MSc Program
Engineering Management

Agile software development within the flat organization Case study: Evaluating the implementation in a small tech start-up

A Master's Thesis submitted for the degree of
"Master of Science"

supervised by
Dr.LarryStapleton

Orthodoxia Papachartofyli

01646734

Vienna, 10.01.2018

Affidavit

I, **ORTHODOXIA PAPACHARTOFYLI**, hereby declare

1. that I am the sole author of the present Master's Thesis, "AGILE SOFTWARE DEVELOPMENT WITHIN THE FLAT ORGANIZATION - CASE STUDY: EVALUATING THE IMPLEMENTATION IN A SMALL TECH START-UP", 88 pages, bound, and that I have not used any source or tool other than those referenced or any other illicit aid or tool, and
2. that I have not prior to this date submitted this Master's Thesis as an examination paper in any form in Austria or abroad.

Vienna, 10.01.2018

Signature

ACKNOWLEDGEMENT

I would like to express my endless gratitude to my professor and supervisor, Dr. Larry Stapleton for his constant patience, guidance and valuable insights throughout the process of writing my thesis.

To my husband, Costas, whose love and support made the completion of this thesis and this master program possible.

And to our daughter Lydia, who patiently waited for this thesis to be completed before making her appearance into this world.

Abstract

The present study focuses on giving a short overview of different organizational structures and of most famous Agile frameworks, in order to provide a background context for conducting research in a small software company. By focusing on a specific software company, the study aims to assess the effectiveness of the organizational structure selected for this company and if and how this structure relates to the success -or not- of Agile implementation. The population of the study includes the aforementioned software company and the participants are 28 employees of this company and the 3 people of whom the company's leadership consists. The literature review of Agile, the differences, advantages and disadvantages of each organizational structure as well as the personal professional experience of the author in the company as well as in the field of Agile helped formulate the objectives and research questions of this study. A questionnaire was designed and provided to the 28 participants and follow-up clarifications, comments and interviews with the leadership were conducted when needed. Literature review demonstrates that choosing the appropriate structure to fit the organization or having the ability to adjust it as efficiently when needed, is crucial for every organization. By analysing the questionnaire results and assessing important findings revealed during the interviews, the study revealed that the employees and the company's leadership perceive the simplicity of structure of the company as a factor that contributes to the successful operation of the company, as they feel it aids in establishing direct communication, cooperation, adaptability, fast decision-making and cross-functionality that are needed in order for Agile software development practices to succeed.

Keywords: Organizational structure, hierarchy, Agile software development, flat structure, Scrum, Kanban

Table of Contents

Chapter 4: Methodology

Chapter 5 : General Observations and Data Analysis

Chapter 6 : Discussion of Findings

Chapter 1

Introduction

1.1 Statement of the problem

Before addressing the research problem, it is important to define the term “organizational structure” and explain why it is important. An organizational structure defines how activities such as task allocation, coordination and supervision are directed toward the achievement of organizational aims. (Pugh, 1990)

Simply put, structure sets the rules in terms of roles, accountability, and authority within an organization: it specifies who does what and who reports to whom so that the organization can operate in a coordinated fashion. A successfully applied structure, suitable to the needs of the organization will provide context for communication and knowledge generation and transfer. (Silvestri, 2012) By grouping and linking people together, structure sets the stage for learning and mutual supporting to take place, which can offer a competitive advantage when learning processes and outcomes are difficult to imitate by competitors. Even deeper, however, organizational structure is critical to identity formation. Individuals identify at different levels with their role in an organization, with their broader community of practice, with the unit in which they are embedded, and with the culture of the organization to which they belong. (Galbraith, 2012) Societal, environmental and technological changes slowly caused a shift from the bureaucracy-supporting theories of Taylor, Fayol and Weber to the more "modern" concepts of divisional and functional structures as opposed to classically hierarchical. (Mohr, 1982) Changes of the 21st century such as global competition, disruptive technologies, shorter product life cycles, and more sophisticated and knowledgeable customers are causing fundamental shifts towards flatter/flat or a combination of product-centric divisionally flat organizations. (Morgan, 2014)

Moreover, tech companies often face other challenges when trying to improve their overall managerial experience by implementing and successfully using ways of organizing themselves and their everyday work according to frameworks and methodologies such as Agile, Scrum, KanBan etc. (Koch, 2005) In a highly

competitive and quickly changing working environment and market, companies are constantly eager to find ways to increase their productivity, the employee performance and ultimately, the customer satisfaction and preference. Especially though, in the world of IT companies, the need for agility and adaptability is even greater, as software companies are the ones that "lead the way" with the development of new disruptive technologies. (Schmidt, 2016) It should be of no surprise that most of these new models such as the flat organization and the holacratic organization originally appeared and were applied in IT companies. (Morgan, 2014)

Choosing the appropriate structure to fit the organization or having the ability to adjust it as efficiently as possible when the previous one is not effective anymore, is of crucial importance for the IT world. A wrong decision from top management regarding how it would be better for employees to structure themselves, what level of decision-making power they can -and should- supply in which level and how deeply should the formalization of procedures run, are all points of the research problem. (Koch, 2005) By focusing on a specific IT company and closely examining its structure, the struggles it is facing as it is growing and the trials and tribulations of applying Agile methodologies both on the level of management and on the level of software development, the present study research aims to shed some light foremost in the importance of choosing the appropriate organizational structure and the existence or not of relations between implementation of Agile and said organizational structure.

1.2 Significance of the study

A simple internet search of the term "organizational structures" will reveal many different layouts of organizational structure, from classical hierarchical to product-centric to current trends using combination of a team-based flat structure. While there is no doubt that each one of these different structures has its own advantages and disadvantages, its efficiency depends on the willingness of the organization to "make it work" and of course, it can be assumed that when this willingness is not present, the complexity of the structure hinders the outcome. (Koch, 2005)

In the world of IT, although there are numerous approaches from many different but almost equally successful organizations, a trending belief is that when trying to

achieve a high-quality and high-complexity product as a software one, simplicity in the structure works best as it cuts out all the middlemen and various escalations of issues and even saves time from endless meetings that do not need to occur since all the interest parties communicate with each other on a day-to-day basis. (Galbraith, 2012) “Hierarchical organizations cannot react to new market opportunities and changes fast enough, this impedes the company’s survival in the long run”, says Michael Dubakov, founder of Targetprocess. “Organizations should distribute managers’ responsibilities among cross-functional teams and boards to become flatter and increase their overall agility”. (Dukabov, 2015) It is not of course to say that traditional models that have withstood years of success were not rightfully successful, but as the software product changes and adjusts rapidly, different approaches can guarantee the best productivity, efficiency and employee satisfaction. Moreover, a vast organization located in many countries with numerous products might not have an alternative other than using a pyramid-like design but even then, in a more locally-flat and product-centric approach. (Galbraith, 2012)

It is therefore, crucial, to first understand the reasons of existence of each organizational design and why the right design criteria are necessary to ensure success in the field of software development. Theory that will be presented in the thesis might be applicable to other non-software products, but the focus in this case is on software companies and their very specific design and production processes.

1.3 Contribution

It is of course easy to theorize about different approaches when it comes to choosing a suitable organizational design, but as real-life experience dictates, textbook solutions and one-fits-all approaches might not be the answer when the theory needs to be practically applied. Therefore, practitioners could be equally invested in finding material and technical experience on similar issues because there is not always enough time to brainstorm over which approach/methodology/practice will work best, and occasionally there is not enough human experience to achieve this result. In those situations, existing material combining theoretical background and an actual example from the real business world, can help professionals save time and make informed decisions as soon as possible. Also, a case study of a company and the

structure-related decisions made there can be helpful for either a small start-up company that struggles to decide which organizational design fits them best or an existing company that has decided to alter their structure. Furthermore, for academics studying current and future trends in the models of organizational structure, a case study of a company and its organizational design can provide valuable insights and help in supporting developed or developing theories.

The study of a company where the Agile framework is being used not in the textbook approach but in a way it suits the employees' skills and mentality, is something that would be useful for students and academics as well as professionals, since the practical information in this section is not that common, especially for European companies, as the majority of resources on the matter comes from the USA. Finally, for a company struggling to adopt Agile, more information on whether they should also consider modifying their structure, could help make a more informed decision. Also, most of existing information regarding the Agile framework and how it can be applied it is either very theoretical or very limited when it comes to real-world examples. And even so, rarely is the correlation done between the chosen organizational structure and the level of application of Agile.

1.4 Research purposes

- Identification of factors that influenced the organizational structure of the software company under examination.
- Identification of a potentially specific correlation between organizational structure and Agile implementation in an organization.

1.5 Research questions

RQ1: What factors informed the organizational structural choices made by members of a software company using agile methods?

RQ2: To what extent did members of the organization in a case study of a software company using agile methods perceive the structural choices made to be appropriate for their business?

1.6 Outline of study

Chapter 1 served as an introduction to the study. A review of related literature is provided in chapters 2 and 3. More specifically, chapter 2 summarizes the most important concepts and forms related to organizational structure and chapter 3 attempts to summarize briefly on the topic of Agile Software Development and Agile practices in general. Chapter 4 presents the methodology that was used in conducting the study and the data gathering techniques used. Data analysis is also performed in chapter 5, as well as the profiling of the specifics of the company under study before discussing the research findings and the conclusions in chapter 6.

Chapter 2

Theoretical background and literature review of organizational structures

2.1 Introduction

The notion of organization structure and its directives - defining how activities such as task allocation, coordination and supervision are directed toward the achievement of organizational claims, is nothing new. Almost a century ago, Henri Fayol proposed formal hierarchy as the most effective mechanism of dividing and coordinating labour. (Miner, 1982) Around the same period, German Sociologist Max Weber introduced the concept of the "ideal" bureaucracy as having a strong hierarchy of authority, extensive division of labour, impersonal rules and rigid procedures. Later, Rensis Likert developed his own vision of organizational structure, named "human organization", with emphasis given on the human factor -as opposed to the approaches taken by Weber and Fayol in the past- the growth of supportive relationships and the employee participation. (Griffin, 2014) Those three approaches, despite having fundamental differences, form the basis of the "classic organization theory". Societal, environmental and technological changes slowly caused a shift from the bureaucracy- and hierarchy-supporting theories of Fayol, Weber and Likert to the more "modern" concepts of divisional and functional structures as opposed to classically hierarchical. During the 1960s, Chandler (1962) originally described and studied three types of structures: functional, multidivisional, and holding company. For each structure, there was a unique strategy. A functional organization was the means for implementing a single business strategy. A multidivisional structure was used for diversification into multiple related businesses. The holding company was appropriate when diversifying into multiple unrelated businesses. (Galbraith, 2012) Burns and Stalkers in 1961 talked about organizations falling under two categories, mechanic and organic. Whereas mechanic structures are more appropriate for environments where change is neither frequent nor rapid, organic structures serve fast-changing environments in demand of great flexibility

(Burns et al., 1961). Since then, all these forms have evolved from “structures” to complete “organizations” and from pure to mixed forms.

Many different categorizations have been presented over the years, presenting the possible structure an organization can have. Most of the possible different organizational layouts can however be placed under one of the following categories: functional departments that are task specialized; self-contained divisional units that are oriented to specific products, customers, or regions; or matrix structures that combine both functional specialization and self-containment. Changes of the 21st century such as global competition, disruptive technologies, shorter product life cycles, and more sophisticated and knowledgeable customers have caused organizations to modify their layout into more integrative, fast-responsive and flexible forms. Customer-centric structures, process structures, flat or product-centric divisionally flat structures are some examples of the most recent shifts.

2.2 Theoretical background

It is evident from reviewing the expert literature that many different approaches, terms and criteria exist in the effort of recognizing and categorizing the various forms of organizational structures. Before presenting some of them, their compatibility to the needs of each organization as well as their respective advantages and disadvantages, there are some other concepts closely intertwined with structures that need to be briefly summarized.

2.2.1 Fundamental Processes - Division and Coordination of Labour

Every organizational structure, from a small 2-person company to a multinational conglomerate, should first and foremost take into consideration that two basic requirements must be met: the division and coordination of labour. By subdividing the work into specific tasks assigned to specific people, they become more specialized in their field of work and then work efficiency gradually increases. Of course, this division of work optimally needs to be in regard with the specific abilities of each individual to obtain maximum potential.

After work is divided, it also needs to be coordinated and monitored. Everyone and everything need to be working in synchronicity. This monitoring and coordinating activity can be achieved through different approaches. Formal and informal forms of communication support the overview of the work activities by engaging people in constant contact on an everyday informal basis or by going through all the appropriate channels of communication as indicated by the hierarchy of the organization, respectively. (George J. et al., 2012)

2.2.2 Key elements of organizational structure

When an organization is being structured, specific aspects need to be taken into configuration. Those are the following:

Span of Control

Span of control refers to the number of people that need to report directly to the level that is higher than them in the company's hierarchy. Over the years, many theories and formulas have been used to determine the best span of control but there is no rule that covers every possible situation. Today, one general rule of thumb is that the choice for the most appropriate span of control depends on the complexity of the task, as well as on the time that is needed to supervise and coordinate all the subordinates' activities and on the proximity of the supervision. (Stroh L. et al., 2002) A wider span of control -meaning more subordinates- is possible when employees perform routine tasks, whereas a narrower span of control -meaning fewer subordinates- is required when employees perform complex tasks. Also, when tasks are highly linked, supervisors theoretically need a narrower span of control to coordinate and manage a high level of interdependent work. (McShane S. et al., 2008)

Centralization and Decentralization

An organization is described as centralized when the decision-making authority is limited at the hands of a small group of individuals at the top of the organizational hierarchy. On the contrary, decentralization allows decisions to be made throughout the hierarchy. There are advantages and disadvantages in both approaches.

Traditionally, centralization is credited with improved coordination of the overall activities, especially under stable conditions in a well-established environment. However, centralization can cause a limited flow of information from the higher to the lower ranks and vice versa, causing communication & performance issues. (Stroh L. et al., 2002) Additionally, centralization is not suitable for environments with high rate of change and uncertainty, as in these cases people must have the authority to make fast decisions without waiting instructions from the official channels in order to keep up with the changing environment. (George J. et al., 2012) Another advantage of decentralization is that workgroups and teams can more easily focus on value-adding processes as they do not have to dedicate as much time in going through several hierarchical levels to get approval for their next move. This is even more accurate when the size of the organization increases, and more time is needed for all the interested parties up and down the hierarchy to be informed. As a result, when the size increases, decentralization to some extent becomes inevitable. (Stroh L. et al., 2002) Although the current trend is slowly decentralizing authority, there are challenges that are coming along with it, so the correct ratio of de- and centralization must be achieved to avoid issues such as too much initiative taken by people and teams -initiative that is not necessarily aligned with the interests and strategy of the organization- as well as problematic performance due to inadequate communication among different teams, workgroups or divisions. (George J. et al., 2012)

Formalization

The establishment of rules and standardized procedures to control all the activities of an organization is called formalization (George J. et al., 2012). The more standardized some procedures get; the less necessary direct supervision becomes. As a company increases in size, formalization might become inevitable as it can help best monitor all employees and coordinate their activities, especially when due to the increasing size, direct supervision or informal channels of communication are not as effective. Thus, as compliance is increased, performance and efficiency also rise. However, it does come at a cost: As formalization applies specific rules and procedures, flexibility and creativity are hindered. In a constantly stable environment this would not be problematic, but in an organization facing constant changes, the

need for a creative and customized solution is higher. An environment of strictly followed procedural instructions can cause many employees to feel disempowered. Moreover, too many rules and processes demanding to be followed to the letter create waste in terms of time management, causing the efficiency to decline. (McShane S. et al., 2008) Finally, it is easy for an organization to lose focus from their original purpose of existence and become too fixated on adhering to the rules.

2.2.3 Contingency theory

According to the contingency theory, the organizational structure that best fits to an organization is the one that matches the factors, otherwise called contingencies, that have the most impact and can potentially cause uncertainty. (George J. et al., 2012) The most important of those factors that subsequently influence the organizational design, are usually the following, shown in Figure 2.1 below:

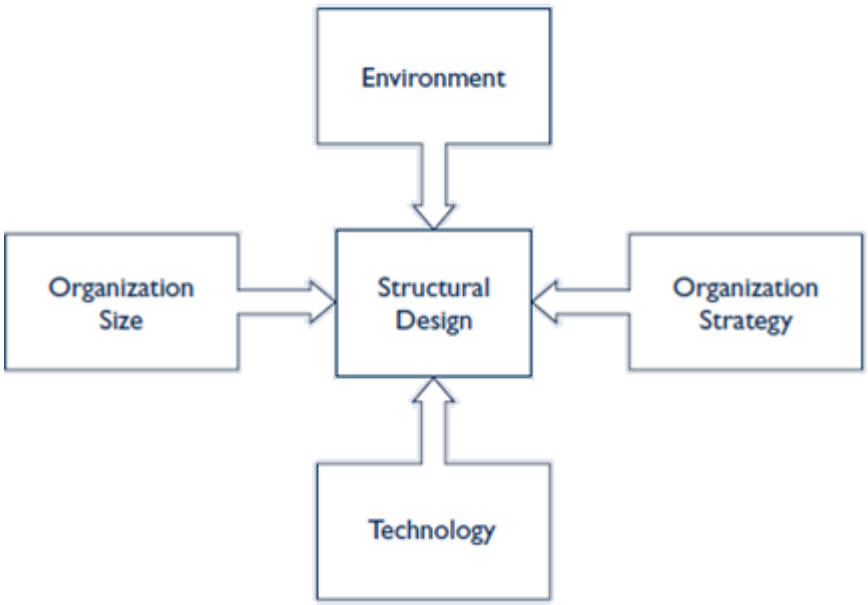

Fig. 2.1 : Contingencies influencing Structural design (Cummings T.G., 2009)

More specifically:

Organizational Environment

Depending on the characteristics of the external environment of an organization such as complexity, dynamic behaviour, simplicity and diversification of products, a different structure might be a better match for each organization. To elaborate further, the complexity and dynamic behaviour of the organization environment indicate a state of constant change that does not create conditions of predictability and identification of commonly appearing patterns. So, for example, an organization existing in such a dynamic and complex environment might benefit from choosing a structure that allows even lower-level -in terms of hierarchy- employees to make decisions in order to accommodate fast pace of changes. On the other hand, when uncertainty and complexity are low, and the stability of the environment is high, the structure can be more formal and clearly defined, as not so much quick coordination is needed to obtain results. (George J. et al., 2012)

Organizational Technology

The set of people's skills and knowledge along with physical infrastructure of tools, machines, computers etc. form the organization technology that is needed in the total making process of products and services for customers. (George J. et al., 2012)

Generally, as the technology gets more complicated and the tasks are more interdependent, more coordination is needed, and the organizational structure needs to be flexible in order to accommodate sudden changes and unpredictable issues. On the other hand, when the technology is simple and there is not a great degree of interdependency between tasks, then the structure can be rigid and more formal. (DeSanctis G. et al, 2005)

Organizational Size

Usually, organizations with many employees need to structure themselves differently compared to small organizations. As the number of employees increases, each employee gets a more specialized task, leading to greater division of labour. Subsequently, this creates the need for a more specific and well-defined coordinating and controlling mechanism. This is the main reason that large organizations

standardize and formalize their processes and work functions as much as possible, mainly focusing on using formal ways of communication. (Hertel G., 2002)

Another way the size of the organization can affect the general structure is through the level of centralization and decentralization that occurs. As can be assumed, the greater the size, the greater also is the need to decentralize, as the whole flow of decisions cannot be limited anymore to a small group of people. (McShane S. et al., 2008)

Organizational Strategy

Strategy defines how the organization can use its resources to gain competitive advantage and to achieve its goals. Subsequently, structure arises -not exclusively but in correlation with more factors as we have already examined- from the organizational decisions. (Cummings T.G., 2009)

For example, an organization might offer massively popular products at prices that lead to maximized productivity, also known as cost leadership strategy, whereas another organization might aim to target specific clients with customized products. This strategy is called differentiation strategy (Porter A.M., 1980). When it comes to structure in those examples, a cost leadership strategy would demand high level of job specialization and standardization, to maximize production and efficiency at the same time. In contrast, a differentiation strategy that calls for constant client communication and flexibility, would demand less formalization and centralization so people can self-coordinate to meet clients' specific needs. (McShane S. et al., 2008)

2.3 Organizational Structures

After examining factors that influence and form specific organizational structures, it is time to focus on the structures themselves. The vast majority of every organization has evolved to its current form from a simple structure. In a simply structured organization with only a handful of employees and usually one product, there is no complicated hierarchy, but direct reporting to the owners instead. There is no need for formalization or specialization and the overall flexibility is high. As the organization grows, direct coordination and supervision from the owner gets

complicated. (McShane S. et al., 2008) In the case of growth, to which structure would the organization migrate? There are many ways to organize and differentiate among all known organizational structures and the types of the organizations that fit in each structure. Apart from the simple structure, we can also isolate the functional, divisional, matrix, team based and network structure.

2.3.1 Functional structure

Probably the most recognizable and mostly used model structure is the functional structure, as presented in Figure 2.2 below. The idea of this structure is grouping together people who have similar knowledge and skills or perform similar tasks, also known as a method of departmentalizing. (George J. et al., 2012) For example, the splitting of the organization into functional units leads to having different entities such as marketing, sales, human resources, R&D, production and finance, while each department head is someone who is familiar and has been trained according to the objectives of each department. (Cummings T.G., 2009)

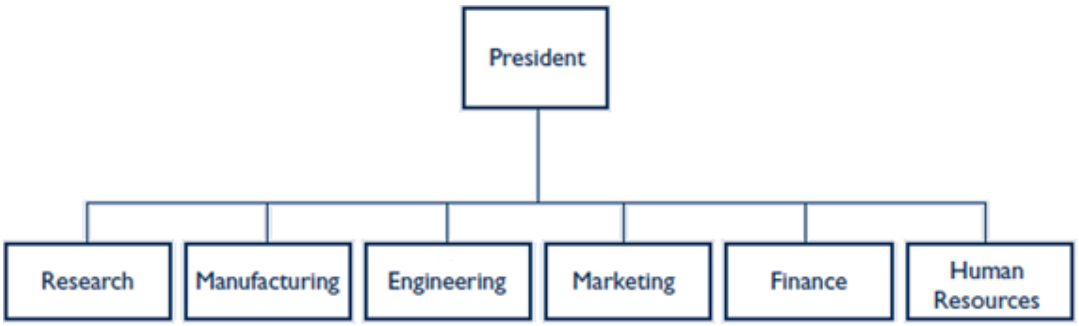

Figure 2.2: Example of a functional organization (Cummings T.G., 2009)

Typically, an organization structured functionally has a high degree of centralization and standardization of work processes to facilitate maximum coordination. (McShane S. et al., 2008) The typical profile of a company structured functionally is usually one of a relatively small to medium size company, operating in a stable environment with routine technology, typically of small complexity. (Cummings T.G., 2009) The high degree of division of labour and of specialization accomplished

by splitting an organization into functions has numerous advantages and disadvantages as shown below:

Advantages of the functional structure

Facilitating of communication: It is easier for people with similarities in their skills and tasks to communicate with each other, as it is easier for people with similar expertise to monitor and overview their subordinates, as well as communicating with them. (Cummings T.G., 2009)

Increasing specialization and performance: Learning from people with similar expertise leads to improvement of skills and overall performance is enhanced. Having supervisors and managers that are experts in this function also leads towards the same result. (George J. et al., 2012)

Promoting career development from the talent pool: Within the same function, talented and motivated employees are enabled to climb the career ladder. Working within the same function, specialists are easily recognized and facilitated to move upwards, while they share their expertise with the rest of the department. (Cummings T.G., 2009)

Reducing duplication of resources: By grouping people functionally, the organization achieves optimal use of people and resources as there is no duplication of services. (Cummings T.G., 2009)

Disadvantages of the functional structure

Focus on routine, repetitive tasks: By working on a specific task on a limited-scope function, employees lose touch with the big picture of the business and could only have very short-term goals related to routine tasks instead of abiding to the organization's goals. (Cummings T.G., 2009)

Insufficient coordination: The limitation of people in functional silos leads to hindering coordination and communication. (McShane S. et al., 2008) Scheduling among different departments can be difficult when each of them aims to maximize its own performance and not the organization's. (Cummings T.G., 2009)

Difficult Management Control: Top management may struggle to maintain control as the organization expands. The more complex the activities become as the company grows, the less suitable is the functional structure to deal with this new complexity and instability. As each functional unit gets more decision-making responsibilities, the coordination of activities becomes more difficult. If the expansion also takes place in new geographical regions, the challenge of maintaining control becomes even greater. (Cummings T.G., 2009)

2.3.2 Divisional Structure

The usual transition from a functional structure to divisional usually occurs in order to cope with the coordination issues that occur when an organization starts expanding. Under the divisional structure the activities are grouped by division, and each division internally retains a functional structure. As division, we define a group of functions dedicated to making a specific product or service. Each division has greater autonomy than a regular functional unit. (George J. et al., 2012) Typically, the divisionally structured organization is large enough to justify the duplication of resources in the different divisions, and operates in an unstable environment, as opposed to the functional structure organization. (George J. et al., 2012) The divisional structure can be product-centric, customer-centric or dedicated to a specific geographical location, depending on the needs of the organization.

PRODUCT STRUCTURE: Grouping functions by types of products, so that each product division contains all activities that are necessary for that specific product. (George J. et al., 2012) Each of these divisions contains its own set of functions, such as accounting, marketing, R & D and so on, as can be seen in Figure 2.3 below:

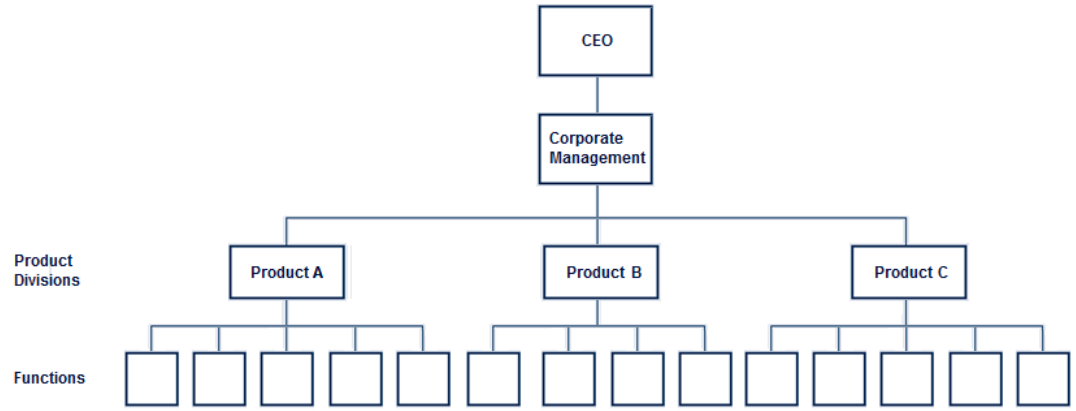

Figure 2.3: Example of a product structure organization (Cummings T.G., 2009)

MARKET STRUCTURE: Grouping functions by types of customers so that each division services a specific market segment, as shown in Figure 2.4 below, with the goal of optimally meeting that specific customer's needs. By following this approach, each market or customer division can specialize in all the aspects of the requirements of this customer. (George J. et al., 2012)

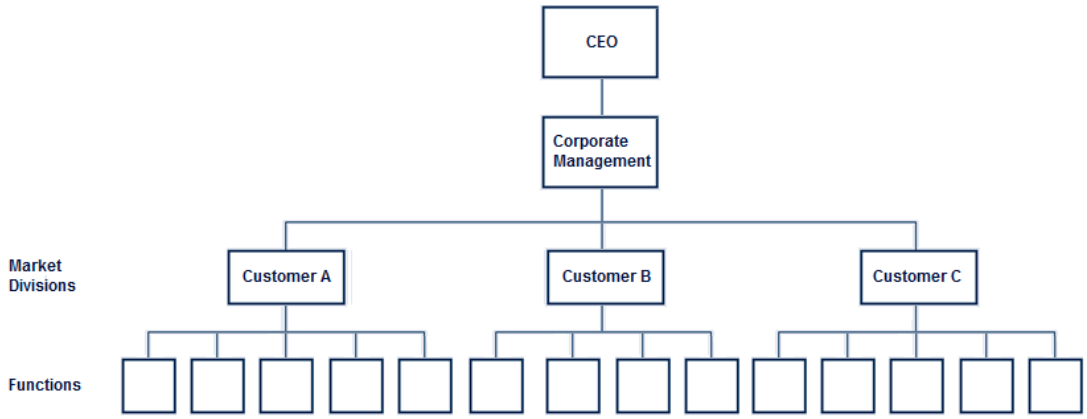

Figure 2.4: Example of a market structure organization (Cummings T.G., 2009)

GEOGRAPHIC STRUCTURE: Grouping function by region so that each division contains services customers in a specific geographic location, as shown in Figure 2.5. When an organization expands geographically, management and overview of the activities becomes increasingly difficult as not all resources are stationed in the same physical location, and each location site can face its own challenges. By splitting the organization in regional divisions, coordination becomes feasible again and also,

each division has the flexibility to adapt to the local preferences and demands. (George J. et al., 2012)

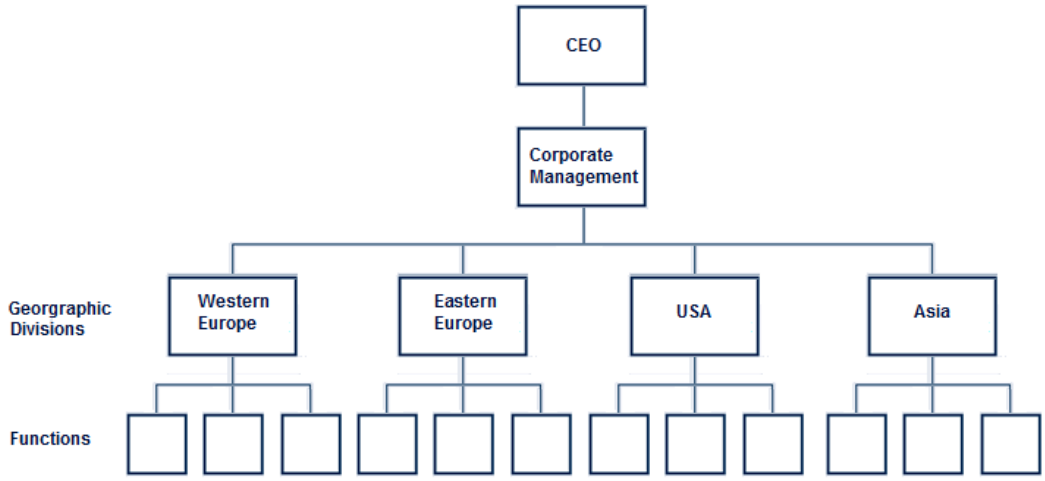

Figure 2.5: Example of a geographic structure organization (Cummings T.G., 2009)

As expected, choosing one form divisional structure depending the needs of the company, is accompanied by a variety of advantages and disadvantages, as are briefly presented below:

Advantages of the divisional structure

Ability to focus on the product or customer: Each division, as it contains its set of functions, is able to focus on a specific product or customer and provide high-quality products and services, with a heavily outcome-oriented behaviour. (Cummings T.G., 2009) In the case of regional divisions, local managers are able to focus on the regional conditions much better than top management and can make better decisions. (George J. et al., 2012)

Growth is easily accommodated: Similar products or clients can be added to existing divisions with minimum discrepancies. Different products or clients can be accommodated by the creation of a new division. (McShane S. et al., 2008)

Communication and coordination: Communication between internal functions is much easier, now that every needed function exists in the same division, regardless of the type of divisional structure. (McShane S. et al., 2008)

Skill expansion and sharing: The existence of many different functions under the same division provides employees with the opportunity to have a general overview of the whole procedure and also, to expand their skills and share their expertise with anyone interested in diversifying his or her abilities, making troubleshooting and brainstorming more fruitful. (Cummings T.G., 2009)

Motivational advantages: Local managers are given great autonomy which can increase their involvement in the workplace and boost their performance. Job satisfaction from the managers often leads to high performances in the division in general, as people are more involved in the decision-making processes and have higher responsibilities. (Cummings T.G., 2009) Lastly, the proximity that each division has to its own product or customer, makes the employees more personally invested, more committed and more determined in reaching the division's goals. (George J. et al., 2012)

Disadvantages of the divisional structure

More resources, more costs: As each division has its own set of functions, the general organizational costs increase due to this duplication of resources. (McShane S. et al., 2008)

Communication issues: Divisional structures traditionally have more management layers and people in management positions compared to functional structures, so communication can be more complex if people from different functions within the same division or between divisions need to communicate and coordinate their efforts. (George J. et al., 2012)

*Knowledge silos***:** Expertise is spread throughout the various business units, which reduces the ability and perhaps motivation of these people to share their knowledge with counterparts in other divisions. (McShane S. et al., 2008)

Inefficient use of resources: Unless the division is quite large, resources are not used as efficiently as in functional structures where resources are pooled across the entire organization. (McShane S. et al., 2008)

Focus on divisional goals: Divisions may start to compete for organizational resources and pursue their own goals at the expense of organizational goals. (George

J. et al., 2012) These conflicts reduce cooperation and can sometimes promote allegiance to divisional rather than organization objectives. (Cummings T.G., 2009)

2.3.3 Matrix Structure

The divisional structure might be appropriate when management needs to be quicker and more flexible when dealing with change, but the more dynamic the organizational environment is becoming, the more need there is for more flexibility. The matrix structure combines characteristics from the functional and divisional structure, while aiming at maximizing the advantages and minimizing the disadvantages of both structures. (Cummings T.G., 2009) The general idea is that a lateral structure focused on product or project coordination is overlaid over a vertical functional structure (Cummings T.G., 2009), as pictured below, in Figure 2.6:

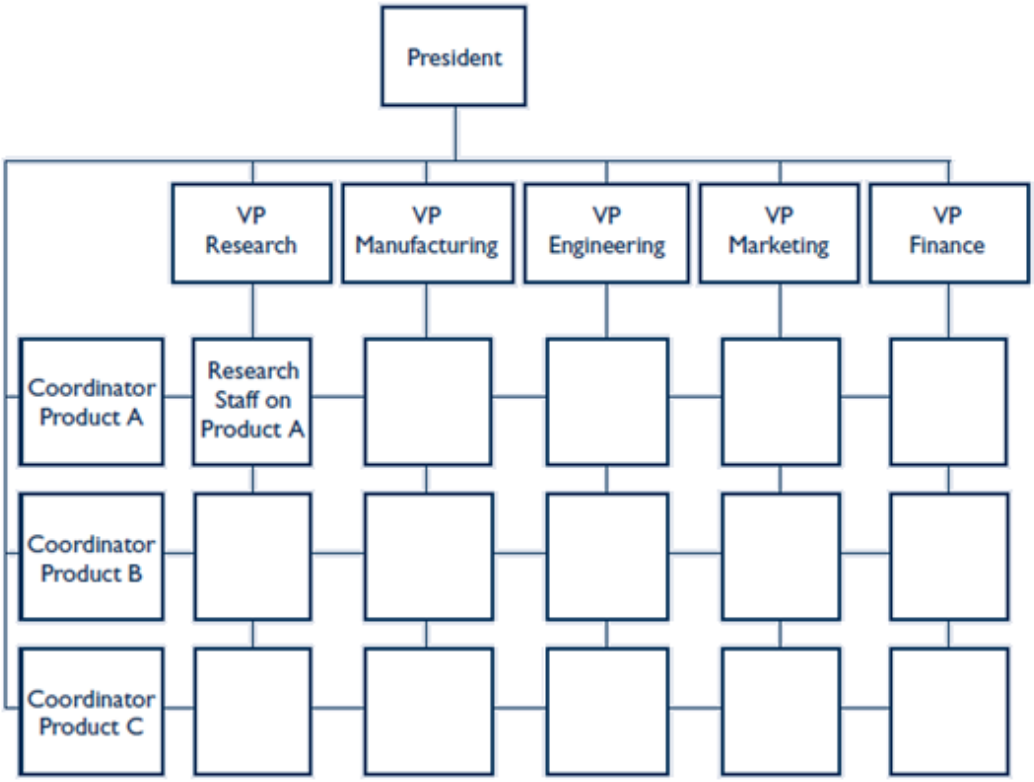

Figure 2.6: Example of a matrix organization (Cummings T.G., 2009)

As pictured above, people and resources are grouped in two ways simultaneously: by function and by product. (Galbraith, 1971) Employees are grouped vertically in a permanent functional unit, and by doing so, knowledge and expertise is easily exchanged, supporting the overall productivity. Furthermore, employees are also grouped horizontally into product teams, in which members of different functions work together to develop a specific product. (McShane S. et al., 2008) Even though the resulting structure seems complex, it is, in reality very flexible. Each person in the product teams pictured in the Figure above reports to two people: the functional boss, who is interested in the performance of the employee from a functional perspective and similarly, the product boss. (Cummings T.G., 2009) Thus, team members are known as two-boss employees. Matrix organizations are typically characterized by their dual focus on very specific product demands which call for technical specialization, in an environment that requires high abilities of information-processing and also calls for shared resources instead of unhindered duplication. (Cummings T.G., 2009) These preconditions are mandatory otherwise the success as it will be soon explained in the following sections, apart from the coordination issues a matrix can face effectively, it can cause motivational concerns. This is the reason why most companies -usually high-tech companies- that depend for their survival on rapid product development designed to meet very specific customer needs use matrix structures. (McShane S. et al., 2008) Flexible and adaptable as it might be, the matrix structure is characterized on occasion not only by its numerous advantages but also by some disadvantages:

Advantages of the Matrix Structure

People can work dynamically and cross-functionally: New products or projects can be implemented quickly by using people flexibly and by moving between product and functional orientations as circumstances demand. Two-boss employees are transferred from team to team when their functional expertise is needed, as product team membership in a matrix structure is not fixed. (Cummings T.G., 2009)

Optimization of resources: The matrix structure offers a way of sharing resources and expertise across departments – which can make a project more cost-effective. Matrix is ideal for project-based businesses with fluctuating workloads. (McShane S. et al., 2008)

Improved communication: Helpful in overcoming traditional department barriers, improving communication and cooperation across the entire organisation. Permits face-to-face problem solving and creates a work setting in which managers with different functional expertise can cooperate to solve decision making problems. (Cummings T.G., 2009)

Motivation that leads to innovation: Likely to result in greater motivation amongst the product team members -usually highly skilled employees- who given the freedom and autonomy the matrix provides, may be able to brainstorm and innovate. (George J. et al., 2012)

Serves two dimensions of interest: The matrix structure is also the most logical choice when two different dimensions -for example, regions and processes or products and processes- are equally important. (McShane S. et al., 2008)

Disadvantages of the Matrix Structure

Issues due to two-bosses reporting: For the two-boss employees it might be common to face goal ambiguity and conflict as their loyalties and capacities are divided between two managers. Often, the demands coming from functional departments and project structures might be conflicting and cause extra stress to employees that are already under heavy pressure of work. (McShane S. et al., 2008)

Difficult to implement: As it may be evident, implementing such a complex structure from a different and most likely much simpler structure requires heavy costs -due to the dual management- and support for the people involved to be able to work around all the inevitable conflicts that will arise. (Cummings T.G., 2009) Additionally, the coordination benefits that occur from working with a matrix structure might take a while before appearing because the new, complex set-up can be very challenging to coordinate. For employees as well it can be confusing, and they subsequently need a long time before they are able to work productively in this structure. (Cummings T.G., 2009)

Accountability becomes unclear: Occasionally, it might not be so clear to distinguish the line of accountability for project teams given the complex nature of the matrix. In a functional or divisional structure, it is always evident who is responsible for

managing a specific department, so there is accountability even for unexpected issues. In a matrix structure, however, the unexpected issues do not get resolved as easily because neither manager might take ownership for them. (McShane S. et al., 2008)

2.3.4 The Process Structure

Also known as the team-based, horizontal or boundless structure, the process structure emphasizes on lateral relationships more than on the vertical with the goal to enhance customer satisfaction, which is the goal. (Galbraith, 1993) Under the process structure, teams are formed around core processes -for example, customer support, sales management, product development- necessary to produce a specific product or service. Each distinctive group of core processes is managed by the "process owner", resulting in a few hierarchical levels compared to other structures. In its core, the process structure has key end-to-end processes -rather than products or functions- which have been simplified by eliminating non-essential tasks but at the same time enriched by combining tasks. The teams that are formed are a key aspect of the success of the process structure. They consist of members with technical expertise and the soft skills needed to make sure that the team is self-managing and maintaining a close relationship with the customers. Performance is rewarded, with more importance given to the performance of the whole team rather than the one of the individual member. An example of the process structure can be shown in the Figure 2.7 below:

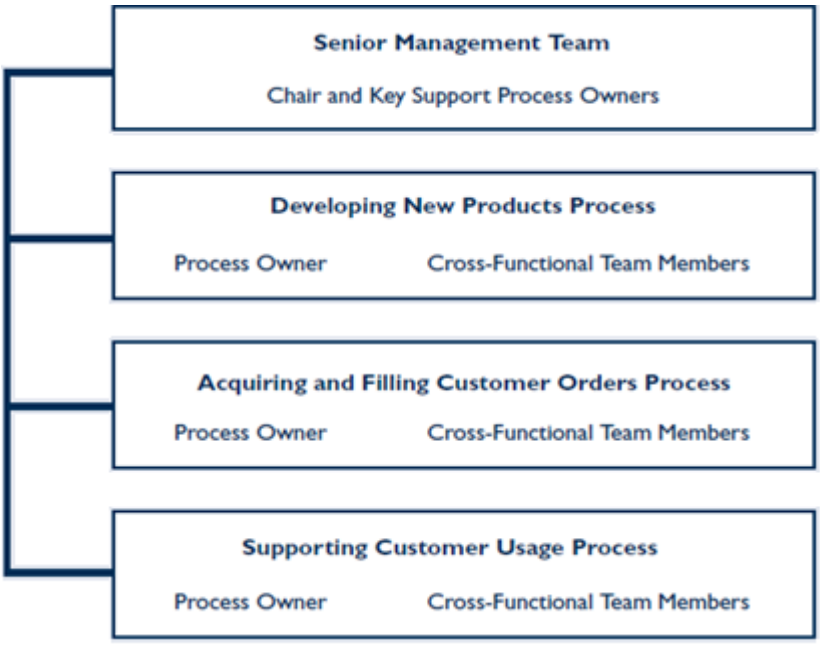

Figure 2.7: Example of a process structure organization (Cummings T.G., 2009)

The organizations that usually choose this structure operate in uncertain and changing environments, are moderate to large, have customer-centric goals and use advanced and highly interdependent technologies. (Byrne J.,1993)

Advantages of the process structure

Focus on customer needs: Most of the resources are dedicated on meeting customer needs. This prioritization results in improved speed, efficiency and as expected, customer satisfaction. (Cummings T.G., 2009)

Improved information flow and coordination: By removing layers of management, information flows more quickly and accurately throughout the organization. Moreover, due to the removal of hierarchical obstacles, coordination, decision-making processes and performance are improved. It is also easier to monitor the total work flow. (Cummings T.G., 2009)

Increased flexibility: Process structures also are more flexible and adaptable to change, compared to more traditional, strictly defined structures. (Cummings T.G., 2009)

Cost reduction: The removal of management layers also leads to reducing the managerial costs of monitoring and managing up and down the hierarchy. (Cummings T.G., 2009)

Employee involvement: By reducing boundaries between departments and giving more power to the teams, job satisfaction is enhanced. (Cummings T.G., 2009)

Disadvantages of the process structure

Difficulties to implement: The process structure cannot be easily applied to an organization as it requires fundamental changes in the working mentality, skills and managerial roles. These changes take time and resistance from existing managers might be expected, as they would need to develop skills in managing lateral relationships. and self-managing teams. (Cummings T.G., 2009)

*Resource duplication***:** A process structure may result in expensive duplication of scarce resources. (Cummings T.G., 2009)

Identification of processes: The success and effectiveness of the structure is greatly affected by the key processes that are identified as the core of the organization's operations in satisfying customer needs. If the processes are not correctly set down, the structure could fail to meet performance standards and customer demands. (Cummings T.G., 2009)

Slow decision making: The teams must consist of highly trained people, able to handle great information flow and quick changes. If teams are not skilled adequately, then this results in slower decision making, especially when a crucial issue comes up. (Cummings T.G., 2009)

2.3.5 The Network Structure

Another structural arrangement that is becoming increasingly popular for organizations operating in highly complex and uncertain environments, using uncertain technologies with goals of achieving organizational specialization and innovation is the network structure. (Cummings T.G., 2009) Under the network structure, the company outsources one or more of their functional activities to other companies, and it is not uncommon for the company to keep only one specific core

competency in which they excel such as product design or R&D, as presented in the Figure 2.8 below:

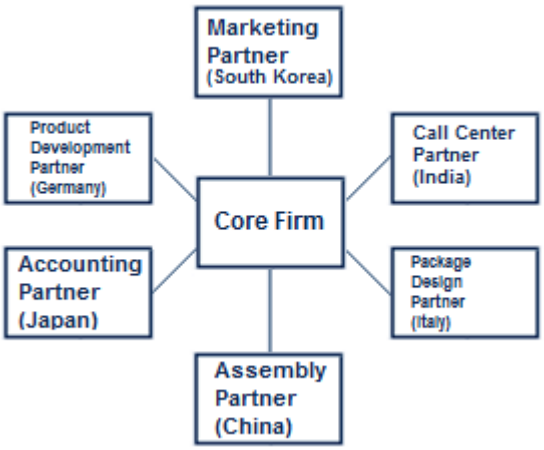

Figure 2.8: Example of the Network Structure (George J. et al., 2012)

By doing so, the company operates in a network of cooperating companies, thus the name network structure. (George J. et al., 2012) Companies nowadays recognize the benefits related to reducing costs and increasing flexibility that comes with outsourcing their non-essential functional activities and focusing on one or two areas where they excel. (George J. et al., 2012)

Advantages of the network structure

Quick response to change: Enables highly flexible and adaptive response to dynamic environments and customer needs. If the customer demands a new product, the company may form new relations with other firms offering the necessary resources needed to make this new product. (McShane S. et al., 2008)

Reducing costs: Searching for partner-companies worldwide that can offer the company the best possible resources at a competitive price, reduces costs and makes the organization more competitive on a global scale. (McShane S. et al., 2008)

Specialization and expansion: The company presents its core competencies as its leverage which makes them recognizable as the best in what they do, offering complete customer satisfaction while giving them on the same time, the opportunity to expand rapidly as there is no need to expand in terms of physical resources as

well, which would be the case if everything was produced in-house. (McShane S. et al., 2008)

Disadvantages of the network structure:

Some loss of control: Regardless of how well a company cooperates and communicates with its subcontractors, it will never reach the same level of control it would have if all functions were in-house. (McShane S. et al., 2008)

Access to confidential information: The company might be forced due to the nature of the tasks needed to be done to share proprietary knowledge with the cooperating organizations, introducing a risk of having some of this information eventually leak to competitors. (Cummings T.G., 2009)

Challenging relations: It can be a troublesome procedure to maintain good lateral relations across other autonomous organizations, as well as motivating members to relinquish autonomy to join the "alliance". (Cummings T.G., 2009)

2.4 Tall and Flat structures

Having already referred to span of control, de- and centralization as factors that affect the organizational structure, it is time to refer to hierarchies. Regardless of how an organization has set up its structure in terms of task organization and coordination, the way this structure is expressed in terms of roles and layers is another issue, related to the hierarchy of the structure. In almost every organization, there is hierarchy, a structure of layers where everyone is subordinate to someone else, in a form of a pyramid, as shown in Figure 2.9:

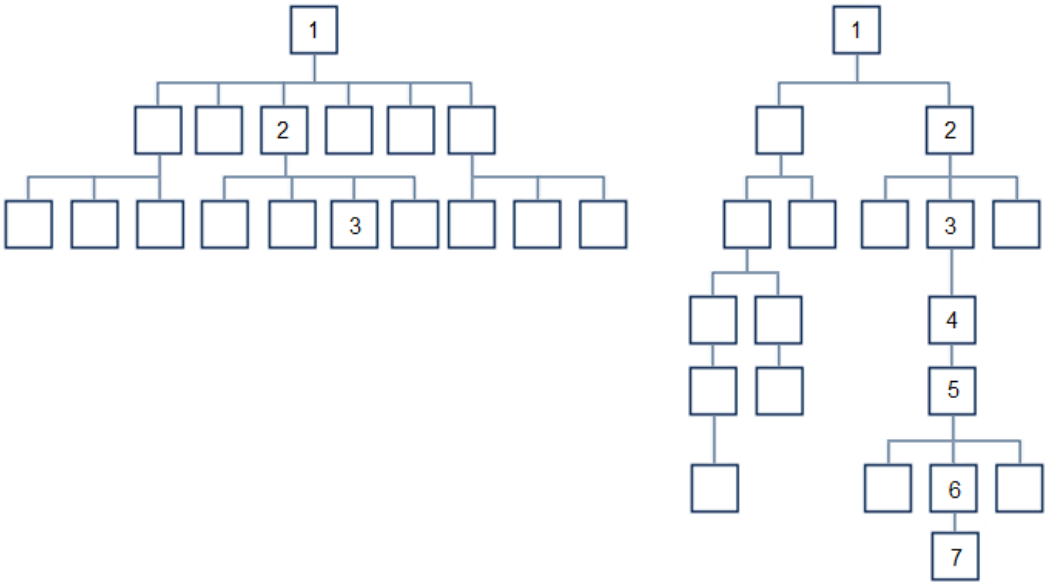

Figure 2.9: Example of a Flat & Tall organizational structure (George J. et al., 2012)

Depending on how many layers there are, an organization can be tall, mid-sized or relatively flat, as pictured above. In the same way it becomes more difficult for an organization to keep for example, its functional structure when it is expanding in other countries as well, it becomes equally difficult to accomplish the same degree of coordination and communication when the hierarchy of the company is too tall. (Morgan, 2014) The high costs that accompany a tall hierarchy, along with the slow decision-making process and the communication problems, have led many companies to analyse their hierarchies to find out if there are levels of management that can be reduced by transferring the tasks of that specific role to someone directly above or below the hierarchy. (George J. et al., 2012) Some models of not-as-tall organizational structures are the following:

Flatter organizations

By eliminating unnecessary layers of hierarchy, the organization gets a flatter structure where hierarchy still exists. There is a strong focus on communication and collaboration and the immediate benefits of taking off some managerial layers are improving the employee job satisfaction and challenging the status quo around traditional hierarchical management models. Instead of completely changing the entire structure and introducing a radical new approach to work, it achieves similar

results in a shorter term and with much less effort and resource allocation. (Morgan, 2014) Of course, for the "flattening out" to be effective, the organization must possess interdependent technologies that allow for maximum collaboration and communication among the employees. However, it needs to be pointed out that as the company expands, it will most likely lead to the hierarchy having to become taller.

Flat organizations

In flat organizations, the number of people directly supervised by each manager is large, and there are few or no levels of middle management between staff and executives. (Ghiselli et al.,1972) Smaller and some medium size companies might be able to operate in this type of an environment but when it comes to organizations with thousands of employees then it becomes challenging, as the manager in a flat organization possesses great responsibility in providing direction, help and support to so many subordinates. (Ghiselli et al.,1972) Flat organizations are sometimes called self-managed organizations, as the employees are supervised less while promoting their increased involvement in the decision-making process.

Latest trends: Flatarchy & Holacracy

New structural models emerge all the time and globalization, disruptive trends in technology and societal changes are some of the reasons for the appearance of new emerging trends. One of those trends is *flatarchy*, shown in Figure 2.10, a hybrid of hierarchies and flat organizations. Flatarchies do not have a permanent structure but they can -very dynamically- switch between small hierarchies or flat formations in an ad-hoc way, depending on the current needs of the company. This temporary creation of internal entities of new structures is becoming popular as organizations -small or big- want to focus on innovation that is not set in a specific department.

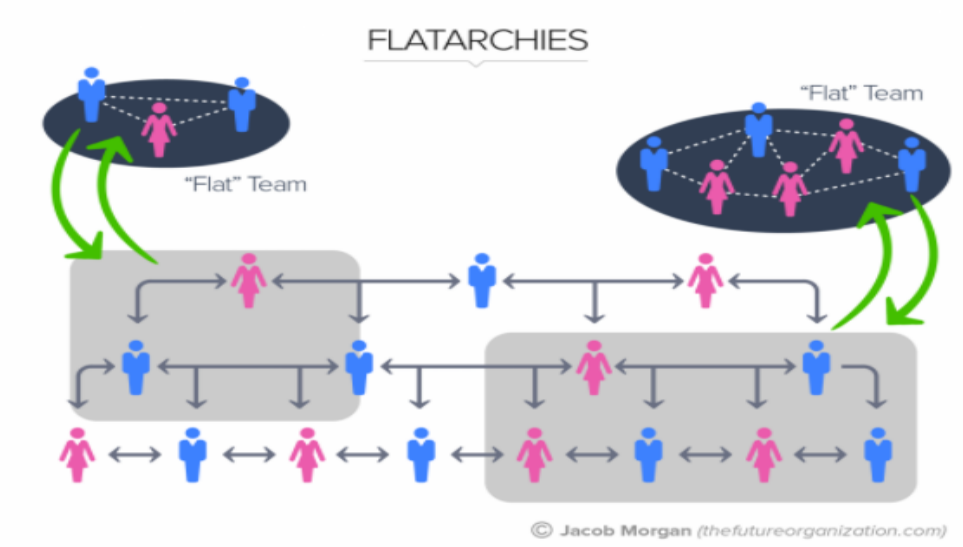

Figure 2.10: Flatarchy (Morgan, 2014)

Another example of an innovative organizational structure, is **holacracy**, shown in Figure 2.11. As Jacob Morgan (2014) in his book, the Future of work, presents it: "The basic goal with this structure is to allow for distributed decision making while giving everyone the opportunity to work on what they do best. There is still some form of structure and hierarchy, but it's not based on people as much as it based on circles or what most people would think of as departments. Information is openly accessible, and issues are processed within the organization during special and ongoing meetings." (Morgan, 2014)

Figure 2.11: Holacracy (Morgan, 2014)

After examining some basic examples of "shorter" organizations and some current managerial trends in the field of new structures, it is important to point out that the issue at hand is not choosing the most innovative, up-and-coming flat structure to base the organization on, rather an issue of choosing the one that best fits the company. Indubitably, reducing layers and decentralizing some of the decision-making authority can lead to rapid decision making, increase in productivity and individual performance innovation, flexibility, more direct communication with the customers and overall improved employee satisfaction. Horizontal structures, as all structures do, have their own implementation challenges as well. Even companies known for their flat structure, were originally created flat and had to adapt as they grew. However, choosing a more horizontal structure just for the sake of becoming flatter, can lead to increased restructuring costs that will not be reimbursed through improvement. Organizational culture can be negatively impacted, procedures and rules that replace previous managerial human positions may be increased and cause an extra working stress for the employees, and the lack of clear accountability and responsibility can easily become an issue.

2.5 Summary and Conclusions:

After a review of some of the expert literature on the various types of organizational structures and all the factors that can impact them, it is evident that there is no one-fits-all solution. Choosing the appropriate structure for each company is a great challenge and a decision that must consider all concepts mentioned in this chapter, such as Division and Coordination of Labour, Span of Control, Centralization and Decentralization, Formalization. Moreover, the organization must remain flexible and always have in mind the contingencies -organizational environment, size, strategy and technology- that can easily change and gravely impact the structure, leading to inevitable changes, such as reducing management layers or transitioning to a geographical divisional structure from a previous functional one. The management of an organization needs to look for practical and scalable approaches when considering removing some of the layers in the hierarchy. Removing middle levels of management and abandoning antiquated formal procedures and bureaucratic obstacles can be an excellent first step but as the company grows, it needs to invest the time and resources necessary to assess possible new set-ups and innovative ideas

on organizing themselves. Especially in the world of IT companies, where the need for adapting to the ever-changing environment is equally important to the development of the product itself, the chosen organizational structure faces the additional challenge of offering the appropriate conditions in which software product is developed. As the problems become more and more complex and the pace of changes increases, traditional hierarchies cannot react fast enough to new market opportunities and cannot offer to the development teams the support they need in order to be cross-functional and agile. As the Agile software development methodologies have become the norm when it comes to creating cross-functional teams that operate flexibly and fast, it should be of interest to examine during this research study how the choice of a specific flatter structure fits with the implementation of Agile.

Chapter 3

Theoretical Background and literature review of Agile

3.1 Introduction

In the world of software development, there is a relationship between a company's structure and its ability to adapt to changes. As already discussed, for example, a company with a matrix structure fits better to an organization operating in a dynamic, unpredictable environment when compared to a organization with a functional structure that needs stable conditions to thrive in. Organizations must be able to change their structure and operations dynamically in order to respond to the ever-changing customer, market pressures and competition. So, how can an organization operate with maximum adaptability and agility in an environment of high complexity?

Implementing methodologies that fall under the Agile Software framework can provide the support needed to bridge the gap between business and development. Agile software development -mostly referred to as Agile- has had a major influence on the ways software development takes place. It is a broad umbrella term that describes a group of software development methodologies that rely on iterative and incremental cycles to build and deliver products, through the coordinated and collaborative work of self-organizing, cross-functional teams. (Elshabrawy, 2012) Although Agile, as it is known today, has existed for the last 15 years, its roots go back much earlier. More than 30 years ago, classical plan-based methodologies for managing software development projects, such as the Waterfall model were the accepted and expected route. (Elshabrawy, 2012) However, even from the early 1990s, as the PC industry started demonstrating its dominance, it was obvious that those approaches were dramatically slowing down the process of releasing the software product. As a result, developers and researchers introduced new lightweight software development methodologies. (Cockburn, 2001) However, these efforts to develop software product in increments and iterations had not been so successful till Agile Manifesto, a set of values and principles describing the ideology of Agile software development for faster and more efficient product development, was created

in Utah in 2001 from a group of software developers. (Kent et al., 2001) Agile and its various methods, such as Scrum or KanBan, are broadly used in the area of software development. Although each Agile practice varies greatly from the others, having its own terminology and tactics, they all follow the same basic principle of abiding to an iterative and incremental process during which all of the aspects of product development run in parallel. (Elshabrawy, 2012)

3.2 What is Agile?

3.2.1 Traditional Software Development Process - Waterfall

To understand the importance of Agile, it is crucial first to understand the usual plan-based approaches that were used in the world of IT that Agile was created to replace. Agile emerged in the early 2000s with the publishing of the Agile Manifesto as an alternative to the traditional Waterfall-style management that seemed to be the root cause of common issues such as delays in the software release process and poor internal communication. Traditional methodologies such as Waterfall are characterized by a sequential series of steps such as requirement definition, planning in early stages, building, testing and deployment, as shown in the Figure 3.1 below. As presented in the diagram, the model begins with establishing system and software requirements and continues with architectural design, detailed design, coding, testing, and maintenance. The basic principle is that a detailed visualization of the final product takes place before even starting working on it, and work according to the specifications of the visualization in order to ensure that all design flaws will appear during the visualization phase. First, the client specifies the requirements that are thoroughly documented. Then, planning of how the general architecture of the software needs to be created takes place and the actual work starts. After testing the product in various ways to ensure it meets the client requirements, it is eventually deployed to the client. (Nabil M et al, 2010) The above process is visualized shortly in Figure 3.1 below. In the past, when even in the world of IT the environment in which the organization would operate in was not as dynamic and unstable as it would later become, Waterfall theoretically was ideal for various reasons: it emphasized on clear requirement specification in advance and on a clear outline of the product before implementing it, and it is clear to understand with specific milestones to

anticipate. However, reality states that in most scenarios, this model cannot be applied. It is usually unrealistic to expect that all customer requirements will be precisely stated with no possibility of changes during the implementation. Equally unrealistic is to expect that all the design specifications will be set in advance with nothing changing during the implementation. Also, the fact that software is delivered all at once and at the end, means that there is no possibility to discover possible serious defects early. Before moving to the next separate phase, the previous one must be documented and formally reviewed, meaning great costs if for some reason there are corrections of the requirements needed to happen. Many people also think that the amount of documentation is excessive and inflexible. (Nabil M et al, 2010)

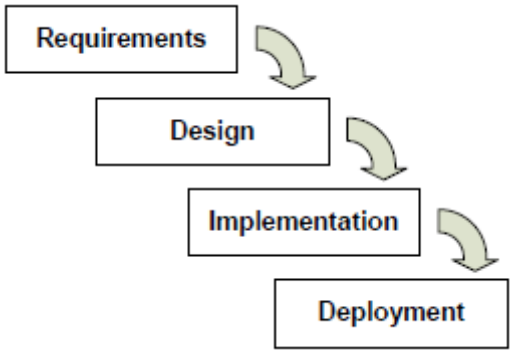

Figure 3.1: Waterfall model (Laman et al, 2003)

3.2.2 Agile Software Development

Agile software development originated as an umbrella term containing many different methodologies such as Scrum, KanBan and Xtreme Programming but eventually evolved into a philosophy about software development, holding specific beliefs, values and practices.

As mentioned in the introduction, in 2001 the Agile manifesto was published by a group of software experts, also known as the Agile Alliance, introducing for the first time the four core values that would allow for quick, flexible and adaptive to change, development of software:

Manifesto for Agile Software Development, as published from the Agile Alliance (2001):

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan (Agile Alliance, 2001):

The *Manifesto for Agile Software Development* is based on twelve principles:

- Customer satisfaction by early and continuous delivery of valuable software.
- Welcome changing requirements, even in late development.
- Working software is delivered frequently. (weeks rather than months)
- Close, daily cooperation between business people and developers.
- Projects are built around motivated individuals, who should be trusted.
- Face-to-face conversation is the best form of communication. (co-location)
- Working software is the primary measure of progress.
- Sustainable development, able to maintain a constant pace.
- Continuous attention to technical excellence and good design.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- Best architectures, requirements, and designs emerge from self-organizing teams.
- Regularly, the team reflects on how to become more effective, and adjusts accordingly. (Agile Alliance, 2001):

3.2.3 Agile Terminology

By using some of the basic Agile terminology as it is described by the Agile Alliance, it is easy to describe in general how the agile development of a software product would progress (Agile Alliance, 2001):

User Stories: In consultation with the customer or product owner, the team divides up the work to be done into functional increments called "user stories." Each user story is expected to yield a contribution to the value of the overall product.

Daily Meeting: Each day at the same time, the team meets so as to bring everyone up to date on the information that is vital for coordination: each team member briefly describes any "completed" contributions and any obstacles that stand in their way.

Incremental Development: Nearly all Agile teams favour an incremental development strategy; in an Agile context, this means that each successive version of the product is usable, and each one builds upon the previous version by adding user-visible functionality.

Iterative Development: Agile projects are iterative insofar as they intentionally allow for "repeating" software development activities, and for potentially "revisiting" the same work products.

Team: A "team" in the Agile sense is a small group of people, assigned to the same project or effort, nearly all of them on a full-time basis. A small minority of team members may be part-time contributors, or may have competing responsibilities.

The team members along with the help of the customer or another managerial position individual will meet with the objective of breaking down the whole product in small, independent segments that can be separately accomplished and shipped to the customer. The team works in specific time frames and by the end of each of those iterations, the specific tasks are accomplished, tested and can be delivered in order to receive feedback from the customer. The product is eventually completed in an incremental way by the end of those iterative loops. The team also meets in a daily basis to discuss the progress and any impediments but usually at the end of each iteration as well. This process is shown in the Figure 3.2 below:

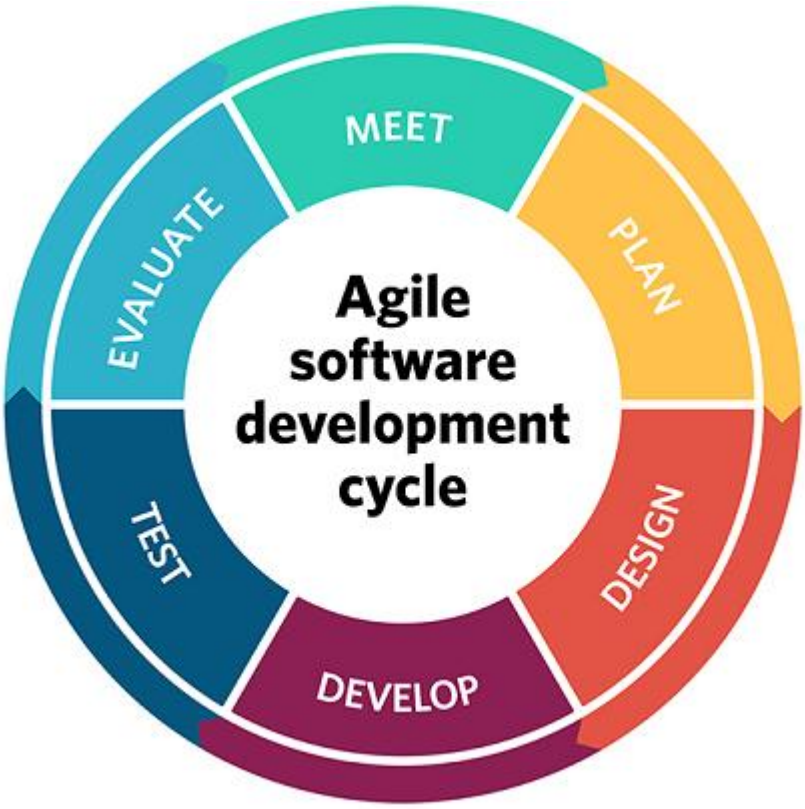

Figure 3.2: Agile Process in software development (Rouse, 2017)

3.3 Agile methodologies

As it has been already established, Agile is an umbrella term containing many different frameworks and methodologies with the goal of achieving the desired level of flexibility and adaptability when developing a software product. Those frameworks can be perceived as process tools that provide the organization with a guideline of how things should be done, to a certain extent. (Kniberg et al. 2010) Perhaps the two most famous and broadly used of those frameworks are Scrum and Kanban which are shortly introduced in the sections that follow.

3.3.1 Scrum

Scrum is an iterative and incremental agile software development framework for managing product development. (Verheyen, 2013) Scrum recognizes that customer requirements will most likely change and that unpredictable challenges will appear. Scrum can be considered empirical as it is accepted that some problems that might appear are not possible to be anticipated or defined from the beginning. In this case,

the team's constant experience and contact with problems in general will lead to better problem-solving skills, improved abilities of adapting to quick changings and increased confidence in working under ever-changing conditions. (Kniberg et al. 2010) Scrum, on its core, uses small teams that are creating small increments of product in short periods of times instead of having one large team that creates a product end-to-end in an extended period. The main points can be shortly described as Henry Kniberg (2007) presented it in his work**:**

- Scrum teams are small, cross-functional development teams, meaning that no technical specialists are added, creating knowledge silos.
- All work tasks are put in a list split down into small, independent deliverables.
- The list, called "Product Backlog" is being sorted by priority and each task is assigned by the team with a relative time effort, in a meeting called "Sprint Planning Meeting".
- Time is split into short fixed-length iterations, called "Sprints", with potentially deliverable code created at the end of each iteration and presented at the "Sprint Review" milestone meeting,
- The code that is created during each one of those Sprints corresponds to tasks, also known as User Stories, selected according to prioritization from the Product Backlog, thus creating the "Sprint Backlog".
- Customer gives feedback after each iteration, helping the team to optimize the rest of the workflow till the completion of the product and focus at the potentially new findings.
- The team meets daily to discuss shortly on the current and future tasks to be done, as well as on any potential problems that need to be addressed and they also meet retrospectively at the end of each Sprint to reflect on how to work better in the future, in a meeting known as the "Sprint Retrospective".
- Team uses various metrics and tools to help them evaluate whether their estimation of the complexity of each User Story and their actual performance was satisfying.
- Roles that are introduced under Scrum and are part of the Scrum Team along with the developers, are the one of the Scrum Master who facilitates the

Scrum processes and eliminates impediments and the one of Product Owner, who represents the customer within the team, defines the requirements and helps the team prioritize their work. (Kniberg, 2007)

The whole process is also presented in Figure 3.3 below:

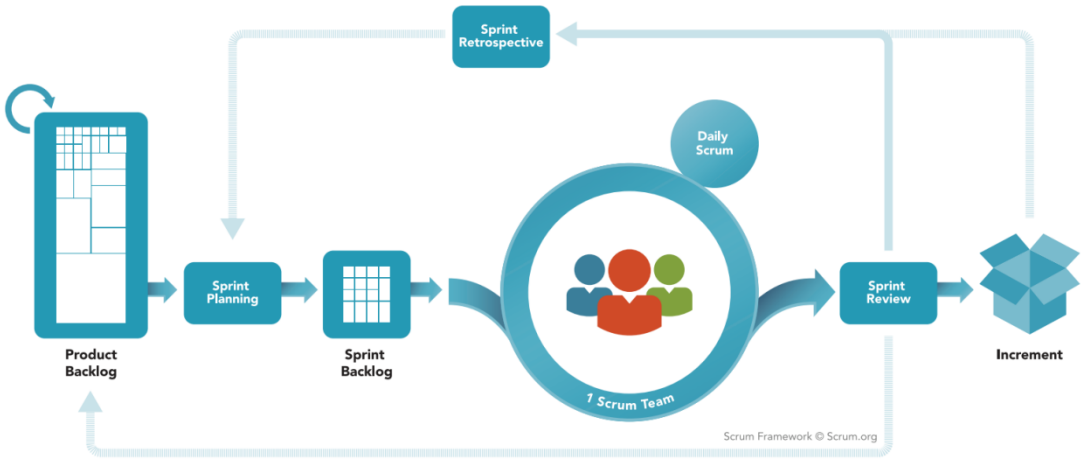

Figure 3.3: Scrum framework (Scrum.org, 2018)

3.3.2 Kanban

Kanban in software development is an approach used for managing and optimizing the flow of information or materials in a process, where the goal is the delivery of a software product. (Klipp, 2014) Kanban is not a software development framework strictly defined as Scrum is, with specific steps to follow, rather an approach to introducing and dealing better with change that occurs inevitably to an existing software development lifecycle or to a project management methodology. The main idea of Kanban is limiting work in progress and adding something new in the workflow only once a current task is ready for delivery or ready to move to the next step of the workflow process. Limiting work in progress assists in reducing waste that would occur due to constant multitasking and task switching. (Anderson et al., 2005) Mapping of the value stream leads to understanding the process and then each stage in this process has a specific limit of tasks in progress. (Kniberg et al. 2010) There are three basic principles related to Kanban:

Visualize Workflow: It is important to visualize the tasks in the process in a way that depicts their state, for example “done”, “in progress” or “blocked”. The

complexity of a process makes the importance of a visual workflow even greater, even though Kanban can be used even for a process with a few steps. In software development, however, there are traditionally many different stages (plan, design, draft, approve, schedule, implement, test, integrate, deploy) that can be visually represented. There are many different ways of visually representing the workflow, such as a traditional board on the wall, or software tools. Some examples of both are presented below in Figures 3.4 and 3.5:

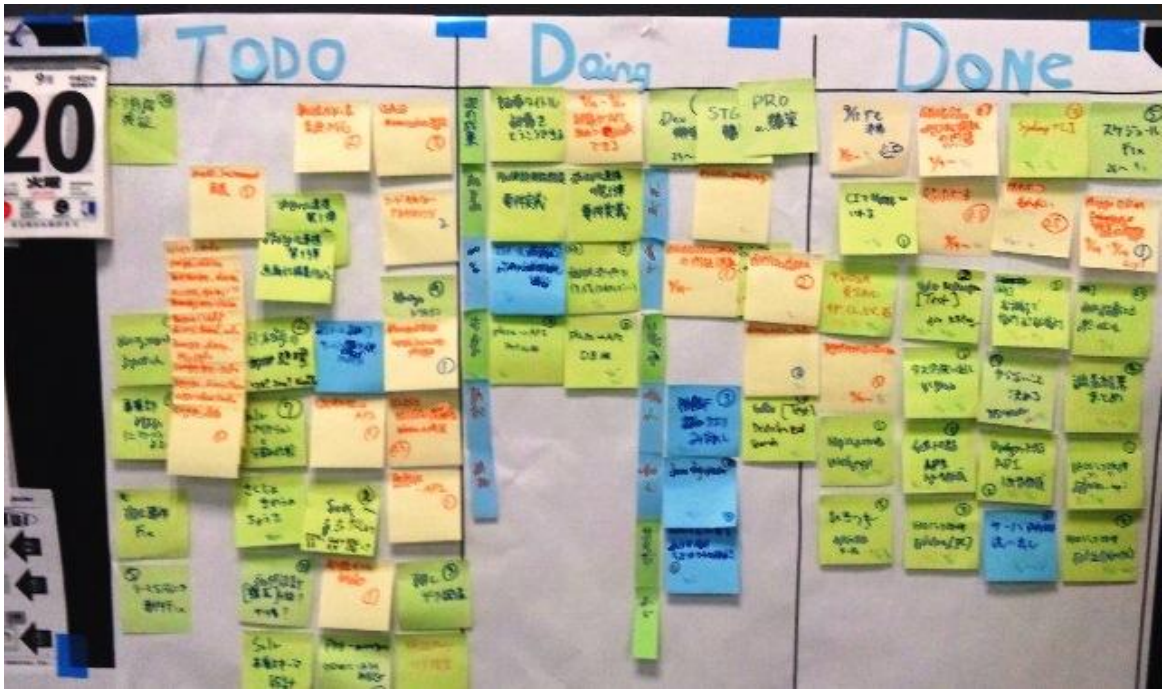

Figure 3.4: Kanban board. Tasks are moving through different phases of completion. (Kanban Tool, 2015)

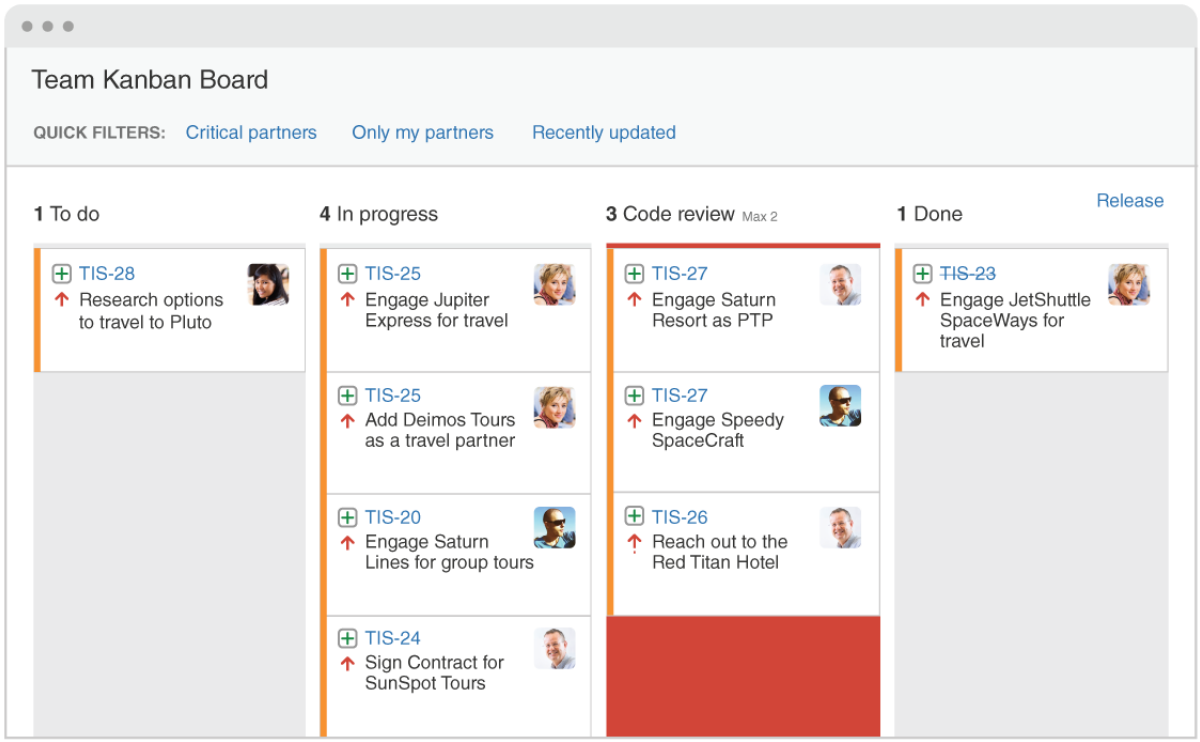

Figure 3.5: Kanban board JIRA software. Similarly, tasks are moving through different phases of completion. (Radigan, 2014)

Limit Work in Progress: Specific limits must be assigned to how many items may be in progress at each workflow state at any given time. (Kniberg et al. 2010) While seeming restrictive, it is a principle that can optimize the workflow as there is a limit to how many tasks can be performed simultaneously without overwhelming the development team and making committing to unrealistic goals. (Klipp, 2014)

Measure and Improve Flow: By estimating the average time needed to complete one item which is also called “cycle time”, the process can be optimized to make that cycle time as small and consistent as possible. (Kniberg et al. 2010) In general, good metrics are needed to make sure that the processes are monitored so they can eventually be optimized and the whole flow can be improved.

One reason behind the popularity of Kanban is that -as opposed to Scrum- it does not need special measures to be implemented. Instead, it can simply be applied to the existing processes, in order to locate areas of improvement and areas that perform already optimally. (Klipp, 2014)

3.4 Common practices of Agile teams

As already established, there are many ways an IT organization can be Agile. However, the vast majority of Agile organizations can be characterized by some common practices that embody the Agile principles and can help the teams to optimize their work without having to reinvent all their processes from scratch. These practices are summarized in Allan Kelly's work, Changing Software Development Learning to become Agile (2008):

Focus on Quality: High quality standards during development assist in significantly reducing faults in the software developed. Less faults equals to less disruptive and time-consuming rework needed, which would demand new testing of all changes, which of course translates to additional costs. Usually, agile software development teams use specific practices to improve quality and to reduce rework:

- *Code Review*, one of the most famous of those practices is where developers review the code written by others to uncover shortcomings. It is a valuable quality-assuring practice, as many times, it is easier for someone else to detect an error than the person who wrote the code.
- *Pair programming,* when two developers work side by side to develop and evaluate the code they are producing, an approach that can be greatly beneficial especially when one of the pair is more experienced so he or she can assist the other developer to learn. Also, this can be greatly beneficial in case a developer wants to work in a new area that he or she is not so familiar with, as by working side by side with another more experienced developer can accelerate greatly the learning process.
- *Automated testing* is used daily to test the system under development and to make sure that changes added did not cause any issues. All changes -after their integration are tested as a part of the system using specific scenarios to do so. These scenarios usually check the basic functionalities of the system.
- *Automated unit testing* is done by each developer on their own code, before the general automated testing takes place. This way, the developer can be more assured that the changes he is about to integrate at the system are functional and will not cause any issues.

- *Test Driven Development (TDD)* suggests that the developer will start by performing a failing test before writing code and then work on making the test pass. This is done by slowly writing the code needed to make the test pass. This assists in making sure that the changes made will be as short as possible and no extra functionality will accidentally be created apart from the one that the developer is asked to create.
- *Continuous integration* dictates that every piece of software developed be integrated to the general system as soon as possible to detect possible issues - with the help of automated testing**,** instead of everything being integrated at the end. Depending on the nature of the software product, integrating the new pieces of code might happen up to several times per day.
- *Quality before features,* where developers have to stop working on developing new features in the case a fault is detected only after it has been integrated in the general system. Development is resumed once faults are fixed, as quality is seriously compromised if new features are developed in the presence of known errors.

Focus on Customer Priorities*:* Agile aims to involve customers as much as possible throughout the whole procedure and all stages of software development, in the form of a product manager or a proxy customer. Agile teams focus on making sure that the customer is responsible for providing and devising the requirements, making clarifications regarding the requirements at any given time during development, helping the prioritization of tasks and accepting the code developed. By making sure that the customer stays involved, the team is better informed about the requirements, potential ad hoc changes are communicated directly and on time and disputes are resolved earlier.

Focus on Design: In traditional software design approaches mostly used in the past, developers theoretically should complete the whole design of the code at the very beginning, based on the original requirements given to them. In reality though, requirements usually change on the way and that causes the code quality to decline as rework and work-around solutions need to be found to meet the original design criteria. Moreover, usually coding would begin before even

design was completed due to lack of time and vague requirements. Agile, on the other hand, takes for granted that the requirements will be vague and the change inevitable and aims to deliver high quality -internal and external- working software while accommodating change as best as possible. So usually, agile teams focus on developing earlier the simplest code that could work given the current requirements. In order to make sure that internal quality will not be an issue causing the external quality to decline, Agile teams use the method of *refactoring*, where existing code is revisited and internally improved without affecting the external functionality with the goal of not fixing a fault but rather making an improvement caused by a change in requirements. Refactoring keeps code flexible which improves its ability to be delivered.

Predictable Time Boxes: In Agile, teams work in short-time boxed iterations, at the end of which software is delivered and feedback from the customer as well as possible changes in requirements are supplied. Those short iterations help the teams to stay focused on the tasks at hand and facilitate easy communication of progress internally and externally.

Focus on Communication: Agile teams focus greatly on internal and external communication. Various visual means are used -as the Kanban board presented earlier- to depict progress and task status so that the information is accessible for anyone to see. Daily meetings support collaboration by communicating progress and impediments. As mentioned already, customer feedback is evaluated and sought for by frequent delivery of software.

Reaching a New Agreement: Instead of having false expectations that the requirements are in their final form in advance and that documentation is complete, both the customer and the Agile organization must come a to new understanding that allows for greater productivity. The customer works closely with the developers, provides them with the resources they need and trusts them to do the best work possible. The Agile team prioritizes work according to the customer's wishes and delivers software as often and early as possible, communicating risks in transparent ways.

3.5 Agile and Organizational Structures

One important aspect to understand is the relationship between a company's structure and its ability to adapt to changes in processes. Agile methodologies and the simplicity behind their main principles imply that they are easy to implement and that they fit any and every company structure, although that is hardly ever the case, when there are so many different organizational approaches and environments. (Elshabrawy, 2012) In older, traditional approaches of software development, the most common organizational structure was some bureaucratic form with high formalization, as opposed to Agile software development which has been observed as a better match for the organic, flexible and dynamic organizations. (Anderson D., 2005) Literature review of the research focused on the field of software development depicts that there has been interest in detecting any existing relations between the structure of the organization and its ability to adapt to new processes. (Schmidt, 2016) However, in most of those studies as they are summarized by T. Dyba and T. Dingsoyr, it is evident that those studies focused on development teams isolated from the structure in which they belong. (Dyba T., 2008) In recent years, however, experts in the field state that one of the considerations that accompany Agile adoption from an organization is that, empirically found, the organizational structure might have to change. (Kumar et al, 2014) More specifically, as an example, it is pointed out that organizations structured around functions have a greater difficulty in successfully implementing Agile, compared to those structured around product lines, due to dependencies that occur within a functional structure. Of course, in the software industry, by "product" may not be something as clearly defined as a physical, tangible product as it can be a shared software platform, a specific service or a new release of a program. That is why it is crucial for the unit of delivery to be established, and after the products have been re-categorized, the company can assign to them an agile team, or clusters of agile teams, that will be in charge for the development and maintenance tasks associated with those products. (Comella-Dorda S., 2016) It is suggested that if possible, the organization makes its structure as simple as possible and centred around specific products handled by stable end-to-end teams, as structural simplicity reduces dependencies -which can pose a great obstacle to Agility- and creates an environment that is safe for running experiments that even if they fail, new solutions can quickly be adopted. (Kumar et al, 2014) According to

LeadingAgile CEO and Founder, Mike Cottmeyer, the first thing to focus on when Agile implementation is the desired outcome, would be to "begin by focusing on your business goals, articulating a strategy to create a team based organizational model, and model based on iterative and incremental delivery principles, one that uses agile and lean methodologies for delivery and governance, but that operates within the operational and cultural constraints of the existing organization and its policies", and then move on to working on organizational culture and practices. A flatter product-based structure, where value flows across teams horizontally and not across countless vertical lines, where it is easier to identify and resolve bottlenecks and capacity and demand are balanced, is a good starting point for the implementation of Agile. (Cottmeyer, 2014) After all, one of the cornerstones of Agile is the existence of small, self-organizing cross-functional teams that can make their own decisions, with less layers of bureaucratic reporting and delaying, as is the case in flatter structures.

3.6 Synthesis of Hypotheses

Based on the literature review provided already, an attempt can be performed to set out and summarise what the experts would predict in the case study findings before we move any further, regarding a small Agile software development company that has progressed from a handful of employees to 50+ in less than 5 years, maintaining its structure as horizontal as possible. As has been already emphasized in the previous and current chapter, a flat organizational structure has a strong focus on collaboration, communication, flexibility and adaptability of the workflow processes of an organization. Apart from that, people are empowered to self-manage their work more, leading to a higher job satisfaction. All these points established regarding flat organizational structures have already been emphasized as the base as well as the environment in which Agile software development practices can flourish. However, also from the literature review so far it has been evident that taller organizational structures are a better fit for expanding organizations as the workload of monitoring and supervising all processes cannot remain the responsibility of few people.

3.6.1 Research Propositions

H1. Participants in software development projects perceive that the need to improve productivity necessitates decreased hierarchical structures in their organisation.

H2. Participants in software development projects perceive that increased hierarchical structures in their organisation cause problems for agile methods as deployed in software projects.

- **H2.1** Increased hierarchy hinders initiative in challenge undertaking.
- **H2.2** Increased hierarchy hinders fast decision-making processes.
- **H2.3** Increased hierarchy hinders employee communication and cooperation.

H3. Participants in software development projects perceive that flat organisational structures are needed to enable agile values to be effective during software development.

- **H3.1** Fast-decision making process is necessary in enabling effective agile processes during software development.
- **H3.2** Allowing people to self-manage their job is necessary in enabling effective agile processes during software development.

H4. Participants in software development projects express high job satisfaction when working in a company with a flat structure.

H5. Participants in software development projects perceive that a flat organizational structure facilitates easy adoption of Agile software development processes (TDD, refactoring, code review etc).

3.7 Summary and Conclusions

Agile in its core advocates quick response to change, adaptive planning, fast delivery times and a focus on constantly optimizing the product and the processes that lead to it. Agile was developed when increase in the complexity of products and in the pace of changes dictated a move towards accepting uncertainty as part of the software development process. Instead of assuming that change can be predicted or even avoided by upfront planning or strict policies, focus was shifted towards forming a closer and more essential communication with the customer. Agile developers, instead of focusing on endless documentation or a detailed design phase, would

rather spend time on the product itself. Agile software development frameworks - most famous of which being Scrum and Kanban- continue to evolve constantly in order to make a better fit to the current needs of the organizations. Undoubtedly, many companies can rightfully claim that they have teams or specific divisions working according to the Agile framework principles. However, having a daily stand-up meeting, a Scrum Master or a Kanban board, does not equal agility in the veins of the organization grid. In practice, agile development involves teams that are self-managing and change-adaptive to a large extent, though agile development is commonly still practiced within a hierarchical organization. However, to this day there is no specific evidence to what is the perfect recipe for Agile in a traditionally hierarchical organization or which is the perfect structure for optimal Agile implementation. What seems to be empirically shown through years of real-life applications, is that simplicity in the structure with a focus on a specific product entity helps create an environment where self-managing teams can operate better. Self-managing is, after all, one of the main benefits of adopting a flatter company structure.

Chapter 4

Methodology

4.1 Introduction

Research methodology is important for meeting the scientific objective of the study. The research objectives and the type of data that need to be collected will determine the specific research methodology that will be used. Conducting a specific study, the researcher should propose a set of propositions that could be formed based on what the expert literature is suggesting, and then analyse the findings gathered from the research to test whether the propositions were accurate or not. This leads to determine whether there are answers to the research questions. This chapter presents the research methodology, study participants and how the analysis of the results gathered was performed. From the beginning, it was decided that this study was exploratory. The study's purpose was to unveil the reasoning behind the structural choices regarding this particular software company, and how do the employees perceive those choices. From the findings perhaps, some conclusion can be reached on the effectiveness of Agile in a horizontally structured organization. It was apparent that this research needed to be approached using a qualitative exploratory method rather than a quantitative one, as a qualitative approach is more suitable for the understanding of a topic where the situation needs to be analysed and understood.

4.2 Research method

The present study is a descriptive survey that collects the required data through questionnaires and informal interviews of the company's top leadership. There are no metrics or variables that are measured or monitored, which would mean that the methodology is purely qualitative but some quantitative interpretation of the results - average, standard deviation- is also performed. Participants in the survey were given a 25-question anonymous online questionnaire to complete. Those questions aimed to cover in a thorough but short manner all major considerations related to the company's structure and the practices used daily, as seen from the perspective of an employee. The questionnaire was designed to provide a closer look at the level of satisfaction the employees feel with the current structure and at the way they

perceive the impact of the structure, as well as the efficiency of the Agile practices that are implemented within the specific structure.

The interviews of the people at the top level of leadership of the company aimed to unveil background information about the company, the reasoning behind the structural choices of the company, the challenges that management faces as the company expands and what are the plans for the future of the company's structure.

A few factors helped greatly in narrowing down the scope of items on which to focus. These points were:

Being part or not of the development team: All company employees are of course a part of the company's structure and have their own opinion on the advantages and disadvantages of the company's horizontal structure. The questions that are related to the Agile implementation of the company are addressed solely to the development team, as they are the ones that are using Agile in their everyday work life.

Personal professional experience of the author: Having worked in software development for seven years under both Agile and traditional (Waterfall) software development methodologies, it was easier for the author to isolate as targeted as possible the most important Agile practices under examination. Moreover, being an employee of the company under study provided an opportunity to extract very specific questions about the benefits as well as the limitations of a simple, flat structure.

Availability of data: Obviously, the vast majority of the data gathered by this study was provided by the questionnaires completed by the employees of the company. However, informal discussions with the top leadership were valuable in filling in the blanks on any misconceptions the author had regarding some processes and challenges faced by the company. Also, the high availability of online resources such as the online Kanban board as well as other project related materials, provided further details and a deep insight on the way of operations. This insight assisted in pointing out more areas of interest.

4.3 Data collection instruments

As already mentioned, the present qualitative survey employed a questionnaire given to the company's employees for data gathering. Furthermore, informal interviews were conducted with the top management of the company, providing more data for the survey.

4.3.1. Questionnaire format

The participants were asked to complete an online anonymous questionnaire (Appendix A) of 25 questions. The questionnaire was deliberately short as more questions might tire the participants, leading them to answer in a random pattern instead of expressing their honest opinion. The objectives of this research study were explained to the participants in advance, as well as given as a statement in the questionnaire itself. The first three questions have a purely demographic value, determining the sex, the group age and the educational level of each participant. The rest of the questionnaire was split into two sections and the participants had to evaluate several statements presented to them and express their disposition towards each statement, choosing among "Strongly Disagree, Disagree, Neutral, Agree, or lastly, Strongly Agree". This is also known as rating based on the five-point of the Likert scale. Table 4.1 illustrates the five-point Likert scale rating:

Strongly disagree	disagree	No opinion	Agree	Strongly disagree
1	2	3	4	5

Table 4.1: Five points of the Likert scale

In both sections, all statements were expressed either positively or negatively to make sure that the participants would not be prejudiced by an exclusively positive OR negative representation of each statement. The first section, containing of 10 questions, was dedicated to statements regarding the organizational structure of the company, the efficiency of that structure and the impression that employees have of it. The second section, containing of 11 questions, focused on the Agile practices

that are used daily and the opinion of employees regarding those practices. As Agile practices mainly include software development techniques, i.e. TDD, the statements that should be evaluated only by the development team had an extra "N/A - I am not a part of the development team" added in the Likert scale to differentiate from people of the testing, documentation etc. teams without depriving them of the opportunity to evaluate statements of everyday practices that are Agile but not necessarily related to code development.

4.3.2 Informal Interviews Format

As already mentioned, people that also offered valuable insight were the people of whom the top leadership of the company consists. The scope of the thesis was described to them and they were helpful in pointing out specific aspects that could be of interest. Discussions and informal interviews were necessary for the details of the "big picture" to be revealed, as well as specific challenges and general observations regarding the leadership's view of structure and practices. This provided the necessary missing information about the structural choices made so far, and the ones likely to happen in the future. A high-level description of how the company handles new projects and customer requirements was provided, as well a description of the most distinct Agile practices used by mostly the development team. This provided a deeper understanding of the level of Agile implementation, an understanding necessary to evaluate whether the structure -at least at a first glance- seems to agree with Agile in general. Identification of challenges relevant to this study were also identified at this point, giving the opportunity to compare them with the challenges revealed by the data gathered from the questionnaire. Most important questions asked, can be found in Appendix B.

4.4 Participants

Participants that completed the questionnaire are a subset of the company's employees, and are part of different teams, i.e from the development team, the team of testers, IT team or team of documentation experts. As the survey was anonymous, a categorization can be performed only between members and non-members of the development team. 21 males and 7 females were included among the people who participated in the survey with varying educational levels and ages. The reason for

choosing to include everyone who wanted to participate was to ensure that many different points of view of the human workforce of the company will be presented, not just those of the members of the development group. Anonymity was also offered to the participants to ensure that they would feel comfortable to express their opinion honestly without the exposure of a normal interview.

The participants that participated in the informal discussions and interviews are the three people -one female, two males- of whom the top leadership of the company consists. They are people of high academic educational qualifications, that have focused in different parts of the company, and more particularly, one of them is the leader of the development team, one of them the leader of the testing team and the third handles mostly customer issues and project negotiations. Even though all provided their insight, the head of the development team provided more details regarding the Agile software development practices of the company, as well as his vision for the future expansion of the company.

4.5 Analysis and Interpretation of Questionnaire Results

The answers of the participants provided, first of all, some demographic graphs that displayed descriptive criteria of percentage, mean and standard deviation. All scale data can be easily handled using the mean, but the standard deviation will provide a measure of spread, revealing -if standard deviation is low- a high degree of agreement among the participants or vice versa. Tendency was also calculated to depict what the general inclination is at each question, especially in cases where the results were neutral and harder to interpret.

Chapter 5

General Observations and Data Analysis

5.1 Introduction

Before proceeding with the analysis of the study results, it is important to understand the background, nature and ways of working of the company. The information presented in the first part of this chapter is a result of personal observation and working experience in this company, as well as a result of informal discussions with the leader of the development team whenever some clarification of the company's environment and practices was needed. At the next part of the chapter, data gathered from all sources are analysed and grouped by main findings.

5.2 Background and context: The Company

The company under study is a software company in the field of telecommunications, that develops an automated testing platform. It was founded in 2012 and started with just 3 people. Today, there are more than 50 employees and the company has an impressive hit rate of projects acquisition (>90%), which is a result of the combination of very high-quality products and of the limited supply of automation solutions in the market. The company consists of a development team and a testing-support team. Development team includes some sub teams, such as the IT support team, the documentation experts team etc. Along with the company's employees at the headquarters in Vienna, the company also has an office in Germany and an outsourcing partner in Croatia. As the development of the products is done exclusively in Vienna, the focus of the survey conducted was on the workforce of this site.

5.3 Background and context: Agile software development

The company is developing its product in an Agile way, using Kanban approaches, as the author knows already from her current professional experience as a testing engineer there. From members of the leadership team it was mentioned that the company operates in an Agile way because Agile characteristics are evident:

“We are Agile. We respond to customer needs quickly and appropriately, we constantly optimize and enforce our workforce with new technology. We are able to scale our capacity quickly and easily in a satisfying extent and we aim for integration with line of business applications”. A member of the development team was recorded saying that “We do Kanban, but it is not in a prepackaged can, we do it the way it makes sense for us”.

As the number of customers is increasing, the flow of requirements is continuous and there is no finite set of features, the management leadership of the company has decided that there is little value in using formal estimation techniques as the ones utilised by Scrum. Kanban and its less formalized procedures that can be customized even more, are a better match for the dynamic environment the company operates in. All features from different projects that belong to different customers are prioritized and defined as requirements. Requirements are then split -with the cooperation of a few senior developers- into smaller tasks, called epics, which are independently releasable. Each epic is then split into smaller user stories by the team leader and with the help of the input of the development team. The cycle time for the completion of each task is monitored and is depending on the person who has undertaken the task.

Every week, a new internal release is created, which is an opportunity to check whether new additions to the code have caused issues or have uncovered the existence of bugs. This process ensures a frequent feedback loop, both internally as well as externally, once the product is deployed to the customers.

5.3.1 Agile practices used

To ensure the quality of the code, several Agile practices are utilised:

- Continuous integration
- TDD
- Unit testing
- Pair programming OR code review
- Refactoring

Another Agile practice that the company employs to make sure all issues are communicated and people are heard, are the daily stand-up meetings where progress is also reported verbally, along with any impediments that may have occurred.

5.3.2 Agile variations and modifications

Agile advocates for the elimination of knowledge silos but in reality, this cannot be completely avoided. The company considers as a better approach to accept the existence of shallow silos that can be beneficial through cooperation and knowledge shared by the "experts". The CEO and leader of the company states that "there is no realistic way to avoid silos entirely. What you can do however, is to keep them as shallow as possible and make sure everyone can benefit from their existence by sharing the knowledge". However, cross-functionality -one of the cornerstones of Agile- is taken highly into consideration by the use of workshops called "hack days" where normal flow of work is interrupted, and all members of the team must cooperate in small sub teams in order to create end-to-end small new projects. Indication of the success of this approach was the creation of an internal tool created to help testers manage in an optimal way the results of the automated testing that was so successful, it eventually was separately launched as a new sub product of the company. Apart from those "hack days" other days are frequently reserved solely for fixing known bugs, which -apart from the obvious quality increase- aim to give to the team members the opportunity to deal with an area they might not be so familiar with.

Instead of using a physical Kanban board, teams use an online progress tracking tool, where all tasks are visible to everyone, as well as their progress and any potential issues that may have arisen. This level of visibility helps team members be acquainted with each other's work and promotes cooperation and easier problem solving as it is easy for someone to find out who has dealt with a similar problem in the past or who might be dealing with a similar issue currently.

5.4 Background and context: Organizational structure of the company

Regarding the company's structure, as the evolution transitioned from a handful of employees to approximately 60 people split between developers, testers, IT, back

office employees etc. that all operate on the same level of accountability and responsibility, leadership considered the only logical structural choice to be as simple and organic as possible, with the goal of remaining flexible and removing communication barriers. This was supported by them saying that "we want everyone to feel free to express themselves. It is not possible to have a complete democracy, the final say in the majority of decisions falls on us, but we want to hear from everyone first".

As the company expanded, no middle layer of management was introduced, which meant that the structure remained flat but the extra responsibilities that would fall upon this middle layer, were split among all product levels.

The leadership of the company views as one of the greater challenges that come along with an expanding, flat organization the increasing difficulty to communicate intentions and the vision of the company. The two CEOs mentioned that "the more we expand, the more difficult it is for everyone to stay on the same page". Also, according to them, hesitation & dissatisfaction has been expressed from some of the employees as the company expanded, as they felt that the company might lose its family-feeling approach. Apart from that, due to the high level of familiarity surrounding such a family-feeling environment, decisions of the top management are frequently exposed to criticism and negative comments: "Sometimes, when everyone is so close to each other, people tend to be quick on making negative comments. We want to have a family-like environment company but sometimes the criticism can be counterproductive for us", stated the leader of the company.

Leadership always struggles to make sure that all ideas are communicated, and people feel safe to give feedback on the way of operations. That is why there are often private meetings between each employee and the top leadership where mutually constructive feedback is exchanged. These are called by the *CEO appraisal meetings,* where "we say to each employee what we think of their performance and we encourage them to share openly with us their complaints, thoughts and concerns". That, in combination with the improved communication, the responsibilities that people are encouraged to undertake and the lack of a complex hierarchy, are some of the factors of flat structure that seem to have contributed to a generally high employee workplace satisfaction, as evidenced from the results of the questionnaire

and from the consistently good feedback the company receives internally from its employees.

As the company operates on high level of personal and direct communication, the need for extra meetings is limited to the aforementioned daily stand-up meetings and to company-wide workshops that occur a few times per year, where people are working in different teams and in various brainstorming projects that help in knowledge sharing and company bonding, as people have the chance to cooperate with colleagues that they do not normally or frequently work with.

5.5 Interview Data

5.5.1 Descriptive Statistics of demographic features of the participants

The following charts 5.1 to 5.3 provide a visualization of the demographic features of the research participants, and more specifically, information on their gender, age and educational degree:

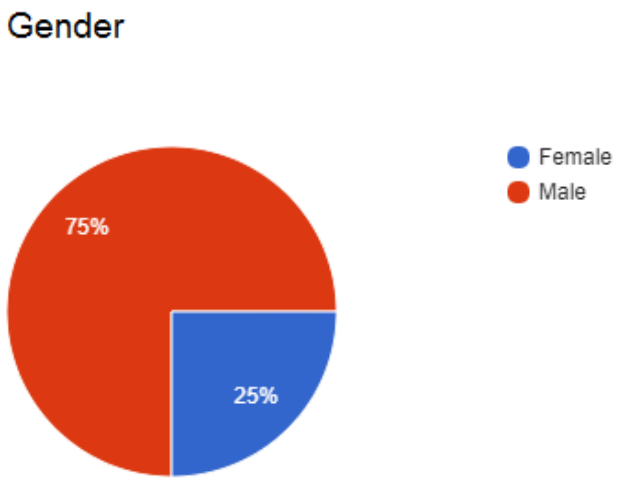

Figure 5.1 : Percentage of the gender variable of the participants

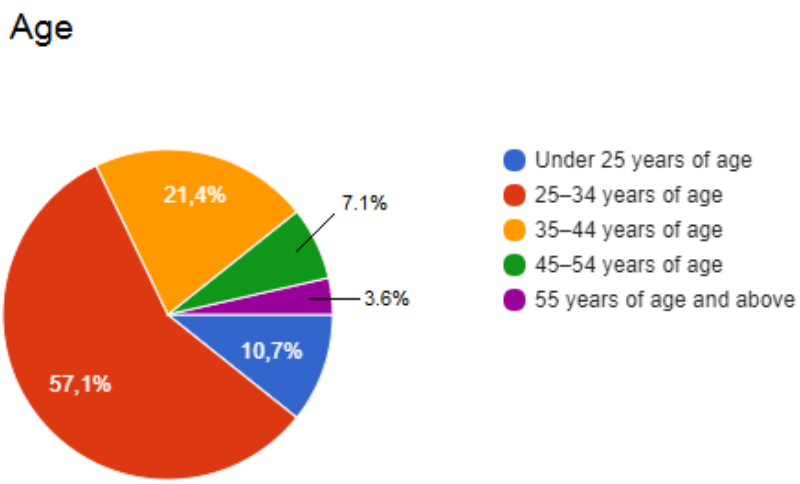

Figure 5.2 : Percentage of the age variable of the participants

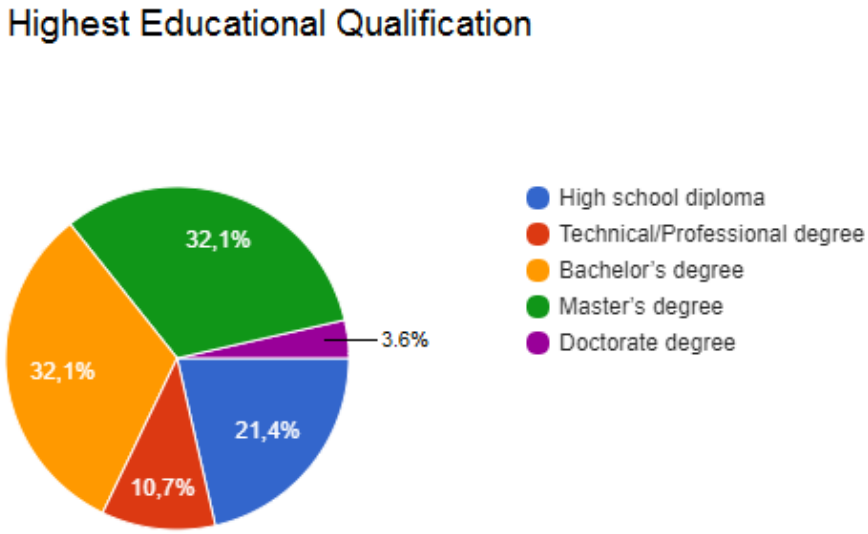

Figure 5.3 : Percentage of the educational degree variable of the participants

5.5.2 Summarization of answers regarding company's structure

Question	Mean of Answers	Tendency	Standard Deviation of Answers
1. The flat structure of the company works to its advantage	3.70	4	0.94
2. Flat structure does not offer career advancement opportunities	2.67	3	**1.09**
3. I enjoy working in a company with a flat structure	4.33	4	0.61
4. Lack of middle management means less bureaucracy	4.00	4	0.94
5. Lack of middle management means easier communication	3.48	3	**1.09**
6. Lack of middle management means easier cooperation	3.47	3	**1.05**
7. Lack of middle management equals more responsibilities for me	3.56	4	**1.07**
8. I view extra responsibilities as a positive challenge	4.25	4	0.69
9. Lack of complicated hierarchy helps productivity	3.70	4	0.81
10. I feel my opinion & concerns are heard	3.85	4	0.93

Table 5.1: Calculation of Mean of Answers, Standard Deviation and Tendency of the answers related to the company's structure.

As it was already explained in section 4.3.1, the value "1" of the participants' answers shows strong disagreement, "2" depicts disagreement not as strong as before, "3" is translated as having no particular opinion rather than a neutral one, "4" shows agreement on behalf of the participant, and "5" indicates a strong agreement.

For the purposes of analysing the results, the "Tendency" column was added to help evaluate the general trend of the participants' answer. Tendency is explained in similar way as above, ranging from "1", which would depict a strong disagreement,

to “5” which is an indication of total agreement. As it can be observed in the table above, most answers have a relatively low (<1) standard deviation, which means that the low spread of the answers is an indication of an agreement among the answers. Answers having SD> 1 have been highlighted and some comments of the participants were noted, in order for the greater spread of answers to be explained:

Flat structure does not offer career advancement opportunities: SD = 1.09

While most participants explained that the flat structure offers them the opportunity to stand out easily and gain recognisability for their accomplishments which can translate to better career advancements, there is some deviation here as some participants disagreed with this statement. More specifically, it was mentioned from a female coder that “when I perform well, it is much easier for that to be immediately recognized and maybe next time this will mean that I will be sent away on the customer demonstration business trip”. However, it was also stated from an older employee of the company that “No other layers means that there is nowhere for me to move up to”. The above quote depicts that some of the participants feel that there are not many opportunities in the company itself as there is no particular professional ladder to climb on officially.

Lack of middle management means easier communication: SD = 1.09

Lack of middle management means easier cooperation: SD = 1.05

The slightly bigger spread of those two answers was explained as a difference between the work processes of development team members as opposed to the testing team members, and the comments made by the participants were similar for those questions. “We can talk straight away to each other, no need for someone to intervene”, was mentioned more than once. This quote shows that development team members need to communicate mostly among each other with no need for someone higher in the hierarchy to intervene or monitor the process. On the other side, testers have to be on frequent communication with the customer base, potentially causing issues in the information communicated. One example from the personal experience of the author as a testing engineer would be a tester that communicates directly with a customer and does not share the information with the

rest of the team, something that not would happen if the communication was handled from higher up.

Lack of middle management equals more responsibilities for me: SD = 1.07
Not all employees share the same level of technical experience, meaning that often more experienced employees undertake more responsibilities and challenges that occur due to lack of middlemen in the hierarchy. A newcomer made the observation that "At the beginning, I was occasionally overwhelmed by my tasks and I constantly had to run to others for help and they ended up doing my task for me."

5.5.3 Main findings related to company's organizational structure

Working on extracting the main findings by combining or grouping the answers of the participants regarding the company's structure, we can isolate the following, considering a statement as a finding when the tendency of the answers is equal to or greater than 4:

Less bureaucracy but not necessarily easier cooperation and collaboration: "Not reporting to several people above you, saves time. Not having to attend meetings after meetings to keep everyone updated, saves time", was stated in similar variations in more than one occasions. The flat structure of the company focuses less on bureaucracy by removing waste from the everyday work processes. There is ambiguity regarding cooperation and collaboration, depending on which employees answered the relevant questions, as it was already explained on section 5.5.2, using direct employee quotes. Apart from the quotes provided, the tendency of the questions 4,5 and 6 of Table 5.1 in section 5.5.2, solidifies this statement.

More challenges and visibility:
The flat structure of the company offers more responsibilities, challenges and visibility to the employees as they are involved more in all processes. Quotes provided in section 5.5.2 provide additional evidence apart from the questionnaire itself, as can be seen from the positive tendency of questions 7 and 8 in Table 5.1, section 5.5.2.

High job satisfaction:

The job satisfaction the employees working in a flat structure is high and they do not perceive the simplicity of the structure as an obstacle in their career advancement. Quotes from employees given in the previous section support this finding. Questions 2,3 and 10 of the questionnaire, presented in Table 5.1, section 5.5.2 are also supporting this finding with their general combined tendency.

High productivity:

The increased flexibility and higher job satisfaction of the company can also be measured by its productivity which the employees consider high, while contributing it to its flat structure. One of the employees stated, on the relation between structure and productivity, that: "The products speak for themselves. I don't think it would be like this if things were more complicated". Tendency of the answers of questions 1 and 9 (shown at Table 5.5.1, section 5.5.2) support that same conclusion.

5.5.4 Summarization of answers regarding company's Agile practices

Question	Mean	Tendency	Standard Deviation of Answers
11. I feel as I can decide which task I want to undertake	3.92	4	0.8
12. I am able to self-manage by job without strict monitoring	3.89	4	0.96
13. I am able to work on several things, gaining expertise in different fields	3.92	4	0.90
14. I believe our company workshops are beneficial	3.34	3	**1.27**
15. I do not see the value of daily stand-up meetings	2.40	2	**1.16**
16. I think we perform adequate unit testing to ensure quality	4.04	4	0.72
17. I wish we worked more with pair programming	3.11	3	**1.31**
18. Code reviews are contributing to quality assurance	4.32	4	0.63
19. Messy code is refactored frequently	3.32	3	0.86
20. I use TDD when I implement new code	3.11	3	0.81
21. Frequent releases help us locate bugs sooner	3.90	4	0.75
22. Decision-making processes are fast	4.04	4	0.72

Table 5.2: Calculation of Mean of Answers, Standard Deviation and Tendency of the answers related to the company's Agile practices.

As explained in the previous table, the answers are ranging from "1", which would depict a strong disagreement, to "5" which is an indication of total agreement.
The tendency column has the same value-to-agreement interpretation as above. Again, the low standard deviation among most of the participants' answers, expresses a general agreement among them. As before, the highlighted higher values of the standard deviation (SD>1) are explained shortly according to the participants' comments:

I believe our company workshops are beneficial: SD = 1.27

This deviation is expected, as obviously not all participants find always value in the company workshops. Depending on the nature of the work that each member usually performs, the workshop material can be less interesting for someone. Direct quote supporting this statement, from a male tester, who during one of the workshops was put in a team that had as a workshop objective the development of a mini educational project out of his expertise area: "I was bored for two of the three days as I did not find it interesting for me".

I do not see the value of daily stand-up meetings: SD = 1.16

"It used to take 10 minutes, now it takes at least 20 minutes for everyone to speak!" This quote from a member of the coding team shows that as the company expands and the development as well as the testing teams grow, the daily stand-up meetings can add a bit of waste in the everyday work processes, as it takes longer for everyone to report on their status. More importantly, it was pointed out that as in the existing teams there are always sub teams working on their own subprojects, it might not be of interest to them to be familiar with what another sub team is currently working on. Quoting from the same member as the previous quote, "I am working on feature X, so I really don't care so much about what the documentation team is working on or about the progress of the people working on feature Y as it is completely irrelevant to what I always do".

I wish we worked more with pair programming: SD = 1.31

This deviation is expected, as obviously there is also a matter of personal preference if someone wishes to work frequently with pair programming. Even though it can be extremely beneficial for less experienced developers, some members with high expertise feel that they could be faster and more productive if they worked alone: "When there are deadlines, I lose so much time working with someone else that everything takes longer", stated one of the senior coders of the company.

5.5.5 Main findings related to company's Agile practices

Similarly, regarding the company's practices, we can isolate the following:

Self-managing employees:
Employees are mostly empowered to choose the task they will work on without strict supervision and monitoring. Specifically, it was mentioned as a comment during an informal talk with a participant:
"I have worked in the past having someone constantly looking over my shoulder and I had to report to him every day. It is nice to feel that they trust you to do what you are supposed to do and manage your work and choose more often than not what you want to work on", male, software developer. The tendency of the answers of questions 11 and 12 in Table 5.5.2, section 5.5.4 verifies this result.

Cross-functional teams:
Employees are offered the opportunity to gain expertise on different fields by working on different areas. This is clear from the answers of the participants in question 13 stated in Table 5.5.2, section 5.5.4 but additional evidence was provided by participants stating that "Hack days and workshops are an opportunity to work on different areas".

Successful implementation of Agile practices:
Agile practices (TDD, pair programming, code reviews, code refactoring etc) are mostly successfully implemented although not all of them in the same degree. For example, on the one hand employees wish to work more with TDD but on the other hand, they are satisfied with the level and value of code reviewing that takes place.

Evidence of that is provided by the consensus of the answers of the participants in questions 15-21 of the questionnaire as presented in Table 5.5.2, section 5.5.4. As stated already in section 5.5.4, the deviation of the answers is explained (quotes from developers were provided in section 5.5.4) mostly depending on personal preferences and level of experience of the developers.

Fast-decision making process:

Employees feel that the company is flexible, adaptable to change and also quick to reply to customer demands, as proved by the answers of the participants in question 22 (can be found on Table 5.5.2, section 5.5.4). One of the senior developers stated on the topic that "We respond fast anytime a decision has to be made!".

5.5.6 Key findings extracted from discussion points of the informal interviews

Split into more companies:

As the company expands and starts losing its familiar atmosphere, management is considering splitting the existing structure into smaller companies, dividing them by different products, instead of choosing a traditional divisional organizational structure. Answering the question whether expanding goes in hand inevitably with a more complicated structure, it was stated from the CEO and founder of the company that "We want to remain as flat as possible in order to keep the job satisfaction and our flexibility high. If we keep growing with the same rate, I am thinking of splitting the company into smaller ones, each one working on a specific product".

Successful operation as an Agile company due to its flat structure:

Company considers itself Agile in terms of flexibility, adaptability, integration of new technologies and quick response to customer demands. This is contributed -by the people interviewed- to the flatness of the organizational structure. The co-CEO of the company stated during our talks that "The feedback that we receive during the appraisal meetings with each employee is consistently good. The feedback that we have from the customers and our high project acquisition rate is also good and keeps

getting better. This structure works for everyone, there is no need to complicate it more unless we absolutely must".

5.6 Summary and Conclusions

From the data gathered from the questionnaire provided to the company's employees and the interviews conducted with the top leadership, many interesting findings were extracted that can be used in the next chapter, in order to investigate whether the research propositions can are supported or rejected, as well as what are the answers to the research questions formulated on chapter 1. Moreover, some possible existing relations between the flat structure of the company and the degree of successful Agile implementation might be revealed. As was expected from the literature review, the company's flat structure has emphasized a lot on improving communication and collaboration among all different parties involved in the development, testing and deployment of the products. People are also able to undertake more responsibilities and to self-manage their work and are generally more involved which translates to a higher job satisfaction and thus, higher productivity. It was also apparent that the company performs well in the field of Agile software development as the general tendency was positive towards the implementation of Agile practices for code development as well as Agile values such as, cross-functionality and fast-decision making. An interesting finding is the future intention to split the company into few smaller ones, by specific products or services, as the expanding continues, to make sure that the benefits enjoyed by the flat structure approach will not cease to exist as the company expands. As expected, there was some deviation among the responses of the people that participated in the survey, but further clarifications were provided to explain the cause of this deviation.

Chapter 6

Discussion of Findings

6.1 Introduction

Ideally, the structure under which a company operates is a perfect match for the needs of the organization and helps in achieving optimal ways of operation and maximum productivity. In the software development industry world, the complexity of the products, the fast pace of change and the high need for adaptability create an ideal environment for the adoption of Agile software development as Agile was created to tackle those issues precisely. The combination of a specific organizational structure with the use of Agile is an interesting topic, one that has not been explored adequately. The data gathered from this study aim to shine a light on the existence of any relations between the two concepts, even though the study is limited to a specific type of structure of an Agile company. The next section of the chapter introduces the limitations of this study. This will be followed by examining the hypotheses, contrasting and comparing them against the main findings extracted in the previous chapter. Moreover, an attempt to answer the research problems will be performed, as well as an examination of whether the research objectives were reached. Finally, it will be attempted to identify and explain any existing patterns and suggestions will be made for future research.

6.2 Research Limitations and Difficulties

Conducting any research is usually accompanied by several limitations and difficulties inflicted by internal and external factors that are not easy to control. The first limitation is related to the research methodology and design used in this study. The study relied mostly on a qualitative approach, which, unlike quantitative methods, is more open to interpretation by the researcher compared to more strictly defined methods. Although a few quantitative variables were exploited, the comprehension of the results was partially based on the researcher's understanding of

the situation. For this reason, it is possible that the findings in this study related to uncovering any relation between organizational structures and implementation of Agile may be based on incorrect interpretations of the data. Another similar limitation is the limited supply of resources focusing on the examining those relations between structure and Agile. Lack of substantial literature on this matter affected the formulation of research questions and research hypotheses, thus affecting data gathering techniques and subsequently, the findings themselves. A third limitation is the limited number of participants in the survey, who are a subset of the company's employees. This establishes the comments and feedback provided by them as limited data to extrapolate generally from on the software development industry. Another limitation of this study would be the gathering of findings that are relevant only to a specific combination of an organizational structure with the Agile framework. Lastly, the data originating only from the questionnaire gauges and the talks with the company's leaders, set another limitation, as the information gathered by them depended mostly on the formulation of the appropriate questions from the researcher.

Apart from the limitations presented above, difficulties also appeared while conducting this study: Originally, it was challenging to detect whether the company of the structure is simply a flat structure, or it could be considered process structure or just a simple structure, as there were similarities with other structures as well. A closer look of the company's practices provided the correct answer, which was important in order to set the appropriate route for conducting the research and gathering the needed information. Another difficulty was the formulation of the questionnaire, as it had to be as short as possible in order to ensure that participants would devote the time needed to complete it. This meant that the questions had to be as specific and targeted as possible, with little room for interpretation on behalf of the participants. Of course, in order for the questions to be formulated, the author needed to develop a deep understanding of implications and requirements that accompany this specific type of structure, as well as a familiarization of all Agile practices that the development team works with.

6.3 Examination of the Hypotheses

Based on the main findings in section 5.5 of the previous chapter, it will be attempted to examine whether the research hypotheses are supported or rejected.

H1. Participants in software development projects perceive that the need to improve productivity necessitates decreased hierarchical structures in their organisation.

Based on the findings presented in 5.5.3 and the evidence provided by the quotes and the questionnaire results in 5.5.2, participants consider the flat structure to be one of the reasons for the company's high levels of productivity. Thus, this hypothesis is supported.

H2. Participants in software development projects perceive that increased hierarchical structures in their organisation cause problems for agile methods as deployed in software projects.

- **H2.1** Increased hierarchy hinders initiative in challenge undertaking.
- **H2.2** Increased hierarchy hinders fast decision-making processes.
- **H2.3** Increased hierarchy hinders employee communication and cooperation.

Findings on 5.5.3 and 5.5.4, and the evidence provided by the employee quotes and the questionnaire results, support the above sub-hypotheses H2.1 and H2.2, as it was commonly expressed that people feel empowered to undertake more responsibilities as well as they feel they are a part of a fast-decision making environment, supported by the decreased hierarchy of the company. However, H2.3 must be rejected as there is not enough evidence to support it. The general tendency of the answers to the questions aiming to reveal the perception of the improved -or not- cooperation and communication within the flat structure was neutral as it depends on whether the participant was a development team member or a testing team member. Thus, no definite conclusion can be achieved and H2.3 is rejected.

H3. Participants in software development projects perceive that flat organisational structures are needed to enable agile processes to be effective during software development.

- **H3.1** Fast-decision making process is necessary in enabling effective agile processes during software development.
- **H3.2** Allowing people to self-manage their job is necessary in enabling effective agile processes during software development.

Findings presented in section 5.5.5 support both sub-hypotheses H3.1 and H3.2. Self-managing employees and fast-decision making are two of the very fundamental processes of Agile and participants feel that the company and its way of operations support those processes, showing that the structure of the company is a key contributor to this fact.

H.4 Participants in software development projects express high job satisfaction when working in a company with a flat structure.

It is clearly supported by the findings in section 5.5.3 as well as by the corresponding quotes of the participants, that the flat structure of the company contributes to the high job satisfaction the employees enjoy. Hypothesis H.4 is thus supported.

H.5 Participants in software development projects perceive that a flat organizational structure facilitates easy adoption of Agile software development processes (TDD, refactoring, code review etc).

Hypothesis H.5 is supported, based on findings of the section 5.5.4, as well as on the quotes of the participants. Section 5.5.3 refers to the spread of answers regarding the degree of use of TDD, but this was explained as a matter of personal preference and not a matter of not successful implementation of Agile coding practices. Thus, the hypothesis is supported as participants perceive the structure a success factor of the total sum of Agile coding practices (TDD, refactoring, code reviewing etc).

From the findings presented in section 5.5.6, extracted from the informal discussions, one new hypothesis can be formulated, based on the literature review:

H.6 Participants in software development projects perceive that the expansion of a company necessitates a taller hierarchy to maintain its productivity.

However, from section 5.5.6, it is evident that the leadership of the company stated clearly that as the company expands, instead of adding more management layers, they would split the company into smaller ones, and each one of these would handle end-to-end a specific product. This is against the theory presented in chapter 2 where experts claim that a bigger company needs a taller hierarchy. Thus, this additional hypothesis is rejected.

Another hypothesis that can be retrospectively created according to the theory of chapter 2 and can be supported by evidence provided by the informal talks, is the following:

H.7 Participants in software development projects perceive that the structural simplicity of a flat company assists in the successful implementation of Agile.

It was revealed by discussing with the company's leaders that they perceive the successful implementation of Agile is largely due to the structural simplicity of the company, as summarized in section 5.5.6. However, apart from this supporting evidence, findings in section 5.5.5 claim that this is the perception of the employees as well. Thus, hypothesis H.7 which was not formulated in advance, is supported.

6.4 Research objective and questions

According to the previous section of this chapter, it is evident that the participants of this study have a generally positive inclination towards the flat structure of the company, considering it a factor that contributes to the high productivity and their high job satisfaction. They also perceive that the structure is a good match for the adaptation and successful use of Agile practices within the organization. The above are opinions shared not just by the employees of the company, but also from its leaders who seem inclined to maintain the structure as flat as possible. At the beginning of this study, the research objective stated was to answer the research questions posed in chapter 1, section 1.5. Using the findings gathered from all data and the hypotheses examined in the previous section, it can be attempted to answer the research questions:

RQ1: What factors informed the organizational structural choices made by members of a software company using agile methods?

The choice of the flat structure of the company was influenced by the fact that the company operates in a highly competitive and rapidly changing environment and market. As a result, it is extremely important to maintain the flexibility and adaptability with which the company must operate in order to remain highly productive. As it was clearly stated by the company's founders and leaders, choosing a flat structure helped in easily integrating new technologies -something very crucial in the world of IT-, and quickly responding to customer demands. As a flat structure emphasizes more on establishing clear and direct communication and cooperation channels, it was evident that this, as well, was one of the factors that informed this particular organizational structural choice. The high degree of job satisfaction that the employees enjoy and the opportunities that everyone has of expressing his or her concerns are additional factors that influenced towards this structural decision, as the lack of formal hierarchies creates an environment where all employees feel that they are equal to each other, thus important enough to be heard. Another important discussion point would be that despite the expectation that leadership should have already considered inevitable to start adding middle layers of management as it was expanding the last recent years, their trust and satisfaction of the efficiency of the flat structure prevented them from doing so and will do the same in the future, as the objective is to keep the simplicity of the structure by splitting into more companies.

RQ2: To what extent did members of the organization in a case study of a software company using agile methods perceive the structural choices made to be appropriate for their business?

As it has been stated multiple times already, Agile, in its core, represents flexibility. Flexibility translates to many aspects of organizational processes: fast decision making, quick response to change, people working on different fields to gain expertise that will help them become cross functional and so on. From the vast majority of the data gathered by this study, it is evident that the general agreement of the employees is that the structural choices are in favour of the successful implementation of Agile. The deviation that was present in a few cases -and was explained by using quotes and comments from the participants in each case- is not a measure of inconsistency of the results, rather than the expected varying perspective of people working in different teams. It is quite reasonable that members of the

testing team would have a different view of some of the practices or view of even the structure itself. The fact that the everyday work practices and processes are not similar between the two teams, impacts the way each team perceives of the agility of the company within a flat structure. However, the general consensus is that the simplicity of the structure aids in establishing the much-needed direct communication, cooperation, adaptability and cross-functionality that are needed in order for Agile software development practices to succeed. It can be thus concluded that the members of the organization perceive the structural choices as mostly and generally appropriate, when it comes to the Agile methods practiced by them daily.

It is evident from above that the questions posed at the beginning of this study were answered to a certain degree, when it comes to the narrow scope of this specific organization. In this sense, the objectives of this study were met. Of course, as the population under examination in this case is small, it is not wise to extrapolate and make generalized statements as this would demand a wider research spectrum, as suggested in the next section.

6.5 Suggestions for further research

The field of studying differences, advantages and disadvantages of the various organizational structures has many theoretical resources and practical examples of real-world companies to support it. However, when it comes to newer organizational models such as flat, flatter, flatarchic or holacratic organizations, there has not been much material covering and examining those newer trends. The same statement is valid also for when Agile comes into consideration: Although the literature in examining the theory of Agile or ways of successfully implementing Agile is abundant, there are not enough sources covering how Agile interacts and integrates itself in a specific organizational structure. This is a gap in literature that this case study hopefully contributed in addressing. This case study could be a helpful tool in the hands of professionals trying to determine whether the flat structure could also be applicable in their own software company. Furthermore, for academics studying the future of organizational structures, a case study of a company and its organizational design can provide valuable information. Regarding Agile, it is always of interest both academically and practically to examine how Agile practices are implemented,

what variations exist and how Agile can be modified to suit the specific needs of a particular organization. Combining the topics of structure and Agile, it is obvious that as the literature examining any correlations is inadequate, any information, especially of practical nature, is important for academics and professionals alike. The study providing combined information on those topics, can give some insight o the degree of Agile implementation and successful use of Agile practices within the scope of a specific organizational structure, in the field of IT world.

As IT companies can benefit greatly by adopting Agile software development methods on the one hand, and IT companies are companies that can be created and expanded quicker than traditional-product companies, it is of great interest for future researchers to study different organizational structures within which Agile can flourish. Suggestions for future research would be the following:

- Examining more IT Agile companies with flat structures in a similar way as this current case study.
- Examining how and why the perception of the degree of successful Agile implementation is influenced by specific teams within a company, as it was shown in this case for the testing team compared to the development team.
- Making sure that those companies cover a wide range of Agile methodologies such as Scrum, Xtreme Programming etc., to examine how well -or not- a more strictly defined Agile framework can fit in a flatter company.
- Expanding the above-mentioned suggestions into IT companies with other structural choices to determine what is the relationship of Agile with taller hierarchies.
- Expanding, finally, even more into non-IT companies, in order to determine whether the nature of the product or service that the company provides is a factor that impacts the extent of successful Agile implementation.

6.6 Summary and conclusions

As it has already been stated, a suitable structure that applies successfully to the needs of the organization will provide context for communication and knowledge generation and transfer. (Silvestri, 2012) Choosing the appropriate structure to fit the

organization or having the ability to adjust it as efficiently as possible when the previous one is not effective anymore, is of crucial importance for the IT world. By focusing on a specific IT company and examining closely its structure, the challenges it is facing as it is growing and any issues occurring when applying Agile methodologies for software development, this study aims to focus on the importance of choosing the appropriate organizational structure and the existence or not of relations between implementation of Agile and said organizational structure. The study revealed that the employees and the company's leadership perceive the flat structure of the company as a factor that contributes to the successful operation of the company. The leadership aims to give feedback power to the employees and to make sure that all opinions are expressed. They plan to maintain the flatness of the structure even as the company expands, to make sure that the communication environment they have cultivated, will remain intact despite the challenges that may occur. Participants state high job satisfaction that stems mostly from the lack of constant supervision and reporting to superiors, the chance to gain visibility and recognition by undertaking more responsibilities and the sufficiency of Agile techniques used. Despite some deviation -that was explained each time it occurred-, the consensus of the participants is that the simplicity of the structure aids in establishing direct communication, cooperation, adaptability, fast decision-making and cross-functionality that are needed in order for Agile software development practices to succeed. One interesting finding is the company considering splitting itself into smaller companies, each one of those handling a specific product, in order to ensure that the hierarchy remains as flat as possible. This is an indication of the extent of the value that the company has assigned to the benefits of the flat organizational structure and is something that can be of interest to academics studying current and future trends in the models of organizational structure and to professionals struggling to make structure-related decisions. In the especially competitive and quickly changing environment of IT world, companies always aim to increase their productivity. The need for agility and adaptability is also great, as software companies are expected to respond quickly to change and be as flexible as possible at any given time. Although each company might have its specific way of arranging their structure and achieving their goals, it can be concluded from the findings of this study that when trying to achieve a high-quality and high-complexity

software product, flatter structures can be beneficial, as this way, middlemen, escalations and endless meetings are reduced significantly as all the interest parties communicate with each other on a day-to-day basis. Of course, this depends on each particular company as the product, the organizational environment, organizational technology as well as the company's objectives and vision are different in each case and this is something that interacts with the structure as well. Morever, traditional organizational models that have been successful over the years, have been rightfully so, as for vast organizations or for organizations with more traditional products, having taller hierarchies might be inevitable. Even in the case of a vast organization though, an attempt could be done to establish locally flatter or product-centric approaches. However, it is believed by some, as stated in Chapter 2, that hierarchical organizations cannot react to new market opportunities and changes fast enough. (Dukabov, 2015) Distributing managers' responsibilities among self-managing, cross-functional teams could assist an organization in becoming flatter and in increasing their overall agility. Increased agility, cross-functionality and flexibility are important steps towards implementing Agile software development practices. Agile advocates quick response to change, adaptive planning, fast delivery times and a focus on constantly optimizing the product and the processes that lead to it. Even though flat structure is definitely not a prerequisite for becoming Agile, it was shown in this case study that it can be a step towards the right direction.

References

- Agile Alliance (2001), available online at https://www.agilealliance.org/agile101/
- Anderson D. and Dragos D. (2005), *From Worst to Best in 9 Months: Implementing a Drum-Buffer-Rope Solution at Microsoft's IT Department*, TOC ICO World Conference November 2005.
- Radigan D. (2014), *Kanban: How the kanban methodology applies to software development*, https://www.atlassian.com/agile/kanban, last retrieved on November 10, 2016.
- Beck, Kent, et al. *Manifesto for Agile Software Development*, http://agilemanifesto.org, 2001, last retrieved on October 10, 2017.
- Burns T. and Stalker G. (1961), *The Management of Innovation*, Social Science Paperbacks, London.
- Chandler, A.D. Jr. (1962). *Strategy and Structure: Chapters in the History of the American Industrial Enterprise*. Cambridge, MA: MIT Press
- Cockburn A. (2001), *Agile Software Development*, 3rd ed., Addison-Wesley
- Comella-Dorda S., Lohiya S. and Speksnijder G. (2016), *An operating model for companywide agile development*, Digital McKinsey, https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/an-operating-model-for-company-wide-agile-development,last retrieved on November 3, 2016.
- Cottmeyer M. (2014), *Some Thoughts on Agile Transformation in big companies*, Leading Agile, https://www.leadingagile.com/2014/11/thoughts-agile-transformation-big-companies/, last retrieved on December 12, 2017.
- Cummings T., Worley C. (2009), *Organization Development and Change*, South-Western Ohio, Cencage Learning
- DeSanctis G., Glass J.T. and Morris I.M (2002)., *Organizational Designs for R&D*, Academy of Management Executive, vol. 16, 55–67.
- Dubakov M. (2015), *Building Flat Organizations with Cross-functional Teams and Fewer Managers*, Interview with Ben Linders for InfoQ, available at: https://www.infoq.com/articles/flat-organizations-teams-managers
- Dyba T. and Dingsoyr T. (2008), *Empirical studies of agile software development: A systematic review*

- Elshabrawy W. (2012) *The Effect of Organizational Structure on the Adoption of Agile Methodologies: A Case Study*, University of Waterloo, Ontario, Canada

- French R, Rayner C., Rees G. and Rumbles S. (2008), *Organizational Behaviour,* Wiley

- Galbraith J.R. (1971), *Matrix Organization Designs: How to Combine Functional and Project Forms,* Business Horizons 14 : 29–40.

- Galbraith J.R., Lawler E. (1993), and associates, *Organizing for the Future: The New Logic for Managing Complex Organizations*, San Francisco: Jossey-Bass

- Galbraith R.J (2012), *The Evolution of Enterprise Organization Designs*, Journal of Organization Design, JOD, 1(2): 1-13 (2012), DOI: 10.7146/jod.6342

- George J.M and Gareth R.J (2012), *Understanding and Managing Organizational Behaviour*, 6th ed., Prentice Hall, New Jersey.

- Ghiselli, Edwin E.; Siegel, Jacob P. (1972). *Leadership and Managerial Success in Tall and Flat Organization Structures*. *Personnel Psychology*. **25** (4): 618.).

- Griffin R. W. and Moorhead G. (2014), *Organizational Behaviour: Managing People and Organizations*, 11th ed., South-Western Ohio, CENCAGE Learning

- Hertel G., Geister S. and Konradt U.(2005), *Managing Virtual Teams: A Review of Current Empirical Research*, Human Resource Management Review 15, pp. 69–95.

- John A. Byrne (1993), *The Horizontal Corporation*, Business Week, December 20, 1993

- KanBan Tool, *An Online Kanban Board*, https://kanbantool.com/kanban-board-online, last retrieved on December 14, 2017.

- Kelly A. (2008), *Changing Software Development: Learning to become Agile*, West Sussex, Wiley

- Klipp P. (2014), *Getting Started with Kanban*, CreateSpace Independent Publishing Platform

- Kniberg H. (2007), *Scrum and XP from the Trenches,* C4Media

- Kniberg H. and Skarin M. (2010), *Kanban and Scrum - making the most of both,* C4Media

- Koch S.A. (2005), *Agile Software Development: Evaluating the Methods for Your Organization*, Artech House

- Kumar A., Castagna F. and Maher J. (2014), *Enterprise Agile Adoption: Challenges and Considerations*, Scrum Alliance, https://www.scrumalliance.org/community/articles/2014/october/enterprise-agile-adoption, last retrieved on December 4, 2017.

- Larman C., Basili V. (2003), *Iterative and Incremental Development: A Brief History*, IEEE (June ed.). 36: 47–56. doi:10.1109/MC.2003.1204375.

- Martin R. (2003), *Agile Software Development, Principles, Patterns, and Practices*, New Jersey, Prentice Hall

- McShane S. and Von Glinow M.A. (2008), *Organizational Behaviour: Emerging realities for the Workplace Revolution*, McGraw-Hill Irwin

- Miner J.B (1982), *Theories of Organizational Structure and Process* (Chicago: Dryden),pp. 358–66

- Mohr, L. B. (1982). *Explaining Organizational Behaviour*. San Francisco: Jossey-Bass Publishers.

- Morgan J. (2014), *The Future of Work*, 1st ed., New Jersey, Wiley

- Nabil M. and Govardhan A. (2010), *A Comparison Between Five Models Of Software Engineering*, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010

- Porter A.M. (1980), *Competitive Strategy New York*: The Free Press

- Pugh, D. S., ed. (1990). *Organization Theory: Selected Readings.* Harmondsworth: Penguin.

- Schmidt C. (2016), *Agile Software Development Teams: The Impact of Agile Development on Team Performance*, Springer

- Scrum.org (2018), What is Scrum?, https://www.scrum.org/resources/what-is-scrum, last retrieved on December 24, 2017.

- Silvestri L. (2012) *The Evolution Of Organizational Structure*, Footnote

- Stroh L.K, Northcraft G.B. and Neale M.A. (2002), *Organizational Behaviour: A management Challenge*, Lawrence Erlbaum Associates

- Rouse M. (2017), *Agile Software Development*, http://searchsoftwarequality.techtarget.com/definition/agile-software-development, last retrieved on September 24, 2017.

- Verheyen G. (2013), *Scrum: Framework, not methodology*, Self-publication

List of Figures

List of Tables

List of Appendices

Appendix A

This part contains the research questionnaire given to the employees of the company:

Dear participants,

This questionnaire is provided to collect data for the master 's degree thesis entitled "**Agile software development within the flat organization. Case study: Evaluating the implementation in a small tech start-up**". You can help us to provide rich findings by giving honest answers to each question. Thank you in advance for your sincere cooperation. Please complete the following information before answering the questionnaire:

1) What is your gender?

Male ☐

Female ☐

2) How old are you?

Under 25 ☐

25-34 ☐

35-44 ☐

45-54 ☐

55 or above ☐

3) What is your highest educational qualification?

High school diploma ☐

Technical/Professional degree ☐

Bachelor’s degree ☐

Master’s degree ☐

Doctorate degree ☐

You are asked to carefully study the following questions and give your answer by choosing one of the options from Strongly Disagree to Strongly Agree.

Regarding company's structure:

Question	**Strongly Disagree**	**Disagree**	**Neutral**	**Disagree**	**Strongly Agree**
1. The flat structure of the company works to its advantage					
2. Flat structure does not offer career advancement opportunities					
3. I enjoy working in a company with a flat structure					
4. Lack of middle management means less bureaucracy					
5. Lack of middle management means easier communication					
6. Lack of middle management means easier cooperation					
7. Lack of middle management equals more responsibilities for me					
8. I view extra responsibilities as a positive challenge					
9. Lack of complicated hierarchy helps productivity					
10. I feel my opinion & concerns are heard					

You are asked to carefully study the following questions and give your answer by choosing one of the options from Strongly Disagree to Strongly Agree. An option "Not part of the coders' team" has been added for the appropriate participants.

Regarding company's practices

Question	**Strongly Disagree**	**Disagree**	**Neutral**	**Disagree**	**Strongly Agree**	**Not part of the coders' team**
11. I feel as I can decide which task I want to undertake						
12. I am able to self-manage by job without strict monitoring						
13. I am able to work on several things, gaining expertise in different fields						
14. I believe our company workshops are beneficial						
15. I do not see the value of daily stand-up meetings						
16. I think we perform adequate unit testing to ensure quality						
17. I wish we worked more with pair programming						
18. Code reviews are contributing to quality assurance						
19. Messy code is refactored frequently						
20. I use TDD when I implement new code						

21. Frequent releases help us locate bugs sooner						
22. Decision-making processes are fast						

Appendix B

This part contains questions of the interviews conducted with the leadership of the company:

1) Can you explain generally how does the company operate?

2) Do you think the company operates in an Agile way? If yes, how so?

3) What is the general process of creating new features, in terms of Agile (Kanban) practices?

4) Which Agile practices are used?

5) Why the application of Agile does not follow strictly the textbook instructions?

6) What modifications that can be considered Agile have you adopted?

7) What are challenges that you have encountered as the company grows and the structure remains flat?

8) Do you think that the flat structure works in advantage of the company?

9) How do you measure the success of the use of the flat structure?

10) What are your future plans for accommodating the growth of the company?

11) Is it inevitable for the structure to change?

12) Do you think the structure and the successful implementation of Agile are related? If yes, how so?