

Agile Project Management Methodology applied in large software development projects

Dissertation submitted in part fulfilment of the requirements for the degree of
Master of Business Administration (MBA) in Project Management.
at Dublin Business School



Jeison Cepeda
10372190

Declaration

I, **Jeison Cepeda**, declare that this research is my original work and that it has never been presented to any institution or university for the award of Degree or Diploma. In addition, I have referenced correctly all literature and sources used in this work and this work is fully compliant with the Dublin Business School's academic honesty policy.

Signed: **Jeison Cepeda**

Date: 26.08.2019

Acknowledgments

I am very grateful to have gained and expanded my knowledge during my Master of Business Administration (MBA) at Dublin Business School. I would like to express my sincere gratitude to my supervisor **Dermot Boyle** for his continuous cooperation, support, and valuable advice during this research on Agile Project Management Methods in large software development Projects. Also, I would like to thank the Project management department, the Research Methods Lecturers and the Library team for their great help and guidance to DBS students in their researches.

My huge gratitude to my **mother, girlfriend, brother, and family**, who have been my support and the reason to keep fighting for my dreams and values in the midst of difficulties and challenges that I have faced since 2013, when I left my country Venezuela. Today, I can say that for the achievement of victory has always been essential to go through the sacrifices and difficulties. Thanks to them for helping me to overcome these barriers and make me believe that everything is possible.

Last, but not the least, I would like to thank my classmates, friends, colleagues and all DBS family for their positive energies and good vibes during this year at Dublin Business School.

Abstract

Traditional project management methodology – being inflexible- was unable to respond to varied and continuous customer requests in a quick changing software development environment. In contrast, Agile project management methodology provides a set of methods that have been successfully applied to small and medium projects, but there is doubt as to the suitability of this approach to larger projects. Therefore , the main aim of this research is to conduct an identification and analysis of benefits obtained when Agile Project Management Methodology is applied in large software development projects. The study also strives to determine if agile methods work successfully on a large scale. Qualitative research was carried out as a multiple – case study. Primary data was collected through five semi-structural interviews to software professionals with huge experience and knowledge of agile methodologies. The main benefits obtained were visibility, Quality, employee and customer satisfaction. Moreover, this paper gives reasonable evidence that supports the success of agile project management methodology in large software projects.

Table of Contents

1. INTRODUCTION	01
1.1 Research Background and rationale	01
1.2 Research Aim	04
1.3 Research Objectives	04
1.4 Research Question	05
1.5 Thesis Structure	06
1.6 Summary of chapter	08
2. LITERATURE REVIEW	08
2.1 Introduction	08
2.2 Agile Methodology	09
2.3 Agile Manifesto	20
2.4 Benefits of agile Methodologies	22
2.5 Agile vs traditional methodologies	25
2.6 Agile at scale	27
2.7 Limitations of agile	32
2.8 Literature conclusion	36
3. METHODOLOGY	38
3.1 Methodology Introduction	38
3.2 Research Design	39
3.2.1 Research Philosophy	39
3.2.2 Research Approach	41
3.2.3 Research Strategy	42
3.2.4 Sampling – Selecting Respondents	45
3.3 Data Collection Instruments	47
3.4 Data Analysis Procedures	48
3.5 Research Ethics	48
3.6 Limitations of Methodology	49
3.7 Summary of chapter	50
4. DATA ANALYSIS AND FINDINGS	51
4.1 Introduction	51
4.2 Qualitative Data Analysis	51
4.2.1 Benefits of APM in large software development projects	52
4.2.2 Limitation of apply APM at large scale software development projects	54

4.2.3 Differences of APM in both small and large software development projects.....	56
4.2.4 Comparative Analysis of both traditional (waterfall) and Agile at large.....	58
4.2.5 Does APM work at large scale?	59
5. DISCUSSIONS	61
5.1 Introduction.....	61
5.2 Discussions of data.....	61
6. CONCLUSION AND RECOMMENDATIONS	67
6.1 Conclusion	67
6.2 Recommendations.....	70
7. FUTURE RESEARCH	71
8. REFLECTION	71
REFERENCES	74
APPENDICES	78
Interview questions.....	78
Table of Figures	
Table 1 Scrum Development process	13
Table 2 DSDM Iterative Development Cycle	18
Table 3 Traditional Project Management versus Agile Project Management.....	28
Table 4 Hub Project Management Structure	30
Table 5 Research Process Onion	39
Table 6 Participant codes	51

1. Introduction

1.1 Research background and Rationale

By the late 1990's it had become evident that traditional project management (TPM) methodologies – being inflexible- were unable to respond to varied and continuous customer requests in a quick changing software development environment (Papadopoulos, 2014, p.455). Therefore, the need arose for a new methodology that could meet the needs of customers (Magdalena, 2017, p.21). In February 2001, a group of 17 representatives from different software methodologies such as Scrum, Dynamic System Development Methods (DSDM), Extreme Programming (XP) and Crystal, met at The lodge at the Snowbird ski resort in the Wasatch mountains of Utah, USA. They met to discuss what their successful projects had in common and the need to implement lighter approaches to the heavyweight and documentation-dependant traditional methods. This group of representatives then formed the 'Agile Alliance' and published their manifesto that outlined the four core values and principles of agile software development philosophy (Larson et al, 2018, p.593). The manifesto focused on the need to respond swiftly and efficiently to evolving changes in the scheduling of small projects concentrating on the area of budgetary requirements, manpower needs and the availability of personnel resources and technological advances (Rizwan, 2012,p.1).

During the 18 years since this manifesto was devised, the Agile Alliance has had a huge impact on software development organisations and has seen them evolve from predictive (such a waterfall) to agile approaches (such as scrum, extreme programming, Crystal, Dynamic System Development Method, etc..) (Turetken, 2016,p.1). These agile methods have been successfully applied to projects in what is referred to as the "agile sweet spot", which consists of "small co-

located teams working on small, non-critical, green field, in-house software projects with easy governance rules and architectures with a very little variation" (Hobbs, 2017,p.3).

However, as the key benefit of agile methodology is its adaptability to unexpected changes that appear during the execution of a software project, this has become very attractive for software professionals, who desire to study and apply agile methodologies in all software project sizes, especially in large-scale software projects, where there have been many challenges, obstacles, and a great level of doubt regarding its benefits (Ahmad, 2014,p.1).

The Fourth Industrial Revolution (Industry 4.0) characterised by exponential changes in the way people live, work and interact with one another has impacted directly on the software sector making it a highly demanding and dynamic area with an unpredictable and ever-changing environment (Masood, 2017, p, 20), which has increased the importance and the necessity to study agile methodologies on a large scale. Moreover, in recent years, the Project Management Institute (PMI) has considered APM as one of the hottest topics, publishing a paper on the subject of scaling agility (Petit, 2016) (Note that this publication presents different research questions and objectives when compared to this research). This publication presents some concerns as to whether software quality has become better or worse (Petit, 2016) which opens the door to consider quality as one of the benefits of this research.

Agile methods in large scale projects have been recognised as a separate and interesting research area for the last six years. There have been multiple case studies conducted on software companies such as Ericsson and Telematicum Inc (a global communication software and service company). Ericsson found that APM correlates with a more balanced use of internal software documentation, an increase in the project visibility and coordination effectiveness

among the agile software team, as well as a possible increase in productivity (Lagerberg, 2013, p.348). On the other hand, Telematicum found that APM could improve both quality and employee satisfaction, however, customer satisfaction was not measured as the software product had been recently released to customers and it was not possible to collect the necessary data to make solid conclusions, therefore it opens the door to consider customer satisfaction as another benefit to study (Papadopoulos, 2014, p.462). These case studies showed just a few of the many benefits that Agile methods can provide to large software projects, therefore, it is crucial for this research to continue analysing in order to be able to determine new benefits as well as supporting or contrasting those mentioned above. Moreover, these benefits will have an essential impact to determine if APM works in a large software project. As Jorgensen (2018, p.01) mentioned in his research, benefits obtained and delivered to the client during the development of a product, such as the cost and time control are key to measuring the success of a software development project.

Other researchers displayed a great interest in assessing APM on a large scale, however, there are some contrasts in the conclusions. Farooq and Masood highlight that a significant benefit of APM is that it provides a cost reduction by decreasing costly changes and reworkings, as well as saving resources by reducing the need for rigorous activities during the planning process (Masood & Farooq, 2017, p.22). Additionally, their investigation describes the fast development of software, along with the improvement of scope and customer satisfaction as other important benefits. In contrast, Ahmad in his research argued that the fast development of the software and scope of projects are not considered clear benefits when applied to large and complex software projects, directly affecting the quality of the product (Ahmed, 2014, p.9). Moreover, Maneva found in his research that large projects require a more rigorous approach than agile can provide, since they involve more personnel resources, and therefore, need a

better coordination and communication process (Maneva, 2017, p .21). This is also supported by Turetken and Trieneskens, who mention that agile methods do not address problems arising from communication and coordination between multiples teams in large projects (Turetken and Trieneskens, 2016, p3).

1.2 Research Aim

Based on the number of issues previously mentioned, the main aim of this research is to conduct an identification and analysis of benefits obtained when Agile Project Management Methodology is applied in large software development projects. The research also strives to determine if agile methods work successfully on a large scale.

1.3 Research Objectives

In order to successfully conduct this research and to determine whether agile methodologies work efficiently on a large scale or not, it is necessary to follow a well-designed structure and specific objectives that will lead towards answering the research question. The objectives for this investigation are:

- To analyse the benefits of applying agile project management methodology in large scale software development projects.
- To identify the limitations faced by software development teams during the agile transformation journey at large-scale.

- To determine if the agile project management methodology works successfully on a large scale.
- To do an in-depth comparative analysis of traditional and agile project management methodologies in the software development industry.
- To understand why to apply agile instead of traditional project management methodologies in large scale software development projects.
- To understand how project management methodology works at large scale.
- To gain a deep understanding and knowledge of agile project management methodology which can help the researcher in their professional development.

1.4 Research Questions

Structuring the most appropriate research questions are extremely important to address the research objectives and achieve the research aim. Two main research questions have been proposed:

- What are the benefits of applying agile project management methodology in large scale projects in software development industry?
- Do Agile methods work at scale/have they been applied successfully to large projects?

1.5 Thesis Structure

In accordance with the objectives defined on this project and research question, the dissertation has been divided into seven chapters.

Chapter One outlines the Introduction. This first chapter provides an overview on the origins of the Agile methodology and how its implementation became so popular in the software development industry. This chapter also provides the rationale for this research and the importance of studying possible benefits of agile on a large scale. Moreover, the research aim, research objectives and research questions have been outlined in this section providing a clear guidance that will be a starting point that will inform the entire development process of this study.

The literature review, Chapter Two, is considered the most valuable section in the research process. The learning obtained within the theory allows one to determine the scope of the research and formulate the interview questions that will aid in the collection of primary data. This chapter summarises secondary resources used in the research. It starts by defining each of the agile methodologies, how they emerged and their implementation in the IT sector. This is followed by a discussion of different theories of the application of agile methods in small and large software development projects including their benefits and limitations as well as a comparison between traditional and agile approaches.

Chapter Three contains the Research Methodology. This chapter defines the research methodology selected in this study; it describes the procedure undertaken to collect and analyse data that confirms whether agile methods can be successfully applied on a large scale or not.

All different sections of the research process such as the research philosophy, research strategy, research design, research approach and primary and secondary data are also defined in this chapter. By using the onion research process where each of the layers are discussed, selecting the most convenient methods for this type of study emerges.

Chapter Four describes the data analysis and findings. In this chapter, the information collected is analysed based on the results and relevant points obtained from the semi-structured interviews that were carried out with software engineers who have extensive experience in agile methodologies. Raw data was processed in order to obtain a better understanding of the interviewee's perspective regarding the topic of the research, while identifying common patterns between respondents that can lead to an accurate conclusion. The analysis has been summarised in this chapter.

Chapter Five presents the Discussion. After identifying and analysing relevant information from interview's respondents, a discussion and comparison with the previous theories and studies based on the experiences of IT professionals with agile methods at scale is described in this chapter.

Chapter Six contains the Conclusion and Recommendations. In this chapter, key points and relevant information are outlined and the research is concluded based on the analysis and comparison of primary and secondary data that was previously collected. A summary of the achievements and contributions made are also outlined in this chapter as well as recommendations and thoughts from the researcher's perspective. Based on the results, this study can act as a reference for future works and as a measure that software development

managers can use to rapidly identify and evaluate any limitation when applying agile methodology in large projects.

Chapter Seven contains the Reflection of the Researcher. In this final chapter, the researcher provides a retrospective or self-reflection of the research journey and research process taken. It describes the benefits obtained from the study of the application of agile project management methodology and how it contributes to the professional and personal growth of the researcher. The knowledge acquired will have an impact on the researcher's career progression.

1.6 Summary of chapter

The research topic is introduced to the reader by providing a background of the agile methodologies and their proven benefits in the development of software projects based on previous studies and theories. Sections such as research aim, objectives and research questions are clearly presented as well as a brief overview of each of the chapters that constitute the entire research. It is necessary to understand how the agile methodology emerged and why its implementation has increased over the six years. For this reason, a wide definition of the Agile Approach, its different methods and benefits as well as the limitations are outlined in the next chapter.

2. Literature Review

2.1 Literature Introduction

This section presents five topics with summary information about agile methodology, agile manifesto, benefits of agile methods, traditional vs agile approaches, agile methods at scale and the resulting limitations

During the first two topics, a description will be given of how agile methodology was originally formed and emerged in response to the regularly changing customer needs that traditional methods could not meet. Also, it will illustrate a brief description of software development life cycle, as well as a description of the most famous agile methods such as: Scrum, Extreme Programming, Crystal, Dynamic System Development Methods and Lean Software Development, followed by the values and common strategies based on the twelve golden principles stated in the Agile Manifesto.

The third topic will discuss the important benefits that Agile Project Management (APM) methodology provides to the development and management of software projects as supported by various research. The fourth topic juxtaposes agile and traditional approaches with emphasis on the defining relevant aspects and situations where agile can be appropriate. This juxtaposition will be followed by the identification of obstacles that may be present during the implementation of Agile approaches in large software projects and possible solutions.

Finally, a discussion of the limitations of Agile methodology in large software projects comparing different case studies presented. This section will help to identify relevant factors within the implementation of the Agile Methodology for the elaboration of the interview's questions and support of findings and conclusions.

2.2 Agile Methodology

Agile Project Management (APM) methodology was encouraged by the famous Toyota Production System in 1940 when the philosophy of lean management emerged in Japan after the Second World War. Post-war Japan's economy was characterized by low demand as mass production was not an efficient system. In this context, the Toyota Company then developed

an on-demand manufacturing system in the 1940s. This was achieved by reducing the size of the lots resulting in an inventory reduction and lower investment in machines, workforce and space. (Layton, 2012, p.67). For this reason, Lean Manufacturing is considered a precursor to the agile movement.

In the 1990s came the so-called "Lightweight Software Development Methodologies", such as Crystal (1992), Scrum (1995), Dynamic System Development Methods (1995), Extreme Programming (1999) Lean Software Development (first applied in software development in 2003), among others. According to Augustine (2005, p.23) APM methodology is a group of methods that originated due to the demanding environment and the rapid expansion of consumer computing. The 90's forced the software industry to create agile solutions, based on incremental development to deliver business value by involving customers and rapidly adapting to their changing needs. Under this philosophy, the requirements and solutions evolve through the collaboration of self-organized multifunctional teams.

The term 'Agile' defines a process in which a team can manage a project by dividing it into several stages. Constant collaboration of members and participants is encouraged in the continuous improvement and iteration of the phases. According to Martin. J (2017, p.39) the objective of an agile project is not to control the execution according to processes and compliance with plans, but to provide the greatest possible value to the product. The agile process begins with the description of the final product that the client expects, its use and the problem to be solved with it. In other words, like traditional approaches all requirements are provided before the project planning is organized and participants are assigned.

Once requirements are prioritized, the team undergoes a process of planning, execution and evaluation. Said process may change the final product so that it adapts better to the needs of the client. Carroll et al (2015, p.14) and Larson et al (2018, p.580) agreed that the continuous collaboration from both the customer representative and project team members during the entire process is fundamental to APM's success. This continuous collaboration yields versatility throughout the development process to deliver the expected end product on time and to subsequently engage the customer with the business. Furthermore, in accordance to this, Augustine (2005, p.30) affirms that this practice drives rapid and reliable customer-value by "enabling feedback, collaboration, self-organization, learning, adaption and continuous improvement".

To describe in detail the development process in an agile environment, Popil et al (2013, p.119) state that the Agile Software Development Life Cycle (SDLC) consists of six phases. The first phase, known as 'pre-project planning', is a phase whereby the objectives of the project and aspects of the market are defined. The second phase, known as "start". This phase's main goal is to understand the problem and solution domain. The third phase, known as 'construction', is focused on the quality of the software and any necessary changes in the requirements. The fourth phase is known as 'end game', wherein the system is on the way to production. The fifth phase, known as 'production' and is focused on keeping systems productive and useful, even when the software is used by customer. The sixth phase known as retirement, is focused on removing the items from production.

According to Layton (2012, p.15) agile approaches are based on an empirical control methods. In the context of software development, agile methodologies have been proven to be effective both for the development of a new project or for the upgrade of existing projects. These

empirical controls involve three important steps such as Transparency; every member or participant know about the progress of the project, Frequent Inspection; the client regularly evaluates the project and the process, and finally, Adaption; any adjustment needs to be made according to the feedback provided by the client in order to minimize problems. To implement the latter two empirical controls, the project needs to be developed in iterations also called sprints. In other words the project needs to be segmented. The project will have the same type of work that is involved in a traditional software development approach but instead of implementing above steps for the entire project at once, they will be implemented by segment as the project progressed.

Additionally, Misra et al (2013, p.972) affirms that the main change from traditional software development approaches to the agile ones is placing less focus on heavyweight processes and documentations and paying more attention on quickly developing products while satisfying clients as well as allowing changes even late in the software development (Misra et al, 2013, p.973).

In the software industry, the most famous lightweight agile methods are described as follow:

Scrum: an agile methodology takes its name and principles from studies conducted on new production practiced by Hirotaka Takeuchi and Ikujiro Nonaka in the mid-1980s. The term ‘Scrum’ came from the rugby approach wherein a cross-functional team “tries to go the distance as a unit, passing the ball back and forth” (Carroll et al, 2015, p.41). This agile methodology is based on transparency, inspection and adaptation to achieve its main goal. Agile employees make frequent inspections of the work done. This enables future adjustments for optimizing predictability and risk control of software. (Maneva, 2017, p.23).

In Scrum, a project is executed in short time cycles of fixed duration. These durations vary from two to four weeks as per Larson et al (2018, p.586). Each iteration must provide a complete account of functional features and advances in the final product that are deliverable to the client via the path of least effort. Each feature of the product can be considered as a mini project and is created according to four distinct phases: Analysis, Design, Build and Test as shown in figure N.1 (Larson et al 2018, p.586).

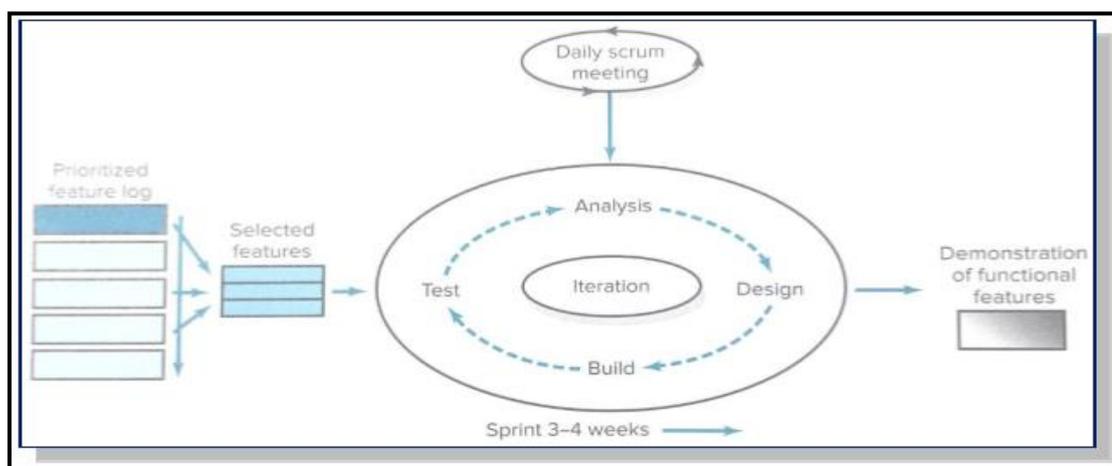


Figure N.1 Scrum Development process

Source: Larson and Gray (2018, p.586).

In the first phase, the team will analyse all functional requirements that are required to complete a feature. The second phase entails the development of a design that meets the requirements previously reviewed. The third phase regards the actual building of the feature. In the fourth and final phase the feature is tested and documented (Larson et al 2018, p.586).

According to Layton (2012, p.73) there are three key roles and responsibilities in the scrum process. Firstly, the product owner or client; who represents the business needs of the project. Secondly the development team; who performs the day-to-day work, they are dedicated to the

project and cross-functional. Finally, the Scrum Master who ensures that the team maintains the focus to meet the objective, facilitate the process and resolves impediments at the team and organization level. The team holds a daily synchronization meeting, which typically lasts fifteen minutes. In this meeting the team will inspect the work that each member is carrying out, identify dependencies between tasks, progress towards the goal of the iteration and identify any obstacles that may impede the objective in order. The latter is done to make adaptations that allow the completion of the task.

Also, there are four key events (meetings) in the scrum framework. The first event is called ‘Sprint Planning’ wherein the product owner and team define all tasks required for the development of the feature in said Sprint. The second event called: ‘Daily Scrum’, consists of the aforementioned meeting the team holds daily to share the progress of the feature and possible problems. Three questions are usually addressed such as: “What the team member did yesterday?” “What the team member will do today?” and finally “What problems the team is facing?” (Layton, 2012, p.74). The third event called: ‘Sprint Review’ takes place at the end of the sprint wherein the team shares with the stakeholders what has been completed within the Sprint. Finally, the fourth event called ‘Sprint Retrospective’, which is an internal meeting held at the end of the sprint to analyse the achievements during the sprint and what possible improvement needs to be made for the next one. (Carroll et al, 2015, p.41).

In the same way Scrum defines four key events, it also defines three tangible deliverables, called Artefacts. Firstly, ‘Product Backlog’ which is the list of requirements and priorities that are expected to be delivered. This is usually documented as user stories that describe the product. Secondly the ‘Sprint Backlog’ which is the list of tasks to be completed in each Sprint. Finally, the ‘Burn - Down Chart’ which shows visually the status and progress of the work in

the sprint. By adding this second chart the progress of the entire project can be shown. (Carroll et al, 2015, p.41).

Conforming to Larson et al (2018, p.588) and in consonance with Carroll et al (2015, p.54), the value of Scrum is that it leads to the implementation of a daily mechanism to quickly inform the team about the state of the project and propose resolution of problems in real time. Its theory is based on an experiential learning circle.

Another approach to product development and particularly to software is **Extreme Programming**, which is focused on improving the quality of the software and respond positively on changes in customer requirements (Maneva, 2017, p.22). According to Carroll et al (2015, p.88). Extreme Programming is designed to deliver the software that customers need when they need it. It encourages developers to respond to changing customer requirements, even in later phases of the development life cycle. It delivers what customers want, at a determined date, in small releases in a frequent manner. This technique was first used by Kent Beck on the Chrysler Comprehensive Compensation (C3) payroll system in 1996, it was named Extreme Programming as the fact that it is said to take best practices to extreme. (Carroll et al, 2015, p.88).

Extreme Programming Methodology also called XP is often used inside the scrum framework for software development due to their similarity. In contrast to Scrum wherein the priorities are defined by the client, in XP the sequence in which the product is developed are determined by the development team. In XP the team works in a strict order of priority as the features of the product are defined by the client and the team must work in that same order. (Carroll et al, 2015, p.41).

XP defines five values: Communication, Simplicity, Feedback, Courage and Respect. XP is mainly based on the first three values. In an XP environment, the communication in the team is key for a successful project, as well as the constant interaction between the client and the development team. The client's feedback and the constant assessment of the work determine if the project is going to the right direction (Carroll et al,2015, p.92).

As well as in Scrum, in Extreme Programming the project begins by collecting the stories of users, which constitute the traditional 'use cases' to describe the requirements. Once obtained these stories the programmes evaluate quickly the development time of each one, this is the first phase of the approach: Planning.

Following the collection of details the second phase is applied, the Design of the project, in XP the design should be as simple as possible. According to Carroll et al (2015, p.92), XP puts special emphasis on simple and clear designs "a good design is a simple design", that not only improve developers performance and delivery time but also reduce costs and complexity. In addition to this, Layton (2012, p.73) affirms that the simpler the design, the lower the cost to modify the software code if customer requests it.

The third phase is the Coding, which is the main product of the system development process as stated by Layton (2012, p.70) "Code not only delivers the solution but can also be used to explore problems" additionally Carroll et al (2015, p.91) affirms that "the code can even be used to illustrate a problem which is difficult to explain". In this phase the product is built. Pair Programming technique is usually implemented, which consist of two people working together on a coding task with one person writing the code while the other review each line of code that

is being typed (Layton, 2012, p.73). Moreover any developer will have access to the code and modify it to accelerate the progress of the project. Finally, the Testing phase, in this phase the team will test that the product works as expected. Several tests are then made to identify bugs and improve the system.

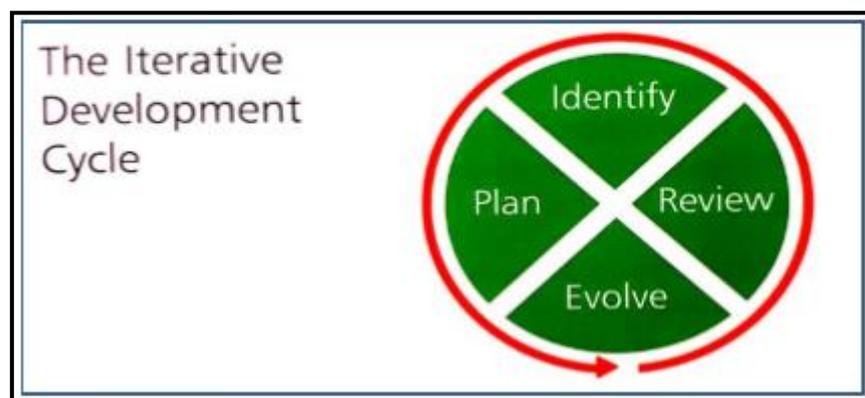
According to Carroll et al (2015, p.97) XP works best with small teams and short project timescales with dynamic requirements and critical delivery date. However according to Layton (2012, p.70), XP has been proven to provide a greater developer productivity by increasing the intensity of the practices along with the efficiency and success of the project development.

Among the methodologies successfully used in Software Development, is the **Crystal** agile methodology. Created in 1992 by Alistair Cockburn and like other agile methodologies, it is based on the communication of all people related to the software project and the frequent delivery of usable code to customers (Girvan et al, 2017, p.14). It is characterized by being focused on the project participants and the reduction to the maximum number of artifacts produced. In a Crystal environment, the development software is considered a cooperative game of invention and communication, limited by the resources to be used. The development team is a key factor. Efforts should be made to invest in improving the development team's skills and abilities, as well as having work policies in team defined.

Although this agile methodology has been effective for small software projects, according to Maneva (2017, p.23), it has not been proven to be efficient for large software project as by involving more people it requires a better coordination. Dikert (2016, p.88) affirms that Crystal methodology is suitable for projects where no more than 50 people are involved.

Another agile framework that provides the delivery of rapid solutions, is **Dynamic System Development Methods (DSDM)**. DSDM was first launched in 1995 by the DSDM Consortium and is based on the same basic concepts of as such as client involvement, incremental delivery and iteration. According to Maneva (2017, p.24), DSDM is mainly focused on early delivery of the project and definition of strategic objectives. Carroll et al (2015, p.62) also states that DSDM provides a framework for developing business solutions in a short time frame while still maintaining quality with an incisive focus on information systems projects that are characterized by tight budgets and agendas.

As well as Scrum and Extreme Programming principles, DSDM involves customers throughout the development of the project and focuses on frequent product delivery. The DSDM development is iterative; all changes during development are reversible and tests are carried out throughout the life cycle of the project (Maneva, 2017, p.24). The iterative development cycle consists of four stages: Identify, which is basically to set the objective; Plan, that defines what needs to be done; Evolve, which is the actual project development according to the plan, and Review which presents the results and feedback.



*Figure N. 2 - DSDM Iterative Development Cycle
Source: Carrol and Morris (2018, p.74).*

Lean software development (LSD) emerged from lean production processes implemented by Toyota in the automobile manufacturing industry in Japan. Since the 1940s, Toyota Production System (TPS) based on just-in-time process has brought great results to the company which allowed it to become one of the world's largest automobile manufacturer (Layton, 2012, p.67).

LSD presents the following seven principles: Elimination of waste, which is possible by reducing delays, irrelevant code and deficient communication; amplify learning, which is possible by sharing learning as frequent as possible with each agile team members and clients; *Decide as late as possible*, delays in the decision making can be beneficial as it allows to make decisions under a deeper understanding of the problem; *Deliver as fast as possible*, delivering as quickly as possible a software is beneficial for the customer, as not only will receive the product within the timeframe, but provide sooner feedback to the development team; empower the team; *Build integrity* and see the whole, by seeing the big picture and keep each member of agile team working to achieve the same objectives (Carrol & Morris, 2015, p.106).

As shown above, there are a variety of agile methodologies. All agile methodologies adhere to the Agile Manifesto and the Agile Principles. This leads to universal characteristics present within agile methodologies such as customer focus, communication, product and flexibility (Layton, 2012, p.15). To have a deeper knowledge of these principles, it will be necessary to present the agile manifesto as part of the literature review.

2.3 Agile manifesto

The agile methodology merged from the need of overcoming some challenges and difficulties that arise during the application of traditional methodologies in software projects. As per

Larson et al (2018, p.593) in February 2001, a group of 17 representatives of the different software methodologies such as Scrum, XP and Crystal, met in Utah, US, to discuss what their successful projects had in common and the need of implementing lighter approaches to the heavyweight and documentation-dependant traditional methods, this group of representatives then formed the ‘Agile Alliance’ and published the manifesto that stated **the four core values and principles of agile** software development philosophy :

“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.”

The 12 principles stated in the Agile Manifesto are formally written below (Beck et al , 2001,p.975):

1. “Highest priority is given to satisfying the customer through early and continuous delivery of valuable software.
2. Changing requirements are welcome, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Working software is delivered frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers work together daily throughout the project.

5. Projects are built around motivated individuals. They are given the environment and support they need and trusted to get the job done.
6. It is believed that the most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. It is believed that working software is the primary measure of progress.
8. It is believed that agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. It is believed that continuous attention to technical excellence and good design enhances agility.
10. It is believed that simplicity – the art of maximizing the amount of work not done – is essential.
11. It is believed that the best architectures, requirements, and designs emerge from self-organizing teams.
12. It is believed that success is achieved when at regular intervals the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

The agile manifesto presents some important and relevant principles characterized by continuously accepting changing requirements. In the final phase of software development, high customer satisfaction, simplicity in development of software, development of motivated people and finally in-person communication (Papadopoulos ,

2014) are given the same level of importance as the speed, the flexibility and the quality of the designs of software projects (Misra, 2012).

It is important to mention that the Agile project management methodology, through its principles, encourages the reduction of unnecessary overhead. This helps to maintain simple software designs. In addition Misra (2012, p.979) affirms that, agile approaches encourage innovation and creativity through a high flexibility in the roles and responsibilities during software development. Finally, this approach maintains that agility is reinforced by good quality designs.

2.4 Benefits of Agile Project Management Methodology

The APM methodology emerged to cope with the strong criticisms to which the traditional methodology (waterfall) was subjected (Carroll and Morris, 2015, p.16). These criticisms were mainly related to customer satisfaction, project quality, project cost and duration. Therefore, software organizations due to highly competitive and growing markets, have different reasons to apply this methodology. be able to achieve their objectives. However, it is not clear yet how beneficial it is for large project and large traditional organizations, originating a great doubt and the reason of this research. This section is focused on assessment of the benefits of the agile methodology in the software development industry.

2.4.1 Higher Customer Satisfaction

According to Maneva et al (2017, p.29) Agile methods are focused on delivering a high-quality software in shortest time possible and with a continuous improvement based on customer feedback. In order to meet new customers' requirements during the software development, this methodology has. Therefore a high degree of adaptation driven by customer feedback are causally linked with a higher degree of customer satisfaction. Additionally, Carroll et al (2015, p.16) points out that in agile methodology, the customer is involved and engaged in making

decisions and setting priorities throughout the project. Therefore, customers will be committed to the final product as it will meet their requirements while increasing their satisfaction.

2.4.2 Better Quality

During the implementation of APM methodology, the continuous testing of the product and the customer's feedback are keys to not only obtain an excellent product but also develop what the customer really needs (Carrol and Morris, 2015, p.16). Additionally, Layton (2012, p.310) highlights that taking a proactive approach to quality prevents product problems and embraces a sustainable developments. These are some important things that an Agile teams need to do to contribute with the high quality of the product.

2.4.3 Project cost reduction

The reduction of project costs is one of the most significant benefits of agile methodologies over traditional ones. According to Layton (2012, p.320) the budget required for the development of an agile project is directly proportional to its duration. Since agile projects are faster than traditional projects, these can also cost less.

Agile methods are designed to respond and adapt to unexpected changes. Therefore, when the need arises to make a change or adjustment in the software specifications, the agile team has the ability to handle it without massive rework. This minimises costly changes and reworks (Farooq and Masood, 2017, p.22).

Finally, Farooq and Masood (2017, p.22) posit that a large percentage of traditional methodologies do not meet budget objectives. While Agile approaches decrease this percentage significantly, which supports the theory that agile costs less than traditional.

2.4.4 Project timeframe reduction

The agile methodology significantly increases the probability of delivering a software project on time and at a much quicker rate than traditional methodology (Layton, 2012, p.319). This benefit can be achieved mainly through the use of short – term scope, plans and design. This instils versatility in project and also cost saving. Moreover, it permits the elimination of last-minute changes in the software project and promotes continuous improvisation with minimal documentation. This crucial for the agile team to save time and deliver the software in the shortest possible time (Masood and Farooq, 2017, p.22).

2.4.5 Increased Creativity and Innovation

APM methodology works in an environment that supports creativity and productivity. This approach fosters the creation of flexible, open and informal team structure, which allows greater interaction of ideas and knowledge leading to higher innovation (Massod, 2017, p.22). Moreover, this approach indicates the need to give autonomy to each member of the agile team, which improves morale and work commitment, enhancing creativity and innovation during the project development.

Although agile methodologies have shown flexibility in software development projects, most benefits have been achieved within sweet spot projects, therefore it is important to assess these benefits when applying them in large software development projects which according to researchers shown in the Agile Limitations section of this paper, remain debatable as their effectiveness may be challenged as the project grows.

2.5 Traditional vs Agile project management Approaches

Traditional and Agile approaches are methodologies mainly applied in the software development industry. These methods involve activities and processes that are predisposed to errors. Therefore it is essential to choose the most appropriate model, as well as to test and validate each software before moving to the production phase (Stonica, Mircea and Ghilic, 2013, p.74).

Traditional project management methods in the software development industry were based on principles from the construction sector (Carroll and Morris, 2015, p.22). These traditional methods focus on planning, executing the plan and taking corrective action or deviation from plan such that there is a high probability of success in the project. Therefore thorough planning in the beginning is key and fundamental in these methods (Larson and Gray,2018, p.580). During the planning phase, the project scope is defined via goals, deliverables, features, functions, tasks, deadlines and costs. Every detail of the project is established through the Work Breakdown Structure (WBS). These traditional methodologies have the characteristic of identifying and evaluating the problems and risks pre-emptively (Larson and Gray,2018, p.580). However if the project scope and/or technology is not totally known, the project could have a high degree of unpredictability and plan-driven methods suffer.

The most common traditional methodology is Waterfall, which follows mainly five logical and sequential phases. Firstly, it begins with the strategy phase, through the development and agreement of requirements. Secondly, the analysis phase, which establishes what would have to be done to meet those previously agreed requirements. Thirdly, the design phase, which is focused on how it would be done. Fourthly, the build phase, which is focused on creating the solution. The final phase entails the client testing and approving the software as per (Carroll

and Morris, 2015, p,22). This methodology is considered progressive, since it is based on the waterfall principle, this is, the water flows from highest altitudes to lowest altitudes and once it has reached the lowest point, the water cannot return to a higher point. In other words, during the development of software after moving on to the next phase, there is not option to go back to the previous stage.

The development of software following the phases shown above (Waterfall Methodology) assumes that essential requirements can be defined upfront. If customers require changes, this could cause project delays or cancellation (Larson and Gray, 2018, p.580).

After evaluating both traditional and Agile methodologies, it will be necessary to compare these approaches that will contribute to the conclusion of this research:

In general, Traditional methodologies are based on predictive approaches, while agile methodologies are based on adaptive approaches (Stonica, Mircea and Ghilic, 2013, p.70). Additionally, traditional methods were created to operate in a predictable zone, while agile methods live in an unpredictable zone, where the scope could present some changes or not be well defined (Larson and Gray, 2018, p.580).

According to Masood et al. (2017, p.21) one of the most distinctive advantages of Agile over traditional project management methodologies, is a higher level of flexibility, which has a positive impact on the project due to the “embracing changes, delivering part of functionality incrementally, and reflecting and learning continuously”. Additionally, Takeuchi (2016) mentions that agile presents better benefits over traditional in terms of employee’s performance, customer satisfaction and productivity. On the other hand, Papadopoulos

highlights (2014, p.459). The main advantage of waterfall or traditional methodology over agile methodology presents an easy transition from allocated and distributed teams which is possible as waterfall has a clear structure for controlling the activities.

In traditional methodologies, teams follow a detailed program with a list of tasks and activities that they must complete during the entire life of the software development project. Whereas agile teams do not have a detailed program, but clear future activities related to the feature that must be developed. Furthermore, Stonica, Mircea and Ghili (2013, p.70) posit that agile teams, by following a more flexible structure, have as a main advantage the ability to adapt to dynamic changes in the product requirements. Unlike in traditional approaches where conforming to Massod (2017, p.21) for being methods focused on developing the software according to the planning status of the project, disable the option to adopt changes.

Additionally, in Agile environments, stakeholders are continually involved in the development and installation of the software Which allows them to have a higher interest and motivation in the development of the product, unlike traditional methodologies (Massod, 2017, p.23).

A brief of differences between traditional and Agile project management Methodology are displayed in the follow table 2.

Traditional	Agile
Design up front	Continuous design
Fixed Scope	Flexible Scope
Deliverables	Features/requirements
Freeze design as early as possible	Freeze design as late as possible
Low uncertainly	High uncertainly
Avoid change	Embrace change
Low customer interaction	High customer interaction
Conventional project teams	Self-organized project teams

*Figure 3: Traditional Project Management versus Agile Project Management.
Source: Larson and Gray (2018, p.582).*

The traditional approaches are known for being scalable to large projects. In contrast agile approaches are more suitable for small and medium software projects. According to Larson et al. (2018, p.592) most of Agile Project Management Methodologies, especially scrum, are perfectly suitable for projects that can be carried out by small teams between five to nine people. However, these methods can be also applied on large – scale projects through a condition called “Scaling”, which presents integration as a chief challenge. Moreover, Stonica (2013, p.70) highlights that there is not enough evidence of agile success on a large scale, supporting the theory that agile is more suitable for small projects and the great doubt about agile at scale. Therefore it is imperative to assess Agile Project Management Methodology in large projects and the necessity of including Agile at scale as a part of the literature review.

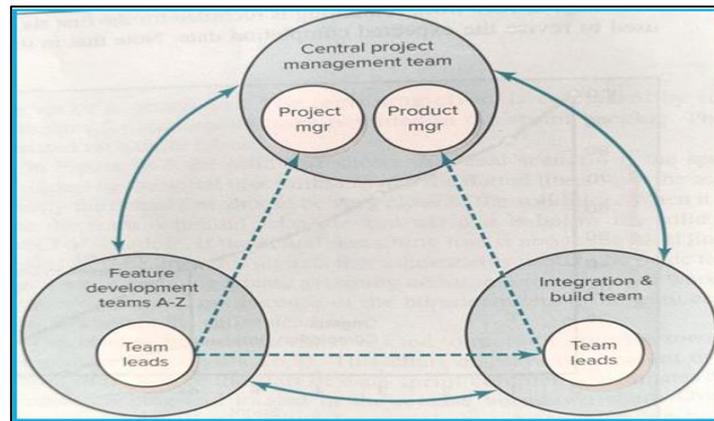
2.6 Agile at scale

Nowadays, information flows at great speed and processes require greater dynamism. In this regard, traditional management methods have become obsolete. New techniques and resources must be used in order to meet changing customer requirements, however there are software companies that are reluctant to implement agile methodology in their usual processes and still using traditional methods. This preference will always lays on whether the project needs to follow a more structured plan, in which requirements are unchanging, or the project is considered extremely risk averse, this reluctance increases when implementing Agile methods at large scale and may be difficult to manage. Therefore a further analysis based on empirical evidence is required to identify the benefits that agile methods brings to large software development projects.

Prior to exploring the elements of agile at scale, it is important to understand that a large or complex software project may be measured by budget, line of code or employees required. According to Rizwan (2011, p.2), if a project has over 100,000 lines of code, it will be considered as a large project, which requires a large amount of working time and employee's dedication. Moreover, Dikert et al (2016, p.88) asserts, projects that need 50 or more employees including all project personnel, or at least six teams, can be considered as large-scale.

During the development of large-scale software, the term of integration arises as the chief challenge of scaling. For this research, it is very important to clarify both the term of integration and scaling. Integration is about making sure that the different features being created work in harmony with each other. While scaling is when "several teams are working on different features of the project at the same time" (Larson and Gray, 2018, p.592). In order to find a solution to the integration challenge, the agile project management method needs to manage the interdependence of different features through significant upfront planning called "Staging" (Larson and Gray, 2018, p.592). Wherein efforts and actions are coordinated to ensure compatibility.

Additionally, Larson and Gray (2018, p.592) present the Hub Project Management Structure as a suggestion to overcome this issue (see figure X below). This pertains to the roles and responsibilities during the development of large projects, showing different feature development teams and one independent integration and build team. Such a structure addresses the integration issue through testing and establishing requirements for the several feature development teams. There is a central management team conformed by the project manager, who represents the several feature development team. The product manager, who represents customer needs. This new team has the responsibility of coordinating multi-team structure.



*Figure 4 : Hub Project Management Structure .
Source: Larson and Gray (2018, p.5 92)*

As the project scales, it becomes very difficult to hold meeting with all members of the agile teams that participate in the large software project. Therefore, Papadopoulos (2017, p.458) highlights in his research, that a solution to overcome this difficulty is to conduct sprint planning. This entails daily stand-up and retrospective meeting individually per team. Also, the Scrum Master is responsible for summarizing the acquired feedback and use it as input on the agile team level.

During the application of Agile principles to large projects, different types of conflicts associated with culture, structures and the overall development process may arise (Petit and Hobbs, 2017). An example of this, according to Lagerberg et al. (2013, p.351) an internal software documentation challenge was faced during the development of two large software projects at Ericsson Telecommunications Company; each project used different approaches, Project A used a waterfall approach while Project B used scrum approach. A survey conducted to members of both projects resulted that, a 58.3 percent of members spent a daily average of 15 minutes working on internal software documentation in project A. While in Project B, a 43.5 percent of members spent a daily average of 15 minutes. Although traditional methodologies have been considered to spend more time on documentation than Agile

methodologies, surprisingly, according to this study, Project A (Waterfall) spent less time on documentation than Project B (Scrum). This result may be caused due to general Ericsson policies and directives that affected the software documentation in Project B. (Lagerberg et al. 2013, p.354)

Although there was some challenges and limitations during the implementation of agile approach in the development of large-scale project at Ericsson, a significant number of benefits have been also obtained. It is important to mention that, this case study consisted of two large software projects that were developed using two different methodologies. This allowed the identification of possible advantages that agile methodologies had over traditional methodologies in the development of large software projects and possibilities for the company to replace their usual method.

Project A served as a control and as such was developed using classical waterfall methodology, as mentioned before and which also included few agile practices, while Project B was entirely developed with Scrum Agile Methodology. Both projects were developed simultaneously during a period of one year between 2012 and 2013, project A consisted of 420 members and 15 teams which 56 percent were seniors developers with over one year Agile Software experience, and project B consisted of 120 members and 14 teams which 37 percent of the members had the same level of expertise as members in project A. The result of the Agile implementation in project B can be summarized in an increase of communication between different functional roles. Which resulted in a higher knowledge sharing where members gained more insights from other functional areas as well as an increase in the project visibility, coordination effectiveness and productivity, members were aware of the status of every task distributed within the different teams that subsequently generated a higher intra-team and inter-

team coordination effectiveness, unlike project A members who considered not to have awareness of other member's advancements (Lagerberg, 2013, p.352-353).

As mentioned before that the benefits and challenges in adopting agile methods in software development vary further with the necessity of applying this methodology in large projects (Oktaay, 2017). These variations depend on the changes in budget, schedule, resources, technology, requirements and teams (Qureshi, 2012), which can be addressed through an agile methodology that deals well with changes (Ahmad, 2014). According to Papadoupoulos (2014, p.459) a study conducted at Telematicum Inc, a global communication software and services company, concluded that one of the most relevant benefits of applying agile methods while developing a large software project was the flexibility of dealing with changes. The study consisted of the development of a new product called 'TeleWeb' by using Scrum Agile Methodology and the re-design of an existent project called 'Smart Client' by using the traditional Waterfall approach. It was conducted to reduce the development timeframe and provide flexibility which by implementing traditional methodologies were quite difficult to achieve. The project was developed in a period of 10 months and consisted of 100 members divided in several agile teams (Papadoupoulos, 2014, p.459). Changes requested by product owners during the project development were made directly and without reopening phases or re-analysing requirements as in traditional process which affects not only delivery times but also development costs.

Other benefits obtained, was the employee satisfaction which was measured through a survey conducted to employees. A majority of members agreed to have achieved a higher satisfaction when using scrum agile methodology, keeping in mind that these members had a long experience using traditional methods, their answers also indicated that team's collaboration

was improved with agile methods. (Papadopoulos, 2014, p.461). Additionally, Papadopoulos (2014, p.461) concluded that the delivery time of 'TeleWeb' was improved by the reduction of time wasted on tasks, also an improvement of quality was achieved due to the flexibility that scrum provides when allowing frequent tests or revisions with product owners and team members, that subsequently reduced costs in rework.

Furthermore, in concordance with Papadopoulos (2014), a study conducted at Cisco Cloud and Software IT (CSIT), Pandey (2015) affirmed that quality improvement and employee satisfaction were the most evident benefits acquired from agile implementation. Two large and complex products called 'Cisco Subscription Billing Platform' (SBP) and 'WebEx App for Samsung', were developed by implementing agile methods and moving to Scaled Agile Framework (SAFe) to simplify and accelerate development process. The SBP was delivered on time with all planned capabilities, quality was improved through the reduction of 40 percent of defects as well as employee's satisfaction which increased by eliminating the need of employee overtime and extra meetings. Moreover, on the 'WebEx' App product, a 25 percent of the quality assurance defects were reduced and the development process was accelerated by doing a daily check on the code which allowed developers to review new features earlier in the process (Pandey, 2015).

2.7 Limitations of agile

Although Agile approaches has been proven to bring numerous benefits to the product development in smaller projects, its benefits and implementation in large projects remain unclear. As mentioned before that one of the advantages that APM offers to agile teams is the adaptability to changes according to Masood et al (2017, p.24). This technique may be a

double-edge sword that may lead to a significant increase in cost and time by encouraging excessive changes to the project in every development phase, changes that sometimes cannot be differentiated by the team whether is needed or if it is a discretionary change. In addition to this, Ahmed (2014, p.07) states that some critical tests can be missed with the continuous change of the project and that can lead to a code break due to frequent builds that will bring more challenges to the test and subsequently an increment of costs and time.

However, according to a survey conducted by Jorgensen (2015, p.182) to participants at a seminar on management of software development projects in Oslo, Norway in March 2015, concludes that a 64% of respondents agreed to have acceptable or successful experiences in controlling cost in large software projects. Moreover, a research conducted by Saeed et al (2015, p.72) where data was collected from 51 different sources distributed in conferences, journals and thesis to detect possible limitations of Agile approaches, states that only 14% of researchers declared a budget overflow when using agile at large scale.

Another factor that limits software development teams in implementing Agile methodologies in terms of the scope of the project when it comes to monitoring and controlling. The lack of documentation and the absence of predetermined standards and high-level requirements. According to Masood et al (2017, p.24), as agile teams tend to reduce the documentation work, one's ability to evaluate the team's performance and determine whether the project is going to the right direction or not decreases. As such more challenges to test the final version of the product emerge. Additionally, Dikert et al (2016, p.98) states that "large development projects demand high-level requirements management", which need to be elaborated to be useful by agile teams affirming in his research that these requirements refinement were reported as challenges. On the other hand, according to Ahmed (2014, p.05) Although it is likely to skip

some details when large teams are distributed leading to a misunderstanding of the project due to a gap in the documentation, these misunderstandings can be reduced and improved by providing users stories, scenarios, use case diagrams, etc. in a globally accessible backlog, furthermore, Saeed et al (2015, p.72) concludes in his study that only 24% of researchers agreed that the documentation is a limitation in the development of large scale software project.

According to Larson et al (2018,p.594), some companies have succeeded using Agile approaches. While others have been gradually introducing Agile in their programs and others still struggling with the scaling challenges, as an example Larson states that in 2004, Siemens Medical Systems started with one scrum team and it was progressively increasing until 2007 where 97 scrum teams were reported.

Likewise, other possible limitations can be defined in general terms as: lack of training, adaptability of the development team to changes, poor communication in large or numerous teams and distribution of work, as an example of this, Larson et al (2018, p.594) states that a survey conducted to over 3,900 software and financial representatives. The results were that 94% percent of the respondents affirmed to have used Agile approaches, however 53% were reported as successful, and the main reasons of the failure were due to lack of experience, company culture and lack of management support.

2.8 Literature Conclusion

The literature review presents evidence and arguments about the important of assessing the benefits of agile project management methodologies in large software projects.

Agile methodology originated with the purpose of responding to the needs of customers that traditional methods were unable to meet. Flexibility and adaptability is one of its greatest abilities. Communication between developers or employees in general, and feedback from customers are present in every phase of the project even late in its development (Misra et al, 2013, p.973). The result being, it is much easier to carry out modifications as the development of the product progresses. This is unlike traditional methodologies that only allow to make changes in the first phase of the project.

The agile methodology is clearly beneficial, when it is applied to small software projects. However, the benefits in large projects are not yet proven. This warrants further research. The most common benefits of this methodology include cost reduction, customer satisfaction and timely completion, which have been supported or contrasted with a qualitative research in the software development industry.

The most prominent traditional and predictive approach, is waterfall, which is basically a document driven model in which proper sequence is maintained (Popli et al,2016, p.118). Whereas Agile approaches, based on adaptive approaches presents Scrum, Extreme Programming, Crystal, etc. as the most used. Scrum works with small tasks called sprints which allows to optimize and to control the development of the project from beginning to end, it uses four types of inspirations: Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective (Maneva, 2017, p.23). Extreme Programming is mainly focused on improving the quality of the software project (Maneva, 2017, p.22). Finally, Crystal which is focused on the communication between teams and customers through the entire development life cycle (Maneva, 2017, p.23)

These methodologies were implemented by several organizations such as Ericsson, Cisco and Telematicum to provide empirical evidence that can determine whether the full implementation of agile methods in large software projects brings positive results or not. Based on the results obtained from the Ericsson case study, Largerberg et al (2013, p.353) concluded that there were strong evidences through the entire project which demonstrated that agile methods can be beneficial in large scale implementation. Moreover, according to the results delivered during the development of the two different products using agile methods at CSIT, Pandey (2015) concluded that there was a positive impact in the product quality and employee satisfaction. Additionally, the Telematicum case reaffirmed quality improvement and employee satisfaction as main reason to apply APM in large Scale software Project.

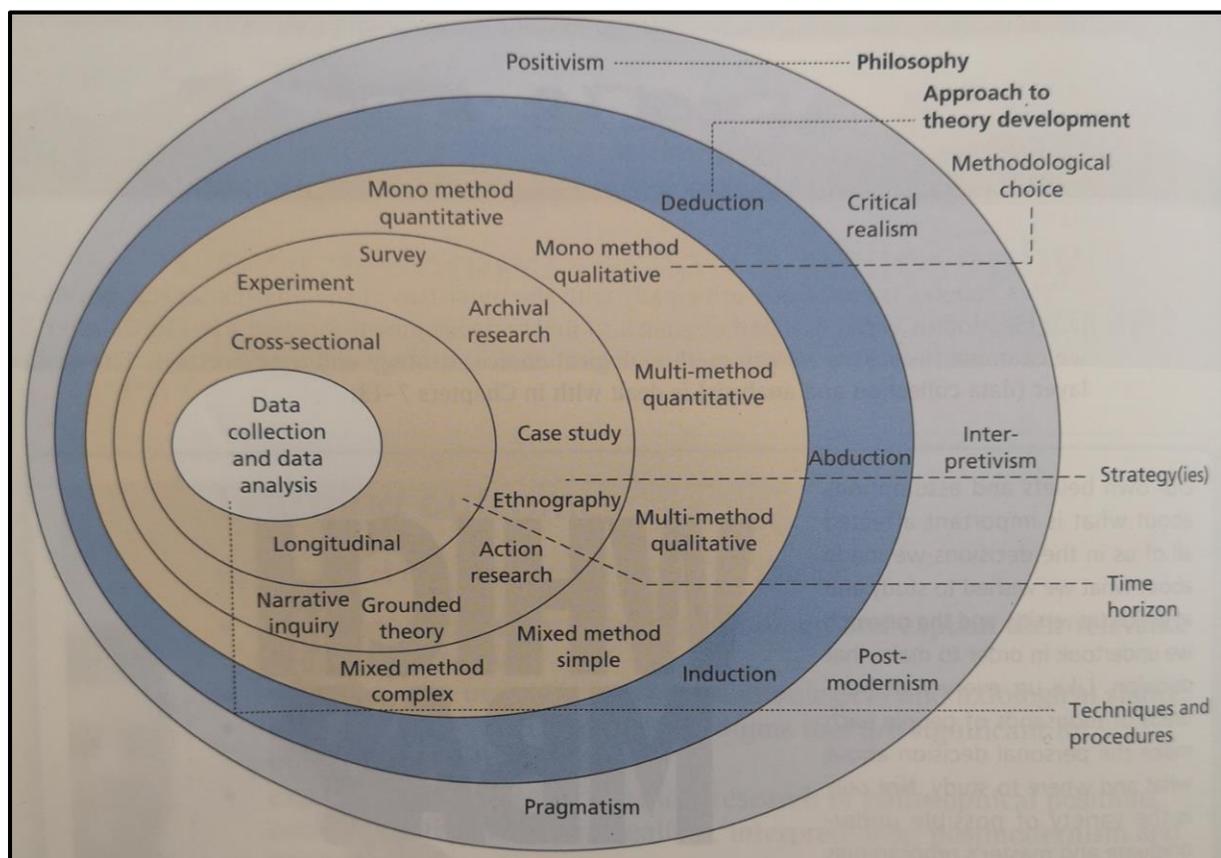
Although there was some evidence of APM benefits at large scale, it is recommended to carry out a further evaluation of possible benefits and limitations that every company has depending on their policies and usual work structure. As stated by Papadopoulos (2014, 462) software companies with long history of applying Traditional methods need to carefully plan activities and ways to overcome possible limitations.

3. METHODOLOGY

3.1 Methodology Introduction

Given that the area of investigation of this research is the analysis of benefits that APM Methodology brings to software projects and possible limitations of its implementation at large scale, a set of related data can be collected and which will require specifics techniques or procedures to be processed and analysed, therefore this research can be successfully conducted only if the methodology is clearly defined.

This section describes the research methodology used to achieve the aim proposed in the research and it will follow the research process onion proposed by Saunders, Lewis and Thornhill (Saunders et al, 2016) as shown in Figure N.1. A deep dive of every aspect of the research process will be provided, this involves the research philosophy and strategies used, followed by sampling and data collection process and finally the analysis process and ethical issues that the research techniques may imply as well as the limitations of the research methodology.



*Figure N.5 Research Process Onion
Saunders, Lewis and Thornhill. (2016, p.124)*

3.2 Research Design

3.2.1 Research Philosophy

The research philosophy known as the first layer of the research 'onion' is a system of beliefs and assumptions about the development of knowledge in an specific field (Saunders et al, 2016, p.124). During the development of the research, it will be necessary to make different types of assumptions, which may be related to the human knowledge (Epistemological), the realities encountered (Ontological) and the ways that my own values influence (axiological). According to Bell et al (2019 , pp.18) an Epistemological philosophy in a business research, refers to theories about what is known or what can be known, it has knowledge as a main objective (Bell et al, 2019,p.18), moreover Saunders et al (2016, pp.127) consider the epistemology as assumptions about knowledge and how these can be communicated to others. Another philosophy is Ontology, which refers to theories about the nature of reality (Bell et al, 2019, p.18), it influences the way the research objects are seen and studied by the researcher, from a business and management perspective, these objects include organizations and individuals (Saunders et al, 2016, p.127). Finally, Axiology, which is the term that refers to the study of the nature or essence of the values and judgments within the research process. (Saunders et al, 2016, p.127).

The researcher's beliefs and assumptions influence directly the philosophy choice, which will underpin the methodological choice, the research strategy and the data collection techniques. Although, scholars and gurus of research methods have not found agreement about what is the most suitable philosophy for business and management researches, a consistent set of assumptions will form a credible research philosophy, which is required for a coherent research project. According to Saunders et. al (2016, p135), there are five main philosophical concepts

a business and management research can employ: Positivism, interpretivism, pragmatism, critical realism and postmodernism. However, positivism and interpretivism are the most acceptable and common philosophies apply in a master's level.

Positivism, this philosophy known as scientific approach is usually related to a Quantitative research, where the researcher is not involved in the data collection process. According to Blumberg et al (2014) philosophical and scientific activities must be carried out within the framework of the analysis of real events verified by experience. A suitable way to collect data is by observing these events directly or by measuring them using surveys or other instruments. It follows the same sequence applied in experimental research in the physical sciences, framing hypothesis, gathering data and testing it to determine its veracity to then accept that the hypothesis represents a temporarily true statement of reality (Bell et al, 2019, p.30)

Interpretivism, in contrast to Positivism, it consists of a high involvement of the researcher in the course of the process, the researcher seeks to understand social phenomena from the actor's own perspective. According to Bell et al (2019, p.31) Interpretivism requires a strategy that respects the contrasts between individuals and the objects of the natural sciences, the compression of individual's experience predominates and for this reason it requires the researcher to get involved and understand the personal meaning of the social action.

This research aligns and draws from the research philosophies of interpretivism which is usually associated with a qualitative research strategy. This philosophy has as a main objective to understand opinions, interpretations and experiences of the population interviewed, whom have a depth knowledge and a huge experience in the application of agile project management

methodology in the software development industry. The use of interpretivism will be more relevant to develop an understanding of the subject from an open analysis.

3.2.2 Research Approach

There are different approaches that can be applied in qualitative researches such as Inductive, Deductive and Abductive as shown in Figure N.3. An **inductive** approach uses arguments to obtain conclusions that start from particular facts which are accepted as valid to reach conclusions whose application is of a general nature. This method begins with the individual observation of the facts, the behaviour and characteristics of the phenomenon are analysed, and comparisons, experiments etc. are made to finally come to a universal conclusion (Blumberg et al. 2014, p. 21-22).

A **Deductive** approach different from an Inductive, begins from a general frame of reference to a specific. According to Saunders et al (2016, p. 145), in a deductive inference, conclusions are valid or accepted depending on the reasons (premises) given. It begins with the analysis of postulates, theorems, laws, principles, etc. of universal application, and through deduction, arguments, assumptions, etc, its validity can be verified to apply it in a particular way. On the other hand, in an **Abductive** inference Saunders et al (2016, p.45) states that the data collection, identification and analysis process is used to generate a new or to modify an existing theory to subsequently test it through additional data collection.

After evaluating these three different approaches, the Inductive approach was adopted to conduct the research as it is clearly linked to the qualitative research strategy to be implemented, as stated by Saunders et al. (2016,p.147) that an inductive research approach will

be more suitable with an interpretivism philosophy. This approach allows formulating the theory after the collection and analysis of data, which will be compared with the literature review of agile project management methodology presents in this research.

3.2.3 Research Strategy

There are different research strategies whose implementation will depend on whether the investigation is exploratory, descriptive or explanatory and whether it follows a qualitative or quantitative approach (Saunders et al. 2016, p.177-178). A qualitative research is an exploratory investigation that is used to get a deep understanding of latent motives, opinions and motivations while a quantitative research is based on statistics to analyse and verify data or specific information (Blumberg et al. 2014. P. 148). According to Saunders et al. (2016, p.178) Experiment and Survey are the strategies that are exclusively linked to a quantitative approach while Case Study and Archival & Documentary Research can be linked to either quantitative or qualitative and finally Ethnography, Action Research, Grounded theory and Narrative Inquiry that are exclusively linked to a qualitative research.

Since the research seeks to analyse the behaviour, facts and events that software development professionals have experienced during the implementation of Agile Methods in the development of software projects, a qualitative research strategy will be the most appropriate to provide an in-depth understanding of the investigation. A qualitative strategy would result in valuable insights for accurate conclusions rather than quantitative research strategy, which could limit the scope of the research.

Additionally, the research will be developed through the multiple case study strategy, according to Saunders et al. (2016, p.184) a case study allows to elaborate one or multiple hypotheses or theories through the study of a determined reality or situation, as well as to confirm these hypotheses and existing theories. To study real-life cases is what distinguishes this strategy from others. Saunders et al (2016, p.184).

Moreover, as mentioned in the previous section that an inductive approach is usually linked to a qualitative research, Bell et al (2019, p.64) states that *'When the predominant research strategy is qualitative, a case study tends to take an inductive approach to the relationship between theory and research'*. Through this strategy will be possible to examine successful and unsuccessful software companies in the application of agile methodology in the development of software projects and what prevents them to implement this methodology at large scale. According to Bell et al (2019, p.63) by the implementation of case study strategy, semi-structured interviews and documentary data collection in a research conducted to a UK retail bank, findings permitted to identify insights into why so many quality management programmes had failed, for this reason a case study strategy is considered to be the most suitable for this type of research.

Time Horizon

According to Saunders et al (2016, p.200) there are two types of time horizon: Cross-sectional studies and Longitudinal studies, and the decision to follow one or another will depend on the research question. A Cross-sectional study refers to a particular time in which the research is complete, while in a Longitudinal study the research can be conducted in a prolonged period of time (Saunders et al. 2016, p.200). As this research is conducted for academic purposes,

time is usually a constraint, therefore a cross-sectional study will be considered as the most appropriate approach due to the nature of the research. Although this approach usually employs survey strategies, according to Saunders et al. (2016, p.200) it can be employed also for qualitative researches as a number of cases are based on interviews that are conducted over a short period of time.

Secondary Data

Secondary data involves qualitative data which has been extracted from books, journal articles and any published data from reliable academic sources that have been already studied and analysed by other researchers in order to identify different perspectives that will address the conduction of this particular research. This data was selected based on similar studies and publishing year; data with no more than six years old has been considered for this research, since Agile Methodologies is widely used in the management and development of software projects, articles must be as recent as possible to avoid inaccurate or invalid information.

3.2.4 Sampling - Selecting Respondents

Sampling techniques will allow to reduce the amount of data to be collected by only taking into consideration the data from a sub-group instead of all possible cases or elements (Saunders et al, 2016, p.272). These sampling techniques are widely classified into two types: non-probability sampling and probability or representative sampling. Probability sampling technique allows to know the probability that an individual has to be included in the sample through a random selection, whereas in Non-probability sampling technique, the selection of respondents will depend on their knowledge and experience on the topic. (Blumberg, 2014)

This research seeks to get a detailed overview of the benefits that Agile methodologies brings to the development and management of large software projects. To gather information on the research topic, the population selected for the study involves Agile managers, Scrum Masters, Agile Architects, Agile coaches, Software Developers, etc, who have a huge knowledge and experience in the application of agile methodology, therefore, as respondents with certain criteria (knowledge, expertise, etc) is required in this research, a non-probability sampling technique has been used. According to Saunders et al (2016, p.297) a minimum non-probability sample size for semi-structured/In-depth interviews is 5-25 and in the case the research requires comparison between distinct groups a larger sample size will be needed. A sample size of 5-7 respondents has been selected for this research which involves only participants with experience in the topic and who know the implications of Agile methodology in large software development projects. It is important to mention that no comparisons between groups will be considered in the research.

There are different methods used in the non-probability sample selection process in a qualitative research such as: Convenience, Purposive and Snowball. According to Blumberg et al. (2014, p. 193) Convenience sampling are unrestricted non-probability samples where researchers have the opportunity and autonomy to select whoever they can find to be part of the sample, this includes friends, neighbours and other people found in public places, therefore is the cheapest and easiest sampling technique to conduct but also the least reliable design. In contrast, using Purposive sampling technique, the sample is selected according to certain criteria, participants are selected to represent pre-determined parameters and conditions. It consists of 2 major types: Judgment and quota. Judgment sampling, where the sample is chosen based on the researcher's knowledge of the population, its elements and nature of the research objective. And, Quota sampling, where the researcher ensures equitable and

proportional representation of the subjects, depending on which condition is considered the basis of the quota. (Blumberg et al, 2014, p. 193-194). Finally, the Snowball sampling technique that refers to the accumulation of results that can be found elsewhere, it is used when it is difficult to locate members of a certain population, to carry out a data collection few members of the target population can be located and asked for the necessary information to locate other members who know of that population (Blumberg et al, 2014, p. 195). Experts that can provide valuable information based on their experiences in the management and development of large software projects and the implementation of agile methodologies is the most appropriate population for this type of research, however this participants may require to be contacted through third parties as there is no a direct relationship with the researcher therefore the technique used was Snowballing sampling, experts with huge interest in agile methodology at large scale were contacted through referrals at Dublin based tech companies such as: Workhuman and Guidewire Software.

3.3 Data Collection Instruments

Data collection refers to the use of a wide variety of techniques and tools that can be used by the researcher to develop information system. (Blumberg et al. 2014, p. 154). One of these techniques is the interview which, depending on its purpose, may contain highly structured and unstructured sections (Saunders et al. 2016, p.390).

As mentioned in the previous section, since the investigation was defined as a qualitative research, it will require to provide certain flexibility during the interview process in order to collect all relevant information needed, therefore the primary data for this research has been collected through semi - structured interviews. Through this type of interview, it will be possible to give to the interviewees the freedom of express their ideas and point of views in

their own terms, so that it will contribute to provide reliable, comparable qualitative data. In addition to this the researcher will be able to adapt questions to the different situations and particular characteristic as the interview progresses.

These semi-structure interviews will be conducted face to face or by skype calls depending on the choice of the interviewee. During this process, it is planned to interview between 5 and 7 people with experience and knowledge of agile methodology in the software industry, which provides in-depth understanding of benefits of agile methodology in large software projects. These interviews will be voice-recorded to engage with the participant during the dialogue and then transcribed for data analysis, it is important to mention that notes will be taken during the interview in case the interviewee disagrees with the recording of the conversation.

3.4 Data Analysis Procedures

Given that the investigation is a qualitative research, the analysis of data is quite complex and laborious as a large volume of data is generated, due to one of its main characteristics, the circularity of data. The method of obtaining the primary data was semi-structural interviews, which were transcribed through a software called 'Otter'. Then, the data analysis performed would help to identify emerging themes or patterns and trends of responses that agile software professionals have given about APM in large software projects.

3.5 Research Ethics

In this type of research is common to find ethical issues, therefore it is important to take into account the ethical principles when performing the investigation, Bell et al (2019, p.114) divides these principles in 4 main areas: harm to participants, lack of informal consent, invasion

of privacy and deception. The risk of harm to participants involves many aspects such as physical harm, harm to participant's self-esteem, stress and the most common for qualitative researches are the issues of confidentiality and anonymity, which in turn, involve legal and ethical considerations (Bell et al. 2019, p.117).

In order to minimising the risk of harm and following the ethical standards of Dublin Business School, as much information as possible will be provided to obtain informed consent from participants. Blumberg et al (2014, p. 122) also states that by giving sufficient information and a justification of the study will motivate the interviewee to answer questions truthfully. The explanation of the purpose and process of the research as well as the importance of the interviewee participation will be explained before commencing the interview, each professional to be interviewed will be provided with a consent form by email which states that the data will be treated confidentially and will be used exclusively for the purpose of the research, also participants will be informed that the conversation would be recorded for data analysis, and it will only be done after obtaining consents from the interviewees. Additionally, participants will be informed that the data collected will be stored in password protected devices and if after the interview, any of the respondents asks that the information provided should not be considered or it should not be used, it will be its right and it will be removed completely.

3.6 Limitations of Methodology

As any other researches, there are a number of limitations faced in this particular investigation. The research is focused on highlighting the benefits that Agile Project Management Methodology has brought to Software Development Projects, specifically for small projects and the advantages and limitations that can bring at large scale, however the scope of this

research may be limited due to the lack of previous investigations or literature available of agile project management methodology in large software projects.

The information gathering process may represent a risk to obtain reliable information as the quality of the data collected depends on the respondent's perceptions of events and opinions, this is another limitation that can be found during the collection and analysis of data as well as relevant details that can be missed or not shared during the interview process if the respondent does feel intimidated with voice-recorded interviews. Other important limitation is the time frame to complete the research as the strategies to be developed must comply with academic exigencies, it is necessary to ensure there is sufficient time to analyse data to reach a conclusion that meet all requirements.

3.7 Summary of chapter

This section has provided a detail description of the various types of methodologies by following the onion research process proposed by Saunders et al (2016), all aspects of this technique were explained as well as a justification of the selection of specific methods and strategies were provided.

Overall, the research will employ an Interpretivism philosophy, which has as a main goal to understand opinions, interpretations and experiences of professionals or experts in the application of agile methods in software development projects, followed by an inductive approach which allows formulating the theory once data is collected and analysed. Also, multiple case study has been chosen as a research strategy, as well as semi-structured interviews which will be helpful to study agile methodology benefits and implications in some

software companies like Work human and Guidewire Software. Finally, an overview of the ethical issues that may be present during the information gathering and limitations that this research may face.

4. DATA ANALYSIS AND FINDINGS

4.1 Introduction

This chapter presents the findings of the study which were obtained through semi structured interviews conducted with software development professionals with proven experienced in Agile environments. A brief description of respondent's profile is provided, followed by a detailed description of the answers and relevant data obtained from interviewer's experience. Each of the interview's questions is related to the research objectives so that responses obtained can provide a more accurate answer to the research question.

4.2 Qualitative Data Analysis

Five semi-structured interviews were conducted with software engineers in order to collect relevant data for this research. Respondents were technology professionals with experience in the development of software projects under agile methods. No personal or sensitive data was required from participants and pseudonyms have been used so that anonymity of respondents could be maintained as detailed in table below:

Participant	Job Title	Details
H1	Senior Software Consultant	Profession: Software Engineer.
H2	Senior Software Consult	Profession: Java Developer
H3	Senior Architect	Profession: Computer Science Engineer
H4	Senior Software Development	Profession: Information Technology
H5	Senior Software Development	Profession: Software Engineer

Figure N.6 . Participant codes

Interview questions were mainly focused on five aspects related to the research objectives and which the structure of this chapter was based on:

- Benefits of APM in large software development projects.
- Limitation of apply APM at large scale software development projects.
- Differences of APM in both small and large software development projects.
- Comparative Analysis of both traditional (waterfall) and Agile at large scale software development projects
- Does APM work at large scale?

4.2.1 Benefits of APM in large software development projects

According to H1, in large projects, it is usually quite difficult to make sure that each team member sees the big picture of the project, so that the entire team is aware of the status of the project, if it is going in the right direction or not. H1 affirmed that by using agile methods, it is possible to see how big the backlog is, how big the list of features is and what still needs to be delivered. From that, it is possible to make a proper plan by prioritizing the features that are a necessity to deliver first as well as giving an estimated time of delivery. Also, H1 mentioned

that by using sprints, in the case of scrum teams, is possible to agree if one item is more important than other, this feature will be then the top priority which is delivered as soon as possible. Moreover, sprints help to identify items that were not planned to be delivered. In that way, it is possible to quickly re-plan and give these items priority so that they can be rapidly delivered without impacting the rest of the items. In addition, H1 was also of the opinion that quality improvement does not necessarily depend on the methodology used. However agile methods do help if, for instance, quality issues are encountered during the product development, it is possible to start working on the issue directly. This is indeed one of the benefits of using agile as it is necessary to ensure, always, that everything that is being delivered is of the best quality, and if not, it allows to make adjustments as soon as possible.

According to H2, in large software projects the risk of failure is quite high. For this reason it is crucial to be able to detect these risks during the development process. The key point of agile is that allows to identify and face possible issues as the project progresses, so that all team members are aware of what the project's challenges are early in the process, therefore H2 remarked the risk management as a key benefit of agile in large software development projects. Also, H2 mentioned that being aware of these problems at the beginning of the project helps not only to lower costs and reduce delivery time but to minimize team frustrations, which is one of the most valuable characteristics of agile. H2 emphasized that agile improves employee's satisfaction, working with this method is less frustrating than working with waterfall as the team is aware of the challenges that they can face. Moreover, H2 mentioned that quality is also one of the benefits of agile even when organizations with a well-established waterfall culture can deliver huge quality. The interviewee stated that "it depends mostly on the time or type of project, but the quality in software company is much better with agile than

with waterfall, however, in other industries waterfall could deliver the same quality as agile, that's my experience”.

According to H3, from a management level, agile gives the whole picture of the project at a certain time. For a one-year duration project with around 100 user stories to complete or main features to complete, agile provides a better visibility of how far the team is from the goal. H3 mentioned that overall in a large project, this is an easy way to provide visibility to the executives or to the people that are financing the project. Which also provides flexibility when the project is delayed or when adding extra scope is required and the deadline needs to be extended, in other words visibility helps to make decision on the budget and resourcing. Moreover, H3 concluded that by providing a better picture of what needs to be done and who the owner of each task is, the customer and employee satisfaction is improved. Finally H3 pointed out that, in the insurance industry, delivering on time cannot be considered as a benefit due to the dynamic of this particular sector, customers may need to add even more functionalities before the product goes live and the reason is that they want to achieve a higher customer satisfaction. H3 stated that “Our customers are more focused on providing a better user experience, rather than the budget that is being spent to make that happen”.

According to H4, agile makes team member's job easier, in terms of knowing exactly the work that needs to be done, otherwise, the team will be in a position on not really knowing what they want to do, or what they must do. The interviewer affirmed that “When you have it on a board, you can come in, you know things that are available for you to do”.

According to H5, in the agile approach client's feedback is provided soon enough; insights from clients are received when something is not going as they expected, or when new

requirements need to be included. The team has time to react and change the plan without going through heavy work. Also, H5 mentioned that a better visibility is given by having a plan with all stories which the team needs to work on and which the team can always see.

4.2.2 Limitation of apply APM at large scale software development projects

According to H1, one limitation is that, in a large project, there are more people that need to be managed, this can be a difficulty, however it is a small limitation that can be . Also, H1 mentioned that one of the main limitations is that as long as there are people not using agile or not fully convinced that agile works in a larger project, the entire team will feel that the project might fail. Moreover, H1 stated that is very important that the first people working in agile way are the management of the project and the company.

According to H2, the limitation of agile is really lack of commitment from the management level; if the people working in agile do not embrace agile and make a commitment on working in agile, agile could be even worse than waterfall. Each member has to play the game. If, for any reason (cultural, monetary, etc), most of the people within a project or within a company do not want to play agile, it is better not to go agile.

According to H3, having experience mostly in large projects, no limitations were found except for the inspection plans which are quite big and the commitments for the sprint. H3 concluded that the larger the project is the larger the number of points to be discussed and inspections to be done. Also, H3 Highlighted that the limitations are related to the management, maybe companies still using traditional methodologies as that is the way they have always been doing

it or perhaps the software or project does not have many features and it is oriented to only one feature so that they can go over and over again through the same process.

According to H4, the bigger the project, the bigger the possibility that the same backlog of work is crossed over by multiple teams and there could be a lot of crossover between teams working at different levels, therefore the estimation process can be a bit out of kilter, as they can estimate stories differently. H4 highlighted that it starts becoming a bit of a problem when multiple teams work on the same backlog, and that could prove to be an issue initially of some sort of baseline that all teams follow, for instance five story points, essentially two weeks of work. Then they just set the baseline across teams, the interviewer affirmed that may not work too well.

According to H5, if there is a really big team that has a lot of different pieces or sub-teams, they will need to be in constant communication in order to avoid time delivery issues as there may be teams working on something that is impacting the other's work and this sometimes can slow down the process of delivering a feature. Also, H5 considered lack of management as a limitation. The interviewee stated that if the management is not doing things as they should, agile methods will not be useful in large projects. As an example, according to the interviewee's previous experience having worked in a company where they were supposed to be using agile methods, their sprints were every three-six months, this being the case, once an item was released and the customer started logging into the application and testing it changes can be requested if they disagree with what was delivered to them or with what the team did in six months. In other words six months of work ended up being a waste of time, the interviewer concluded that this is due to a matter of some management issues as they say to be using agile methods when they actually do not. H5 was also of the opinion that lack of experience is one

of the limitations of agile in large projects, most of new companies may start developing a new large project using agile methods with un-experienced people in charge which can lead to a project failure and not necessarily due to the methodology itself but because of the lack of experience of members.

4.2.3 Differences of APM in both small and large software development projects

According to H1, there is a huge difference in small and large projects. The first difference, for example, if there is a very small project sometimes the team may consider not implementing every single line of agile, as in that way, it would take even more time managing rather than developing, and in some cases delivering features quickly is the most important requirement. On the other hand, when applying agile in a larger project, the perspective would be the opposite, it will be necessary to make sure that everything works in a very agile way, and it will be required to make sure that the whole team and project can embrace the agile methodology.

According to H2, the key idea beyond agile is to divide and conquer approach to handle different difficult things to do, and what agile does is to split all these challenges into small chunks and handle every chunk individually, the agile methodology creates a way to handle all these small chunks. This is, in other words, a large number of small chunks working together. For small projects, this is actually not necessary, divide and conquer approach, it can be made, requirements are handled even better as the number is not beyond the human capacity to handle everything at the same time. The interviewer stated that for every project, less than a month and less than 5-6 people can be handled without agile. However, when going to a larger project,

agile will help better. The idea of agile is having multiple teams, multiple projects in different releases working together, this is how agile will perform.

According to H4, in small project, communications are better, however, when dealing with larger projects, off site workers can be also involved, and communication can be a difficulty. In this case, it is necessary to storyboard so that each member is able to see what the breakdown of the stories are, what is the available to get done, and what it has been already assigned, this makes collaboration easier. From that point of view, the interviewer affirmed that in agile, using tools such as JIRA, facilitates communication.

4.2.4 Comparative Analysis of both traditional (waterfall) and Agile at large scale software development projects

According to H1: One of the benefits of agile in largest projects compared to traditional is that as soon as an issue is identified, changes needs to be done to the project, it is easier to do these changes in an agile approach, rather than in a traditional approach, sometimes doing a change, or identifying a problem in a traditional methodology is more costly, costly in time, resources and planning. In traditional, changes can be only made at the end of the process.

According to H2: Waterfall has very few benefits compared to agile in large software development projects. However, waterfall has something that agile does not have which is, from the customer's point of view, they could go waterfall in order to have a fixed price. Customers will know the price of the product in advance, they will give all requirements and the company or service provider will then fix a price, and then the customer can be relatively sure of what the cost of the project will be, and obviously, if the cost and time increase or the project gets delayed, the contract will be then the responsible for these additional costs. H2

concluded that customers want to go waterfall mostly because they want to reduce price and negotiate with multiple services provider in order to get the cheapest.

H3 mentioned that the main concern with traditional methods is that when developing one piece with a very good visibility, and even having that visibility as a developer, the team may be considering only one use case and missing the rest, H3 affirmed again that agile helps to get better visibility in large projects. Also, the interviewer believes that agile helps to do things better in large projects.

According to H4, Traditional methods take a lot of work up front, members are not sure how long an aspect would take and start working on the coding, everything is done as a whole. with agile, the project is broken into smaller building blocks, which gives the whole team visibility into the time estimates for each piece of work.

Teams work from the perspective of this little bit of software that needs to be released, that fits into the whole project, build a bit of work that you're doing so that it allows to easily break it up.

H5 affirmed not to have seen many benefits of using waterfall, for instance, when planning the project, there are many factors involved. The interviewee mentioned that if the project is not broken down or releases are done bit by bit, and instead, everything is done all at once, the project might end up being a huge mess. This is because it requires a lot of preparation time before starting, architects are required as well as paperwork needs to be done and other different requirements before the work can be started. Moreover, the interviewer stated that if changes come in between, costs and rework will be higher, and here is where traditional waterfall approach fails.

4.2.5 Does APM work at large scale?

According to H1, it is necessary to make sure that everything works in a very agile way. The whole team and whole project need to embrace the agile methodology. The interviewer provided an example based on his experience where in a project of 10 different teams, each team had eight different people working. There was a total around 80, 90 people in the whole project, it was a very large project, by the end of the third month, the team started to encounter some issues, which were due to a bad implementation of agile. After reviewing the possible causes of the issues, the team was required to be re-trained, so that each of the member would be able to use agile properly. Finally, the team started working properly on the practice and everything went smooth in the end. The interviewer concluded that although there are differences, the larger the project is, the more important is using agile. H1 was also of the opinion that one of the difficulties is the confidence of each member on using agile in a larger project. It is crucial that if the project is large every single person works in agile way. Also, H1 considered that it is even more important that the most experienced people working in an agile environment are the ones managing the project and the management of the company as well.

H2 mentioned that according to his experience working in small and large projects, agile works much better when is applied to large projects. Small projects can also work with an agile method, however agile creates a number of different redundancies and extra tasks to do in comparison with waterfall that for small project may not be necessary. The interviewee affirmed that when going to larger projects, agile works better, having multiple teams, multiple projects in different releases working together, this is how agile performs.

According to H4, agile in large projects works better than traditional methods. It is true that there are some issues present in terms of forecasting and story point estimating, however, this does mitigate when using tools such as JIRA, which helps with this type of issues. H4 also stated that for large projects, there are always going to be problems such as different teams working at different sort of cadences or having different story points for each of their stories that would not correlate to other teams, this can affect the reporting and may be an issue, however it can be managed.

According to H5, if companies have its internal structure well defined and from the software point of view, the architecture is well defined, there can be as many teams as you needed as long as they are independent. Team members will need to be able to work in small projects that are part of a big one.

Also, H5 mentioned that the culture and policies of the company, lack of experience or lack of management support at some points impact the success of applying agile in a big project. H5 described his experience in a company where they were supposed to use agile, their sprints were around 3-6 months, they ended up having issues when delivering a feature after six months of developing, customers may disagree with the result. H5 concluded that it was a management issue, because in fact it was not the software company, but businesspeople trying to run a software company and they did not have the experience, or they did not fully understand what agile was for. In addition, H5 mentioned that companies want to apply, a sort of old-fashioned school of ways of doing business combined with new business techniques, but this works in a different way.

Finally, according to H5 agile in large project should not be an issue if it is used properly.

The interviewee mentioned that from his experience, the idea of applying agile to large projects means that the project needs to be broken down into smaller bits so that can be easily controlled.

5. DISCUSSIONS

5.1 Introduction

This section examines the information gathered the Literature Review chapter by creating a discussion on the results collected during the primary data collection process and the responses obtained from software development professionals interviewed in order to achieve the research aim.

5.2 Discussion of the data

Given that the aim of this research is to identify and analyse the benefits that Agile Methodologies provide to large software projects as well as to determine if this approach can work successfully on a large scale, it is necessary to interpret and discuss the findings obtained in relation to the previous theories and studies. The collection and analysis of this data allowed the researcher to identify patterns in relation to the benefits and limitations provided by the agile approach on a large scale.

One of the most relevant patterns is the visibility of the project. According to H1, by using agile methods it is possible to see how big the backlog is, how big the list of features is, and what needs to be delivered. Moreover, H3 mentioned that agile methods provide the whole

picture of the project at a certain time; it provides visibility to executives and investors about how far they are from the goal, and if extra scope needs to be added or if the deadline needs to be extended. In addition to this, H5 stated that better visibility is offered when a plan is in place as it informs all members of what they need to work on. This is also supported by Lagerberg (2013 p.352-353) who, during the implementation of agile methods in an Ericsson case study, concluded that the visibility of the project was increased.

Project quality was another significant pattern found during the analysis of the findings. According to Petit (2016), in a research conducted by PMI, he stated that there are “still some concerns as to whether software quality has become better or worse” when adapting agile principles to large projects. On the other hand, during the development of two projects ‘Subscription Billing Platform’ and ‘WebEx’ at Cisco Cloud Software company, Pandey (2015) affirmed that projects quality was improved by the reduction of 40 percent of critical defects and the decrease of 25 percent of the quality assurance defects, respectively. In addition, the development process was accelerated by doing a daily check on the code which allowed developers to review new features earlier in the process. Furthermore, Papadoupoulos (2014, p.461) during a case study conducted at Telematicum Inc, concluded that due to the flexibility that Scrum provides, by allowing frequent tests and revisions with the Product Owner and team members, an improvement in quality was achieved. H1 also agreed that a better quality level can be achieved as Scrum allows for the early identification of defects and fixes them as soon as possible. In addition to this, H2 stated that quality is also one of the benefits of agile even when organizations with a well-established waterfall culture can deliver huge quality; the interviewee concluded, according to his experience, that the quality in agile is much better than waterfall in relation to the software sector.

The Employee and Customer Satisfaction elements also had relevant patterns throughout the analysis of data. According to Papadopoulos (2014, p.462), in a large software project study conducted at Telematicum Inc., he concluded that the employee satisfaction rate was successfully improved. This was measured through a survey conducted with employees on which a majority of the respondents stated they experienced a higher level of job satisfaction when using Scrum. However, in this study, the customer satisfaction rate was not measured as it was not possible to collect the necessary data to make solid conclusions due to the recent release of the software product. Moreover, H2 agreed that the employee satisfaction rate was achieved as people were aware in advance of the challenges that they had to face which resulted in less frustration among employees. In addition to this, H4 stated that agile methods make the team member's job easier as each employee knows exactly what the status of the project is at any particular time and the work that needs to be done, concluding that agile increases employee satisfaction. This is also supported by Pandey (2015), who outlined that during the development of two large and complex products at Cisco, eliminating the need for employee overtime and extra meetings resulted in higher employee satisfaction rates. Finally, H3 mentioned that both customer and employee satisfaction were improved due to the benefit of visibility; the interviewee stated that the team was able to provide the customer with visibility on the project and could detail how far the team was from the goal at a certain time. Additionally, from the developer and management perspectives, a good picture of what needs to be done as well as identifying the owner of each task are important features of the APM.

One of the most relevant aspects that can determine whether the agile approach works successfully on a large scale or not, are the limitations. According to H2, one of the limitation of agile is the lack of commitment from the management level; if people working in a project do not embrace agile and make a commitment to working in an agile environment, this

methodology could have a more negative impact than the waterfall method. Everybody must co-operate. H2 concluded that if for any reason (cultural, monetary, etc.), some members within a project or within a company refuse to use agile, it is better not to use it. Moreover, H5 agreed that the lack of management buy-in is a limitation; the interviewer affirmed that if the project is not managed in an agile way, these methods will not be useful in large projects as a number of issues may appear during the development of the project. In addition, H1 mentioned that the main problem is that as long as there are people not using agile or not fully convinced that agile works in larger projects, the entire team will not feel positive about the success of the project. Therefore, H1 stated that it is very important that people with more experience working with agile methods manage the project and the rest of the team. This limitation is also supported by Larson et al (2018,p.594) who, in a survey conducted to over 3,900 software and financial representatives, concluded that the main reasons for its failure were due to a lack of management support, company culture and a lack of experience.

As mentioned above, a lack of experience and a company's culture are other main reasons why agile fails when it is being applied to larger projects. According to H5, some new companies may start developing a new large project using agile methods with inexperienced people in charge which can lead to project failure and this is not necessarily due to the methodology itself but because of the lack of experience of the staff members. Moreover, Petit and Hobbs (2017) highlighted that during the application of agile principles to large projects, different types of conflicts associated with culture, structures, and the overall development process may arise. As an example of this, Lagerberg et al (2013,p.351) mentioned that a survey conducted with members of two projects (project A using waterfall and project B using Scrum) developed simultaneously, surprisingly project B spent more time in the documentation phase than project A. However, the researcher concluded that the result was impacted by policies and

directives from within the Ericsson company. This is supported by H5, who mentioned that the culture and policies of the company at some points impact on the success of applying agile in a big project.

Other outcomes that emerged during the analysis of the findings was the level of communication that existed and the management of people in large software projects. According to H4, in a small project, communication levels are better. When dealing with larger projects, off-site workers can be also involved, and communication can be a difficulty. However, according to Lagerberg et al (2013, p.354) in the project developed at Ericsson, the communication between different functional roles was increased when using agile methods. In relation to the management of people and meetings, H1 mentioned that more people need to be managed in larger projects and this can sometimes be a difficulty. However, it is a small limitation that can be overcome. In addition, Papadopoulos (2017, p.458) highlighted that as the project scales, it becomes very difficult to hold meetings with all members of the agile teams, therefore, to overcome this difficulty it is necessary to conduct sprint planning.

After discussing the benefits and limitations of APM at large scale, it is necessary to discuss and determine if agile works successfully on a large scale based on the findings and previous researches. According to H1, it is necessary to make sure that everything works in a very agile way. The whole team and the whole project need to embrace the agile methodology. Therefore, H1 provided an example based on his experience in a very large project. By the end of the third month, the team started to encounter some issues, which were due to a bad implementation of agile. After reviewing the possible causes of the issues, the team was required to be re-trained, so that each member would be able to use agile properly and the process could be improved. Additionally, H1 emphasized that the larger the project is, the more important it is to use agile,

but it is crucial that every single person works in an agile way. In addition to this, H2 mentioned that according to his experience working in small and large projects, agile works much better when it is applied to large projects. Small projects can also work with an agile method, however agile creates a number of different redundancies and extra tasks to do in comparison with the waterfall method that for small projects may not be necessary. Therefore, H2 affirmed that when working on larger projects, agile works better as having multiple teams and multiple projects in different releases working together is how agile performs at its best.

On the other hand, H4 stated that for large projects, there is always going to be problems such as different teams working at different paces or having different story points for each of their stories that would not correlate to the other teams. This can affect the reporting and may be an issue. Also, H4 commented that there are some issues present in terms of forecasting and story point estimates, however, this is mitigated when using tools such as JIRA, which helps with this type of issue. According to H5, the culture and policies of the company, a lack of experience or lack of management support at some points can impact on the success or not of applying agile in large software projects. However, the interviewee emphasized that if companies have well defined internal structure, and from the software point of view their architecture is well defined, there can be as many teams as are needed as long as they are independent. Team members will need to be able to work in small projects that are part of a bigger one.

Finally, According to H2, the risk of failure is very high in large software projects. The key point of agile is that it allows workers to identify and face problems early in the process; all members will be aware of what the challenges are in the project so they can be faced sooner. Therefore, H2 remarked that risk management is a key benefit of agile in achieving success in

large software development projects. This is supported by Jorgensen (2018, p. 189), who concluded his research on the failure risk that is reduced rather than increased when the agile methodology is applied instead of traditional methods in large software projects.

6. CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

Based on the analysis of the benefits when applying APM methodology in large software projects, it is possible to conclude that visibility, quality, employee satisfaction, and customer satisfaction can be achieved.

The body of this research paper presents evidence that more visibility was obtained during the development of a large-scale software project conducted at Ericsson, a telecommunications company. Moreover, during the interview phase, H1, H2, and H5 concluded that agile methods provide the whole picture of the project, which gives a better whole vision of the large project. The second benefit obtained was quality; this was improved during the development of the ‘Cisco Subscription Billing Platform’ project at Cisco Cloud software company and ‘TeleWeb’ project at Telematicum Inc company. Also, H1 and H2 agreed that a better level of quality can be achieved when agile is applied to large software projects. The third benefit obtained was employee satisfaction; this was successfully improved in a case study conducted for Telematicum Inc and during the development of two large and complex products called ‘Cisco Subscription Billing Platform’ and ‘WebEx for Samsung’ at Cisco. Additionally, this benefit was supported by H2 and H4, who agreed that agile generates less frustration and more satisfaction in employees. Finally, H3 highlighted that the benefit of visibility positively

influences the employee's satisfaction as it provides a good picture to developers and managers of what needs to be done as well as who the owner of each task is. The fourth benefit obtained was customer satisfaction, which according to H3, higher customer satisfaction is achieved by providing the customer with optimal visibility of the project, which includes resource allocations and potential risks.

The benefits obtained have presented reasonable arguments in favour of APM in large software development projects. However, this research has also demonstrated that agile on a large scale is not straightforward, therefore, some limitations faced by software development teams were identified during the agile transformation journey at large-scale.

The first limitation identified was a lack of management support; the interviewees agreed that if the management team did not have the commitment and motivation to work in an agile environment, then this methodology could be worse than the waterfall method. Moreover, this paper, based on both interviews and research papers, suggests that for a large project to be really successful, all members involved in the project must embrace agile principles. The second limitation identified was a lack of experience; this was concluded after software professionals commented that in companies where large projects are developed using agile methods with inexperienced people in charge, it can lead to project failure and this is not necessarily due to the methodology itself but because of the lack of experience of members. The third limitation is related to the culture and policies of a company; these were identified from both the literature review and interviews. The body of the research papers highlighted that during the application of agile principles to large projects, different types of conflicts clashed or aligned with the culture of the company. Moreover, during the study of two complex projects at Ericsson, the researchers concluded that the software documentation was negatively

impacted by policies and directives of the company. Additionally, during the interviews with some software professionals, they stated that many companies may fail in the attempt of combining both methods, traditional and agile in large projects.

Finally, these limitations identified were also supported by the body of the research reviewed in this paper. Namely, where a survey conducted with over 3,900 software professionals concluded that the main reasons for the failure of the project were related to a lack of management support, lack of experience and company culture.

Furthermore, as a result of a comparative analysis between TPM and APM methodologies in the data analysis and findings chapter, this paper concluded that one of the main advantages of APM compared to TPM is that as soon as an issue is identified, changes can be made to the project. It is easier to make these changes in an agile approach, rather than in a traditional approach and sometimes making changes, or identifying a problem in a traditional methodology is more costly in time, resources and planning. On the other hand, in the traditional approach, changes can only be made at the end of the process. Additionally, this comparative analysis showed that the TPM methodology requires a lot of work upfront. The team members are not sure how long an aspect could take, there is low customer interaction, there is a fixed scope and changes are avoided and everything is done as a whole. On the other hand, the APM methodology requires a continuous level of work, resulting in a higher customer interaction level, a flexible scope, and a willingness to embrace changes.

After analysing and identifying the benefits and limitations of agile principles at large scale, as well as comparing both methodologies, this academic research concluded based on both primary and secondary data, that there is reasonable evidence that supports the success of APM

in large software projects. However, this is only achievable, as long as everybody involved in the project embraces and works in an agile way, and also if the company adopts an agile culture. This is crucial to overcoming the limitations related to the lack of management support, culture, and policies of the company. Finally, despite large projects being inherently risky, this research, based on the discussion chapter, suggests that the failure risk is reduced rather than increased when agile is applied instead of traditional approaches in large projects, which supports the success of agile in large projects.

6.2 Recommendations

This research has identified the majority of benefits and limitations to determine if APM methodology works successfully in large software projects. Therefore, this section provides the most valuable suggestions for addressing these limitations and for being able to achieve the success of APM methodology in large projects.

During the development of a large software project is absolutely necessary that the management team has the knowledge ,experience, motivation, commitment and mindset of working in an agile way. If any of the team managers does not believe that APM methodology works in a large software project, and try to apply Hybrid or TPM methodologies, it is more likely that the project fails. Additionally, in order to reduce the risk of failure, it is recommended that all members involved in the project work in an agile environment.

7. FUTURE RESEARCH

The author believes that some interesting gaps arose in relation to APM methodologies for future researchers.

This research presents as a suggestion the importance of applying agile methods during the development of a large software project. However, it will be interesting to assess the reason why some companies pretend to be 100 percent agile at scale, but they are not. Also, analyse the main reason why some software companies still traditional in large projects, despite APM methodology successfully faced the highly demanding, ever-changing and dynamic Industry.

After analysing the limitations faced by agile teams in large projects, it is important to assess the risks associated with the application of agile at scale. This will allow to have a deeper knowledge and make accurate conclusions of APM at large scale.

Finally, it will be interesting to analyse APM methodology for space operations or organisations where the failure of a project has terrible consequences.

8. REFLECTION

Having received my degree in Mechanical Engineering, I got the opportunity to work for one of the most famous multinational automakers, Ford Motors Company, where I acquired a significant experience and knowledge of manufacturing design and technology. During my time in the company I rapidly got interested in getting to know more in depth about the methodologies used when creating new pieces or products and all the management process that

is involved within it. It was only until I decided to pursue the MBA Project Management when I realized that having the foundations and abilities that this course were going to provide me and along with my background I could actually understand and see a bigger picture of the art of management.

It is evident that, in recent years, technology has advanced at unimaginable speed and many fields have been opened in the labour area related to technology, this was even more evident for me once I arrived to Ireland, learning that the most successful technologies enterprises were based here and this subsequently woke my curiosity up in finding out how my background could fit in it. Having experience working in a Lean environment and after doing a broad research regarding other methodologies used in in the project management in the technology sector, I found out Agile Project Management Methodologies have been used extensively for software development projects with positive results and not only are considered to be effective for software projects but also manufacturing since it emerged from it.

Agile development methodologies are becoming even more popular in recent years. Most leading companies use agile approaches, such as Google, Yahoo, Symantec, Microsoft, and an extensive list of organization follow the agile model. A number of experts perceive these methods as the panacea for all the ills of software engineering. However, there are still gaps in the success of the development process that use agile philosophy at large scale, although there is evidence of the benefits that this approach provides in small projects it is still unclear the benefits of agile at scale, for this reason my interest in studying and investigating the reasons why some organizations still attached to conventional methods when it comes to large software projects.

I believe that getting the most valuable anecdotes from agile experts, getting them to share their experiences and elaborating this investigation having as reference similar cases will definitely provide me an understanding not only on this particular management methodology but also on the traditional project management methodology such as Waterfall which still being one of the most used approach in the software development. By identifying the agile scope, benefits and limitations I will be able to quickly tackle possible issues when implementing the approach in the future.

Overall, my journey at DBS has consisted in unenumerable experiences and learning that not only allowed me to meet and share knowledge with people who had same goals as mine but to meet some expert individuals such as the lecturers and all professionals who were involved in the elaboration of this research from who I learnt not only about the topic but also from their personal perspective of project management in general.

References

- Agile Manifesto (2001) 'Manifesto for Agile Software Development'. Available at: <https://agilemanifesto.org/> (Accessed: 05 May 2019).
- Ahmad, A (2014). Agile Large-Scale Software Development Success Factors, challenges and Solutions. *I-manager's Journal on Software Engineering*, Vol 8 / Number 3, p1-10. Available at: https://www.researchgate.net/publication/292981218_Agile_Large-Scale_Software_Development_Success_Factors_Challenges_and_Solutions [Accessed January 10th, 2019].
- Augustine, S.(2005). *Managing Agile Projects*. Indiana: Pearse Education, Inc.
- Beck, K., Beedle, M., Bennekum A., Cockburn, A. Cunningham, W., Fowler, M., Grenning, M., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D. (2001) - 'Principles behind the Agile Manifesto' Available at: <http://agilemanifesto.org/principles.html> [Accessed June 10th , 2019].
- Bell, E., Bryman, A. and Bill, H. (2019). *Business Research Methods*. 5th ed. United Kingdom: Oxford University Press.
- Blumberg, B., Cooper, D. and Schindler, P. (2014). *Business Research methods*. 4th ed. New York: McGraw-Hill Education.
- BURROUGHS, A. (2019) 'Faster is better: How agile project management can benefit any organization', *Smart Business Cleveland*, 30(12), p. 46. Available at: <http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=138062483&authtype=shib&site=ehost-live> (Accessed: August 2019).
- Carrol, J., Morris, D. (2015). *Agile Project Management for speedy Results*. 2nd ed. United Kingdom: In easy steps.
- Dikert, K. Paasivaara, M. Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations, *The Journal of Systems and Software*, 119 p.89-108, Available at: https://www.researchgate.net/publication/308807209_Challenges_and_Success_Factors_for_Large-scale_Agile_Transformations_A_Research_Proposal_and_a_Pilot_Study [Accessed January 10th, 2019].
- Girvan, L., Paul, D. (2017). *Agile and Business Analysis Practical guidance for IT professionals*. United Kingdom: BCS Learning & Development.

Hobbs, B., Petit, Y. (2017). Agile methods on large projects in large organizations, Vol.48 / Issue 3, p3-19. Available at:
<http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=123094182&site=ehost-live&authtype=ip,shib,cookie,url&custid=s6175963> (Accessed January 26th, 2019).

Jorgensen M. (2018) 'Do Agile Methods Work for Large Software Projects?'. Processes in Software Engineering and Extreme Programming. XP. Notes in Business Information Processing, vol 314. Available at: https://link.springer.com/chapter/10.1007%2F978-3-319-91602-6_12 (Accessed: June 2019).

Kettunen, P. (2007). Extending Software Project Agility with New Product Development Enterprise Agility. *Software Process: Improvement & Practice*. Vol. 12/ no. 6, pp. 541–548. Available at:
<http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib,cookie,url&db=iih&AN=27778510&site=eds-live> (Accessed 15 2019).

Lagerberg, L. Skude, T. Emanuelsson, P. and Sandahl, K. (2013). The impact of agile principles and practices on large-scale software development projects. A multiple-case study of two projects at Ericsson. ACM/IEEE International Symposium on Empirical Software Engineering and Measurement – computer society. Available at:
https://www.researchgate.net/publication/261224350_The_Impact_of_Agile_Principles_and_Practices_on_Large-Scale_Software_Development_Projects_A_Multiple-Case_Study_of_Two_Projects_at_Ericsson (Accessed 29 June 2019).

Layton, M. (2012). Agile Project Management for Dummies. New Jersey: John Wiley & Sons, Inc.

Larson, E., Gray, C. (2018). Project Management The Managerial Process. 7th ed. New York: Mc Graw-Hill Education.

Martin, J. (2017) 'Agile Organizational Change: Leveraging Learnings from Software Development', Vol. 49, no. 3, pp. 39–41. Available at:
<http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib,cookie,url&db=bth&AN=124244057&site=eds-live> (Accessed: 05 June 2019).

Maneva, M., Koceska, N & Koceski, S. (2017). Measuring agility in agile methodologies. *Journal of applied economics and business*, Vol 5 / Issue 3, p21-30. Available at:
<http://www.aebjournal.org/articles/0503/050303.pdf> (Accessed February 2019).

Massod, Z., Farroq S (2017). The benefits and key challenges of agile project management under recent Research Opportunities. *International Research Journal of Management Sciences*. Vol 5 / issue 1, p 20-28. Available at:

https://www.researchgate.net/publication/316239082_The_Benefits_and_Key_Challenges_of_Agile_Project_Management_under_Recent_Research_Opportunities (Accessed March 2019).

Misra, Subhas (2012). Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*, Vol.29 /Issue: 9, p972-980. Available at:
<https://www.emeraldinsight.com/doi/full/10.1108/02656711211272863> (Accessed 10th March).

Pandey, A (2015). ‘Case Study: Cisco. SAFe drives continuous delivery, improves, quality, productivity, and employee satisfaction’, *Scaled Agile*. Available at:
<https://www.scaledagileframework.com/cisco-case-study> (Accessed: 29 June 2019).

Papadopoulos, G (2014). Moving from traditional to agile software development methodologies also on large, distributed projects. *Sciencedirect, Social and Behavioral Sciences* , Vol 2, nember, 455-463, Available at :
https://www.researchgate.net/publication/273579892_Moving_from_Traditional_to_Agile_Software_Development_Methodologies_Also_on_Large_Distributed_Projects (Accessed January: 15th, 2019).

Papadopoulos, G., 2015. Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. *Procedia - Soc. Behav. Sci.* 175, 455–463. <https://doi.org/10.1016/j.sbspro.2015.01.1223> (Accessed June 2019).

Petit, Y., Hobbs, B. (2016). Scaling agility adapting agile principles to large projects in large organisations. *Project Management Institute*. Available at :
<https://www.pmi.org/learning/library/adapting-agile-large-organizations-projects-10214> (Accessed July 2019).

Popli, R., Chauhan, N (2016). Mapping of traditional software development methods to Agile methodology. *Computer Science & Information Technology*.V1/ P118-122. Available at:
https://www.researchgate.net/publication/269207459_Mapping_of_Traditional_Software_Development_Methods_to_Agile_Methodology (Accessed January 2019).

Project Management Institute (2013) ‘A Guide to the Project Management Body of Knowledge’ (PMBOK® Guide) , 5th edn. United States: American National, Project .Management Institute

Qureshi, M (2012). Agile software development methodology for medium and large projects. *IET-Journals*, vol 2, p1-6. Available at :
https://www.researchgate.net/publication/260649945_Agile_software_development_methodology_for_medium_and_large_projects (Accessed December 10th ,2018).

Saeeda, H. Ahmed, M. Khalid, H. Sameer, A. and Arif, F. (2015). Systematic Literature Review of Agile Scalability for Large Scale Projects, International Journal of Advanced Computer Science and Applications, Vol.6, No.9/ p.63-74. Available at: https://thesai.org/Downloads/Volume6No9/Paper_8-Systematic_Literature_Review_of_Agile_Scalability_for_Large_Scale_Projects.pdf (Accessed February 11th, 2019).

Saunders, M., Lewis, P. and Thornhill, A. (2016). Research methods for business students. 7th ed. England: Pearson.

Stoica, M. Mircela, M . Ghilic, B. (2014). Software Development: Agile vs. Traditional. Informatica Economica Journal, Vol 17/ p.65-76. Available at : <http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib,cookie,url&db=edsdoj&AN=edsdoj.26bc4d0c5984a45b903410ba39fadbe&site=eds-live> (Accessed February 10th, 2019).

Turetken, O., Stojanov, I., Trienekens, J. (2017) . Assessing the adoption level of scaled agile development: a maturity model for scaled Agile Framework. Journal of Software: Evolution & Process, vol 29/ issue 6, p18. Available at : <http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib,cookie,url&db=iih&AN=123822324&site=eds-live> (Accessed February 2019).

APPENDICES

Interview questions

- 1) Given that you have experience in applying Agile Project Management Methodology in Software development projects. Could you tell me if you are a fan – Do you like working this way?
- 2) Why does your company / project use Agile?
- 3) Can you tell us about any choices your company has made – which bit of Agile do they use and which bits of your process are company specific?
- 4) What benefits do you see from the application of Agile in large projects?
- 5) Do you have experience in applying APM in both small and large software development projects? If so can you talk about the differences in application – are there limitations to agile at scale?
- 6) As a member of an agile software development team, do you think these limitations are related mainly to the culture and policies of company, lack of experience or lack of management support?
- 7) Based on your experience in both Agile and Traditional. How could you describe and compare the benefits of both methodologies in large software development projects?
- 8) What do you think is the main reason that some companies still traditional/waterfall?