



Venla Manninen

THE AGILE TRANSFORMATION

THE AGILE TRANSFORMATION

Venla Manninen
The Agile Transformation
Spring 2018
Business Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Business Information Technology

Author(s): Venla Manninen

Title of Bachelor's thesis: The Agile Transformation

Supervisor(s): Ilkka Mikkonen

Term and year of completion: Spring 2018

Number of pages: 58

Within the past ten years, the ways of developing products and services has underwent significant changes. The software and service development environment has been transforming, manifested in technologies, people and processes. As a result, organizations have been facing pressure to be highly adaptive to change. To study the changes, it is necessary to examine the development processes as well as human and business-related aspects.

This research began with a purpose to study the characteristics of agile transformation according to a set of predefined values gathered in the Agile Manifesto. Recent studies of Agile have not seemed to focus on changes in the organizational level. Insights about agile transformation are significant as the environment is in a state continuously evolving, which will affect the methods and frameworks chosen to develop products and services.

The framework of this thesis consists of a set of literature, online and video material as well as qualitative methods, by including an interview. The thesis has been able to distinguish factors, which differ between a traditional and an agile organization. Conclusion suggests that qualitative research support the findings in the theoretical framework. Further research on the topic of agile transformation is encouraged.

Keywords: Agile, Scrum, project management, change management

CONTENTS

1	INTRODUCTION	5
2	AGILE DEVELOPMENT	6
2.1	Software Development Life Cycle	6
2.2	Traditional software development – the Waterfall model	7
2.3	Moving into Agile Development	9
2.3.1	Scrum.....	11
2.3.2	Lean Software Development	13
2.3.3	Kanban.....	14
2.5	DevOps	16
3	THE AGILE TRANSFORMATION.....	18
3.1	What makes an organization agile?	18
3.2	Agile culture change	21
3.3	Management and Leadership in Agile.....	23
3.3.1	Change management.....	27
3.4	APM Framework and Agile Delivery	30
3.5	Case: Spotify’s Agile Model	31
4	INTERVIEW.....	38
4.1	Goal and scope of interview.....	38
4.2	Company A.....	39
4.3	Company B.....	42
4.4	Company C	48
4.5	Summary of results.....	51
5	CONCLUSION.....	53
6	DISCUSSION	55
7	REFERENCES	56
8	APPENDICES	59

1 INTRODUCTION

Today we are facing an environment, which has become increasingly difficult to predict. Agile was a new concept, aiming to respond to such uncertainty and the methodology has since been adopted as a preference to developing software. According to some studies, it has a long history, with earliest origins tracing back all the way to 1940's. The principles and values, which would provide a definition of Agile, referred to as "the Agile Manifesto" arose in 2001. At that time, the concept of Agile became widely known and the implementation of the different frameworks began. Of those frameworks, Scrum has been the most widely used framework in the world. (cPrime Worldwide. What is Agile? What is Scrum? Retrieved 1.6.2018). As the methodology has since evolved, it has created other frameworks, such as Lean software development and Kanban.

One definition of the agile transformation describes it as "an act of transforming an organization's form or nature gradually to one that is able to embrace and thrive in a flexible, collaborative, self-organizing, fast changing environment." (Agile Transformation: Understanding What it Means to be Agile. Retrieved 1.6.2018). Upon reflecting the agile transformation, the matter is seen as something more than simply choosing a set of methods or practices to deliver in an agile environment. Rather, it also involves the issues of culture change and mindset, to be able to support a self-organized, collaborative environment inside an organization.

The theoretical framework of this thesis consists of a basic overview to agile software development, considerations of the characteristics of an agile organization and discussion about the role of management, including change management. A Project Management Framework is included as well. The purpose of this thesis is to study the impacts of Agile not only in software development but on an organizational level, to identify the differences between traditional and agile organizations and to examine the benefits and challenges of agile adaptation. This research includes qualitative methods. The agile practices of three large software companies examined by interviewing individual employees. The chosen methods aim at gaining a deeper understanding of the topic, rather than attempting to generalize the results. The organizations involved in the interviews have chosen to stay anonymous.

2 AGILE DEVELOPMENT

This chapter will provide a basic, methodological overview on software development, introducing the life cycle -model of software development as well as short introduction and comparison between traditional software development and the adaptation of a more modern approach in the form of Agile development. There are several different agile software frameworks or methods, which may be implemented within an organization. A few of the most common frameworks and their characteristics will be introduced and discussed in this chapter.

2.1 Software Development Life Cycle

Software Development Life Cycle, also referred to as SDLC, is a process including a series of steps or phases, which provide a model for the development and lifecycle management of an application or piece of software. Depending on the industry or organization, the process may vary while certain standards, such as ISO/IEC 12207 provide a mode for the developing, acquiring and configuring of software systems. The SDLC process aims to ensure a high quality of software, with other major benefits including for example cost-effectiveness and efficiency. The SDLC process typically consists of five different stages. The illustration below describes the process (Airbrake Blog, 2013. What is the Software Development Life Cycle (SDLC)? Retrieved 1.9.2017.)

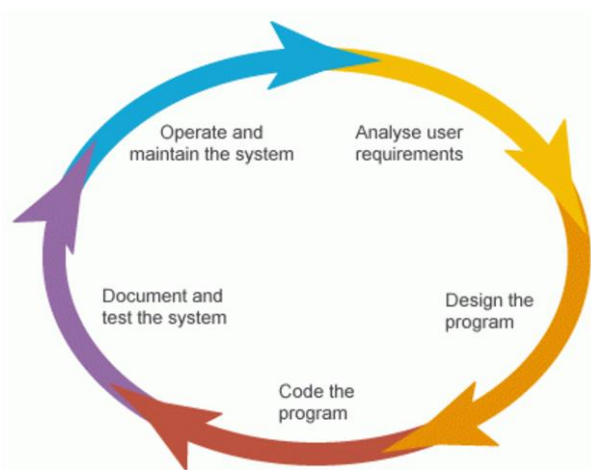


FIGURE 1. (Airbrake Blog, 2013. Software development life cycle. What is the Software Development Life Cycle SDLC? Retrieved 1.9.2017).

Analyzing the user requirements is a critical stage of any software project. This phase involves defining and documenting the expectations of clients or team members. The process is iterative, as an extensive amount of communication will take place between stakeholders, end users and project team members. Different techniques may apply for gathering the data, such as customer interviews and surveys, building of use cases or demonstrating the design by using prototypes.

When designing the program, technical design requirements as well as business requirements need to be prepared. Software architects and lead developers are typically in charge of the technical side of design. Some of the activities defined in this stage include conducting risk analysis and mapping out functional as well as non-functional specifications, which may be interface requirements and other significant details, analyzing of database capacity needs and performance or response times. Risk analysis is crucial as potential threats and vulnerabilities may occur when introducing software or a piece of software to other systems. Depending on the level of risk in privacy matters, some projects may require legal assistance. Legal review may relate to the collecting of personal data or acquiring of permissions and gaining authorizations. (Airbrake Blog, 2013. What is the Software Development Life Cycle (SDLC)? Retrieved 1.9.2017.)

The actual coding phase is typically the longest phase of SDLC. It usually includes conducting something called unit testing. This means that the smallest possible parts of an application (units) are tested in isolation. In this development stage, some changes and adjustments can occur. The outcome of the coding moves into the next phase, which is the documentation and testing where a variety of testing methods may be implemented, for example integration and system testing as well as end user testing. The testing phase determines whether the system needs further analysis, design or coding (Rouse, Margaret. TechTarget. Retrieved 1.9.2017.)

The final phase of SDLC is the implementing or deploying of the system. Training the users or employees may be necessary in this stage. When the system will be made available depends on the size and form of the organization. Waterfall and Agile methodologies are options to use as part of SDLC process. I will next shortly introduce these two different methodologies, including some of their advantages and disadvantages. (Rouse, Margaret. TechTarget. Retrieved 1.9.2017.)

2.2 Traditional software development – the Waterfall model

The Waterfall model represents a classical engineering approach to software development. It is an example of a sequential process control where each step of the development is finished before moving into the next. The Waterfall model is rather simple to demonstrate and understand, as the different steps included in the Waterfall process are visible below.

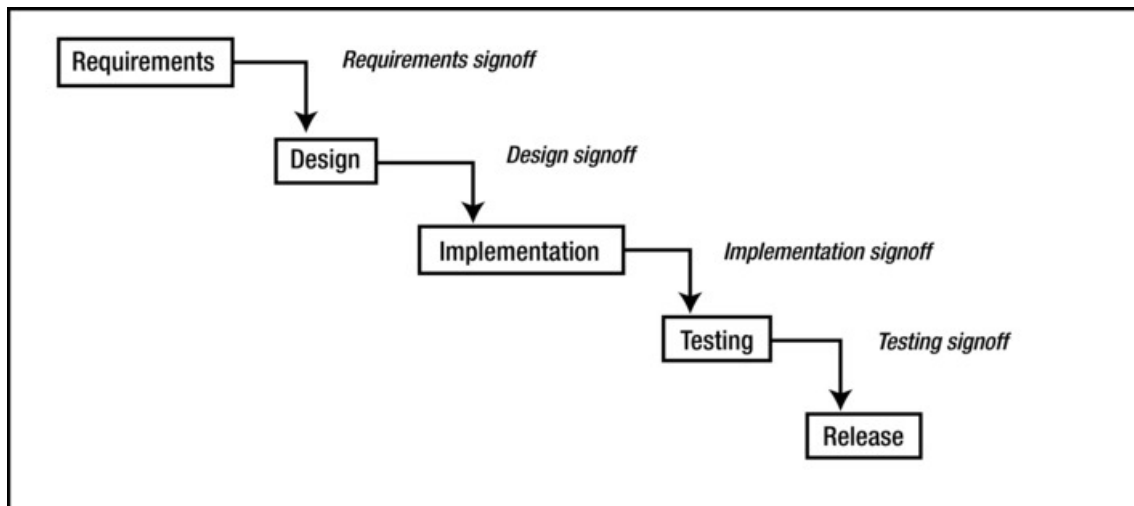


FIGURE 2. Waterfall process. *Simplicity Through Breadth. Software Development Life Cycle (SDLC) Waterfall Model. Model. Retrieved 30.5.2018.*

A fundamental characteristic of the Waterfall model is the careful preparation of documentation between different stages of development. As there are multiple separate teams working on the projects, they must rely on receiving and recording of approvals before moving on to the next stage of development. It is worth noticing that these teams may often operate in isolation from another, having very little or no communication with one another. Once the requirements for the system has been set and documented and the technical team have made a design to meet those requirements, many different management teams will review the completed design. Representatives of technical management and business users as well as project management need to assess the design before the implementation stage can begin. Something referred to as “Gap Analysis” also takes places to address some requirements should they be missing from the design. The final design version needs to cover all needed requirements. (Crookshanks, E. 2015, p.91.) Before release, the code needs will acquire testing to see if the set requirements apply. Following a typical Waterfall model, this would be the first time that any testing is done.

Now the question becomes, what happens if the outcome does not match the expectations? The Waterfall method has been widely criticized for its' inflexible nature, which may often result in creating high costs. As it is possible to detect any mistakes late in the testing phase, they may prove difficult and expensive to fix, especially if there have been fundamental issues very early on when designing the concept. In some projects, expectancy to changes in requirements may take place. Therefore, a more flexible approach is necessary. Other critique regarding the Waterfall model has been made about high-level of risk and uncertainty factors as well as not seeing an immediate return on investment (ROI) (Crookshanks, E. 2015, p.92.)

Based on the critique, it can be discussed whether the Waterfall-model should be disregarded for being too old-fashioned or inadequate. Perhaps that is not entirely the case. There are definite positives related to the method as well, for example being easy to manage and often seen as a suitable option for small projects. Some organizations have modified the method with adding a more iterative approach to it, referred to as "iterfall", where the development project divides into smaller phases, following the idea of a waterfall process while breaking it down to some extent. Another way to address the issue of inflexibility has been to cut releases into stages, keeping the documentation and development according to waterfall method with the project fully designed in advance. This is part of the "Big Design Up Front" -model. (Crookshanks, E. 2015, p.92.)

While modifications to the more traditional ways have taken place to fit the needs of current software development, there has been a growing need to implement a style, which is faster, more iterative, encourages more co-operation and faster knowledge transfer between project teams, while also responding better to customer needs. For this purpose, agile methodologies are gaining popularity among different organizations, for different projects. Whether this transfer is successful or not, may often depend in how organizations are able to handle the cultural change involved.

2.3 Moving into Agile Development

As developers recognized some of the challenges related to waterfall development, new ideas and methods began to surface around 1980s and 1990s. Rapid Application Development (RAD) was one popular, new method built around creating prototypes for requirements specification and design validation. Other approaches began to emerge around a concept of developing software in

an incremental way, focusing on reducing of risk by foreseeing the result. (Girvan & Paul. 2017. p.80.) As different ways to approach the development of software products emerged, the new ways of thinking finally led to Agile emerging in the early 2000s.

The core idea of agile methodology is breaking larger tasks or features into small pieces, built in short cycles, most typically lasting from one to four weeks. The planning, requirements specification, design, coding and testing involves working in small teams. Agile was initially about developing code and creating systems *fast*, with emphasis on high quality and the gaining of better customer satisfaction. It is a method, which is open and flexible to changes during development and doing so, can help minimize risk (Girvan & Paul. 2017. p. 85.)

Something referred to as The Agile Manifesto explains the core ideology of agile. Created in 2001, it identifies four unique values and a set of 12 principles, which lie at the core of agile development and delivery. The values defined in the Agile Manifesto include the following; *individual and interactions* over processes and tools, *working software* rather than clear documentation, *customer collaboration* over negotiating contracts and finally – *responding to change* over following a plan. Thus, while it is recognized that the latter mentioned do possess value, agile will place greater emphasis on the before mentioned.

Successful results with Agile include delivering products to customer fast and frequently, being able to learn effectively, saving considerable amounts in software development costs and enhancing of teamwork and co-operation. (Saffer, D. Designing for Interaction. 2010, p.191). However, the agile methodology possesses some challenges as well. Particularly, designers may find it challenging as it may not allow much time to be put into the ideation process or considering between different options and may involve miscommunication or designs being implemented differently to the designers' intent. (Della Tore, L. How to become a design-driven company in an agile world, 2017. Retrieved 20.5.2018)

As already mentioned, one of the core principles of Agile includes that changes in development must be *adding value* to the customers. The development cycles are short due to enabling of regular feedback from stakeholders, for keeping them engaged in the project. In agile development, small changes made into the software during the development are valuable as they prevent customers from making many change requests once the design is completed, thus resulting in better customer satisfaction on a longer term. Ability to respond to change relies in a focus on

technical excellence and attention to good software design, allowing the software to be easy to maintain.

Co-operation between business management and developers is also key. In agile projects, face-to-face communication may often be the most effective way of communicating and interacting between stakeholders. The input given by individual team members is also meaningful. A successful project requires the team to be motivated, with the ability to self-direct and organize itself. How to manage the teams depends on how the teams collaborate. (Girvan & Paul 2017. p.82). A few, popular agile development frameworks shall be introduced and discussed next.

2.3.1 Scrum

Scrum is a lightweight framework used for developing products, typically software but also services, marketing or some other desired result. Lightweight, from a software perspective, refers to the reduction of waste from having to do rework caused by insufficient planning or unnecessary documentation. (O. Coplien, J.; Bjørnvig, G. 2010. p.3).

The Scrum method has been said to suit best when working in small teams. However, there can be many Scrum teams within a project, especially when the team size exceeds ten members. Three main principles act as guidelines for a Scrum project: transparency, inspection and adaptation. As for transparency, every stakeholder of the project has access to it. To keep up with progress, regular inspections are put in place to making sure that goals are met while adaptability allows for adjustments to be made if any problems should occur.

The Scrum organization generally consist of only three different roles: Product Owner, the team and Scrum Master. The Product Owner is sometimes referred to as product manager while rather often the two roles are integrated into one. The Product Owner may be coming from a business or product development background and carries the bigger vision for a piece of software without having to own knowledge about how to write the actual code. The Product Owner creates the requirements for the software as well as organizes, prioritizes and evaluates it in each sprint. The requirements which are needed for the software are managed in something called a product backlog, which helps to prioritize and plan the development tasks for each sprint. The Scrum Master

has the role of a project manager, which differs from a traditional way of managing a team. The Scrum Master is more of a coordinator or a coach who makes sure that everyone is following the rules of Scrum, such as attending certain ceremonies like regular meetings or using necessary tools. While interacting daily with the team, the Scrum Master would not interfere too much in the work produced by it.

The work of a Scrum project is developed in cycles, which are called sprints. At the end of each sprint, the team creates a potentially shippable product, called a sprint increment. (Canty 2016, p.70.) Every sprint is expected to produce an outcome, which is regarded as the goal of that sprint. There are certain rules, which teams must follow during the sprints. After a goal has been set for each sprint, it is crucial that no changes should be made, which would risk achieving it. Quality assurance is also a core aspect, which is why Scrum does not allow the decreasing of any quality goals. However, in the case that a sprint goal is not achievable or becomes unnecessary, the Product Owner has the authority to cancel the sprint.

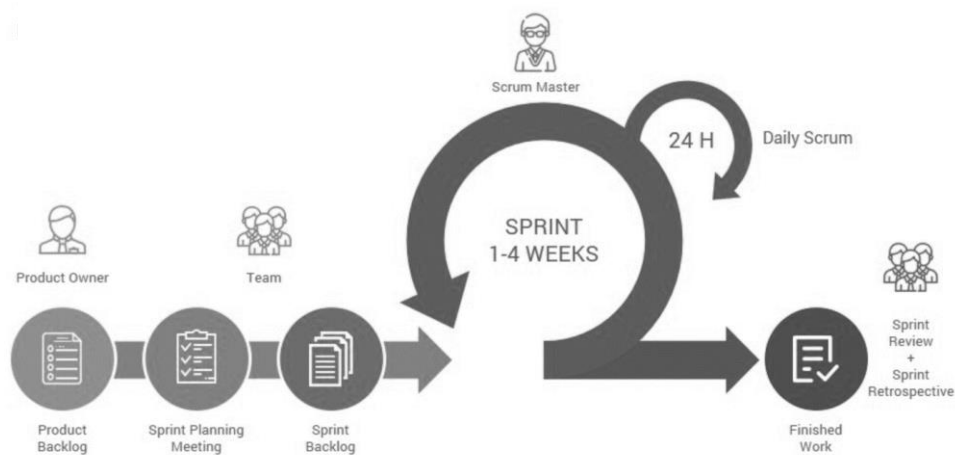


Figure 3. Scrum Process. (Warcholinski, M. Differences Between Lean, Agile and Scrum. Retrieved 1.6.2018).

The team members working in Scrum may possess different skill sets, which are not defined as a standard. They might include programmers, usability experts, marketing people, software architects etc. Basically, the Scrum team should include all those vital skills that are needed to provide the functionality set in each sprint. On top of specialty skills, cross-functionality is often a requirement inside a team. (Crookshanks 2015, Chapter 4.) Depending on the organization and

number of Scrum teams, the same teams may operate for long periods of time without shuffling their team members. This ultimately depends on how experimental the organizations wish to be.

The Scrum teams work in a self-directed manner with independent decision-making being encouraged. Other main characteristics of the Scrum framework include observing and experimenting rather than carefully planning projects in advance and relying greatly on collaboration and interaction. As already discussed, prioritization is a key aspect as the highest possible value needs to be created for customers as quickly as possible. (Canty 2016, Chapter 3.)

2.3.2 Lean Software Development

Initially created for automotive industry needs in Japan in the 1950's, Lean was a movement aiming at reducing losses as well as providing a more sustainable way of production. It was utilized in software development around the year 2000 and upon discovering its' benefits in multiple industries, the startup industry began to apply Lean in 2008. At that time, Eric Reis developed the five Lean principles, stating that the method was a way to develop "new products and services in circumstances of extreme uncertainty".

Lean is rather a philosophy focused on smart development, which means improving performance by eliminating any activities that do not add value for the end user. To work "smart" according to Lean principles refers to working in a disciplined, focused manner while relying also on decision-making based on common sense. It regards everyone involved, one way or another, as a stakeholder and invests in achieving long-term results. The Lean ideology consists of a cycle of learn, measure and build. Typical lean companies will conduct a lot of testing and tend to work closely with their customers. Unlike Scrum, Lean prefers planning designs up-front. This includes bringing a team together at the very beginning of the project to introduce everyone to designing the software despite of what roles the team members have. (Lean Architecture: For Agile Software Development, p.2)

Agile and Lean relate with one another as they both strive for achieving short-term and long-term goals and providing clients with competitive, high-quality products. While the Agile process aims at flexibility, the Lean process holds sustainability in the highest regard. In terms of implementing

different tools, Lean uses for example hypotheses, customer interviews and analyzing customer success, while Agile involves sprints, boards and user stories (Nedre, Natalie. Retrieved 1.9.2017.) While sharing some similarities, some specialists argue that the two are rather philosophies or mindsets, which merely implement different methodologies and tools. The agile mindset is a topic of discussion in chapter 3.

Lean	Agile
Thinking and doing	Doing
Inspect-plan-do	Do-inspect-adapt
Feed-forward and feedback (design for change and respond to change)	Feedback (react to change)
High throughput	Low latency
Planning and responding	Reacting
Focus on Process	Focus on People
Teams (working as a unit)	Individuals (and interactions)
Complicated systems	Complex systems
Embrace standards	Inspect and adapt
Rework in design adds value, in making is waste	Minimize up-front work of any kind and rework code to get quality
Bring decisions forward (Decision Structure Matrices)	Defer decisions (to the last responsible moment)

Table 1. Comparison between Lean and Agile. (Coplien J; Bjørnvig G. Lean Architecture: For Agile Software Development, p12, Retrieved 1.6.2018)

2.3.3 Kanban

Kanban is a production-focused system, which is based on the principles of Lean. The term itself consists of the Japanese words *kan*, which translates as “visual” and *ban* meaning “card”. It was created in the 1940’s when engineers of Toyota were observing supermarket clerks stocking items in the store, noticing how the inventory was refilled based on store supply instead of going to vendors, which meant stocking items only when close to selling out the product. This sparked an idea of coming up with a new system where inventory would meet demand and result in higher quality.

To create the new system, visual management was applied to enhance communication. The production workers at Toyota were using actual cards to demonstrate completed tasks as a supply

of materials or components was needed to continue further in the process. This would allow to keep minimal inventory and effectively showcase line production problems as they occurred. By limiting the number of visual cards, overfeeding the system could be avoided. (Leopold, K; Kaltenecker, S. p.12).

Later, the software industry began to implement this “just in time” -process and currently kanban remains a popular framework in agile software development. The ideology involves around the following principles:

- Visualizing the work
- Limiting the work in progress
- Managing flow
- Making policies explicit
- Implementing feedback mechanisms
- Striving for continuous improvement through collaboration

(Planview LeanKit. What is Kanban? Retrieved 25.5.2018). For these purposes, development teams use something called a kanban board. The boards may be physical but rather often virtual boards are regarded as more convenient in terms of collaboration and accessibility. The workflow of a typical kanban board categorizes tasks as phases, describing the task which remain, those that are in progress and those that are completed. A team may edit the workflow according to their specific needs. Visualizing the work tasks of the team by using a kanban board is done to allow complete transparency and to explore capacity needs in real-time. (Atlassian. What is kanban?). The kanban cards showcase details about the work items, what is their current state and which work responsibilities belong to which team member. This is regarded as a good practice for increasing of focus, keeping track of the progress as well as identifying which factors may hinder a project. Direct communication between team members can solve issues of delay.

Limiting the work in progress in Kanban is referred to as a “pull” system, where a completed task pulls the next item of the product backlog. As the number of unfinished products or products features will increase their delivery time, it is necessary to oppose limitations to the number of active operations in the system. In Kanban, this means limiting the number of visual cards (tasks), having team members focus on the tasks assigned to them and not picking up any work assigned to others if their individual task is not completed. An unfinished task can slow down the project by blocking

the workflow and all team members will be able to see where the block occurs. Kanban encourages team members to commit to achieving their task without risking quality as this could reflect poorly to maintaining customer relationships when customers do not get what they are expecting. Developers may sometimes feel tempted to mark the tasks as completed when they are running out of time, despite not reaching the desired level of quality. This can happen especially when many operations are running at the same time. It is not considered as sustainable and therefore it is advised to resist owning too many tasks. Completing tasks or iterations does not follow a strict time-frame in Kanban. As a result, it has been criticized for not being agile enough. (Highsmith, J. 2016. p.197-198).

Kanban, like Scrum, encourages self-organized groups by having them establish necessary policies themselves. These policies should be followed and made visible for everyone, while allowing changes if they were to become redundant, for supporting continuous improvement. If a policy should not be accepted or followed, it would be fitting to raise objective discussion about the policy itself rather than point the blame on individuals. As for constantly learning and improving, feedback sessions by having daily stand-up meetings is encouraged. Other specialty meetings and reviews can be set up to gather high-quality feedback, ideally having a wide range of participants involved.

The differing needs of organizations is recognized in the core practices of Kanban and therefore it will not give a direct answer as to which methods should be implemented and how. Organizations may adjust the tools as they see fit, since kanban will merely encourage to reviewing the existing processes and experiment to see if improvements are needed (Leopold, K. 2015. p.18-23.)

2.5 DevOps

DevOps was created upon discovering the major incoherency issues that were facing the IT and software industry between 2007 and 2008, when the waterfall-method was still largely implemented. At the time, the traditional software development methods received criticism for not including co-operation between relevant parties, which included entirely disjoint operations between coding and those deploying and maintaining the code. This issue extended to development and operations being isolated from another by having completely different views, leaders, objectives and not even physically working under the same roof. This reflected negatively

on what was happening inside the teams as well as to the customers. As members of the IT community got together to discuss the huge communication issues, a new concept was born. (DevOps: Breaking the Development-Operations barrier. Retrieved 30.5.2018).

DevOps is not a specific method but rather an umbrella term, which covers all the activities related to development and service production. As put in the name, it combines development and operations with an attempt to join them together seamlessly. It is a procedure, which emphasizes speed and reliability when building, testing and releasing software and aims to automatize the process using a set of practices. Continuous integration and delivery are at the core of DevOps principles. (DevOps - jatkuvan kehittämisen tukena, 2017. Retrieved 30.5.2018)

DevOps, similarly to Agile and Lean, is a philosophy. It aims to building a culture and a mindset that reinforces and encourages collaboration between teams. This would be beneficial for establishing trust, according to the DevOps philosophy. Other benefits are being able to release software more frequently, anticipating and managing tasks and fixing high-priority issues efficiently. DevOps shares many similar values, already familiar from the agile principles and practices, such as:

- Team collaboration
- Fast releases with attention to high quality
- Transparency and efficient communication
- Prioritizing and managing the work

While DevOps implements agile methods, ultimately it aims to combine Agile together with continuous delivery and process automatization (DevOps: Breaking the Development-Operations barrier. Retrieved 30.5.2018.)

3 THE AGILE TRANSFORMATION

So far, this thesis has introduced agile as a faster, more adaptable way to creating software in a collaborative environment. It is fact that agile methodologies have been widely adopted when managing software projects. However, the core idea of Agile may be included into many business environments, alongside but not limited to software. This chapter will expand the ideology by identifying some of the characteristics, which are essential for an organization to call itself agile. The related concepts will include assessing the changing role and significance of management as well as introducing the Agile Performance Model -framework (APM). Finally, a case company example will aim to provide a practical example regarding agile transformation.

3.1 What makes an organization agile?

Before moving further with the concepts of agile change, it is interesting to consider what kind of agile definitions there exist. As discussed earlier, agile refers to adaptability, flexibility and delivering solutions at speed. The Agile Manifesto defined the core values and supporting principles, which act as a guideline to introduce what agile is fundamentally about. This provides a good framework for assessing how agile an organization is. However, to study how the agile values and principles manifest inside organizations, it is necessary to examine which other agile definitions there exists.

Despite a software development related or a process-oriented perspective, Agile can be explored a mindset – a way of thinking. Having an agile mindset involves absorbing agile into one's identity to the extent that becomes the new norm. While an organization may implement different tools, practices and support various agile principles and values, the agile mindset is seen as sitting on top of everything while wrapping everything together (Measey, P; Radtac. 2015. p.11.)

Consequently, for Agile to find success within an organization, it can often be a question of adopting the mindset. For example, when a new framework is introduced, individuals may begin to implement it, however if not understanding why it is being used, the temptation of gradually going back to old habits can be high. To look at the issue more practically, it is worthwhile to consider

how the agile mindset compares with a “fixed” mindset, which refers to the non-Agile way of thinking.

	Fixed mindset	Agile mindset
Ability	Static, like height	Can grow, like muscle
Goal	To look good	To learn
Challenge	Avoid	Embrace
Failure	Defines identity	Provides information
Effort	For those with no talent	Path to mastery
Reaction to challenge	Helplessness	Resilience

Table 2. Fixed and Agile mindset. (Agile Foundations: Principles, practices and frameworks. p.12)

The differences between the mindsets are evident from the examples mentioned in the table. The agile mindset is about evolving continuously rather than remaining at a certain level, welcoming and overcoming challenges instead of backing away from them and taking failure as a chance to learn. Where a fixed mindset sees threats, an agile mindset may see opportunities. The goal of continuous improvement lies at the heart of the agile ideology. Agile organizations do not tend to punish employees for their mistakes. This is due to accepting the idea that to be constantly able to improve can involve things occasionally going wrong. This applies particularly to software design as no system is without flaw, but expectancy to having flaws, will encourage putting in place necessary practices to monitor and respond to vulnerabilities. In terms of project management, agile allows to experiment, then analyze whether experiments are bringing value and abandon them if that is not the case.

“Business agility” is one term, created to describe the adaptability of businesses to an ever-changing environment. Organizations that have the capabilities to act and adapt when facing changes operate under an agile mindset. These organizations welcome new ideas and support flexibility in their processes and systems. Openness and adaptability are also characteristics of their corporate culture. Simon Sinek (2011) has argued that the values of the company lie at the very core of the agile business -principle. According to his theory, organizations should have a clear idea about the reasoning behind their existence before focusing on the practicalities of their operations. This rationale along with the values of the organizations should thus be the driver for their decision-making and operations. Another argument by Sinek emphasizes the importance of

delivering products and services, which respond to customer needs. An agile organization should always place the customer in the center of what they do. Many organizations that have first adopted agile software development methods, are now considering how to introduce agility into their business operations.

To discuss briefly about agile and business, a set of core business objectives identify, which are the most relevant when discussing agile projects. Agile stresses the following five as the most meaningful: *continuous innovation*, *product adaptability*, *improved time-to-market* (including return on investment), *people and process adaptability* and *reliable results*. An agile mindset may connect with innovations, since the self-organizing nature of agile enables to set up an environment to innovate new ideas. Agile delivery of products requires adaptability as it strives for technical excellence, using customer value and adaptation as ways of measurement. (Highsmith, J.2016. p.10-11.)

As discussed in the first chapter, agile development involves prioritizing product features and delivering them in small, frequent increments. This will push the teams to consider the number of features, which should be included in the releases and eliminating less valuable requirements. Concentrating on value-adding activities and including the necessary skills to complete a project, would result in improving the time-to-market in agile. The people and processes need to adapt, similarly to products, to create value for customers. Processes in agile is a topic, which have been under debate. Many organizations tend to include repeatable processes into their development. This may respond well to situations where expectancy to change is low. As agile expects changes to happen any given moment, it may prefer reliability for processes. Reliable processes operate under certain boundaries while aiming to meet deadlines and expecting changes to occur. (Highsmith, J. 2016. p.11-12.)

A very recent article has identified five trademarks, which an agile organization possesses in terms of strategy, structure, process, people and technology. Without going to too many details, the organizations were discovered to have an overall purpose and ambition by which it navigates (The North Star), a network of empowered teams, supporting of rapid decisions and learning in their processes, having dynamic, passionate people including a cohesive community as well as highly advanced technology. (Aghina W.; De Smet A.; Lackey G; The five trademarks of agile organizations. 2018. Retrieved 3.6.2018.)

	Trademark		Organizational-agility practices ¹
Strategy	North Star embodied across the organization		<ul style="list-style-type: none"> • Shared purpose and vision • Sensing and seizing opportunities • Flexible resource allocation • Actionable strategic guidance
Structure	Network of empowered teams		<ul style="list-style-type: none"> • Clear, flat structure • Clear accountable roles • Hands-on governance • Robust communities of practice • Active partnerships and ecosystem • Open physical and virtual environment • Fit-for-purpose accountable cells
Process	Rapid decision and learning cycles		<ul style="list-style-type: none"> • Rapid iteration and experimentation • Standardized ways of working • Performance orientation • Information transparency • Continuous learning • Action-oriented decision making
People	Dynamic people model that ignites passion		<ul style="list-style-type: none"> • Cohesive community • Shared and servant leadership • Entrepreneurial drive • Role mobility
Technology	Next-generation enabling technology		<ul style="list-style-type: none"> • Evolving technology architecture, systems, and tools • Next-generation technology development and delivery practices

¹The 5 trademarks include 23 practices for organizational agility; 18 are based on survey research. Five additional practices are included that have emerged from recent experiences with large global companies transforming into agile organizations.

Figure 4. Five trademarks of agile organizations. (Aghina W.; De Smet A.; Lackey G; The five trademarks of agile organizations. 2018. Retrieved 3.6.2018)

3.2 Agile culture change

It can be said that an organization is defined and shaped by its' culture, which can manifest itself in many aspects – work roles, processes, frameworks, tools etc. While being visible in many day-to-day practices, ultimately culture will always come down to people and interaction. Understanding what kind of a business culture dominates a business is seen as vital before implementing Agile, however it can be quite challenging to identify and visualize the subtle elements that affect how people interact. (Measey, P; Radtac. 2015. p.29)

There are many ways in which culture effects the operations of an organization. It can include the following basic characteristics: mission and direction, adaptability and flexibility, involving and engaging the people and creating consistency from core values. Culture is a complex entity, which

includes internal factors, such as the core values and capabilities as well as external factors like strategy. (Denison D; Hooijberg J; Leif C.; Lane, N.; & Lief, C. p. 7-8.) The issue of corporate culture may be difficult to define and describe, despite of the fact that it is present everywhere in the workplace. That is because corporate culture includes certain characteristics, which make it difficult for individuals to give precise descriptions about it.

According to Edgar. H. Schein, titled as a “leadership guru”, culture is deep in the sense that it is very challenging to manipulate it. It is also broad, as instead of having people controlling culture, usually it is the case that culture ends up controlling the people. Culture tends to remain relatively stable due to people naturally tending to prefer predictability. (Kanban 2015. p.136). Furthermore, culture includes a large variety of influential environmental factors, for example the market situation, social change or political climate. It exists through having a context and that context is much wider than people usually realize.

The Schneider culture change model offers one definition, which identifies four types of cultures that define factors of an organization aiming for success. The four cultures include the following: collaboration, control, cultivation and competence. Collaboration emphasizes succeeding through working together in teams, valuing matters such as building trust and encouraging diversity. Control refers to stability, power and being able to draw predictions and implement clear processes. Cultivation regards learning and growing while having a sense of meaningfulness. Competence values the aspiration to be the best. The four cultures are further divided into x and y-axis according to their orientation. The x-axis is divided into people or company oriented while y is either reality or possibility oriented.

The model enables to study what kind of characteristics organizations may have and where their biggest values lie. The agile culture has been the subject of inspection through using this model for research about culture. Results from a survey (Spayd, M. 2011) suggest that agile culture strongly invests in collaboration and cultivation and is therefore highly people oriented. In addition, agile tends to steer away from control, instead focusing on creating value with learning through co-operating. (Measey, P; Radtac. Agile Foundations: Principles, practices and frameworks, 2015. P.30-31). Leadership may have a pivotal role in setting the tone for this type of a change in culture. That aspect will be examined next.

3.3 Management and Leadership in Agile

Management in Agile is a very broad subject, which involves the organizational culture, tools and frameworks, project and product management and leadership. This thesis will merely consider a few, relevant concepts. As mentioned, inspection and adaption are some of the core agile principles. Moving away from top-to-down management and into empowering teams is characteristic to forming an agile style of managing projects and people. Motivating employees and providing them an environment built on support and trust is a fundamental part of the agile principles. Therefore, it is relevant to study the issue of motivation and attitude between management and workers.

The Human Side of Enterprise is a publication created in 1960 by Douglas McGregor, which has later been revised (McGregor, Gershenfield, 2006) and introduces a model to study how managers interact with their employees. The model includes two different management perspectives, divided into Theory X and Theory Y. The Theory X is an example of a control oriented -view, which was included in the Schneider model of culture change. The Theory X involves a set of beliefs that assume employees needing strict supervision due to their lazy tendencies, lack of ambition and a tendency to avoid responsibility unless encouraged with centralized incentives. It regards that employees should be controlled as their individual goals would not be meeting the needs of the organization. Quite on the contrary, Theory Y will assume that employees are able to motivate themselves, be responsible for their learning and having a positive reaction to allowing them the freedom to exercise their talents in the workplace. Employees would not need to operate in a “stick and carrot” manner like in Theory X, as their goals would align with the organization through commitment (Measey, P. p.94-95.)

Theory X managers believe that employees...	Theory Y managers believe that, given the right conditions, employees...
Hate work	Like and need work
Seek money and security	Seek to be involved and realise their potential
Have to be forced to work	Drive themselves and work effectively
Prefer to be told what to do	Take initiative
Are rarely creative	Are naturally highly creative
Are selfish	Commit themselves to larger goals

Table 3. Theory X and Theory Y. (McGregor, 1960).

The theory regards the attitude of the management as key to how employees will act and deliver their work. It assumes that when management has a predetermined attitude and imposes it in the workplace, the employees will end up acting exactly the way the managers presume. This means that in Theory X, the employees would expect to be told what to do, having a negative mindset and regarding the work as merely a source of income and not as a mean to express their creative needs. This is due to management having such a key role in setting the underlying cultures and atmosphere at the workplace. As such, the managers' input will typically manifest in what the workers output. Organizations following a model closer to Theory Y are being more productive, according to studies (Measey, P. 2015. p.95). This would be particularly true for agile organizations, as already discussed in the previous section. The Theory Y aligns more conveniently with Agile, which values motivated individuals before processes and expects that teams consisting of individuals will be able to self-organize.

When discussing agile management, the teams and their level of collaboration is an essential part for consideration. The issue is thus about team dynamics. For teams to function well together, certain functions either help or prevent teams from achieving a high level of performance. Patrick Lencioni (2002) has made such a list of characteristics, which includes the following: trust, conflict, commitment, accountability and attention to results. According to the theory, these can be both positive and negative. A team can be regarded dysfunctional if it is not invested in achieving results, avoids being accountable, is fearful about conflicts and lacks commitment and trust.

Accountability refers to owning responsibility so that other team members as well as management can expect the individual to complete their tasks without too much involvement. With attention to results, the teams can practice a form of shared accountability. When speaking of conflict, it is generally associated as a negative term. This is not always the case where teamwork is concerned. When members of a team wish to keep their opinions to themselves due to fear of getting into a debate, it can result in the team operating from a very narrow point of view. It would be better to raise discussion even when having conflicting views to avoid falling into a trap of “group thinking”, which may not create very innovative ideas, for example.

Openness towards failure and shortcomings is one relevant point when discussing the building of trust. As already acknowledged in this thesis, agile leaders will allow employees to make mistakes without the need to punish or be very critical or harsh. They too, should acknowledge not being perfect and realize how much they need to develop personally, just like any other worker. It can be a good idea to communicate this to the teams as well. Leaders can learn a great deal by observing what is happening around them and in other organizations and communities. Where commitment goes, agile leadership will aim to define and effectively communicate to the team about goals, making sure that the team is heading towards them together.

Gathering the different functions into a form of a pyramid demonstrates a value structure with trust forming the base, upon where everything else is constructed. When colleagues trust each other, they are more open to share ideas and opinions even when it may result in debate, which may have a very fruitful outcome. When a team is committed, it is more eager to take initiative and share accountability, which will keep the team focused on achieving results. The management’s role then becomes more about enabling and encouraging all these, fundamental aspects. (Measey, P. 2015. p. 100-101.)



Figure 5. A functional team according to principles by Lencioni. (Measey, P. Agile Foundations: Principles, practices and frameworks, p. 101.)

Going back to the core agile principles, a publication titled Agile Project Management, written by Jim Highsmith offers a simple yet effective quote about the difference between how a traditional manager plans projects compared with an agile manager. It goes: “A traditional project manager focuses on following the plan with minimal changes whereas an agile leader focuses on adapting successfully to inevitable changes”. This stresses the issue that almost every project requires at least some amount of planning, but the differentiating issue lies in the perception of the plan and the expected outcome. Highsmith also points out three main values, which an agile leader should have: delivering value over constraints, leading the team over focusing on tasks and adapting to change over complying on plans. These values are familiar from The Agile Manifesto and are good indicators for examining how agile the style of leading is. (Highsmith, J. 2016. p.17)

As emerging trends have been transforming and continue to transform the way organizations operate and act, the organizations of today are almost like “living organisms”, in need of some stability while being able to function dynamically. (Aghina W.; De Smet A.; Lackey G; 2018. Retrieved 3.6.2018) In an agile organization, which is marked by less bureaucracy than before and an effort to act in a quick and flexible manner, the leader is an enabler, with a clear long-term goal to lead the direction. The theory about organizations being living organisms is demonstrated in the illustration below.

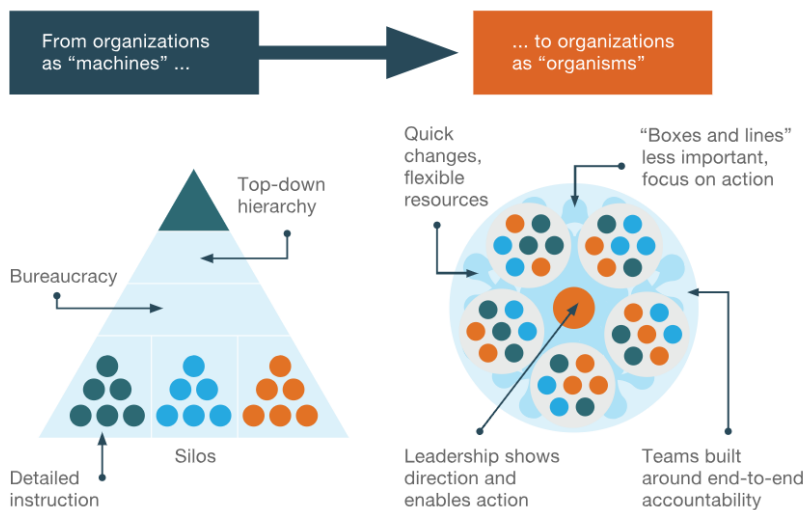


Figure 6. Agile organization as a living organism. (Aghina W.; De Smet A.; Lackey G; *The five trademarks of agile organizations*. 2018. Retrieved 3.6.2018)

3.3.1 Change management

During the latest few decades, organizations have been battling with a variety of significant changes, which touch on many different levels such as economic, demographic, political, financial, collaborative as well as individual. The environment where products and services are developed has become increasingly competitive and the results may have taken a big toll on morale, for example through reducing staff due to outsourcing. Rapid advancements in technology have transformed the way organizations run their operations, aiming to produce feasible outcomes in an environment marked by a great deal of uncertainty. Many companies have failed to react to the changing requirements promptly enough and have died out as a result, while others have blossomed being able to grow even stronger with innovations, a focus and strategy, which supports continuous improvement.

For an organization to stay competitive in times of extreme uncertainty, it needs to be able to adjust to a changing environment, which may often involve putting forward the change forward from within the organization. Sometimes a matter of choice, while other times being more of a matter of force. Managing change includes many challenges, it requires short-term as well as long-term planning and particularly in an agile environment, the objectives may include a great deal of inconsistency. This is because the future has become increasingly difficult to project, but projections are necessary

nonetheless, to form sense of direction. The complexity of the issue becomes clear in the table below, referred to as a “Dance of Change”. (Leopold, K; Kaltenecker, S. 2015, p. 95).

Long-term planning	
Changes require time	Short-term results
Strategic parameters	Pressure to make changes quickly
Focusing on clear goals	Openness to unplanned processes
Parameters of project management	Focusing on flexibility
Step-by-step improvement	Creative room for action
Customer or market oriented	Radical innovation
Managing risks	Employee oriented
Precise problem analysis	Capitalizing on new opportunities
	Quick fixes

Table 4. Priorities in Change Management. Source: Leopold, K; Kaltenecker, S. 2015. Kanban Change Leadership: Creating a Culture of Continuous Improvement p.95)

Examining the table, the different considerations for long-term planning seem to include elements known from both Waterfall-method and Agile. Scrum for example, prefers to develop iteratively in short-cycles, highlighting how important it is to be fast and innovative, which sometimes allows leaving decision-making at the last minute. However, even in Scrum projects and particularly dealing with complex issues in large organizations, long-term goals and broad guidelines are put to place, guiding the teams to aim towards a unified vision that is managed from above. Agile is not as chaotic as it is sometimes interpreted to be. Keeping in mind that Agile tends to rely on technical excellence, this means that a level of precision and risk management must be involved. Not to mention, organizations must always bind to certain laws so there are always limitations to how they can operate and act.

Dr. Paul Evans (2000) has studied the paradoxical nature of the requirements for planning projects, extending on the ideology by creating the “The 11 Paradoxes of Leadership”. It introduces the following traits, seen in the figure below, as suitable for change leaders and managers of today. IT may act as a “checklist”, for examining the considerations how an agile leader should behave, recognizing the conflicting nature of the values. (Measey, P. 2015. p. 115).

1. To be able to build a close relationship with one's staff, and to keep a suitable distance.
2. To be able to lead, and to hold oneself in the background.
3. To trust one's staff, and to keep an eye on what is happening.
4. To be tolerant, and to know how you want things to function.
5. To keep the goals of one's department in mind, and at the same time to be loyal to the whole firm.
6. To do a good job of planning your own time, and to be flexible with your schedule.
7. To freely express your view, and to be diplomatic.
8. To be a visionary, and to keep one's feet on the ground.
9. To try to win consensus, and to be able to cut through.
10. To be dynamic, and to be reflective.
11. To be sure of yourself, and to be humble.

Figure 7. The 11 Paradoxes of Leadership. Source: Measey, P. Agile Foundations: Principles, practices and frameworks. 2015. p. 115).

Regarding change, the team and its' leader can be imagined sailing in a boat with changing weather conditions surrounding them. The team nor the leader can control the weather, but a boat typically has someone directing it, even when facing a storm or other unexpected event. It could be said that this is when the one in charge of directing the boat can become particularly focused and invested on the job, having to make decisions about how to approach a challenging situation. This describes the characteristics of change management. The more uncertainty there exists in the real world in terms of tech developments and trends, market fluctuations or growing demands, the more need it creates for managing change.

Change always involves culture. It is complex to manage because culture itself includes levels of complexity. Returning to the model of the four major aspects of culture: adaptability, mission, involvement and consistency, an organization would benefit from reflecting on how it is addressing these issues. A mission will typically be focused on the long-term, setting the direction where the organization is headed. (Denison D. 2012. p.7.) To demonstrate the complexity of this theory, the following figure will highlight the issue.

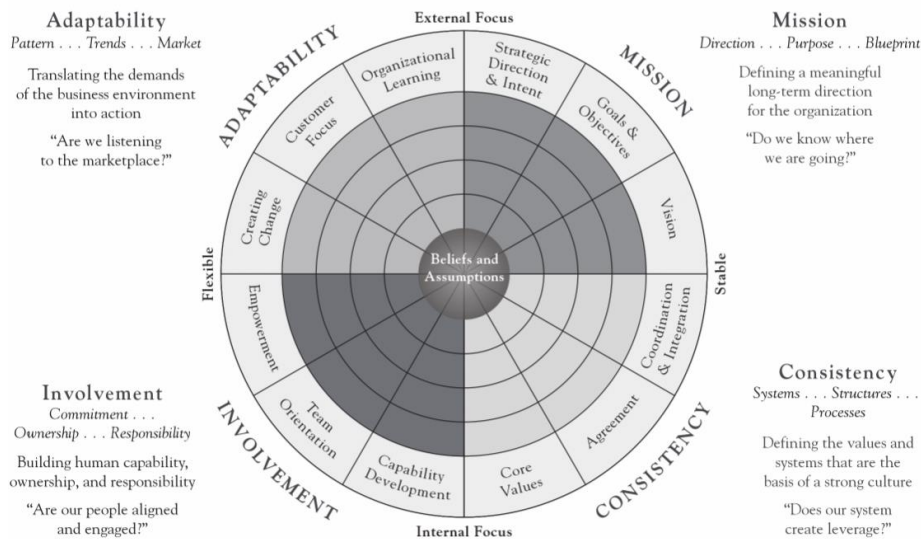


Figure 8. Organizational culture and business performance. Source: Denison D. 2012. *Leading Culture Change in Global Organizations: Aligning Culture and Strategy*. p.8.

Adaptability may refer to a variety of things, which relate to the environment where the organizations operate. Involvement refers to commitment and responsibility, which have already been discussed. It has been established that agile organizations require a level of consistency, despite of seeking for high adaptation and being on stand-by for unexpected changes. For the agile leaders, it is relevant to study the aspects of how adaptable the organization with regards to the changing environment. This includes operating under a meaningful long-term mission, which is guided by vision and reflected in the goals and objectives of the organization. It would also be worth considering how engaged and capable the people are and if the systems and processes support the organizations' culture. Agile will encourage organizations to awareness by constant reflection on these issues, as that it how an organization can achieve continuous improvement. The agile leaders have an important and challenging role in practicing and sharing of awareness.

3.4 APM Framework and Agile Delivery

As mentioned in this thesis, Agile has some constraints despite its' flexible and adaptable nature. While the planning of agile projects may often include having to deal contrasting values, agile project planning includes some of the same objectives as traditional projects. Measuring agile

performance is regarded as necessary to create a correlation between what the self-organizing teams are aiming to achieve and what the managers regard as a successful outcome. All projects tend to have some constraints, typical examples about these are: requirements, time and cost. To explore the issue of constraints in agile projects, something called “the Iron Triangle” is used to demonstrate the differences between agile-, and traditional projects. Technical quality has also been added as new addition to the figure by the authors. (Measey, P. 2015. p.18.)

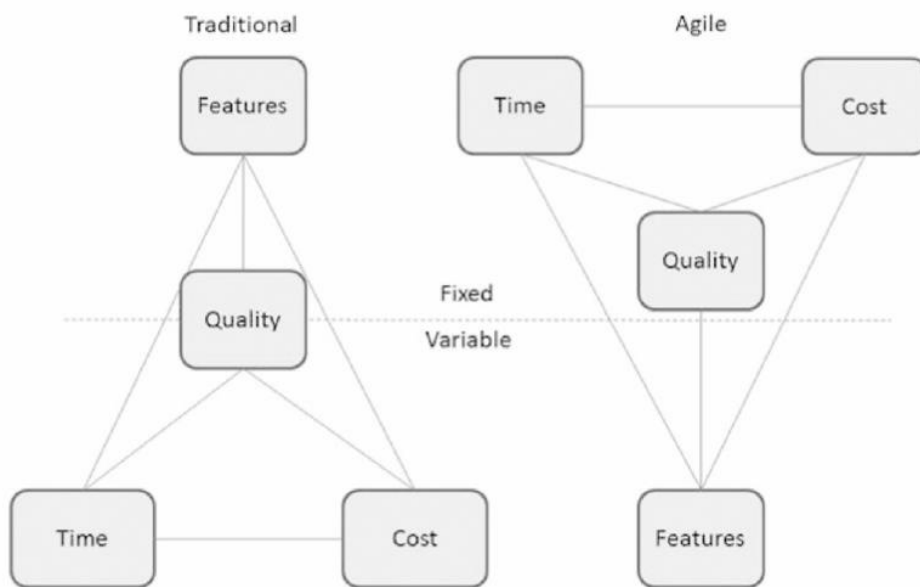


Figure 9. The Iron Triangle. Source: (Measey, P; Agile Foundations: Principles, practices and frameworks, 2015. p.18.)

The traditional model is consistent with the Waterfall method. As introduced in the first chapter, projects following a Waterfall approach include defining the features and the project in advance and assuming, that the time and cost may change, while the requirements stay the same. In the Agile model the pyramid has been turned upside down, having the expectancy that the requirements will change while cost and quality can be relatively fixed. The goal of managing the time constraints into short sprints is referred to as “time-boxing”. Depending on the product and project, time-boxes can vary from days to weeks, and sometimes months while tending to prefer short cycles. Before a project begins, a high-level design or a prototype is introduced. Regarding the designs and prototypes, Agile prefers simplicity as requirements may change. For drawing estimations about the project constraints, Agile prefers that the team, together with customers and other stakeholders, collaborate and experiment with products to make effective decisions about requirements and choice of technologies. (Measey, P; 2015. p.20-21.)

The Agile Project Management (APM) Framework is a model, originally introduced by Jim Highsmith, in the publication “Agile project management – creating innovative products”. It describes the lifecycle of a project as consisting of the five following steps:

1. *Envision* – determining the vision and objectives of the project
2. *Speculate* – creating a capability or feature-based plan
3. *Explore* – planning and delivering of tested stories
4. *Adapt* – reviewing the results and team performance
5. *Close* – concluding of the project

In the envision phase, the teams will figure out what will be delivered, who will be the people involved and how the teams plan to work together for achieving the vision. The success of the project relies greatly on this first stage where a vision. Highsmith regards this speculate phase as “to conjecture something based on incomplete facts or information”, which is in fact how the dictionary defines the word speculate. This is to address the issue of having unknown factors involved, which replaces planning as more of gathering a collection of assumptions. The speculate phase will include having a wide set of product requirements, which are put in a product backlog, having a release plan based on the requirements and estimating potential risks as well project costs (Highsmith, J. 2016. p.84.)

The explore phase includes user or product stories. A user story is known from engineering but can typically be created by product managers in Agile projects. They can follow a structure of: as a <type of user>, I want <some goal> so that <some reason>. (Mountain Goat Software, User Stories. Retrieved 1.6.2018). The teams must decide how many stories are possible to deliver in the iterations or sprints for which they will seek reference from 1-5 earlier sprints. The term for this sort of retrospection is called “velocity”. (Measey, P. 2015. p.64). The explore phase also has the project leaders form a collaborative, self-organizing community. How the customers, product managers and stakeholders interact, is also managed in this phase. (Highsmith, J. 2016. p.85.)

The issue of adapting has been discussed in many instances in this thesis. It is mentioned in the Agile Manifesto that “responding to change is more important than following a plan”. In the adapting phase, a plan can be revised according to feedback from customers, tech people or as a result

from process performance -evaluation. The project or iteration is then ended with a goal to having learned from implementing the previous steps. The APM-framework is not expecting to complete these steps continuously in this exact order, the loop of speculate – explore – adapt can be repeated until enough data is gathered to form a good view of the final product. (Highsmith, J. 2016. p.85.)

The APM Framework is a model for agile delivery. Large organizations or enterprises may have hundreds or more projects, which can use a mixture of agile and traditional practices. The transformation of a large enterprise may include using several methods and learning with time about which work best. Highsmith has suggested an Agile Enterprise Framework, which operates on several different layers. They touch on governance, project management, iteration management and technical methodologies. Without going further into this theory, Highsmith has stated (2016. p.81.), that a framework should support and include the following:

6. A culture of envisioning, exploring and adapting
7. Self-organized, self-disciplined teams
8. Reliability
9. Flexibility and easy adoption
10. Visibility
11. Learning
12. Practices for supporting each phase
13. Management review

Another framework for examining aspects of agile delivery is the Cynefin framework (Snowdon and Boone, 2007), which separates environments into domains according to how simple or complex they are. A *simple domain* operates on a cause-and-effect basis, allowing to project the results with relative ease. The teams operating in a simple domain can draw a defined delivery plan up front. Such as situation would be an example of a Waterfall-approach. A *complicated domain* is less predictable, but a defined plan can be used after spending some effort on analysis and accepting that some flaws may be included. This domain works for both Waterfall and Agile as well as Lean. The third domain is called *complex domain*, where cause and effect no longer apply or can be accepted to change rapidly. Up-front planning is not suitable where so many complexities are

involved, therefore the Waterfall method would not work well in this domain, while it would be ideal for Agile (Measey P. 2015. p.15.)

In the fourth domain, which is *chaotic domain*, cause and effect have no place, which makes planning obsolete. Teams working in a chaotic domain will rather conduct experiments and try to get into another, more manageable domain. Kanban can be an option for a chaotic domain. It may also be an option for innovative brain storming -sessions. The final domain is called *disorder* and it does not have a definition. This would mean that team members would use a working style that comes naturally for them but may not meet the needs of the project (Measey, P. 2015. p.15.)

3.5 Case: Spotify's Agile Model

This example is used to introduce and consider agile transformation through a real-life company case. The information provided in this section is largely based on video material: "Spotify Engineering Culture part 1", which is openly available online. It describes the company's culture of engineering as a continuous agile journey, which is constantly being refined.

Spotify is an established entertainment company, which provides music, podcast and video streaming content. It operates on a freemium basis, offering basic features free of charge with advertisement while a subscription payment enables users to download, and stream content with a higher quality (Harris, M. 2016. Retrived 30.5.2018.) It is the world's biggest streaming company with 35 million songs uploaded to the service and having about 170 million monthly active users, of which 75 million were paying for its' premium, ad-free subscription in 2018. Users can access Spotify using their computers, smartphones and tablets. Browsing or searching for content is enabled by using parameters such as artist, album, genre, playlist or record label. Spotify allows users to create, edit and share content on social media as well as making playlists with other users. Spotify has grown rapidly into a clear market leader in the music streaming sector. Some investors estimate the company to reach a value of 50 billion in a few years (Sassard S; Soderpalm H; Swahnberg O; Reuters. 2017. Retrieved 30.5.2018).

Spotify has an interesting history having moved from a start-up to a global enterprise with users currently in 61 countries. The company hires 180 teams and 1800 people in the field of engineering and R&D. In total, the company employs 3500 people. Spotify adopted Agile in 2008 as they first

started to implement Scrum. With the company growing rapidly, the Scrum teams were soon multiplied. It was discovered at the time that Scrum practices, such as sprint planning meetings and breaking down tasks, were no longer working efficiently. This resulted in a change of culture where it was encouraged to break rules when needed, as agile values would matter more than Scrum itself. Upon reinventing itself, the company changed the role of Scrum master to act as an Agile Coach, which was as a trend at the time agile emerged. The new role of a manager was regarded more to being “a servant leader rather than a process master”.

Instead of having Scrum teams, the company organized development teams into autonomous “squads”. These squads were small, cross-functional, self-organizing team of less than 8 people. The teams would conduct end-to-end development, in charge of designing, committing, building, deploying and maintaining operations. Autonomy meant that the squad would decide what to build and how, as well as learning how to work together while doing it. However, the teams would have some boundaries, such as strategy and short-term goals to be negotiated every quarter. Each squad would also have a long-term mission. The physical office space at Spotify was optimized for collaboration, with members working closely together, having walls acting as whiteboards and including a common area for retrospect sessions. Autonomy was regarded as an important value as it would keep team members motivated and allow them faster decision-making. In accordance to agile values, Spotify wanted to minimize hand-offs and unnecessary waiting, for purposes of efficient scaling. The different squads would be tightly aligned by product strategy, company priorities and focusing on overall mission over individual squads. As a quote from Spotify says: "Be autonomous but do not sub-optimize".

Spotify regards itself as an organization where high alignment would mix with high autonomy. This includes a culture where management figures out which problems to solve but lets the team members do the actual solving, a practice which is very much in accordance with agile principles. As far as development methods go, some may implement Scrums and sprints while others use Kanban. The methods are not standardized, but as certain tools are increasingly adopted, they may spread between teams and become a standard. A balance of delivering consistently while remaining flexible is a main goal at Spotify.

Spotify has over a 100, separate systems, which are coded and deployed independently. While interacting with each other, one system focuses on one specific need, for example play list management and search or monitoring. The systems are small and de-coupled with clear interfaces

and protocols. Each system is owned by one squad while most squads own several systems. Spotify supports an internal open source model, promoting a culture of sharing. If a squad needs help with coding from another squad, they can edit the code themselves while another squad may review it. This is regarded firstly, as saving time since anyone can edit any code, and secondly, providing a culture of peer code review to result in better quality and a focus on knowledge share.

Since Spotify would soon have over 50 squads spreading across different cities, there was a need to develop more structure. As a result, the squads were grouped into tribes. The squads are focused on product delivery and quality while the tribes share knowledge on specific areas of expertise, for example web development or management etc. This enforces an idea of having communities rather than hierarchical structures. As it is believed at Spotify, a strong enough community would be able to operate in a way that is less formal.

The teams deliver small but frequent product releases. There used to be bigger investments having only a few coders but as Spotify grew, it became a problem as dozens of squads had to synchronize with each other for each release and it would take months to get a stable version. To solve the issue, software architecture was changed in such a way that it would enable decoupled releases. This meant that each client platform would form a client app and would be assigned to a specific client app squad. This would allow easy product releases on one specific client platform (desktop, iOS, Android). The squads were also divided into feature squads, which would focus on one feature area, for example a search-feature. Infrastructure squads were formed to make other squads more effective by providing tools and routines such as continuous delivery, monitoring and testing.

For product release and testing, Spotify implements release trains and feature toggles. The release trains mean that each client app has a release on a regular schedule (every week or every 3 weeks depending on the client). When the releases are kept frequent and regular, it means that less up-front planning is needed. Feature toggle is something that is used to hide an unfinished code in the case that it is not completed for release at the same time with the others. This is regarded as a good practice for integration testing, since feature toggles allow to hide or show features for testing and production purposes. This enables to gradually roll out the features as they are finished.

With assigning different types of squads working on different aspects, Spotify is aiming for a self-service model where handoffs can be avoided by squads rather establishing a system based on

enabling and providing of support. Spotify has invested in creating a liberal culture, which is based on strong mutual respect and motivation. It is tremendously focused in gaining a very high worker satisfaction. This reflects on the way Spotify deals with surveys about worker satisfaction. According to one survey conducted at Spotify, worker satisfaction had gained a high result, with 91% stating to be satisfied, however the first response by Spotify was to raise concern about a 4 %, stating to be unhappy according to the survey. From this strong investment in satisfied employees, Spotify has gained a good reputation as a workplace. (Fernandes, T. Spotify Squad Framework – Part 1. 2017. Retrieved 15.5.2018).

The Spotify Agile Model is a very intriguing one. The company provides open information about the agile journey it is taking, with adjusting the process along the way and not being afraid to ditch things that do not seem to work well. Many could presumably want to imitate the model. By continuing to experiment to gain the best results, Spotify seems committed to embracing agile values in its' operations, development methods and corporate culture. It can be stated that the experimental nature of Spotify may not be suitable for every organization. It is also worthwhile to recognize that despite having a staggering amount of monthly active users, Spotify has not yet managed to become profitable.

4 INTERVIEW

For this research, it was the intention of the author to find technology-oriented organizations and find out about their characteristics. For the thesis, it would be viewed as a preference to seek organizations, which involve Agile practices. The research would not specify certain work titles as special points of interest for study as any information would be regarded equally valuable. The interviews are handled similarly to company cases, categorizing them as Company A, Company B and Company C. Each interview describes the experience of one individual. It is thus understood that the results are not eligible for generalization. However, the experience of the individuals is regarded as meaningful in this research.

One of the organizations has moved from Waterfall to Agile, providing a very specific and interesting description about the process. The other organizations had always implemented Agile, according to the interviewees' knowledge. The researcher would like to point out that not having experienced the traditional method of developing software is by no means a significant issue of agile transformation. Agile exists in the practices, frameworks, mindset and culture and those aspects are all part of the transformation. The amount and quality of data gathered from the interviews was satisfactory to the author. This section will gather and introduce the results of the interviews.

4.1 Goal and scope of interview

For this thesis, three large organizations were chosen for an interview. All of them are technology companies with one being a global enterprise. The interview questions were related to agile methods and practices as well as changes, which have occurred in the organizations as well as to the role of the interviewee. The researcher asked the interviewees to provide an agile definition in their own words, to find out if and how the views would differ. Challenges related to organizational change was also asked. Different characteristics in the organizations are listed under separate headings to describe and discuss how agile may present itself in different areas. These characteristics include:

- Collaboration and communication
- Customer and user involvement
- Corporate culture
- Processes or guidelines
- Management and leadership

The interviewees were asked to describe the mentioned characteristics within their organization. These different aspects were chosen due to the research having an aim at understanding the phenomenon of Agile transformation deeply, through inspecting of several aspects. The goal of the interview was to gain in-depth knowledge about the interviewees' own experience on the topic and related issues. The interview method can be described as a general interview as each interview was constructed with the same structure while allowing freedom to present further questions or have a discussion or deepening of knowledge.

Approximately one hour was spent on each interview, of which two were conducted face-to-face and one via phone. The interviewees were all male and under 40 years old. All interviews have been transcribed. Due to the sensitivity of the subject matter, the employees and organizations shall remain anonymous at their own wish. Some general information regarding the company size, as well as the work title of the interviewee is being described.

4.2 Company A

Company A is focused on providing technology solutions related to security. It hires approximately 100 employees locally while includes over 1000 employees in total. The interviewee has a role of software developer / designer in the organization.

The interviewee has a work experience of ten years in company A, working as a developer. Upon reflecting the changes that have happened inside the organization, the interviewee recalls an incident where five years prior, there was a trend of having large investments towards a single product for a single influential client, which proved unproductive. Since that time, the company has focused its' attention to company segment, which is now the biggest focus area. The marketing inside the company has changed, there is now also less video or viral marketing than earlier and large campaigns have been reduced. This measure has been taken largely due to profit not meeting

the investment. A company needs marketing to get clients, but it is better to focus on gaining ROI with more realistic methods, according to the interviewee.

Another big change also traits back five years ago, when a failed product caused organizational change as employees got laid off from their work. This was the cause of a poorly organized project, which included micro-management and developers were not encouraged to raise their opinions on any matters. The effects of the failure remain as long-term employees were forced to leave the organization along with their expertise. The company is still recovering from the failure little by little. Having learned from past mistakes, new products have since been made differently, in the beginning being very agile, considering the needs of end-users and having a strong focus on UI. The biggest challenge in the company has been how to success globally despite being a very well established in their own country. To achieve this, many different experiments have been attempted, such as changing the pricing model, but the issue of how to push for global success has not yet been resolved.

Where development methods go, Agile and Scrum in particular, are being implemented. Different projects or products may use different methods, they may not necessarily need Scrum. Kanban is also being used in Company A, but the interviewee is not experienced with it. Every project has a daily meeting. The projects are being assessed according to probability curves rather than a very specific and accurate method. The development teams have two-week sprints which are properly scaled but are open to possible changes during development. The company used to conduct a specific hour estimation but has overgrown it since it was not efficient to put so much effort into estimating how much time to use on which task. Instead of being “locked” assessments, they are now using more flexible methods. Tasks were also at one point divided according to difficulty assessment, which did not prove to be a good method and was also abandoned later. Scrum has been always used in company A since the interviewee has started working there, but it has been modified along the way according to the organization’s needs.

The role of the interviewee has gone through some changes since starting to work in the company. While always remaining a developer with various responsibilities, the role included more involvement in test automation in the beginning. Lately there have been more changes to the role, as it used to be very individual but has since become more co-operative, not having to own all the responsibility alone. The shift has thus been from independent to more collaborative. The interviewee also mentions having been able to do distant work and communicate with another

branch more before, while now the focus remains on operating in the local area only. “It is a nice change”, comments the interviewee about the issue. When discussing the definition of agile, it is described as “being able to quickly respond to changing requirements”.

Collaboration and communication

On assessing how agile values are demonstrated in the workplace, as already mentioned, collaboration has increased and includes working closely with colleagues and having a daily face-to-face meeting. The meeting involves discussion about what has been done and what should be done next. Online messaging is said to be kept at a minimum. The communication works well while at times, people may spend a lot of time describing what they do, which others may not be able to understand. The point is to keep the discussions short and compact without getting into too many details.

Team sizes depend on the project, current project includes two Scrum teams with a separate client-team and a back-end team. Scrum Master and product manager may complete the same role. As the process is lightweight, a separate Scrum Master may not be needed as people are managing themselves. Product manager always checks what is needed in the next sprint. Allowing this much flexibility works so far, because tasks are getting done without too much issues. Should issues occur more, a need to update the method may rise.

Customer or end user involvement

The focus in Company A has shifted more towards adding of end-user involvement. A few years back, a big event was organized around end-users, flying people from other countries for a “customer day” -event, which was live streamed. It involved mostly open, feedback discussions. The interviewee points out the difficulty of assessing profitability concerning this type of organized sessions.

Previously there has been a “hype” around end-user involvement which has since faded as strategies tend to change. Now that the focus is on the customer companies, frequent discussions occur between them and Company A. “User is always regarded important as agile is about releasing usable products soon and getting feedback from testing”, states the developer. While the users have demands, prioritization is key in Company A. Product adjustments (architecture) are often not very visible and may be done in a long period of time without end-users noticing a difference as it is mostly about fixing minor bugs.

Corporate culture

Regarding mindset, reducing of staff may have taken an effect in Company A, but interviewee sees no big change in corporate culture. It may relate more to consideration about whether employees wish to change company, however employee turnover has not been evident since the latest staff reduction. Own moral is seen as the issue, which has mostly been affected by the reductions. There has not been an issue of people leaving the company due to Agile or Scrum. The team in charge of the project should own the decision-making, if this is not enabled, it may affect the culture negatively.

Management and leadership

Micro-management used to be implemented for a large project in Company A. It caused plenty of issues and has been abandoned since. Top-to-down commands may still occur, while it has mostly proved non-sustainable. Management is described as flexible – if the job gets done, there is no need for pressure from the outside. According to interviewee, the work atmosphere is good at Company A.

When discussing the matter of organizational change, the interviewee mentions staff reductions having been a cause of pressure on an individual level. To overcome the issue, adjusting the attitude and mindset to things out of own control is regarded as helpful. The interviewee continues that doing relevant tasks is key and persistent work effort is needed as not always being able to see immediate ROI and projects may take years before becoming profitable. Due to people being laid off before, developers may be at risk at being resigned and having to take more responsibility than they should own. The developer mentions that working for a project for years may cause worry about how it will end. It has not affected too much directly however, since a developer's focus is always on the next task.

4.3 Company B

Company B is an enterprise software company, which operates globally having customers in over 150 countries. The interviewee is a software architect in the area Research and Development

(R&D) with over 8 years of work experience in the company. During this time, the organization has gone through a big transformation, moving from a Waterfall method into Agile.

When the architect started working in the company, it had separate R&D (Research & Development) and QA (Quality Assurance) departments, which was typical at the time as already discussed earlier in this thesis. Each had their own, separate staff. At the time when Waterfall was implemented, developers focused on validation, having a specific time of release while working with massive content. Problems in QA were not regarded as a huge issue, but massive delays might have been causing problems. Development was done in cycles with new minor or major releases. It was not uncommon to have products re-designed while other releases were simultaneously being pushed out. This made it troublesome to commit fully to the content.

Roughly two years after agile was being introduced, the company started adopting it by organizing the first Scrum teams. Agile coaches were brought in to introduce and educate the new method and courses were arranged as Scrum masters needed special training. Tools brought in, such as TFS (Team Foundation Server), a source code management -tool. The methodology was adjusted to fit the organization's needs. Marketing features and user stories were being added and teams were re-organized. R&D and QA were also combined at the time. Describing the time as a short evolution, the architect recalls the teams and groups as becoming a mixture of people, including those needed to complete the work, such as testers, people involved with automation, database experts etc.

Each team had a Scrum master, who had certain capacity for project management and could have a background in software development. As the second Scrum team was formed, the interviewee was asked to become a Scrum master. More time could be spent on Scrum at the time as it was new. Adopting Scrum acquired a change of mind set and a skill for estimating time use. Some people could not adapt as they did not have the right mindset. This resulted in several individuals leaving the company, finding it too difficult to adjust.

With agile, the Scrum masters needed not interfere too much with the teams' work, while in waterfall they very much involved. When adopting agile, team leaders were assigned and appointed in charge of several Scrum teams. The teams were expected to own responsibility and were empowered to take leadership and handle the content within the team. Many were happy about empowerment and embraced the change with an open mind. The interviewee was not familiar with

product managers prior to Agile, having never even met with them and now needed to co-operate, learning more about process and prioritization. It meant understanding what the teams would need to do and how to achieve it.

Moving forward, the company recognized not wanting Scrum masters to come from inside the Scrum team. Instead, they brought in people internally from product and project management. At this time, the interviewee switched to other tasks and was no longer involved with the teams. Looking back, the organization regards the change as a mistake. The Scrum masters who were brought in, were mostly coming from a product management background or having been Scrum masters in another organization. They did not aspire to become group leaders and were not interested to learn technicalities or manage R&D. When they left, there was a gap left to fill. Developers did not want the role as becoming a Scrum master was not regarded as a real promotion since it was not “real” management. To become Scrum masters, developers would also have to stop coding, which they did not want. Since the management need could not be filled internally, it created a void, which was not ideal. The company needed to get the people from the outside as no one left in the team possessed the skills for management.

Recognizing the mistake, the company decided to take a step back. Each Scrum team would instead have an internal Scrum master, with responsibilities related to development. They would be developing and leading new designs while managing on the side. Scrum masters were responsible of checking trends and following up with the teams to see whether they are they doing the right things. Basically, this followed a typical style of micro-management. Scrum masters could have two teams working under them. Their tasks involved preparing and planning of sprints, marketing features and sprint reviews as well as checking release content and release times, following progress and making sure there would be no big software bugs. The teams consisted of 4-5 people.

If a team was stuck on some issue, it would be pushed to reach out to someone. The role of group leader was fitted for this purpose. As Scrum masters manage content, group leaders manage the people, by organizing daily meetings as well as answering questions from customers and other organizations. Together with Scrum master, a group leader discusses about content, bugs and other issues. Group leaders own a lot of responsibility, they keep in touch with teams, deal with hiring personnel and tend to have a lot of meetings. The role is a demanding one.

Along with agile transformation, the role of the interviewee had been changing also, having started as a developer and becoming a Scrum master of two teams. As the role of Scrum master had begun to change, the interviewee decided to step out of it while the company needed a new kind of expertise. As they were planning their next major release involving a lot of content, the interviewee owned a lot of knowledge having been deeply involved in content. As a new team was created and growing into a group of R&D people, knowledge transfer to a support group was needed. The organization had created a support layer, where it would enable to “support the support”. The Scrum teams would not work with customer cases and would focus on the content only.

At the time of building this new team, the interviewee took on a role of teaching about content and release, making sure that development would run smoothly. The new role was more of a specialist consultant, helping the teams with knowledge transfer. On a bigger scale it had been regarded as good practice to have a group in R&D, which would handle only customer cases. This would cause issues, which could be divided into two parts. Firstly, it was not ideal for the developers to handle bugs and customer complaints only as it created an issue of motivation. Secondly, if a team member would not be handling an area familiar to them, that area would not get enough care. That is why they began to disassemble and having teams focused on certain areas, owning a certain area of R&D. They would have customer-facing teams working for a certain amount of time before moving on to new technologies, which would keep up the motivation. In Scrum teams, the team members would change also, switching after a short release.

Back then the interviewee was leading the customer-facing teams but would leave the customer side as the company created a new product area moving on to building a new team with new processes for establishing a viable system. Working on a new product area with new developers, this would result in knowledge transfer from other people as: “The best way to learn is to handle bugs and problems”. In 4-6 months the interviewee had the most knowledge about architecture inside the company and was touching on many development areas very deeply, able to learn very fast about the whole, complex picture. In contrast, developers tend to be focused on small areas, not having an end-to-end view. A challenge that the developers faced at the time was creating new content with a need to learn quickly and communicate efficiently. By having acquired the sufficient knowledge by understanding designs broadly, the interviewee gained the current position of software architect.

Currently the interviewee works with five Scrum teams and a customer facing team. Two of the Scrum teams belong to one manager while the remaining three teams belong to another group. This creates two groups, with each group having a group manager. Group managers have the same level of management as group leaders. The group leaders are focused on a certain product and work alongside with the architect.

The architect describes the work as being very technology-oriented, communicating with product managers, own manager and being able to foresee the vision for next year or even five years. One part is working with the teams, with a new feature being provided by the product manager. The R&D need to see how to design from a higher perspective, having a deep understanding and knowledge, which helps to specify the requirements. It involves knowing which team should take which features and introducing a high-level design. By breaking it into low-level, the teams will be able to start working. As the teams are focused on the now, someone is needed to focus on the future. Architects supervise that everything is well designed. They own different product areas in Company B, for example recording, voice biometrics and real-time authentication.

Currently the company uses Agile, Scrum and Kanban and share tasks without time definition. Managing tasks is done according to priority. Graphs are used to put things into time lines and for example Gant, a visualization tool is used for planning and estimating how long an entire project might take. Every now and then, restructuring of R&D will occur and changes are made to the group structure. Keeping the same people in the same Scrum teams is however, regarded as the best way to work in Company B. Mixing up the teams is not always needed, instead it is best to focus on achievements inside the teams. Depending on orientation (for example product-oriented), different people with different expertise can sometimes result in the wrong chemistry. In general, it is preferred that Scrum teams work with same content with the same people, not making too big changes and avoiding rebuilding teams frequently. As agile is about velocity and estimating achievements, breaking teams once they became better, does not make sense for the architect. This is a sort of grey area, where some organizations wish to follow the experimental nature of Spotify, for example.

The interviewee sees the main challenge in successful organizational change as an issue of having a clear purpose, regardless of what the company may shift into. This is being done in levels, which are above the architect. Someone who sees the whole vision of the entire organization will be deciding on change. Shaking teams up might make them think it is time to change. The interviewee

feels that architects need to prove their worth, despite of experience. Leaving the company might happen in situations where change occurs. By default, people may tend to dislike change. That is why keeping them satisfied and happy is the real challenge in organizational change, according to the architect.

Collaboration and communication

The organization is big, which is why proper communication is key. Developers, Scrum masters and architects need to work well together. The product owner or product manager also has a critical role. Technical writers who provide documentation need to understand what teams are doing, translating it into understandable language. People from education teach different fields and need to support group services and sales people who need training for what is coming. Teaching is conducted internally, it includes installing of new features and configuration. Education plays a key role in knowledge-transfer, technical support people are also important as systems need to be maintained properly. Communication methods depend, having a lot of meetings can result in a temptation to reduce. E-mail can be a sufficient way to communicate, a call works better when an issue is complicated. If including only a few people, it is best to organize a meeting face-to-face.

From the point of view of the architect, Agile has resulted in a transition to a more collaborative environment as it used to be more individual. The interviewee is happy with this transition. Challenges in communication can relate to individual people, some need to learn about communication. People are not always accurate in their communication, it can result from many reasons, such as shyness or a wish to be respectful. Training or personal work can help when communication skills do not come naturally.

Customer or end user involvement

The issue may vary, some are more customer-oriented and rely on the company's commitment. Meetings with customers or conventions take place to get ideas. Some are more innovative ideas from working with other organizations. As there is a need to keep up with competition, innovations are important. People in charge of strategy, making market estimations and predictions, work closely with product managers who meet with customers to see if the direction is right. The architect sees customer involvement as more market-oriented. The company will do testing with end users, providing them with new innovative products. These customers will be the first ones to try a beta-version of the product before it becomes globally available. While the product version may have

problems, these customers have the benefit of being the first to try it out. The interviewee does not distinguish a significant change in customer or end-user involvement since moving to agile.

Corporate culture

Culture with a changed mindset is regarded as a result from going agile. It is more flexible and open to change than a traditional culture.

Processes or guidelines

For a large organization, this refers to having security awareness, measurements, reasoning why it is needed and how it affects the company. It is called governance and is being led by architects. Processes and guidelines are needed when choosing the new technologies, which are used to solve problems. They are also needed to perform good code review. Products with open API's enable integration for customers, for example in cloud, where guidelines are needed for creating and documenting an API. Java -cloud products use the Atlassian-package. Bitbucket is used for controlling of code.

Management and leadership

Teams aim to be self-organized and top-to-down management has been reduced. Scrum masters may do micro-management and group leaders are responsible for managing the people.

4.4 Company C

The interviewee has the role Production Officer, which includes product development, support service and customer development. The interviewee has been working for 6,5 years in the company. In that time, the company has always used Agile methods.

Regarding the changes, which have happened inside the organization, the interviewee mentions quick growth, which is the cause for many of the changes. Due to the organization having expanded and hired many new employees, it had to create new processes, increase educative programs, tools such as project management -related and develop different practices. With growth, the addition of regulations has also increased. The interviewee mentions the adding of regulations as

“unfortunate” but necessary. Regarding the processes, the company has made quick decisions once discovering that a process is not functioning for its’ purpose. For identifying issues with processes, internal communication has been effective.

Product development and project management methods include Agile and Scrum, which lately has involved DevOps for automatization. The interviewee mentions that the organization is not fully implementing DevOps, but rather a mixture of Agile, Scrum and DevOps. is also implemented. For customer projects providing software, Agile and Scrum are used. For platform development and updates, for example security updates, the organization utilizes automatization. Regarding tools, Trello has always been used for project management, Redmine has been for product development but has been replaced by Gitlab, which is a revision control tool. The organization also has in-house tools for customer project management, which includes customer support. It is mentioned that Trello is not regarded as fitting the needs of the organization to a sufficient amount and a replacement has been searched for and tested.

Regarding the role of the interviewee, it started with sales. Having a technical background and coding as a hobby, the role has changed to operating between sales and production. As the organization has grown, responsibility is now being shared since increasing of team members and teams. The role has gradually formed into what it currently is. There are nine teams in the unit where the interviewee operates. Each team has a focus area. The work responsibilities of the interviewee include human resources, envisioning the big picture and being the one in charge of funding, equipment and resources. The interviewee has adapted and grown with the organization, with each day bringing something new. The organization has also grown to be more international, having sales offices abroad while maintaining the practical operations in the local area. The organization wants to expand support services to offices abroad.

The interviewee is familiar with agile and mentions the organization being known for it. This includes having lightweight processes where the need for control is highlighted. Agile also demonstrates itself in project management and product development. It provides constant visibility and is iterative. Asking about the definition of agile, it is being described as flexible, providing the customer with something visible in a continuous manner with the ability to make changes along the way. The final output should be as complete as possible when the product is being finished. The organization does not prefer long projects. Agile includes the selection of tools for managing projects, prioritization and scheduling. It should be clear what needs to be done now and what needs doing

next. Logging is made to project management tools to provide visibility to the customer and work tasks should be visible similarly.

With regards to organizational change, the interviewee defines it as a question of change management as opposed to sticking to old habits. A manager can expect resistance but regardless, should lead in a straightforward manner while keeping people motivated. Motivating employees includes activating them, getting the whole work community involved, giving responsibility and providing a chance to influence things.

Collaboration and communication

Regarded as efficient and including the chosen tools to maintaining control and transparency and avoiding of misunderstandings. As makers tend to change, documentation helps to keep track. It is seen as beneficial to mix teams to keep them interested. In development discussions, people may speak about their hopes and interests. Communication includes introducing of work tasks. The company has nine teams with a little over 90 people in the local area. The teams consist of 5-20 members. Check-ups are conducted weekly, where any problems can be solved together. Daily directing and encouragement is enforced, with employees planning their own work, aiming to self-direct. Planning is done in the beginning of a project. Worker satisfaction is a high priority. When employees are not experiencing too much stress, it will demonstrate as a better result.

Customer or end user involvement

The company is focused on SaaS and provides solutions, which are easily updated and include support. Depending on the size of a project, customer involvement may not be necessary for a small project. For large projects, a separate project management tool will include the customer. Project planning considers time management for the customer and defines the necessary resources. Standardized and automatized testing as well as unit testing are implemented in projects. Standardized test environments are established and include different browser options, which are assigned as tasks.

Corporate culture

Culture is described as relaxed and casual. Trusting employees results in freedoms regarding managing of work, enabling employees to also work from a distance. The company allows flexible working hours with some monitoring. Employees may plan their work and notify when needed. The most important thing is that the right things get done, it is not so important how. Students will be

able to work part-time in the company with ease. This has functioned well so far. Hiring more international workers has enriched the culture. A focus on work satisfaction includes for example opportunities to exercise at a lower price through company benefits.

Processes or guidelines

The processes and guidelines aim for flexibility but are strict in nature due to abiding laws. Further development work is needed to create efficient but adaptable processes. The company wishes to upkeep a start-up state of mind and wishes to be free from heavy protocols and bureaucracy.

Management and leadership

The company does not reinforce micro-management. Employees need to have the opportunities to influence the work themselves, the person in charge of the teams need to have advanced technology skills while enabling the employees to do their job, providing them support. Management needs to be in control of the costs. The interviewee defines a good leader as being a good delegator.

4.5 Summary of results

Regarding the organizations, it is clear to see that there are clear similarities as well as differences, according to the answers. The differences might arise from the interviewees having different roles in the organization and of course, personality traits. Company size is also a likely cause of variation. In terms of frameworks and methods, Scrum and Agile were practiced in each organization, whereas one organization had begun to implement DevOps. Lean was not mentioned as a framework that was implemented but Lean principles may still be in effect. Particularly for organization C, where a start-up state of mind was desired.

All the interviewees mentioned some failed experiments, whether it was a choice of tool, marketing method, management product-related issue and so on. Particularly Company B had gone through a variety of experiments when adopting agile. This example was very interesting and informative with correlations regarding the theoretical framework of this thesis. Particularly when considering the Spotify's Agile Model, similarities are evident regarding building of teams that are focused on specific features and having the teams grow and expand. This would require the organizations to

react by either appointing leaders and or creating bigger groups, which the teams would belong to. It seems that large organizations are invested in knowledge sharing by creating a system of effective code review. The need for education was mentioned in two of the organizations. Some regarded learning as an important aspect of raising motivation.

The issue about customer or user involvement does not provide a clear answer as to having increased or not as an effect of agile. All the companies did seem to communicate rather consistently with their customers and or users. The amount of customer collaboration seemed to be dependent on the scope and nature of the project. Team communication and collaboration seemed a very vital aspect.

Regarding management and leadership, the answers were divided. In Company B, a global enterprise, micro-management was implemented while the two other companies responded negatively to the issue. It may be the case that big corporations tend to include micro-management more as the team sizes and number of teams are so high. All the interviewees mentioned teams being able to self-organize, so management may not need get too much involved in the day-to-day activities. This seemed to reflect positively on the employees as well as development.

Flexibility was an important part of each company in their processes. The four, core agile values state: *individual and interactions* over processes and tools, *working software* rather than clear documentation, *customer collaboration* over negotiating contracts and *responding to change* over following a plan. Reflecting on these issues based on the interview responses, it can be summarized that each company demonstrates elements, which support these statements. Agility in these organizations is shown in the flexibility of processes, ability to self-organize, a focus on motivating the employees by allowing freedom and the constant effort to improving the existing model.

5 CONCLUSION

To study agile transformation, it has been necessary to study the technology aspect as well as the definitions of what agile can refer to. Defining agile may depend on which aspect is being under a microscope while it seems that the agile mindset and culture are the underlying principles, which affect how the practices, frameworks and tools are chosen to support organizational culture and values. The issue of mindset was mentioned in one interview as the individual pointed out that some employees had to quit the company as they could not adopt the agile mindset.

The differences between traditional and agile organizations is an issue, which involves the style of management, the ability to adapt to change and to continuously reflect on what could be improved. It includes adding a level of flexibility, which will allow the organization to respond quickly when things, such as product requirements or the service environment, change. Top-to-down management is not supported in Agile as this may reflect negatively on the motivation and attitude of the employees.

The qualitative research seems to support the idea that agile teams are self-organizing and take responsibility for their own doing, which appears to have positive impacts regarding motivation and functionality. The issue of having moved to a more collaborative way of working, seems to also be an accurate statement. It can be said that the way in which teams work and communicate together can play a very important part in a successful completion of a project. The motivation of employees seems to include organizing training and a focus on efficient knowledge transfer. Learning seems to be a big factor of motivation, which corresponds with agile theory.

The question of whether organizations are truly agile or not, will once again return to the Agile Manifesto and its' core principles: *individual and interactions* over processes and tools, *working software* rather than clear documentation, *customer collaboration* over negotiating contracts and *responding to change* over following a plan. The research, both theoretical and qualitative, highlight the importance of an organization being invested in its' employees. This includes having flexible processes, encouraging collaboration with teams and customers and being able to adapt to change through varies ways, such as having a shared mission.

Managers and leaders in Agile have a demanding task, since they are battling between factors, which may argue with each other. In a fast changing, adaptive and flexible environment they are required to provide an element of consistency and constraint. Change management is and will likely continue to be a key issue in how projects and people are managed today. As team sizes increase, Agile seems to become a question of re-organizing and scaling. The experimental natures of organizations will vary, which will result in the agile methods and frameworks varying accordingly.

6 DISCUSSION

Regarding agile frameworks and tools, it would be difficult to define how an organization should make the selection of which ones to implement. Agile will not give a direct respond to that question but it will encourage organizations to practice awareness, define their long-term goals clearly, align the short-term goals to match them and examine the internal and external factors involved. This is a healthy advice for any organization that wants to avoid being stuck on the same level and instead, wants to continuously improve and reflect on how it is performing.

The issue of mindset and culture, while relevant, is also very challenging to measure or provide a definite meaning to it. Culture is something that surrounds people everywhere, perhaps that is why the effects of it are not always recognized although they surely exist and affect how people behave at work. Managers and leaders have an important role when it comes to organizational culture. By acting and managing others according to certain prejudice, they can create a culture marked by negativity and control or vice versa, a blossoming creativity and motivated, self-organized individuals and teams.

Agile transformation is a very broad topic, which can be viewed from many different perspectives, whether it is about the business, the people, products, technologies or project management. This has made it a challenging topic to handle, with touching on several relevant issues. The author is satisfied with having gained a comprehensive, overall view of the topic and believes that further research will be needed to examine the agile transformation and the characteristics of agile organizations. Business agility is also one area, where further research could be done. For the research, it would have been very interesting to interview multiple people working in the same organization in different roles to discover how consistent or differing their answers would have been. The author acknowledges that one individual is not able to give a complete or entirely reliable view about an organization, however the experiences of individuals should be dealt with importance, especially since working in the organizations for so many years and having witnessed the changes during that time.

7 REFERENCES

Aghina W.; De Smet A.; Lackey G; Lurie M.; Murarka M. 2018. The five trademarks of agile organizations. Retrieved 3.6.2018 <https://www.mckinsey.com/business-functions/organization/our-insights/the-five-trademarks-of-agile-organizations>

Airbrake. 2013. What is the Software Development Life Cycle (SDLC)? Retrieved <https://airbrake.io/blog/sdlc/what-is-the-software-development-life-cycle>

Anderson, D & Ackerman Anderson L. 2010. Beyond Change Management: How to Achieve Breakthrough Results Through Conscious Change Leadership.

Atlassian. DevOps: Breaking the Development-Operations barrier. Retrieved 1.6.2018. <https://www.atlassian.com/devops>.

Atlassian. What is kanban? Retrieved 28.5.2017. <https://fi.atlassian.com/agile/kanban>

Coplien J; Bjørnvig G. 2010. Lean Architecture: For Agile Software Development.

cPrime Worldwide. What is Agile? What is Scrum? Retrieved 1.6.2018 <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>

Crookshanks, E. 2014. Practical Software Development Techniques.

Denison D. 2012. Leading Culture Change in in Global Organizations: Aligning Culture and Strategy.

Della Tore, L. 2017. How to become a design-driven company in an agile world. Retrieved 20.5.2018) <https://www.atlassian.com/blog/add-ons/become-design-driven-company-agile-world>

Fernandes, T. Spotify Squad framework—Part I. 2017. Retrieved 28.5.2017. <https://medium.com/project-management-learnings/spotify-squad-framework-part-i-8f74bcfd761>

Girvan, L & Paul, D. 2017. Agile and Business Analysis: Practical guidance for IT professionals.

Harris, M. 2016. Essential Details on the Spotify Music Service. Retrieved 30.5.2018.
<https://www.lifewire.com/spotify-music-service-2438399>

Highsmith, J. 2016. Agile Project Management. Creating Innovative Products.

Leopold, K; Kaltenecker, S. 2015. Kanban Change Leadership: Creating a culture of continuous improvement.

Measey, P; Wolf L.; Berridge C; Gray A.; Levy R.; Les O.; Roberts B.; 2015. Agile Foundations: Principles, practices and frameworks.

McKinsey & Company. 2017. How to create an agile organization. Retrieved 1.6.2018.
<https://www.mckinsey.com/business-functions/organization/our-insights/how-to-create-an-agile-organization>

RealtimeBlog. How to choose between Agile and Lean, Scrum and Kanban — which methodology is the best? <https://realtimeboard.com/blog/choose-between-agile-lean-scrum-kanban>.

Rouse. M. TechTarget. Retrieved 1.6.2018. searchsoftwarequality.techtarget.com/definition/unit-testing

Saffer, D. Designing for Interaction. 2009.

Sassard S; Soderpalm H; Swahnberg O; Reuters. 2017. Spotify is now valued at \$16 billion — and it could be worth a lot more when it goes public. Retrieved 30.5.2018.
<http://www.businessinsider.com/spotify-now-valued-at-16-billion-2017-9?r=US&IR=T&IR=T>

Simplicity Through Breadth. Software Development Life Cycle (SDLC) Waterfall Model. Model. Retrieved 30.5.2018. <https://faisalsikder.wordpress.com/2009/12/18/software-development-life-cycle-sdlc-waterfall-model/>

Spotify Engineering Culture – part 1. <https://vimeo.com/85490944>

Toivanen, I. 2017. DevOps Jatkuvan Kehittämisen Tukena. Retrieved 1.6.2018
<http://www.octo3.fi/devops-jatkuvan-kehittamisen-tukena/>

Warcholinski, M. Differences Between Lean, Agile and Scrum. Retrieved 28.5.2017.
<https://brainhub.eu/blog/differences-lean-agile-scrum/>

8 APPENDICES

APPENDIX 1

1. Describe your role in the company.
2. How long have you worked in the company?
3. What kind of changes have occurred in your organization during your work experience?
4. Can you name product development or management methods, which have been implemented in your organization during your career?
5. Which product development or management methods are currently implemented?
6. How has your work role changed since you started working in this organization?
7. Are you familiar with agile?
8. What is the definition and meaning of agile in your own words?
9. Describe the characteristics of the following in your organization.
 - a. Collaboration and communication
 - b. Customer or end user involvement
 - c. Corporate culture
 - d. Processes or guidelines
 - e. Management and leadership
10. In your opinion, what are the biggest challenges related to organizational change? How to overcome them?