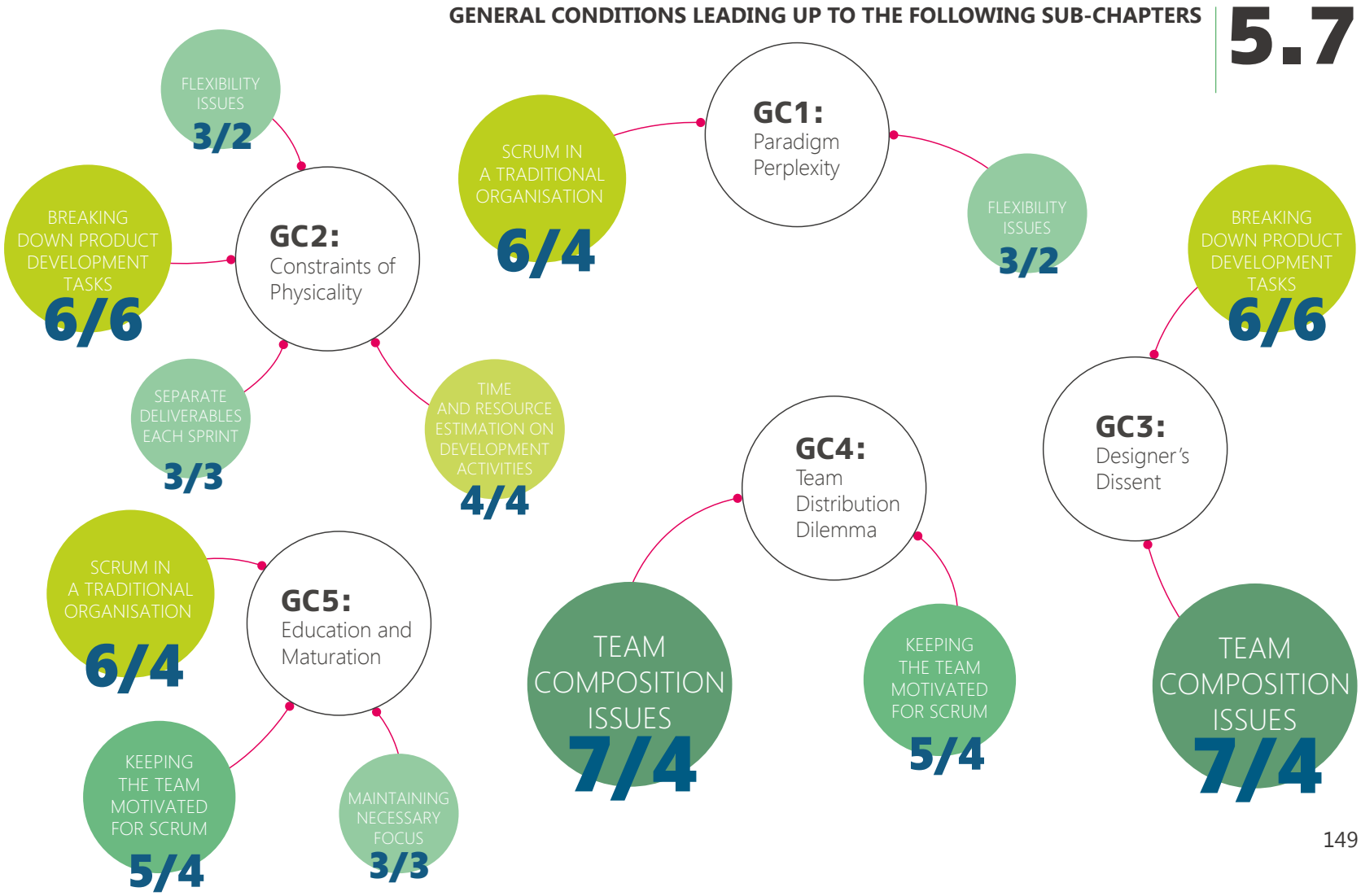
Maintaining flexibility is an organisational challenge because:

- The agile value of embracing change is disruptive to traditional development, and becoming truly open to change will require a break with the defined process control model of Stage-Gate.

5.3 PART 3: SYNTHESISING ANALYTICAL THEMES INTO GENERAL CONDITIONS

In the first part of the analysis, the challenges identified in the interviews were gathered into eight themes, which were then separately examined and explored in part 2 based on their respective level of occurrence. This resulted in the identification of patterns and a division of the various aspects into two separate categories: Developer challenges and organisational challenges. The purpose of this third part of the analysis is to synthesise the identified challenges in order to establish an understanding of the *general conditions*, which in each their way are nourishing the challenges identified in Part 2.

In this research project, a general condition is defined as an underlying condition that notably contributes to the existence of several of the identified challenges. The figure 5.7 below shows the interrelations between the analytical themes. Each of these cross-theme relations illustrates a generally challenging condition and is unfolded in one of the following subchapters.



5.3.1 GENERAL CONDITION 1: PARADIGM PERPLEXITY

Paradigm Perplexity is the first and clearly the most evident general condition. The term covers the basic condition of transferring one methodological framework, representing a radically new development paradigm, into a traditional development environment, which adheres to best practices of another and older development paradigm. The combination of two contradicting paradigms will arguably cause some confusion and conflicts in regard to both the practical development work and the organisational setup.

Clearly, the resulting conflicts and confusion have not put an end to the implementation of Scrum in integrated product development environments, but recapitulating the overview of case compliance with the Scrum framework in figure 5.5 at page 126, it is safe to say that it has complicated the process, and at least prevented the deployment of "full Scrum."

Several conflicting elements complicate the combination of the two process control models. First of all, the inherent conflict between the empirical and the defined process control is found in the Gantt charts prescribing an anticipated course of the project at the same time as they deploy the Scrum framework. This practice reflects heavy planning up front with a prejudice of a known output. Scrum, on the other hand, tries to incorporate a rather dynamic approach with its continuous planning. In other words, the combination of the two approaches is both deflecting and embracing changes in the process. The same conflict is also found the in way that the employees at Company A struggle with combining long-term planning inherited from their old way of working with the short-term planning of the single Sprint period.

Another conflicting element is the organisation of roles and responsibilities, which has also been described in the fifth analytical theme about Scrum in a traditional organisation. In Scrum, project management is a responsibility shared by the development team, which is in contrast to the traditional project management setup. In most of the seven cases, the traditional roles have not been removed as a consequence of the implementation of Scrum. Instead the two sets of roles co-exist side by side in the organisations. The combination of the two sets of roles and responsibilities in some situations create overlapping areas of responsibilities. The challenges of this "dual setup" have been anticipated from a theoretical point of view in chapter 2.

A third difference between the two paradigms is found in the technical development level, and concerns the way that development activities are split up in different ways. According to Interviewee B1 at Company B, Scrum entails a radically different way of developing. While tasks are normally divided by their relation to different platforms, Scrum promotes a division of the development effort into separate functional features across platforms and disciplines. This radical shift in development approach is illustrated in figure 4.8 on page 75.

To some extent it can be argued that Scrum is an island, when implemented in traditional organisations. The Scrum framework does not include any guidance in how it should integrate with an existing organisation or management scheme. In practice, this means that every organisation has to invent its own way of adapting Scrum to fit its organisational legacy. As it has been pointed out in this description of the first general condition, the implementation is therefore often turbulent, due to the radical differences between the two individual systems.

5.3.2 GENERAL CONDITION 2: CONSTRAINTS OF PHYSICALITY

The first general condition focused on the fundamental condition of transferring a methodological framework belonging to one paradigm into a domain that has long traditions of adhering to another paradigm. This second general condition, to a large extent, covers a part of the transference that is just as fundamental: A shift from *the freedom of virtuality in software development* to *constraints of physicality in integrated product development*.

The special circumstances related to conducting Scrum in mechanical development have been mention several times during the interviews. The interviewees have commented this specific issue in relation to several parts of the Scrum framework. The list below illustrates the specific parts of the process that causes difficulties.

- Breaking down development tasks
- Separate deliveries each sprint
- Time and resource estimation on development activities
- Flexibility issues

As it has been mentioned in the previous chapter, 5.3.1, the process of breaking down development tasks is different from that of traditional development, as the separation of the development tasks into functional features conflicts with the high integration between the sub-parts of physical and integrated products. Small and independent work packages may be a viable way in software development, but as Interviewee E2 from Company E states, mechanical designs emerge like amoebae in a non-linear and perhaps even non-rational way.

Another reason that causes the physical aspects of product development to be such a severe challenge compared to the situation in software development is the natural lengths of the separate iteration cycles. As Interviewee D1 from Company D emphasises, a typical iteration cycle in integrated product development is four to six months, and this hampers almost every spontaneous attempt to maintain a truly flexible and dynamic development process in regard to responding to change.

The perhaps most obvious effect of the physical constraints is the disability to present working functionality at the end of each Sprint. Several interviewees point out that this part of the Scrum framework is close to impossible. The reason for this is, once again, likely to be found in the high integration of the physical products compared to software products. It is clear, however, that only a few of the interviewees regarded this as a problematic issues, as they have all opened up for other kinds of Sprint outputs than just potentially shippable product increments.

The Constraints of Physicality is an inescapable and general condition that clearly makes a full implementation of Scrum in integrated product development difficult, if not impossible. The degree of physical limitation is the most distinctive difference between software development and integrated product development. While it has just been argued that development teams may get around the requirement of delivering shippable product increments, aspects such as task breakdown and flexibility towards changes are more critical and cause greater challenges.

5.3.3 GENERAL CONDITION 3: DESIGNER'S DISSENT

While performers of some disciplines in integrated product development thrive with the implementation of Scrum, others are hesitant and generally seem uneasy with this new addition to the development process. The difference is seen almost everywhere in the case material. As it has been stated in the preceding chapter about Constraints of Physicality, the mechanical developers are seemingly the ones struggling the most, due to the very physical character of their resulting work. The case material does, however, also indicate some distinctive characteristics of the industrial designers in the few cases in which the industrial design discipline appears. In this chapter, the mechanical developers and the industrial designers – commonly denoted as product designers – are regarded as one homogeneous group of developers, due to their mutual focus on mechanical design.

According to the interviewees at Company E, the mechanical developers belong to a culture different from that of the other development teams. It is as if they lack the motivation or interest in using Scrum to the same extent as the other teams. Furthermore, the mechanical development team has less compliance with the Scrum framework than the other development teams within software, firmware and electronics. For example, their Scrum meetings are held less frequently.

At Company F, Interviewee F1 goes as far as to state that the team of industrial designers and usability experts are beyond any pedagogical reach, as they never conform to the Scrum practices. Similar patterns are furthermore seen at Company C, where the mechanical design team, which is regarded as a sub-team of the main Scrum Development Team, is disturbingly hesitant in its efforts to supply the rest of the team with specific development tasks at their common Scrum board. According to the interviewees, who both represent the hardware- and firmware disciplines, the mechanical sub-team is relatively isolated and cut off from the otherwise close collaboration across the rest of the Development Team.

Several examples similar to these are given in the case data, and together they seem to indicate a pattern of dissension about the Scrum framework as a beneficial add-on to the existing mechanical development process. Based on this indication it would be natural to ask why the reluctance against Scrum is so distinct in *this* part of the development environment. The case material clearly shows that there are special challenges to this group of developers caused by the Constraints of Physicality, but the examples from the cases do not solely relate to that as a possible explanation. It simply seems as if the product designers work in a somewhat different way than other disciplines do.

The paradoxical part of this is the fact that established versions of creative processes, such as the one promoted by Stanton (1998), resemble the basic incremental and iterative process promoted by the Scrum framework. These fundamental processes would arguably correspond to the design activities carried out by the product designers as described earlier in chapter 2.3.5. However, the empirical findings from the seven cases indicate that this may not be the case.

Unique to the Scrum framework, however, is the heavy focus on estimation and quantification of the development efforts together with the strictly structured iteration cycles. These aspects may be the decisive differences between the separate process models and thereby the parts of Scrum that do not correspond that well with the disciplines of mechanical developers and industrial designers.

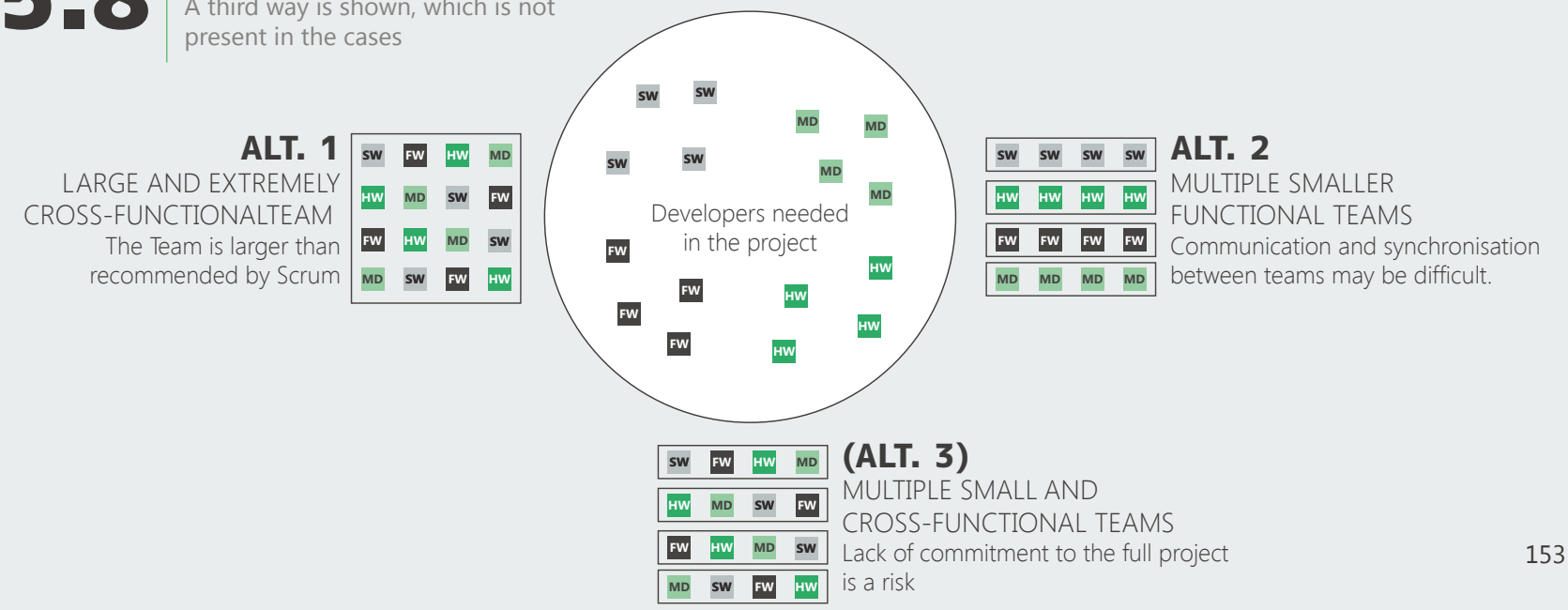
5.3.4 GENERAL CONDITION #4: TEAM DISTRIBUTION DILEMMA

When looking at the composition of Scrum teams in the seven cases, hardly any of the companies have chosen the same strategy. A few companies deploy fully cross-functional development teams, while the development teams in other companies are organised in several parallel development teams divided by disciplines. Finally, there are companies that combine functional and cross-functional development teams in one single project.

This chapter concerns the seemingly domain-specific dilemma of team-distribution that appears as a result of *extreme cross-functionality in* combination with *a large number of involved developers*. According to the Scrum Guide by Schwaber and Sutherland (2011), development teams should be *"small enough to remain nimble and large enough to complete significant work."* This is more precisely specified as a team size of 3 to 9 developers. Fewer than three Development Team members risk encountering skill constraints during the Sprint, and teams of more than nine members requires too much coordination. It is important to remember that this advice has been written with cross-functional software development teams in mind. In most of the investigated cases, project teams have been both significantly larger and more diverse. The conditions mentioned above are regarded as a dilemma because of their negative consequences to team communication and synchronisation, regardless of the chosen approach to team composition. The fundamental choice seems to be between *one large and extremely cross-functional team* and *multiple smaller functional teams*.

As it is seen in the figure 5.8 below, no matter which one of the two basic strategies is chosen, it will evidently entail some kind of communication- or team-synchronic challenge. Company F is an example of multiple teams working together on one project, and according to Interviewee F1 from this case, the communication and task synchronisation between the respective teams is one of the biggest challenges of implementing Scrum. Company D is another example of multiple functional teams. In this case an additional role has been added in order to ensure sufficient cross-team communication and synchronisation: Beside sharing the two Scrum Masters, the three collaborating Development Teams each have a system architect, who is responsible for the technical integration and compatibility with the work of the other teams.

5.8 **TWO WAYS TO COMPOSE TEAMS**
A third way is shown, which is not present in the cases



Company A represents the strategy of fully cross-functional teams, and perhaps because of the size of the company and the character of its products, the project teams are relatively small compared to the teams of the other cases. Despite this fact, Interviewee A2 from Company A mentions that some team members find the high degree of cross-functionality difficult, as everyone is highly specified and not necessarily "talks the same language." This corresponds to the findings of Keller (2001).

The companies mentioned above represent two different approaches to team composition. Each approach has its advantages and disadvantages; hence the dilemma. The discussion about cross-functional and functional teams is related the discussion about experts and generalists. Interviewee F1 from Company F argues that software development teams are in a rather privileged position in comparison to integrated product development teams, due to the relatively high homogeneity found in the software development teams. This means that Development Team members in software have a larger overlap of competencies. The argument seems to be in line with the argument from Interviewee G2 in the Company G case. According to him, it is one the great challenges in regard to team composition that almost everyone involved in a project is an expert in his or hers specific domain and therefore only participates in some small parts of the product development activities.

While the Development Teams at Company A make a great effort to broaden the developers' individual competencies by actively sharing development tasks within the cross-functional team, it is not the typical scenario in the other cases. Company A seems unique in this matter, as the teams are actively trying to create bigger competence overlaps, despite their highly cross-functional compositions. In other organisations it is traditional to employ and promote experts, which may hinder a close communication and collaboration in the cross-functional teams.

To sum up this sub-chapter: The team distribution dilemma is about choosing the right strategy for composing the Scrum Development Teams. It is argued that two basic strategies exist, but each of these strategies has separate drawbacks in regards to team communication and team synchronisation.

5.3.5 GENERAL CONDITION #5: EDUCATION AND MATURATION

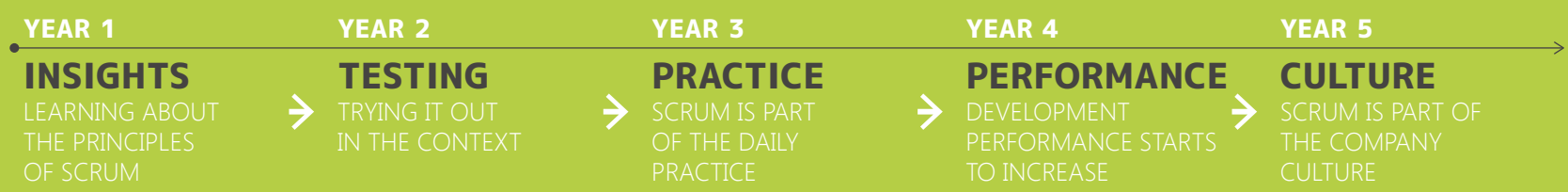
It has already been stated that the implementation of Scrum in a traditional development organisation entails some significant challenges in regard to the co-existence of multiple process-control systems, management systems and distribution of responsibilities. However, these are not the only aspects that require continuous attention if the implementation is going to succeed. This fifth general condition concerns the process of maturing the organisation to Scrum through education, experience, and time.

None of the seven cases fully comply with the rules of Scrum. Each of them deviate in their own individual ways, and full compliance is probably almost impossible due to conditions such as the Constraint of Physicality. It could also be argued that the low compliance with Scrum in some of the cases is caused by the lack of knowledge in how Scrum should be conducted, and how it should be met by the organisation. The fact that it takes time before the organisation is starting to see the effect of Scrum is confirmed by Interviewee G2 from Company G, who has been in the process of implementing Scrum at their client for 6 months.

> *Interviewee G2 (Company G): "It may even take years. Time estimation is a totally new culture, and they [the developers] have never been used to that. We use Scrum in our own office, and we also face difficulties when we estimate time on tasks and activities."*

According to Interviewee G2 the organisation will go through several stages during the process of implementing the Scrum framework as it is describes in figure 5.9 below.

5.9 **EDUCATING AND MATURING THE ORGANISATION TAKES TIME**
Five stages of the maturation-process (Interviewee G2, Company G)

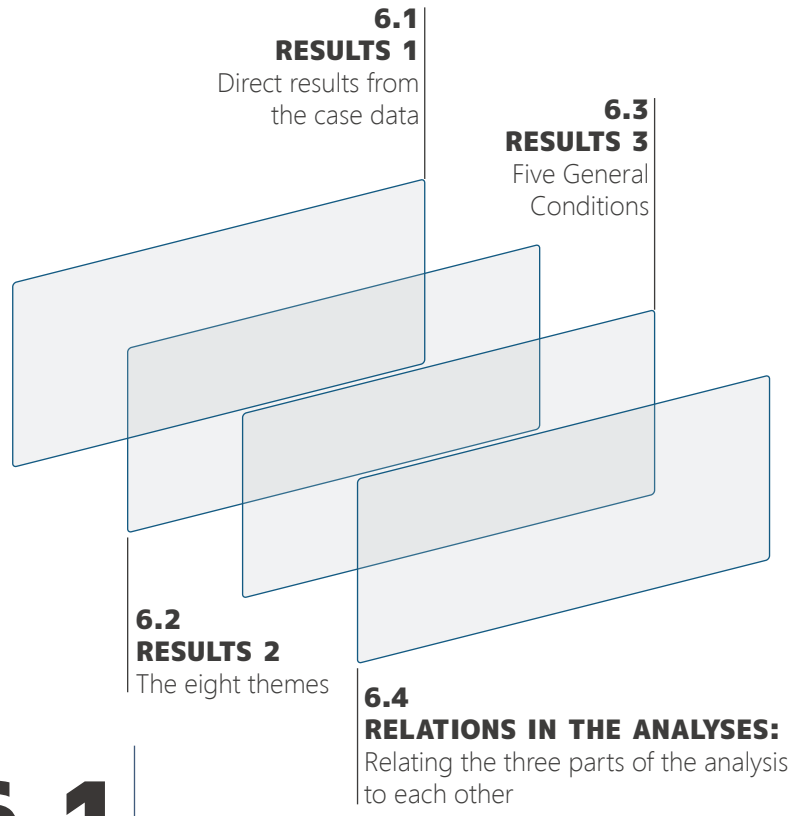


CH.06. RESULTS

The purpose of this chapter is to present the insights which have been identified and unfolded throughout the preceding analysis. More specifically, the chapter will include:

- Results from Analysis Part 1
- Results from Analysis Part 2
- Results from Analysis Part 3
- Relations between the results from analysis

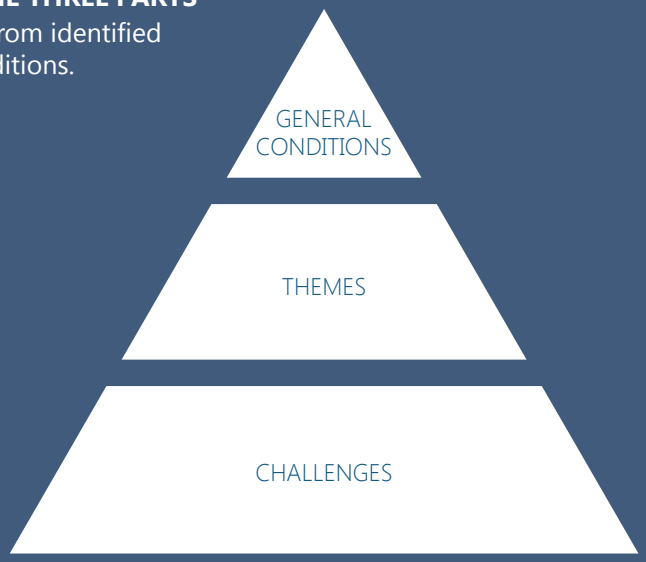
Figure 6.2 below illustrates the relation between the identified challenges, the eight themes, and the five general conditions. The level of generalisation rises for each step up the ladder. The following sub-chapters describe the essentials of each part of the analysis.



6.1 OVERVIEW OF CHAPTER 6

6.2 RELATIONS BETWEEN THE THREE PARTS

Increasing generalisation from identified challenges to general conditions.



6.1 RESULTS FROM ANALYSIS PART 1

The first part of the analysis presents a basic comparison between several aspects of the seven cases as well as a list of convergent themes of challenges.

In Part 1 of the analysis it is found that:

- Scrum experience ranges from only a few months to 4 years
- Sprint cycles ranges from 2 to 4 weeks
- The companies span production of consumer products, industrial components, and healthcare equipment
- The market clock speeds of the industries are considered respectively slow to moderate
- The project teams comprise a wide span of disciplines
- Both functional and cross-functional Scrum Development Teams are commonly used
- Teams are primarily co-located, whereas few are dispersed
- All companies deploy a Stage-Gate process-control model in combination with the Scrum framework

In regard to the level of compliance with the Scrum framework, it is furthermore found that:

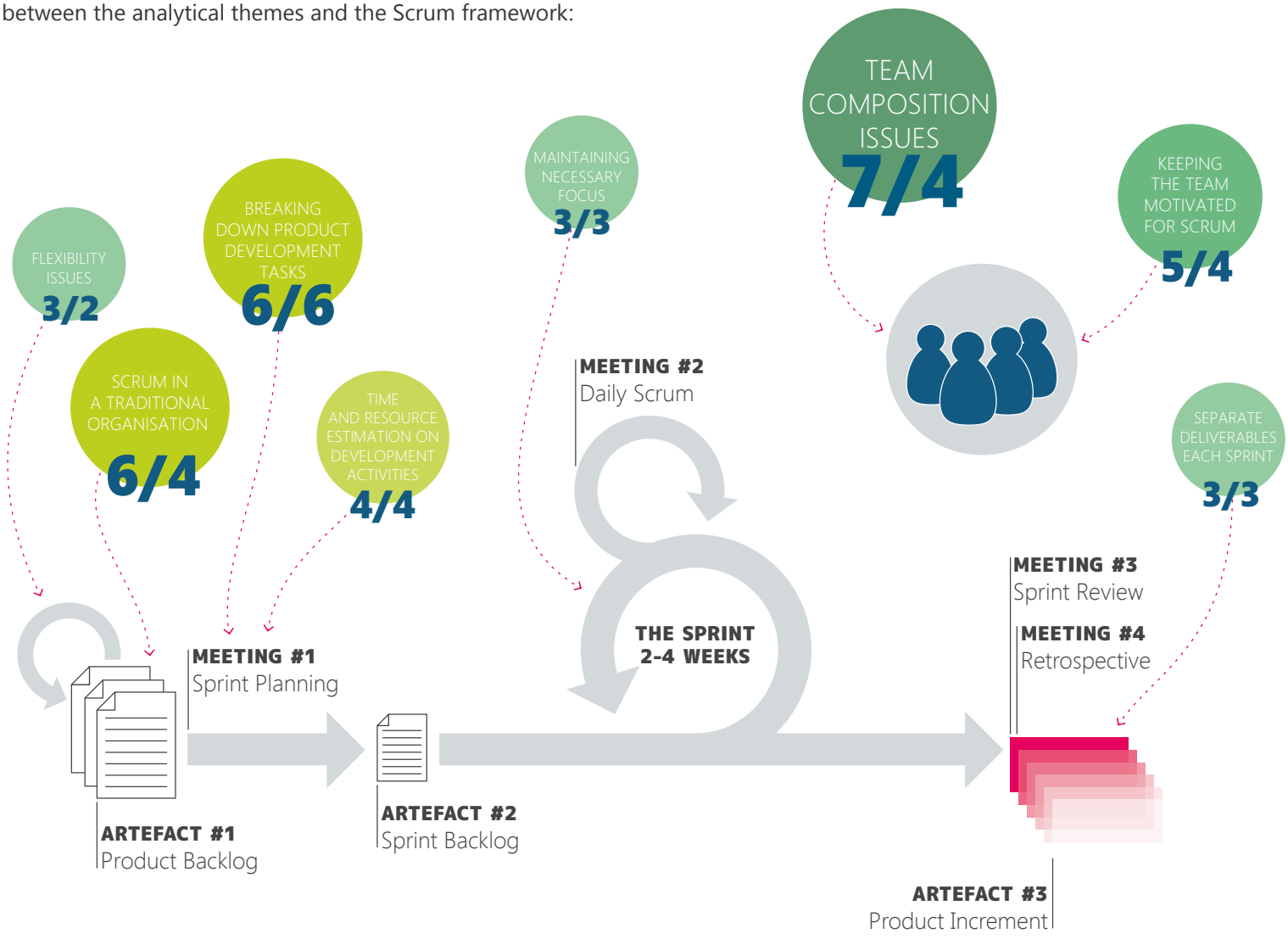
- None of the seven companies comply 100% with the rules of Scrum, however
- The level of compliance varies, and some single cases comply almost fully with Scrum
- Creating potentially shippable product increments each Sprint seems to be the most difficult part to fulfil, but -
- in only a few of the cases the Product Owner succeeds in filling in his role.

Based on the challenges identified throughout the case descriptions in chapter 4, eight themes have been established. The themes are:

- Breaking down development tasks
- Time and resources estimation of development activities
- Keeping the team motivated for Scrum
- Team composition issues
- Scrum in traditional organisations
- Separate deliverables each Sprint
- Maintaining necessary focus
- Flexibility issues

6.2 RESULTS FROM ANALYSIS PART 2

Eight analytical themes have been established in Part 1 based on the case descriptions in chapter 4. These eight themes have been the point of departure in the second part of the analysis, and they all relate to the Scrum framework in various ways. Figure 6.3 below illustrates the relationship between the analytical themes and the Scrum framework:



6.3 **RELATIONSHIP BETWEEN THE SCRUM MODEL AND THE ANALYTICAL THEMES**

Even though the themes stem from many different situations, most of them can be related to specific parts of the Scrum framework

IDENTIFIED PATTERNS (AND SOME METHODOLOGICAL AFTERMATH)

Despite the analysis of all the case material carried out throughout the analytical themes, only a few patterns have been identified as regards the challenges perceived by the interviewees across the cases. A likely reason could be that the separate challenges are, in fact, sporadically appearing without any significant patterns, and that the conditions for each company are unique to such an extent that no correlations between cases exist. However, there could be even other aspects that have had an influence on the lack of identified patterns. Prior to a description the patterns that *do* exist, a few thoughts about the reasons for missing patterns are presented:

First of all, the case material reflects the information that has been the aim to "capture" throughout the interviews. This has all been based on the interview guide. It is therefore evident that if the focus areas of the interview guide have not sufficiently covered all the influential aspects that may cause challenges in the implementation of Scrum, the captured data will not fully reflect the real conditions or contain the actual patterns. There may simply be too many influential aspects in the respective organisations, which makes a real comparison between cases difficult. Even though the theoretical foundation presented in chapter 2 ought to establish the necessary insights in order to "ask the right questions" during the interviews, the actual challenges that the employees experience during the Scrum implementation process may be rather elusive and depend on aspects not examined.

With this said, some patterns *have* been revealed. These are presented in the list below:

Identified patterns

- **Estimating time and resources for development activities.**
  Development Teams with more than three years of experience with Scrum seems to be able to precisely estimate time and resources for development tasks and have actual procedures for handling concept-related tasks with large factors of uncertainty.

- **Keeping the team motivated for Scrum**
  The challenge of keeping the team motivated for Scrum seems to primarily relate to development environments organised in cross-functional teams.

- **Maintaining necessary focus**
  The challenge of maintaining necessary focus only appears in organisations with less than one year's experience in conducting Scrum in integrated product development.

DISTRIBUTION OF CHALLENGES

Each of the analytical themes contains a series of aspects that have been covered through the analysis. Some aspects are primarily related to the practical work of the product designers, whereas some are related to the surrounding development organisation. The diagram below illustrates a distribution of these aspects into the two respective categories. The shown distribution is, however, not definite, but merely represents the primary affiliation of the respective aspects.

DEVELOPER CHALLENGES		ORGANISATIONAL CHALLENGES
● Breaking down product development tasks in the mechanical development environment. ◐ Unique culture or attitude in the mechanical development environment resists exhaustive breakdown of tasks.	Breaking down product development tasks	◐ Polarisation of development team and disbelief in Scrum as a viable solution in mechanical development.
● Developers lack general experience with task estimation. ● Estimation of concept development tasks is subsject to extreme uncertainty.	Time and resource estimation on development activities	◐ Bad estimates hamper correct monitoring of progress in development.
◐ Extremely cross-functional teams may hinder the individual developer's commitment to Scrum. ◐ Habits may result in developers falling back into the old development practice.	Keeping the team motivated for Scrum	● continuous organisational support to the development environment is necessary.
◐ Too many experts may complicate collective commitment to Sprint tasks.	Team composition issues	● Balancing functional and cross-functional teams ● Communication and synchronisation in teams that are dispersed to multiple locations.
◐ Lack of Product Owner or other Scrum elements complicate the development process. ◐ Adhering to two co-existing process models representing very different paradigms and values.	Scrum in a traditional organisation	● Roles and responsibilities of traditional development practice and Scrum seem difficult to fully combine. ● The two process models require different management approaches to driving development efforts.
● Physical aspects of integrated product development complicate the separation of the product in smaller functional elements, which are potentially shippable. ● Creating and evaluating strict done criteria for each small task and sub task.	Separate deliverables each Sprint	◐ Continuous and frequent evaluation of the Sprint outputs is radically different from the practice found in a traditional Stage-Gate process in regards to design reviews.
◐ Disturbances from the surrounding organisation ◐ Estimation of available resources for the Sprint is hampered by not having clear agreements about the extent of sporadic fire fighting tasks.	Maintaining necessary focus	● Establishing the necessary tranquillity around the development team require changed practices about how to treat the Scrum Development team
● High integration in mechanical designs hampers the ability of taking decisions without significantly reducing flexibility	Flexibility issues	◐ Agile value of embracing change is disruptive to traditional development

LEGEND: ● Primary Challenges ◐ Secondary (derived) challenges or consequenses

DISTRIBUTION OF CHALLENGES 6.4

Each theme primarily relates to either a developer perspective or a organisational perspective

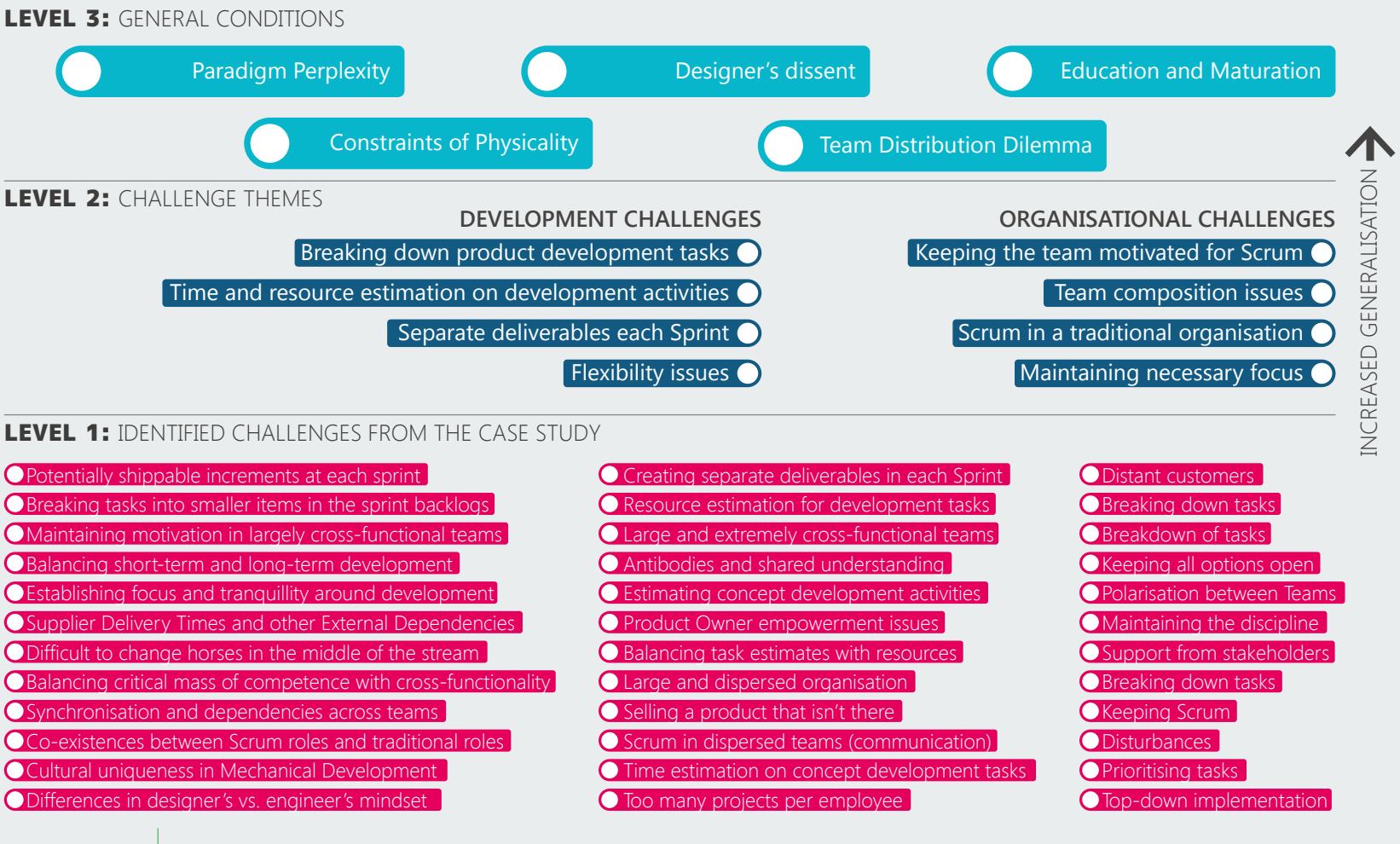
6.3 RESULTS FROM ANALYSIS PART 3

Five general conditions have been identified in the third part of the analysis. These conditions are a result of synthesising the challenges identified in Analysis Part 1 and unfolded in Part 2, and they thereby represents a higher level of abstraction. The general conditions are:

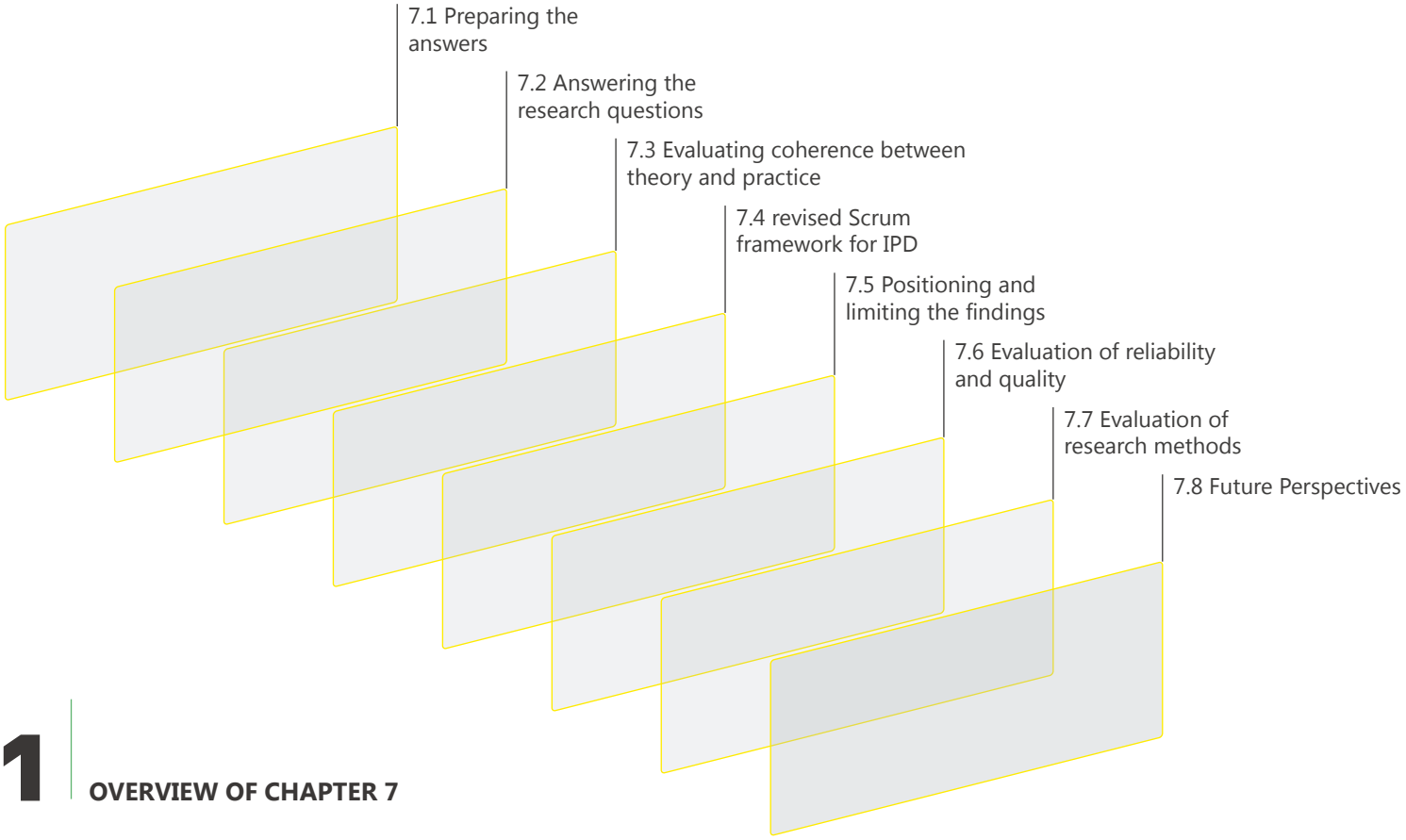
- Paradigm Perplexity
- Constraints of Physicality
- Designer's dissent
- Team Distribution Dilemma
- Education and Maturation

6.4 RELATIONS BETWEEN THE RESULTS FROM ANALYSIS

The purpose of this chapter is to bring together the results from Analysis Part 1, Part 2 and Part 3 in a combined graphical overview shown in figure 6.5 below.



CH.07. CONCLUSIONS



7.1 OVERVIEW OF CHAPTER 7

The objective of this chapter is to present a conclusion of this research project by answering the research questions through an elaboration of the results presented in chapter 6. Furthermore, it is the purpose to discuss the quality and reliability of the research by evaluating the research design and methodological approach. Finally, this last chapter comprises a set of future perspectives in which the present research efforts could be brought further. More specifically, the chapter is structured as follows:

- Preparing the answers
- Answering the research questions
- Evaluating the coherence between theory and practice
- Towards a revised Scrum framework for integrated product development
- Positioning and limiting the findings in relation to the present knowledge
- Evaluation of the reliability and quality of the research effort
- Evaluation of the research methods
- Future Perspectives

7.1 PREPARING THE ANSWERS

The aim of this sub-chapter is to answer the research questions presented in the introduction of this thesis. Even though answers to the research questions to some extent have been unfolded throughout the two parts of the analysis, this chapter will focus on relating answers directly to the questions originally asked.

To recapitulate: The research efforts have been led by the overall question: W*hat are the basic challenges of becoming agile in integrated product development environments using Scrum?* This question is further clarified in the following sub-questions:

- What are the primary challenges to the product designers when working with Scrum in integrated product development environments?

- What are the primary organisational challenges when implementing Scrum in integrated product development environments?

The answers to the questions are not simple. On the contrary, the answers are multifaceted and highly entangled. A multiplicity of challenges have been identified throughout the analysis and the data collection activities, but almost every single one of them encompasses consequences to both product developers and to the organisation as a whole. However, it is argued that some of these challenges are primarily related to the practical activities of the product developers working with Scrum, while other challenges are closer related to the organisational aspects of implementing Scrum.

In order to embrace the complex diversity in the identified challenges a higher level of abstraction has been established by synthesising the challenges into a set of *general conditions*. To a great extent these general conditions actually cover all the identified challenges by representing their origins.

7.2 ANSWERING THE RESEARCH QUESTIONS

Based on the research presented in this thesis it is found that the basic challenges of becoming agile in integrated product development environments using Scrum are:

- **PARADIGM PERPLEXITY**
  "Paradigm Perplexity" represents the challenges of integrating two contradicting process control models with conflicting values, planning schemes and organisation of roles and responsibilities. Due to its influence on such fundamental aspects, Paradigm Perplexity represents a great challenge to the implementation and conduction of Scrum, seen both from a developer perspective and an organisational perspective.

- **CONSTRAINTS OF PHYSICALITY**
  "Constraints of Physicality" represents the challenges of transferring Scrum from a domain of development of virtual products to a domain of development of physical products. This transfer of the Scrum framework entails a number of challenges, primarily to the mechanical developers. The challenges relate to breaking down development tasks, estimating time and resources for development activities, delivering potentially shippable and functional product increments each Sprint, and finally maintaining readiness for changes in the development process.

- **DESIGNER'S DISSENT**
  "Designer's dissent" represents the challenge of motivating mechanical developers and industrial designers to play by the rules of Scrum. It is argued that these groups adhere to a significantly different way of developing products than other groups typically involved in integrated product development. This unique development culture does not exactly comply with the strict estimation and quantification of development tasks as well as the strictly time boxed iteration cycles of Scrum.

- **TEAM DISTRIBUTION DILEMMA**
  "Team Distribution Dilemma" represents the challenge of appropriately distributing all developers involved in a certain project into Scrum Development Teams. Due to the wide span of different disciplines and the large amount of experts often needed in an integrated product development project, it is argued that two fundamentally different strategies seem to exist. However, both of these entail some challenges in regards to team communication and synchronisation.

- **EDUCATION AND MATURATION**
  "Maturing the Organisation" represents the challenge of establishing the Scrum framework as a part of the company culture through education, experience and time. Scrum is often a relatively unknown framework when implemented, and it requires time (several years) and change of habits in order for it to become a deeply rooted part of the development environment.

The mentioned General Conditions are considered the main challenges to overcome in order to be agile, but the second condition, "Constraints of Physicality," is furthermore considered the major critical hindrance to fully complying with Scrum.

The following sub-chapters more specifically present the answers to the two sub questions.

7.2.1 ANSWERING RESEARCH QUESTION 1

Based on the research presented in this thesis it is found that the primary challenges to the product designers when working with Scrum in integrated product development environments are:

- Breaking down development activities during the Sprint Planning meetings into separate and individually defined tasks.
- Estimating time and resources for development activities to a level measured in hours rather than weeks.
- Creating separate deliverables presented as physical prototypes to the Scrum team and the surrounding organisation at the Sprint review meeting at the end of each Sprint.
- Maintaining development flexibility to allow changes in product requirements despite the need to continuously take decisions in order to progress.

As none of the investigated cases fully comply with the Scrum framework, the list above only represents the challenges experienced in situations where the rules of Scrum have been followed to a great extent. However, full compliance with Scrum is virtually impossible in integrated product development, due to the Constraints of Physicality, which is a general condition there.

Each of the four challenges in the list above are not present in all investigated cases, and they may not all be present at the same time in other integrated product development environments.

7.2.2 ANSWERING RESEARCH QUESTION 2

Based on the research presented in this thesis it is found that the primary organisational challenges when implementing Scrum in integrated product development environments are:

- Keeping the team motivated for Scrum despite old habits, reluctance and the negative effect on team cohesiveness due to large diversity in cross-functional teams.
- Composing high-performing teams by balancing the level of cross-functionality and the generalist/specialist relationships.
- Combining roles and responsibilities of a traditional management scheme with the roles and responsibilities from Scrum, due to radically different management approaches.
- Maintaining necessary focus despite disturbances from the surrounding organisation and project accumulation.

It is furthermore argued that the challenge of maintaining necessary focus primarily exists within the first year of implementing Scrum.

As none of the investigated cases fully comply with the Scrum framework, the list above only represents the challenges experienced in

situations where the rules of Scrum have been followed to a great extent. Not all of the four challenges in the list are present in all investigated cases, and they may not all be present at the same time in other integrated product development environments.

7.3 EVALUATING THE COHERENCE BETWEEN THEORY AND PRACTICE

Scrum in integrated product development has been investigated, and the research questions have been answered. This chapter focuses on the level of coherence between *the actual findings* of this project and *the expected challenges* based on theory. In chapter 2.6 a catalogue of gaps, overlaps, and conflicts was presented based on the theoretical foundation. Each of the issues from the catalogue is evaluated against the project findings below.

- *Implementation of Scrum in Stage-Gate gives rise to a conflict between the project inputs of dynamic concepts in Scrum versus precise project definition in Stage-Gate.*
  In practice, a conflict between the Stage-Gate project definition and the concept of a Product Backlog has not been clearly substantiated by the empirical data. This may be caused by the fact that it is only a minority of the case companies that actually deploys a Product Backlog as it is defined in the Scrum Guide. The reason could also be that the development environments do not perceive the two as radically different from each other. Even though the developers are familiarised with the concept of the Product Backlog, they are seemingly not treating it as a dynamic document.

- *The output of a Scrum process follows value-to-customer and contrasts the predefined output from the project definition in Stage-Gate.*
  While this seems as a potential conflict in theory, the statement does not gain significant support in practice. However, one aspect from the project findings that do relate to this conflict is the challenge of maintaining flexibility as development projects are progressing forward. Two companies in the case study experienced difficulties with flexibility issues, whereas the rest of them in general were not trying

to change their course at all. They basically followed their initial plan without any intent to change it. The ability to react dynamically according to last-minute market trends does not seem to be a critical part of the motivation for conducting Scrum, as it was otherwise expected.

- *Stage-Gate lacks a clearly defined retrospective analysis.*
  This claim is not confirmed or disconfirmed by the empirical data. It is, however, clear that the Retrospective meeting from the Scrum framework is widely used throughout the cases, and developers are seemingly welcoming it as a positive addition to their development process.

- *The Scrum framework is expected to match the process of the product designer due to its immediate resemblance with the basic design process.*
  Despite the perfect match in theory, product designers – as in mechanical developers and industrial designers combined – have shown reluctance against the framework. It is not substantiated by the data that these developers have been able to gain an actual benefit from Scrum.

- *Traditional project-organisation would be expected to fit the Scrum framework.*
  The organisation around projects has not shown problematic. In fact, this is one of the aspects from the investigated development organisations that seem immediately compatible, when isolated from some of the more critical aspects mentioned in the remaining part of the catalogue.

- *Single-project focus in Scrum conflicts multi-projects focus in many matrix organisations, which results in lack of needed focus.*
  While a situation where all developers are working on several parallel projects has only been seen in few cases, it definitely hampers the efficiency of Scrum. A core Development Team suddenly becomes difficult to define, and team commitment might be compromised. However, it has not been directly substantiated by the empirical data that multiple projects alone will result in lack of focus.

- *Asymmetry in roles and responsibilities between the two systems is expected.*
  As it has been stated in the findings of this project, combining the roles and responsibilities of traditional management and Scrum is a challenge due to asymmetry between the radically different management approaches. This is in line with the expectations based on theory. It has furthermore been found that it is especially the Product Owner role that has difficulties of fitting into the traditional scheme. On the other hand has the Scrum Master role shown easier to implement, perhaps due to its emphasis on team coaching and guidance.

- *Diversity in cross-functional teams may result in decreased team cohesiveness.*
  This claim is supported by the findings of the project and the issue is found in several of the investigated cases.

- *Tendency to dispersed teams in R&D impedes the close communication of Scrum.*
  Dispersed teams have only occurred in a few cases, but these cases form a clear pattern supporting the claim of dispersed teams being an increased impediment to Scrum implementation.

As the list above shows, the picture is rather mixed. Some of the challenges expected based on theory have not been clearly substantiated by in practice. Other theoretically expected challenges have proved to be real impediments to the implementation and conduction of Scrum in integrated product development.

7.4 TOWARDS A REVISED SCRUM FRAMEWORK FOR INTEGRATED PRODUCT DEVELOPMENT

Up to now, the research project has been focusing on the challenges of complying with the Scrum framework in integrated product development. This has resulted in the identification of several challenges in relation to both practical and organisational aspects of implementing and conducting Scrum. Furthermore, a set of general conditions that complicate the transition of Scrum has been presented. Acknowledging that these general conditions exist in the domain of integrated product development, it now seems appropriate to turn a critical focus toward the Scrum framework itself: Instead of forcing changes on the very nature of integrated product development, should the question rather be if Scrum could somehow be adjusted to fit this domain better?

The objective of this chapter is to critically evaluate the existing Scrum framework and propose changes to it by focusing on some alternative practices and workarounds that the companies are applying in practice. This effort results in a revised version of the Scrum framework, adjusted to integrated product development. The list below presents some of the most interesting alternatives and workarounds from the seven cases.

- **FOCUS ON ACTIVITIES - NOT FUNCTIONALITY.**
  Scrum focuses on working functionality as a result of each Sprint. This is clearly a challenge in integrated product development. At the Company G client, the Development Teams focus on defining *activities, not working functionality* at the Sprint Planning meetings. This minor change in the rules allows the developers to define independent tasks for the Sprint Backlog instead of being force to broadly coordinate the development of separate product features, which may not be possible during a single Sprint.

- **FOCUS ON DONE-CRITERIA – NOT SHIPPABLE INCREMENTS.**
  In line with the shift of focus from functionality to activities in the creation of the Sprint Backlog, the focus should shift from evaluating shippable product increments to evaluating development work based on clearly defined done-criteria at the Sprint Review meeting. In fact, each separate task in the Sprint Backlog could be solely defined by its "Definition of Done" as it is largely practiced at Company F. Thereby it gains resemblance to the concept of Test-Driven Development from Extreme Programming, which is another agile method from the software domain.

- **DIFFERENTIATE BETWEEN TECHNOLOGY DEVELOPMENT AND PRODUCT DEVELOPMENT.**
  It has been argued that time and resource estimation is especially difficult in integrated product development, due to its integrated and complex character. At Company D, the developers from time to time have to re-estimate tasks when alternative solutions to some problems are required, but it is a challenge to estimate something "you don't have a clue about." A viable solution to this could be a clear distinction between technology-development projects and normal product development projects as practiced at Company E.

- **FOCUS ON PRODUCT VISION AND PROJECT DEFINITION.**
  In the majority of the investigated cases it has proved difficult to fill-in the role of Product Owner. It has furthermore been argued that this partly is caused by the asymmetry between the roles of Scrum and the traditional management scheme. At the Company G client an attempt has been made to reduce this challenge by boosting the project definition-efforts through workshops with the project team during the early phases. The project definition-document thereby takes over the function of the Product Owner if this has given to a

person. In addition to this, it could be argued that a Product Vision should be a mandatory artefact in the Scrum framework as actually done by interviewee 1 from Company A.

- **FORMAL FIRE FIGHTING TEAM.**
  In some of the investigated cases the challenge of maintaining the necessary focus has been reduced by establishing a formal fire fighting team, responsible for supporting the surrounding organisation, if needed. While the Scrum Master is actually responsible for protecting the Development Team against disturbances, it may sometimes require more than just a gatekeeper to keep incoming and urgent tasks off the shoulders of the team in some development environments. A formal fire fighting team could arguably be a new role in Scrum for IPD.

This set of workarounds is based on the experiences from the development environments investigated in this project and could very well be implemented in a version of Scrum, customised for integrated product development. The initiatives may conflict with certain aspects of true Scrum, but, as all the cases show, a pragmatic approach to the framework seems to be a necessity.

7.5 POSITIONING AND LIMITING THE FINDINGS IN RELATION TO THE PRESENT KNOWLEDGE

So far, the research project has focused on identifying the challenges of becoming agile with Scrum, and more specifically answering the research questions as it has now been done in chapter 7.2. The research has furthermore proposed a revised Scrum framework for integrated product development. However, a relation between the findings of this project and previous research has not been unfolded yet.

As it has been presented in the theoretical foundation in chapter 2, certain research efforts have been focusing on the challenges of changing development practices from "traditional development" to Scrum in the domain of software development. Boehm & Turner (2005) have described a series of management barriers to agile software development. Cooper (2011) has proposed some changes concerning adaptability, flexibility, and agility, to his widely popular Stage-Gate model. Reinertsen (2009) has invited the lean principles from manufacturing into the companies' R&D environments, presenting the concepts of *Product Development Flow*, which to a great extent resemble practices of Scrum. Finally, Smith (2007) has introduced the concept of *Flexible Product Development*, which is directly inspired by agile development from software and which has been brought into the domain of product development. Just as in the present research project, Cooper, Reinertsen and Smith all seem to be motivated by finding a way to overcome *the challenges of speed*, *change*, and *complexity* in the product development industry. However, none of them have focussed specifically on the Scrum framework, which is the case of this research project, and furthermore they seem to primarily address the challenges from a management perspective.

While the present literature seems to be building on a large amount of industry cases, this project, with its limited basis in only seven Danish companies, has approached the field with a great humility. While standing by the claim that the research has in fact provided

certain clarity to the specific situation of implementing and conducting Scrum in an integrated product development environment, it is acknowledged that the extent of findings may be limited. The gathered data are grounded in a Danish development context and, as such, it cannot claim a general validity beyond this domain. The development environments represented in the data are, furthermore, of a size, which also makes the findings less immediately applicable to both small-sized and extremely large-sized companies. However, it is *likely* that most of the findings will resonate in other contexts as well, due to the fact that *the documentation forms*, *the process control models*, and *the general frame of reference* resemble the conditions of the global market.

7.6 EVALUATION OF THE RELIABILITY AND QUALITY OF THE RESEARCH EFFORT

When evaluating the reliability and quality of the research effort it is necessary to clearly define the criteria against which it is evaluated. However, before the presentation of these criteria and an actual evaluation, a summary of the basic premises of the research is needed:

- The data have been collected from real integrated product development environments
- The data have been collected primarily through the memory and experiences of the interviewees, but also through real-time observations.
- The data have been collected across a varied set of organisational contexts

The quality of qualitative research has been widely discussed and in some research environments the legitimacy of the qualitative approach has even been questioned. It therefore seems appropriate to explicate the criteria of quality for which this research effort has been striving.

Beside a set of quality standards that both quantitative and qualitative research should meet, such as *explicating scientific context and purpose*, *choosing appropriate methods*, *specifying those methods*, and *clearly presenting its contribution to knowledge*, Elliot et al. (1999) have developed a set of guidelines especially pertinent to qualitative research. The following sub-chapters are inspired by these guidelines and present a critical perspective on the quality of the present research effort.

7.6.1 OWNING ONE'S PERSPECTIVE

Authors should describe their theoretical, methodological or personal orientation, as those are relevant to the research.

While personal objectivity in general has been my golden path throughout the course of this research project, I surrender to the fact that the role as a researcher cannot be without a certain personal

influence. I have explicated a theoretical standpoint and methodological approach in respectively chapter two and chapter three, but in order to allow the reader to fully interpret the quality and conclusions of the research, I have disclosed a description of my personal motivation, values, and assumptions as part of the preface at the very beginning of the thesis.

7.6.2 SITUATING THE SAMPLE

Authors should describe the research participants and their life circumstances to aid the reader in judging the range of persons and situations to which the findings might be relevant.

The research effort in this thesis is almost purely grounded in empirical findings obtained through interviews with employees in the seven case companies. The memories and experience of these individuals are therefore the basic building blocks that support the concluding results. Each of the interviewees has been carefully positioned in relation to the surrounding organisation and management system in order to establish a certain transparency in their positions and statements. While Elliot et al. (1999) argue that life circumstances of participants should be presented in order to aid the reader in judging the relevance of the findings in the research, it is important to remember, that these guidelines pertinent to qualitative research originate from the domain of psychology, in which life circumstances very well may be of utmost importance. However, in this project the descriptions of the participants have been limited to the ones mentioned above for the following reasons:

1. The practical and organisational issues of conducting and implementing Scrum in integrated product development have been the focus of this project, not social-psychological issues related to the interviewees. It is not to disregard a real understanding of the participants in the research, but rather an attempt to balance the necessary insights with the researchers' responsibility of protecting the interviewees. This leads to the fact that -
2. some participants have expressed concern about how statements in the interview would subsequently be received by their respective organisations. The interviews have been about issues and insights very close to the R&D activities in the companies and therefore sometimes have touched aspects of deep confidentiality. It is important to mention that no confidentiality agreements have been breached in this thesis. However, the worries of the interviewees have been considered important.

7.5.3 GROUNDING IN EXAMPLES

Authors should provide examples of the data to illustrate both the analytic procedures used in the study and the understanding developed in the light of them.

Through the case descriptions and the subsequent analysis, quotations from all interviewees have been used extensively. It could be criticised that the individual quotes have not been presented in a more extensive part of their original context. However, this may in many cases be compensated for by the coupling quotes from multiple interviewees across cases in order to substantiate the respective arguments.

7.6.4 PROVIDING CREDIBILITY CHECKS

Researchers may use any of several methods for checking the credibility of their categories, themes or accounts.

Credibility check can be made by checking the understandings and insights with the original informants; by using additional analysers; by comparing findings to similar studies; or by triangulation with external factors or data (Elliot et al., 1999). This research project could be criticised for its lack of sufficient credibility checks, as none of the activities mentioned above have been extensively performed. However, several people have been presented to the findings, including other researchers and original interviewees.

A collaborative effort based on initial findings has been made with another internationally based research environment. In this collaboration, the findings of the research project resonated with the similar

experiences and findings from a related domain. The identified challenges and themes have furthermore been presented at a conference about Scrum, to which several of the original interviewees from the case companies attended. The findings were well received both by ordinary participants and the interviewees. With this said, it can always be argued that further efforts should have been made to check the integrity of the data, the analysis, and the final conclusions in order to ensure the credibility of the research project.

7.6.5 COHERENCE

The understanding is represented in a way that achieves coherence and integration while preserving nuances in the data. The understanding fits together to form a data-based story/narrative, "map," framework, or underlying structure for the phenomenon or domain.

Throughout this thesis, an effort has been made to create a logical order in which to present the insights of the research project. A theoretical frame of reference as well as a methodological positioning has been presented prior to the case descriptions and the data analysis. Relations between the theoretical framing and the empirical data have been underlined throughout the analysis, and an explicit structure of the analysis has been implemented. The results of the research effort have furthermore been structured in a hierarchical order as *challenges*, *themes* and *general conditions*, which represent three levels of abstraction, rising towards an increasing generalisation. The results of the research thereby establish a coherent story supported by graphical overviews and illustrations.

Elliot et al. (1999) also emphasises the importance of preserving nuances in the data. In this respect, the present research project may be criticised as the upper levels of abstraction, due to the increased generalisation loses the details unique to each of the seven cases. The extensive case descriptions may compensate for this to some extent, but it is acknowledged that issues highly influential in single cases may not be present in the generalised conclusions.

7.6.6 ACCOMPLISHING GENERAL VERSUS SPECIFIC RESEARCH TASKS

Where a general understanding of a phenomenon is intended, it is based on an appropriate range of instances (informants or situations). Limitations of extending the findings to other contexts and informants are specified.

The research effort in this project has primary been led by the objective of identifying the basic challenges of becoming agile in integrated product development environments using Scrum. Answers have been found through a case study of seven development environments across a varied set of organisational contexts, and the diversity has had the purpose of reaching a general understanding. It is, however, difficult to accomplish this based on only seven cases, and it is therefore important to emphasise the limitations to the applicability of the findings. This has been done in chapter 7.5.

7.7 EVALUATION OF THE RESEARCH METHODS

The previous sub-chapter was about the criteria for quality in qualitative research. The purpose of this sub-chapter is to evaluate the research methods carried out in the project. More specifically the methods used for data collection and the subsequent evaluation and analysis of the data.

7.7.1 DATA COLLECTION

The data have been collected through a case study of seven development environments in Denmark. Each of the seven development environments was investigated through interviews with employees. However, despite this seemingly consistent pattern, deviations from the planned interview situations happened several times. Each interview followed the interview guide that was made in advance, but the sequence of the questions was occasionally changed in order to follow the flow and themes in the on-going discussion. Though the intention was to conduct two separate interviews in each organisation, the situation would sometimes only allow one interview with two or three persons at the same time, due to the tight schedules of the interviewees. Even though this was not planned, it was found that the discussions between the interviewees in such situations resulted in highly valuable data.

Normally, multiple sources of data would be preferred in order to perform credibility checks through triangulation (Elliot et al., 1999). Without doubt, observations of the working Development Teams, or participation in Daily Scrum meetings or Sprint Reviews would have produced data that could benefit the research project. Documentation such as project plans, Product Backlogs, and actual Sprint prototypes could also have provided important insights in order to establish a more complete picture of the individual development environment. However, in most cases the companies found too critical to disclose these elements, which is why they are not part of the data.

In general, the data collection went without problems, and most of the interviewees were enthusiastic about participating, but the fact that several of them regarded their experience with Scrum as highly valuable and a strategic competition parameter, also influenced the data collection. In some cases it took from two to eight months of correspondence with developers and upper management before green light was given for the interviews to take place. Due to a relatively low number of known companies that use Scrum in integrated product development, it was difficult to establish collaboration with the few ones that actually do. This may have influenced the data collection as the participating companies ultimately controlled the data available to the project.

7.7.2 DATA ANALYSIS

The data collected through recorded interviews and observations were examined case by case in a process of both transcription and writing. Each case description was structured the same way by presenting various aspects under a common set of headlines. The recordings and transcriptions were all thoroughly examined for passages where the interviewees talked about challenging aspects of the work with Scrum. The identified issues were then summarised at the end of each case description, which later formed the basis for the emerging themes in the second part of the analysis.

A consistent way of treating each data set is often considered good practice. However, it can also be argued that the systematic approach was an impediment to conducting “deep dives” into specific parts of the data of special interest. Each data set was treated equally, even though the analysis made clear that the compliance with Scrum varied from case to case. In their guidelines to qualitative research, Elliot et al. (1999) argue that conducting an intensive analysis of a diverse subset of the informants, supplemented with a less intensive examination of the rest, is an example of good practice that makes the analysis manageable. This was not the practice in this research project, and it is therefore a question whether the analysis has fully taken advantage of the diversity and inherent details of the data.

The division of the analysis into three parts was done in order to explicate the levels of abstraction on which the analysis was carried out. This provided a clear structure, but also certain limitations in regard to letting the qualitative data reveal its own structure and patterns. However, this was also a way to make the relatively large amount of identified challenges manageable.

As it has earlier been stated in chapter 3 about research methodology, this thesis subscribes to the constructivist approach to producing meaningful knowledge. It also subscribes to its ontological assumptions about reality as social construction. From this ontological stance the analytical efforts may be criticised for not being true in other social contexts, and even that it cannot be rightfully judged, due to the fact that meaning is reached through subjective interpretations of observations. This essentially means that the sender and the receiver of the message of this thesis may fail to get a mutual understanding of it. Even though it may not come that far, both parts of this critique have been met in the thesis. Firstly, the findings are based on a varied set of development organisations, which ensures that they are not simply constructs on the basis of a single context. Secondly, the research findings have gained resonance in presentations and discussions with both people participating and people outside the research project. Furthermore, the collected data have been made quite transparent in the extensive case descriptions in chapter 4.

7.8 FUTURE PERSPECTIVES

So far, this project has been focusing on answering the research questions concerning the challenges of implementing and conducting Scrum in an integrated product development context. In this process several other questions have surfaced from the project. This last part will outline some of the perspectives of the research findings that could be further built on. An important perspective – or perhaps more rightly a potential – of Scrum, grounded in the case study, has already been presented in chapter 7.3. The following sub-chapters will outline some additional future perspectives on the research project.

7.8.1 MORE EMPHASIS ON THE PRODUCT DESIGN AND AESTHETICS

While this project has had its emphasis on the challenges of using Scrum in an integrated product development environment, it has not specifically investigated the implications of using Scrum as regards the quality of product designs and aesthetics in these environments. Further research could be focusing at these aspects, but that would require new cases, as these aspects have been of relatively little importance in most of the cases in this project. A special focus on *products* would supplement the relatively *process-oriented* findings of the present research project.

7.8.2 SCRUM APPLICABLE TO INDUSTRIAL DESIGNERS?

In addition to an increased focus on the products deriving from development efforts conducted in a Scrum framework, further investigation of the applicability of Scrum to industrial designers could prove beneficial. This project has shown a special interest in the broad mechanical development discipline of the "Product Designer." However, the creative, concept-focused, and holistically oriented industrial design discipline has not been contemplated, except for a theoretically based discussion on Scrum versus the basic design process. In order to carry out a more thorough study on how the Scrum framework would apply to the practice of industrial designers, cases

would probably have to be found in other industries than the ones represented in this project. Furthermore, different research methods such as the action-oriented approach could prove valuable in such cases.

7.8.3 DESIGN EDUCATION

Parallel to the research efforts presented in this thesis, certain aspects of Agile Development have been examined in relation to the work of industrial design students at Aalborg University. This work has indicated an interesting potential in relation to project- and problem-based learning environments, and as such, Scrum as a generic framework is not only limited to product development. As it has also been mentioned earlier in this project, the Scrum process *does* to some extent resemble the dual-reflective process of learning and *might* naturally be implemented as a management framework for learning through problem-based projects. A parallel has been seen in a few of the cases in this project, where technology-development projects with a strong learning focus have been conducted through the highly tact-based Scrum process. Traditional project management practices of heavy up-front planning with Gantt charts is often presented as the best practice in the educational environments and Scrum or similar agile practices could provide an interesting counterbalance.

7.8.4 AGILE 2.0 – MORE OF THE SAME?

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

The 12th principle behind the Agile Manifesto promotes regular adjustments in order to become more effective. Surprisingly, more than ten years after the authoring of the manifesto, it has not changed itself. During this time software development has accelerated enormously, and while the manifesto suggests a frequency of software deliveries measured in weeks and months, the most effective software development environments now deliver working software 50 times a day in a continuous cycle of prototyping and feedback data analysis (Boeg, 2012). While this may not be possible or even desirable in physical product development, inspiration could arguably be found in the extreme prototyping frequency and heavily data-based user feedback of these thought-provoking practices. However, this may eventually be just yet another pursuit of bliss by reaching increased effectiveness and quality in product development. The question that remains unanswered is then whether or not effectiveness and quality will stay as the dominant measures of success in integrated product development for a fast-paced and global, commercial market. ■

REFERENCES

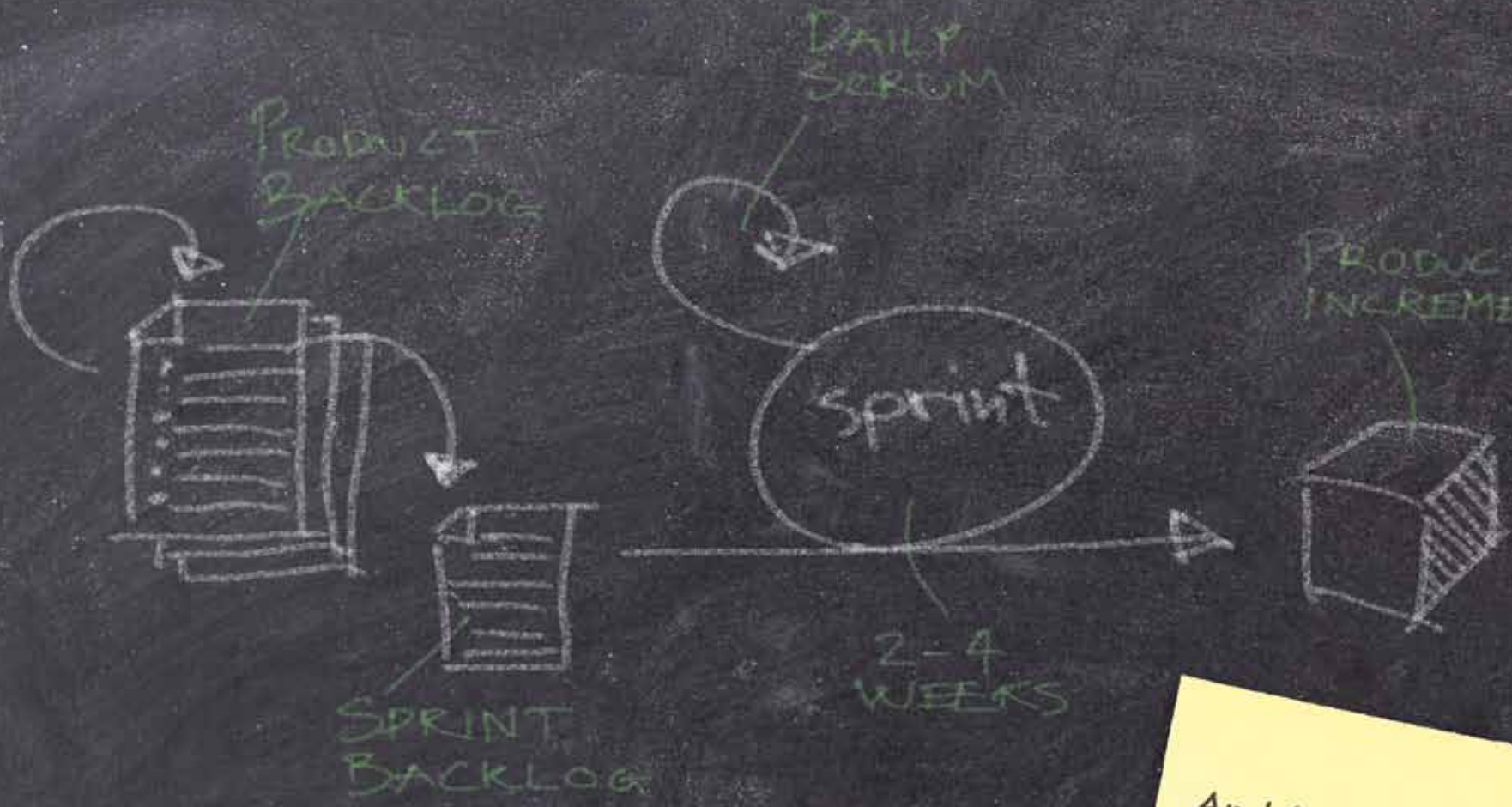
- Abrahamsson, P., Warsta, J., Siponen, M.T. and Ronkainen, J. New Directions on Agile Methods: A Comparative Analysis, *Proceedings of the International Conference on Software Engineering*, 2003 (Oregon, USA)
- Adler, S.P. Interdepartmental interdependence and coordination: The case of the design/ manufacturing interface, *Organizational Science*, 1995, 6, 147–167
- Andersen, H. *Vetenskapsteori och metodlära – En Introduktion*, 1994 (Studentlitteratur, Lund)
- Andreasen, M.M and Hein, L. *Integrated Product Development*, 2000 (IPU, Copenhagen)
- Argyris, C. and Schön, D. *Organizational Learning: a theory of action perspective*, 1978 (Addison-Wesley Co., Mass.)
- Ballé, F. and Ballé, M. Lean Development, Business Strategy Review, 2005, Autumn
- Barry, E.P. *Integrated Product Development Implementation Guide*, 1993, http://www.fas.org/spp/military/docops/smc/ip-dsmc.htm , 22-05-2012
- Bassey, M. On the nature of research in education (part 2), *Research Intelligence*, 1990, 37, Summer
- Bernstein, R.J. *Beyond objectivism and relativism*, 1983 (University of Pennsylvania Press, Philidelphia)
- Blaikie, N. *Designing Social Research*, 2000 (Policy Press, UK)
- Blessing, L.T.M. A process-based approach to computer supported engineering design, *Proceedings of the International Conference on Engineering Design (ICED'93)*, August 1993, 1393-1400
- Boeg, J. *Agile 2.0*, IDA Scrum conference, april 2012 (IDA, Aalborg)
- Boehm, B.W. *Software Engineering Economics*, 1981 (Prentice Hall, Upper Saddle River, New Jersey)
- Boehm, B.W. A Spiral Model of Software Development and Enhancement. *Computer*, 1988, **21**(5), 61-72
- Boehm, B.W. and Turner, R. *Balancing Agility and Discipline – A Guide for the perplexed*, 2004 (Addison-Wesley, Boston)
- Boehm, B.W. and Turner, R. Management Challenges to Implementing Agile Processes in Traditional Development Organizations, *IEEE Software*, 2005, **22**(5), 30-39
- Brannon, Evelyn L. *Fashion forecasting: Research, Analysis, and Presentation*, 2nd edition, 2005 (Fairchild Publications, New York)
- Brügger, N. *Virilio – Essays om dromologi*, 2001 (Inroite! Publishers, Copenhagen)
- Chamorro-Premuzic, T. and Reichenbacher, L. Effects of personality and threat of evaluation on divergent and convergent thinking, *Journal of Research in Personality*, 2008, 42, 1095-1101
- Coombes, H. *Research Using IT*, 2001 (Palgrave, UK)
- Cooper, R.G. *Winning at New Products – Creating Value through Innovation, 4th ed.*, 2011(Basic Books, New York, USA)
- Cross, N. Designerly Ways of Knowing, *Design Studies,* 1982, 3(4), 221-227

- Elliot, R., Fischer, C.T. and Rennie, D.L. Evolving guidelines for publication of qualitative research studies in psychology and related fields, *British Journal of Clinical Psychology*, 1999, 38, 215-229
- Eriksen, K. *Liv eller Brand – Designeren som udvikler af bæredygtige systemer og produkter*, 2003 (Aalborg Universitet, Aalborg)
- Gassmann, O. and Zedtwitz, Mv. Tends and determinants of managing virtual R&D teams, *R&D Management*, 2003, 33(3), 243-262
- Gerwin, D. and Barrowman, N.J. An evaluation of research on integrated product development, *Management Science*, 2002, 48, 938–953.
- Griffin, A. PDMA Research on New Product Development Practices: Updating Trends and Benchmarking Best Practices, *Journal of Product Innovation Management*, 1997, 14, 429-458
- Griffin, A. The Effect of Project and Process Characteristics on Product Development Cycle Time, *Journal of Marketing Research*, February 2005, 34(1), 24-35
- Hartmann, K.L. and Ovesen, N. *Trendspotting – a Designer's Guide*, 2006(Aalborg University)
- Hayes, R.H., Wheelwright, S.C. and Clark, K.B. *Dynamic Manufacturing: Creating the Learning Organization*, 1988, (The Free Press, New York)
- Hertenstein, J.H. and Platt, M.B. Performance Measures and Management Control in New Product Development, *Accounting Horizons*, 2000, 14(3), 303-323
- Heylighen, F. *Epistemology* – Introduction, 1993, http://pespmc1.vub.ac.be/EPISTEMI.html, 13-01-2012
- Highsmith, J., *Manifesto for Agile Software Development*, 2001, agilemanifesto.org, 21-02-2011
- Highsmith, J. and Cockburn, A. Agile Software Development: The Business of Innovation, *Computer*, 2001, 34(9), 120-127
- Hoegl, M., Ernst, H. and Proserpio, L. How Teamwork Matters More as Team Member Dispersion Increases, *Journal of Product Innovation Management*, 2007, 24, 156-165
- Hoegl, M. and Proserpio, L. Team member proximity and teamwork in innovative projects, *Research Policy*, 2004, 33, 1153-1165
- Jensen, L.B. *Fra Patos til Logos: Videnskabsretorik for begyndere*, 2004 (Roskilde Universitetsforlag, Roskilde)
- Karlsson, C. and Åhlström, P. The Difficult Path to Lean Product Development, *Journal of Production Innovation Management*, 1996, 13(4), 283-295
- Kerzner, H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 10th ed., 2009 (John Wiley & Sons, New Jersey)
- Kim, J. and Wilemon, D. Sources and assessment of complexity in NPD projects, *R&D Management*, 2003, 33(1), 15-30
- Kim, J. and Wilemon, D. The Learning Organization as Facilitator of Complex NPD Projects, *Creativity and Innovation Management*, 2007, **16**(2), 176-191

- Koufteros, A.X., Vonderembse, M.A. and Doll, W.J. Integrated product development practices and competitive capabilities: The effect of uncertainty, equivocality, and platform strategy *Journal of Operations Management*, 2002, 20, 331–355
- Kovács, G. and Spens, K.M. Abductive reasoning in logistics research, *International Journal of Physical Distribution & Logistics Management*, 2005, **35**(2), 132-144
- Krafcik, J.F. Triumph Of The Lean Production System, *Sloan Management Review*, 1988, 30(1), 41-52
- Kusar, J., Duhovnik, J., Grum, J. and Starbek, M. How to reduce new product development time, *Robotics and Computer-Integrated Manufacturing*, 2004, 20, 1–15
- Kvale, S. and Brinkmann, S. *InterView – Introduktion til et håndværk, 2nd ed.*, 2009 (Hans Reitzels Forlag, København)
- Larman, C. *Agile and Iterative Development: A Manager's Guide*, 2007 (Addison Wesley, Boston MA)
- Lawson, B. *How designers think*, 1980 (Architectural Press, London)
- Lawson, B. *How designers think – The Design Process Demystified 4th ed.*, 2005 (Architectural Press, Burlington MA)
- Lin, C.T., Chiu, H. and Tseng, Y.H. Agility evaluation using fuzzy logic, *International Journal of Production Economics*, 2006, 101, 353-368
- Losada, M. and Heaphy, E. The Role of Positivity and Connectivity in the Performance of Business Teams, *American Behavioral* Scientist, 2004, 47(6), 740-765
- McDonough III, E.F. Investigation of Factors Contributing to the Success of Cross-Functional Teams, *Journal of Product Innovation Management*, 2000, 17, 221-235
- McMichael, B. and Lombardi, M. ISO 9000 and Agile Development, *Proceeding of AGILE 2007*, August 2007, 262-265
- Menon, A., Chowdhury, J. and Lukas, B.A. Antecedents and outcomes of new product development speed - An interdisciplinary conceptual framework, *Industrial Marketing Management*, 2002, 31, 317-328
- Mikkelsen, H. and Riis, J.O. *Grundbog i projektledelse*, 1996 (Promet, Holte, Denmark)
- Murmann, P.A. Expected Development Time Reductions in the German Mechanical Engineering Industry, *Journal of Product Innovation Management*, 11(3), 236-252
- Myllerup, B. *Kulturen i agile organisationer – en oftest overset succesfaktor*, IDA Scrum conference, april 2012 (IDA, Aalborg)
- Naveh, E. The effect of integrated product development on efficiency and innovation, *International Journal of Production Research*, July 2005, 43(13), 2789-2808
- Nielsen, J.S. *Lean Produktudvikling*, Foredrag hos IDA Nordjylland, 28-09-2011.
- Ogunnaike, B.A. and Ray, W.H. *Process Dynamics, Modeling and Control*, 1994 (Oxford University Press, USA)
- Ovesen, N., Eriksen, K. and Tollestrup, C. Agile Attitude: Review of Agile Methods for Use in Design Education, in Proceedings from *The 13th International Conference on Engineering and Product Design Education*, 2011 (Design Society, UK) 505-510
- Paterson, R. *More on Complex versus Complicated*, 1996. Website: http://smartpei.typepad.com/robert_patersons_weblog/2006/11/more_on_complex.html (17-11-2011)
- Phillips, R., Neaily, K. and Broughton, T. A comparative study of six stage-gate approaches to product development, *Integrated Manufacturing Systems*, 1999, 10(5), 289-297

- Reinertsen, D.G. *Types of Processes – by Don Reinertsen*, 2009. Website: http://www.netobjectives.com/blogs/types-processes-don-reinertsen (22-05-2012)
- Reinertsen, D. G. *The Principles of Product Development Flow – Second Generation Lean Product Development*, 2009, (Celeritas Publishing, CA, USA)
- Rother, M. and Shook, J. *Learning to See: Value Stream Mapping to Create Value and Eliminate Muda*, 2003 (Brookline, Mass, USA)
- Royce, W.W., Managing the Development of Large Software Systems, in *Proceedings, IEEE Wescon*, TRW, August 1970, pp1-9
- Rudman, C. *Answer to Jeff Watson*, 2010 (http://www.linkedin.com/answers/business-operations/project-management/OPS_PRJ/611543-1643668 28-05-2012)
- Schwaber, K. *Agile Project Management with Scrum*, 2004 (Boston, MA, USA)
- Schwaber, K. And Sutherland, J. *The Scrum Guide – The Definitive Guide to Scrum: The Rules of the Game*, 2011 (as available at http://www.scrum.org/storage/scrumguides/Scrum_Guide.pdf 02-01-2012)
- Smith, P.G. *Flexible Product Development: Building Agility for Changing Markets*, 2007 (John Wiley & Sons, California)
- Smith, P.G. and Reinertsen, D.G *Developing Products in Half the Time: New Rules, New Tools 2nd ed.*, 1998 (John Wiley & Sons, NY)
- Snowden, D.J. and Boone, M.E. A Leaders's Framework for Decision Making, *Harvard Business* Review, 2007, 85(11), 68-76
- Sobek, D.K. II, Ward, A.C. and Liker, J. Toyota's Principles of Set-based Concurrent engineering, *Sloan Management Review*, Winter 1999, 40(2), 67-83
- Souza, G.C., Bayus, B.L. and Wagner, H.M. New-Product Strategy and Industry Clockspeed, *Management Science,* 2004, 50(4), 537-549
- Stanton, N. *Human Factors in Consumer Products*, 1998 (Taylor & Francis, London)
- Stolterman, E. Guidelines or aesthetics: design learning strategies, *Design Studies*, 1994, 15(4)
- Striim, O. *Kreativ problemløsning og praktisk Idéudvikling*, 2001 (Nordisk forlag, Denmark)
- Sutherland, J., Viktorov, A., Blount, J. and Puntikov, N. Distributed Scrum: Agile Project Management with Outsourced Development Teams, *Proceedings of the 40th Hawaii International Conference on System Sciences*, 2007, IEEE Computer Society
- Takeuchi, H. and Nonaka, I. The new new product development game, *Harvard Business Review*, 1986, 64(1), 137-146
- Tatikonda, M.V. and Rosenthal, S.R. Technology Novelty, Project Complexity, and Product Development Project Execution Success: A Deeper Look at Task Uncertainty in Product Innovation, *IEEE Transactions nn Engineering Management*, 2000, 47(1), 74-87
- Taylor, S.S., Fisher, D. and Dufresne, R.L. The aesthetics of management storytelling: a key to organizational learning, *Management Learning*, 2002, **33**(3), 313-30
- Tollestrup, C., Eriksen, K. and Ovesen, N. Research Oriented Projects on Design Themes – A Master Semester at AAU, in proceedings from *The 13th International Conference on Engineering and Product Design Education*, 2011 (Design Society, UK) 505-510
- Trost, J. and Jeremiassen, L. *Interview i praksis*, 2010, (Hans Reitzels Forlag, Copenhagen)

- Truex, D., Baskerville, R. & Travis, J., Amethodical systems development: the deferred meaning of systems development methods, in *Accounting, Management and Information Technologies,* 2000, **10**(1) 53-79
- Ulrich, K.T. and Eppinger, S.D. *Product Design and Development, 3rd ed.*, 2003 (McGrag-Hill, Singapore)
- Vesey, J.T. The New Competitors: They think in terms of 'speed-to-market'. *The Executive*, 1991, 5(2), 23-33
- Vliet, H.v., Software Engineering: Principles and Practice, 2007 (Wiley, California)
- Wenneberg, S.B. *Socialkonstruktivisme som videnskabsteori, Working* Paper, 2002, (Department of Management, Politics and Philosophy at Copenhagen Business School)
- Wheelwright, S.C. and Clark, K.B. *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality* (1992) (The Free Press, NY)
- Wisker, G. *Chapter 5: Research Questions and Hypothesis. In: The Postgraduate Research Handbook, 2nd ed.* , 2008 (Palgrave MacMillan, UK)
- Yin, R.K. *Case Study Research – Design and methods, 3rd ed.*, 2003 (Sage Publications, California)
- Zedtwitz, M.v. Organizational learning through post-project reviews in R&D, *R&D Management*, 2002, 32(3), 255-268



And by the way:
Find a way
to handle the
mech.-developers!