“Have you seen the Scrum Master?”: Agile Scrum Deviations and Project Success

par

Salma Hassani Alaoui

Sciences de la gestion
(Analyse d’affaires – Technologies de l’information)

Mémoire présenté en vue de l’obtention
du grade de maîtrise ès sciences en gestion
(M. Sc.)

Juin 2019

SOMMAIRE

Grâce à leur processus itératif et à leur approche collaborative, les méthodes Agiles ont révolutionné l'industrie du logiciel. Au cours des dernières années, l'adoption en industrie des méthodes Agiles a connu une forte progression, privilégiant Scrum comme méthode de choix. Bien que la méthode Scrum ait des directives spécifiques, celles-ci sont souvent modifiées par les praticiens.

Cette recherche exploratoire a été réalisée dans le but de mieux comprendre comment la méthode Scrum est adaptée en industrie, ainsi que les liens potentiels entre les écarts par rapport à la méthode et le succès du projet. Cette recherche vise donc à apporter des éléments de réponses aux questions suivantes: **Comment le cadre Agile Scrum est-il appliqué en industrie? Y a-t-il un fossé entre la théorie et la pratique ? Et y a-t-il un lien entre les déviations par rapport au cadre Scrum et le succès du projet?**

Afin de répondre aux questions de recherche, une approche qualitative a été entreprise au moyen d'onze entrevues semi-structurées. Les résultats de cette étude indiquent une variabilité en termes de conformité aux directives de la méthode. En effet, seul un nombre restreint de directives sont systématiquement adoptées. Par ailleurs, l'association des écarts par rapport à la méthode à des dimensions de succès du projet a permis de relever quatre tendances. En outre, cette étude a permis de soulever d'autres pratiques qui ne figure pas dans les directives Scrum, tel qu'une institutionnalisation de cérémonies de « Grooming» ainsi qu'une mise à échelle des équipes Scrum.

Ces observations sur la manière dont la méthode Agile Scrum est adaptée dans la pratique peuvent aider les professionnels à mieux adapter la méthode Scrum dans des projets de développement de logiciel. De plus, ces résultats s'inscrivent comme une contribution au courant prometteur de recherche sur l'adaptation de la méthode Scrum et ses effets.

Mots Clés : Scrum; Scrum adapté; Scrum en pratique; Adaptation des méthodes Agile; Méthodes Agiles en pratique; Gestion de projet Agile; Succès du projet Agile.

ABSTRACT

Agile methods have revolutionized the modern software development industry through their iterative process and collaborative approach. In the last decades, the adoption of Agile methods has gained momentum in industry, with Scrum being the favored method. While the Scrum method has specific guidelines, these are often changed by practitioners.

This exploratory research was carried out to understand how the Scrum method is tailored in practice and how deviations from the method link to project success. Subsequently this research aims to provide answers to the following questions: **How is the Agile Scrum framework applied in industry? Is there a gap between theory and practice? And how are deviations from the Scrum framework in industry affecting project success?**

In order to answer these research questions, a qualitative approach was undertaken using eleven semi-structured interviews with representatives from organizations who use Scrum in software development projects. The findings of this study indicate variability in the application of the Scrum guidelines, namely, that only a few guidelines are systematically followed, and that some guidelines are rarely followed consistently. Moreover, examining these method deviations and mapping them to specific dimensions of project success led to the emergence of four patterns. Further, this study uncovered practices that are often used in industry but were not part of the original Scrum guidelines, including a formal Grooming ceremony and scaled organization of Scrum teams.

These insights into how the Agile Scrum method is tailored in practice can help industry professionals determine how to best adapt the Scrum method in software development projects. Furthermore, these findings offer a promising agenda for research on Scrum tailoring and its effects.

Key words: Scrum; Scrum tailoring; Scrum in practice; Agile methods tailoring; Agile methods in practice; Agile project management; Agile project success.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Original Term**
IT	Information Technology
CSR	Critical Success Factors

ACKNOWLEDGMENTS

To my eternal cheerleaders, Hasnaa and Ali, because I owe everything I am to you.

First and foremost, my deepest appreciation goes to professor Ann-Frances Cameron, whom I was spoiled to have as a supervisor. I am extremely grateful to you for assisting me into growing an interest into an idea and later a master’s thesis. Thank you for your generous support, your kind-hearted guidance, and for initiating a system that worked for both of us.

Again, I am beyond grateful to my forever loving, enthusiastic, and supportive parents, Hasnaa and Ali, without whom this achievement would have never been possible. Your love is my superpower. Thank you for your silent sacrifices and kind words.

A very special gratitude goes out to my good friends who have provided me with moral and emotional support throughout this journey. Thank you Nisrine and Soukayna for assisting me through my worst days, even across the Atlantic. Many thanks to Antoine, Theophile and Marianne for the best lunch break conversations one could ask for. Thanks should also go to Valérie and Alicia for the last-minute, improvised, yet very helpful, writing retreats.

I am also very grateful to the faculty of the Information Technology department at HEC. Your knowledge, insights, and proximity expanded my vision of this field and made me appreciate it even more.

Finally, last but by no means least, I would like to thank all the participants who have generously agreed to share their experience. Our encounters opened my eyes to the reality of a field I am interested in. Thank you for keeping up with my endless questions. Your contribution is invaluable.

Thank you all!

CHAPTER 1 : INTRODUCTION

1.1 Context

The world in which we live runs on software. From workstations and servers to Internet-of-Things powered sensors and embedded devices, software is present in all forms and shapes in our daily lives (Driver & Klinect, 2018). In addition to products and services, business processes are becoming increasingly dependent on software (Ullah & Lai, 2013; McKinsey & Company, 2015). Software already supports 80% of automobile innovation (McKinsey & Company, 2015). Consequently, embracing information technology, with software projects at its core, is essential to value creation and competition across various industries (McKinsey & Company, 2015). Indeed, a performance gap exists between companies that do leverage technology and those that do not, leaving organizations no choice but to take the leap (Brynjolfsson, 2010). Following this paradigm shift in industries, global information technology spending has increased steadily in the past years, and companies are increasingly investing in their software (Driver & Klinect, 2018; Deloitte, 2017). Indeed, global IT spending is projected to reach $3.8 trillion in 2019 (Garfinkel, 2018), and Gartner estimates that, every year, companies allocate between 20% and 40% of their IT budget to new initiatives (Smith & Proctor, 2013).

Despite significant investments and efforts towards information technology initiatives, success remains a major challenge (Charette, 2005; Smith & Proctor, 2013; Standish Group, 2013; Alami, 2016). In its 2015 CHAOS report, the Standish Group, a primary research advisory organization with a focus on software project performance, studied the performance of 50,000 software projects around the world. Its report revealed that software projects are more likely to fail than succeed based on both project management process (time, budget, target) and the end results of the project itself, with nearly 70% of projects either failing or being challenged (Standish Group, 2015). The report highlights that, in addition to being hard to achieve, success in software development projects is also hard to define (Standish Group, 2015). In fact, although an important theme in project management literature, project success holds no widely agreed upon definition (Jugdev

& Müller, 2005; Lavagon, 2009; McLeod et al., 2012).

To cope with software project challenges, practitioners resort to project management methodologies. In fact, using the appropriate project management methodology in software development is a critical issue and one of the major reasons behind project failure (Alami, 2016; Joslin & Müller, 2015; Shahrbanoo et al., 2012). Agile project management emerged in the 1990s as a set of practices developed by experienced practitioners as an alternative to traditional plan-based methods (Dybå & Dingsøyr, 2008; Birkinshaw, 2018). This values-based approach which emphasizes interactions, customer collaboration, responding to change, and rapid continuous delivery, gained momentum in the software industry following the publication of the Agile Manifesto in 2001 (Birkinshaw, 2018; Dybå & Dingsøyr, 2008; Beck et al., 2001). Owing to the fact that Agile methodologies are more responsive and collaborative than traditional ones, companies adopting them experienced higher productivity within software development teams, faster delivery, and better costumer experiences (McKinsey & Company, 2015; Birkinshaw, 2018). Moreover, several researchers have also established that Agile methodologies contribute to more successful projects (Santos et al., 2013; Dybå and Dingsøyr, 2008; Mann & Maurer, 2005).

Agile is presently entrenched as the most popular project management approach in software development (Reifer, 2017; Rigby et al., 2016). According to the cross-industry State of Agile survey, 97% of the 1,319 organizations surveyed reported practicing Agile development methods (Versionone, 2019). In addition to having revolutionized the software industry, Agile principles have extended into mainstream management thinking and are set to transform work practices in several industries (Rigby et al., 2016). Among companies using Agile, 72% reported using Scrum or a variation of Scrum, making it by far the most popular framework and applied form of Agile in software development (Versionone, 2019).

While the Scrum framework comes with a set of rules and guidelines to follow, these are often modified in practice (West & Grant, 2010; West, 2011). In recent years, a growing body of literature has examined the tailoring of general Agile practices. However, few studies have been published regarding the tailoring of the Scrum framework and its impact on project outcomes such as project success (Ashraf & Aftab, 2017; Hron and Obwegeser 2018). This literature gap is even more relevant given that most organizations choose to adopt the guidelines of a specific methodology rather than general Agile practices (Kiv et al., 2018). Moreover, research has not yet looked into the relationship between Scrum tailoring and project success. In an effort to improve Scrum tailoring in practice, more research is needed regarding how the framework is tailored and how changes to the guidelines link to project outcomes. Several questions remain unanswered such as: What are the practices adopted in Agile Scrum practice, and which ones are most likely to be ignored or modified and why? Are there any additional practices initiated by the teams? Can these changes impact project outcomes? Is there a link between the Agile Scrum framework guidelines and project outcomes?

1.2 Research questions and objectives

To address the research gap outlined above, the objective of this research is to understand how the Scrum framework is tailored in practice, and whether specific modifications made to the Scrum guidelines impact project success. This research also contributes to the discussion regarding how Agile frameworks can be best tailored to accommodate organizational constraints while embracing Agile values and contributing to project success. In so doing, our research questions are: **How is the Agile Scrum framework applied in industry? Is there a gap between theory and practice? And how are deviations from the Scrum framework in industry affecting project success?**

1.3 Thesis structure

This research is structured in five chapters. This first chapter introduces the context, research questions and research objectives. The second chapter consists of a literature review that provides a global overview of the main topics relevant to the research questions. Therefore, the literature review helps lay the foundations of project management with Agile Scrum and project success. The third chapter describes the research methodology adopted, namely the data collection and analysis. The fourth chapter presents the respondents' profiles in addition to the detailed results following data analysis. Finally, the last chapter concludes this research by providing a synthesis of the main results and the contributions made to the scientific literature and practice. Furthermore, the last chapter discusses the research limits and future research opportunities

CHAPTER 2 : LITERATURE REVIEW

The literature review chapter lays the necessary theoretical foundations regarding the most important concepts related to the research questions stated above. Accordingly, this chapter will start by covering a brief history and definition of project management, before diving into the concept of project success and how it has evolved in the scientific literature. Further, this chapter will provide a detailed breakdown of the agile project management approach, its values, principles, and how it compares to the traditional approach. More specifically, the agile Scrum framework will be demystified and clarified. Finally, this chapter will also give detail of the literature on agile and success, as well as agile on practice.

This literature review was elaborated through researching several databases such as ABI/INFORM, IEEE Xplore, ScienceDirect and SpringerLink. Following an iterative process, key words were formed and fed into the search engines. Key words included but were not constrained to: “Information Technology project management”, “Project success”, “Agile project management”, “Scrum software development”, “Tailored agile”, “Agile in practice”, “Scrum in practice”, “Tailored Scrum”. Additionally, forward and backward searches were conducted through search engines like Google Scholar to identify related themes and keywords. The findings were organized, synthesized by themes and presented in this chapter.

2.1 Project Management

While examples of project management go back as far as Ancient Egyptian times and the construction of pyramids, researchers seem to agree that the term “Project Management” only emerged in the latter half of the 20^{th} century with the Manhattan Project (Weaver, 2007; Lenfle & Loch, 2010). With its principles of planning, organization and direction, the Manhattan Project, which goal was to build an atomic bomb in the 1940s, is considered the first “project” and the origin of modern project management standards (Weaver, 2007; Lenfle & Loch, 2010). It was not until 40 years after the Manhattan

Project, in the 1980's, that project management started being applied to the field of software development (Kwak, 2005). Currently, project management is an essential way of organizing work in most organizations (Bakker, 2010). It is a recognized discipline applied across different sectors and industries on a global level (Turner & al., 2010).

The project management institute, which is considered the most influential association governing the project management discipline, defines project management as the "application of knowledge, skills, tools, and techniques to project activities in order to meet the triple constraints of scope, time, and cost" (Project management institute, 2018). With that being said, several variations of the definition of project management exist in literature. The table below assembles key definitions found in literature:

Researcher(s)	Definition of Project Management
Oisen (1971) in Atkinson (1999)	"Project Management is the application of a collection of tools and techniques […] to direct the use of diverse resources toward the accomplishment of a unique, complex, one-time task within time, cost and quality constraints. Each task requires a particular mix of these tools and techniques structured to fit the task environment and life cycle (from conception to completion) of the task."
Turner (1996)	"The art and science of converting vision into reality."
Atkinson (1999)	"Project management is a learning profession. Based upon past mistakes and believed best practice, standards such as BS 60794 and the UK Body of Knowledge continue to be developed."

Ur Rehman & Hussain (2007)	**“**Project management is an iterative process that can be considered as ‘a lock-step sequence of activities with the application of knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholders needs and expectations.’**”**

Table 1 Definitions of project management in the literature

Several contributions have been made to the definition of project management. Early definitions focus mainly on the iron triangle: the constraints of time, cost, and scope (Atkinsson, 1999). In that regard, project management is used as a tool to achieve project efficiencies and focused on aspects of executability, temporality, linearity, controllability and instrumentality (Svejvig & Andersen, 2015). More recent definitions introduce “soft skills” encompassing stakeholders, communication, and leadership (Weaver, 2007). Project management thus evolved from being a set of tools and techniques to being considered a holistic discipline to attain organizational effectiveness, efficiency and innovation involving cultural and interpersonal aspects (Jugdev & Al., 2001).

2.2 Project Success

While all projects aim to be successful, the meaning of success varies in the scientific literature. Indeed, although project success makes for an important and frequently mentioned topic in both the IS and project management literatures, researchers have yet to reach a consensus on its definition (Ika, 2009). The definition of project success has evolved over time, progressively widening its scope as a result of being refined through several researchers’ contributions (McLeod et al., 2012). While project success has for a long time only been concerned by the ability to respect time, cost, and quality constraints, examples abound of projects that have been successful in meeting these constraints and still considered to be failures (Ika, 2009). In this regard, Collyer and colleagues (2010) give the example of Iridium, a multibillion-dollar project to build and launch a satellite infrastructure that succeeded in regard to time, cost and quality, only to be considered an utter failure from a business perspective. On the other hand, several projects exceeded budget and time constraints, but were still considered successful, or as the literature sometimes call them, "successful failures" (Nelson et al., 2014). Project success is therefore linked to broader objectives than simply time, budget or cost. This has led to a notable contribution to project management literature on project success: the distinction between project management success and project success, the latter encompassing the former. Accordingly, Cooke-Davies (2002) distinguishes between the two terms as follows:

> **Project management success**: measured against traditional performance measures such as time, cost, and quality.
> **Project success:** measured against the overall objectives of the project.

This means that project management success can contribute to the project's success but does not guarantee avoiding failure (Davies, 2002). It also means that achieving project success is more demanding than achieving project management success (Davies, 2002). This could be illustrated by the saying, "The operation was a success, but the patient died."

Despite this distinction, project success remains a challenging concept to define. Ika (2009, p. 8) affirms that while project success is a vague concept, it does encompass two definite notions; efficiency and effectiveness:

> The concept of project success remains vague, to the point that the literature on project management does not reach a broader consensus on its definition and measurement than to say that it involves efficiency and effectiveness.

Efficiency and effectiveness are both considered goal-oriented practices related to achieving success (Belout, 1998). Efficiency means maximizing output for a certain level of input, whereas effectiveness looks at reaching goals and objectives. More broadly, effectiveness is known as doing the right thing, and efficiency as doing things right. (O'Shaugnessy, 1992).

Overall, project success has evolved throughout the years from definitions being solely concerned about the implementation phase of the project life cycle to more holistic ones that reflect the whole project and product life cycle (Jugdev & Müller, 2005). Therefore, project success, like project management, has gone from an operational view with the three factors making up the Iron Triangle to a strategic concept (Shenhar et al., 2001). The rest of this section explores the evolution of the definition of project success through three major time periods (inspired by the work of Jugdev & Müller, 2005; Ika, 2009; Mcleod et al., 2012; Davis 2014, 2018; and Albert et al., 2017).

2.2.1 The Iron Triangle (1960s - 1980s)

The Iron Triangle, which is widely known for the three constraints of time, cost and quality (see figure below), represents the early views on project success (Atkinson, 1999; Jugdev & Müller, 2005; Ika, 2009; Albert et al., 2017). These three constraints are also known as the “Triple Constraint”, “Golden Triangle”, “Triangle of virtue”, “Holy trinity”, or simply “time/cost/quality triangle” and focus on process success at the implementation phase (Ika, 2009).

Despite the fact that, according to the definition of Cooke-Davies (2002) presented earlier in this chapter, the three constraints of the triangle are only concerned with project management success, these constraints were used to assess overall project success. The Iron Triangle is therefore considered as a foundation in project management (Atkinson, 1999). Wateridge (1998) justifies the emphasis on time, cost, and quality due to the nature of the work of project managers, who are evaluated on their ability to deliver to these short-term variables. Out of the three constituents of the Triangle, Tukel & Rom (2001) claim that quality is project managers’ primary success measure. Quality, according to Tukel & Rom (2001) “is found to be associated with not only customer focus and rework reduction but also with conformance to technical specifications” (Tukel & Rom, 2001, p. 412).

It is worth mentioning that, while there is a general consensus about the time and cost aspects of the Triangle, the quality vertex of the Triangle has been contested by several researchers (Pollack et al., 2018). While quality is the most used designation for the third vertex, some researchers believe it should be termed “Scope”, “Performance”, or “Requirements” instead (Pollack et al., 2018). In their exploration of the evolution of the concept of the Iron Triangle, Pollack and colleagues (2018), find that the concepts of quality, cost and time were highly correlated in the scientific literature and that the level of interconnection between them is significantly greater than that of the other term used to replace quality.

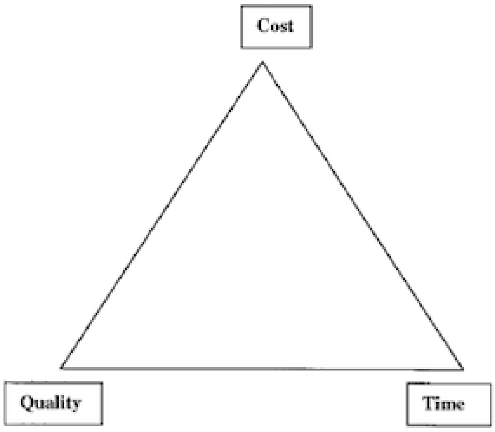


Figure 1 The Iron Triangle

Although a central concept to project management literature, the Iron Triangle receives a lot of criticism (Pollack et al., 2018). Firstly, limiting success to time, cost and quality while overlooking product or service value gives project management a tactical operational value rather than a strategic one (Jugdev & Müller, 2005). Shenhar and colleagues raise the same issue, asserting that the assessment of time, budget and performance in early project management literature reflects an operational mindset. Moreover, Atkinson (1999) considers the Iron Triangle to be incomplete and a type II error in that we are omitting other criteria of success and thus wrongly evaluating project success. He argues that time and budget are nothing more than guesses at the early stages of the project, while quality is a phenomenon that changes across the project life cycle based on people attitudes and beliefs (Atkinson, 1999). In their study, Nelson and colleagues (2014) illustrate the claim by Atkinson (1999) by showing that most IT projects were late (61%), and 40% were over budget. However, for the same projects, the overall stakeholder satisfaction was rated at 91%, making 9 out of 10 projects “successful failures” as they eventually meet the requirements and add value to the organization. Nelson and colleagues (2014, p. 25) conclude that the main reason that these projects fail on process criteria is poor estimation. Indeed, results suggest that “classic mistakes” are made in 64% of projects, causing poor estimations for reasons such as poor problem/ requirement definition, scope creep, lack of requisite knowledge and skills, inadequate estimation methodologies, lack of historical data, lack of visibility or control, and pressure from stakeholders. These results validate the claim of Atkinson (1999) that we cannot rely on time and budget, as they are two guesses made at early stages. A different angle of criticism is offered by Van der Hoorn and Whitty (2015), who declare that the Iron Triangle creates an overly-simplified picture of project work and project managers’ responsibilities. This oversimplification reinforces a thinking that is not aligned with the reality of project practice, causing anxiety among practitioners over unrealistic project control expectations (Van der Hoorn & Whitty, 2015). In spite of the criticism, the Iron Triangle is still considered a preeminent approach to evaluating project success and an essential building block for the definitions that followed (Pollack et al., 2018; Jugdev & Müller, 2005).

2.2.2 Critical Success Factors: Lists and Frameworks (1980s- 2000s)

This second period is characterized by a shift in the focus of the project success literature from trying to assess whether a project is successful to trying to determine the precursors of successful projects (Jugdev & Müller, 2005; Ika, 2009). Accordingly, researchers attempt to answer the question: “What do we need in order for a project to be successful?” Moreover, stakeholders’ satisfaction emerges as a significant indicator of project success, particularly client satisfaction (Jugdev & Müller, 2005). Indeed, the competitive marketplace and the focus on service and satisfaction highlights the importance of client’s satisfaction (Jugdev & Müller, 2005). Therefore, in addition to the criteria of time, cost, scope, and quality, the literature starts focusing on product use, client satisfaction and client and organizational benefits (Mcleod et al., 2012; Ika 2009).

This period can be further divided into two sub-periods as Critical Success Factors lists emerge first, followed by Critical Success Factors frameworks. In the period from the 1980s to 1990s, several Critical Success Factors are identified and described (Jugdev & Müller, 2005). However, Critical Success Factors' lists are produced unmethodically, and are not organized or grouped in a coherent manner, but are rather divided intuitively (Davis, 2014). Pinto and Slevin (1987) are widely recognized for their list of ten Critical Success Factors (Jugdev & Müller, 2005) comprising of: clear project mission, top management support, schedule and plans, client consultation, personnel, technical tasks, client acceptance, monitoring and feedback, communication, and troubleshooting.

Several contributions are made in terms of Critical Success Factors lists, mainly focusing on ensuring operational (Iron Triangle) and stakeholders' success (Jugdev & Müller, 2005). However, researchers later agree that it is impossible to come up with an exhaustive list of success criteria that could work for the needs of all projects (Jugdev & Müller, 2005; Ika, 2009). According to Ika (2009, p. 9), that is due to the fact that CSFs change according to the variables of each project:

> This stems directly from the fact that success criteria and CSFs can differ so much from one project to another due to variables such as project scope, uniqueness, and complexity.

In addition, Pinto and colleagues (1988) argue that even within the same project, critical success factors are not equal, as they vary across the different stages of the project life cycle.

In the 1990s, significant contributions are made to project success literature with the emergence of Critical Success Factors frameworks, and the recognized importance of success as being dependent on internal and external stakeholders (Davis, 2012). Morris and Hough (1987) are considered pioneers in the development of a comprehensive framework of project success determinants, based on eight case studies (Jugdev & Müller, 2005). Accordingly, they divide project success into four categories, answering four questions, as presented in the following table:

Project Success Category	Question
Project functionality	Does the project meet financial and technical requirements?
Project management	Did the project meet the budget, schedule, and specifications?
Contractors' commercial performance	Did the contractors benefit commercially?
Project termination	In the event that the project had to be cancelled, was this decision made reasonably and efficiently?

Table 2 Project Success framework (Morris & Hough, 1987)

In a similar effort to provide more structure to the concept of project success, Belassi and Tukel (1996) categorize the individual success factors found across literature. In addition to providing a categorization of critical success factors, Belassi and Tukel (1996) also attempt to represent the interaction between these factors. Accordingly, their classification comprises four categories of factors: factors related to the project, factors related to the project manager and team, factors related to the organization, and factors related to the external environment (Belassi and Tukel, 1996). Atkinson (1999) takes categorization of CSF a step further, and suggested the Square-Route as a more balanced and realistic way to look at project success. Ensuing from a literature review and analysis, Atkinson (1999) forms his Square-Route model, which starts with the Iron Triangle as a base and adds three other success criteria groups.

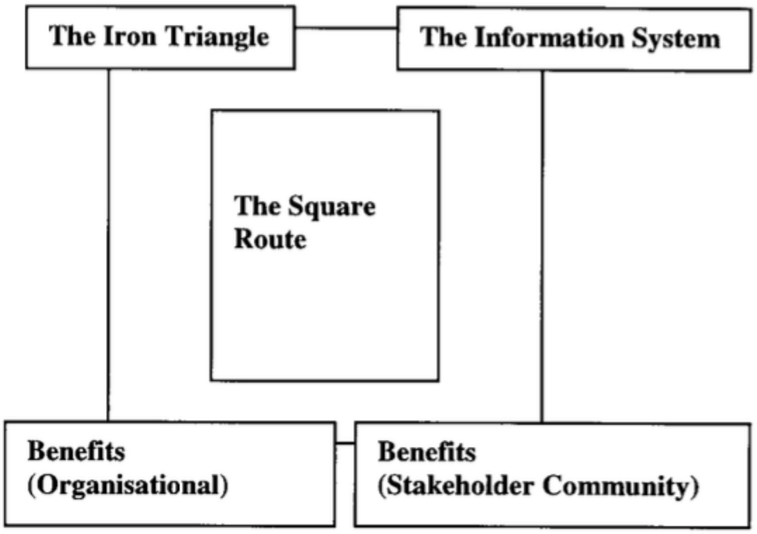


Figure 2 Square-Route model for project success (Atkinson, 1999)

In the Square-Route model, Atkinson (1999) attempts to group and categorize criteria of project success proposed by other authors, namely: Turner (1993), Morris and Hough (1987), Wateridge (1998), deWit (1988), McCoy (1987), Pinto and Slevin (1988), Saarinen (1990) and Ballantine (1996). He suggests three new categories to classify these criteria: the technical strengths of the resulting system, the benefits to the resulting company, and the benefits to a wider stakeholder community (Atkinson, 1999). Benefits to the resulting company are also called direct benefits, while the benefits to the stakeholders are considered indirect (Atkinson, 1999). The information system category includes criteria such as maintainability, reliability, validity, information quality and use of the system (Atkinson, 1999). Benefits to the organization, on the other hand, include criteria such as improved efficiency and/ or effectiveness, increased profits, and reduced waste (Atkinson, 1999). Finally, the category concerning benefits to the stakeholder community carries criteria such as satisfied users, social and environmental impact, personal development, and professional learning (Atkinson, 1999).

In essence, between the 1980s and the 2000s, several researchers contribute to the project success literature, first through the development of CSF lists, and later with CSF frameworks. This has widened the scope of project success from the then-dominant triple constraints view of the Iron Triangle. Indeed, CSF lists and frameworks enable the introduction of stakeholders and the external environment, providing a more comprehensive, less technical, and more realistic view of project success.

2.2.3 Holistic View of Success- 21st Century

The literature of the 21st century expands the definition of project success even further, considering project management as a strategic asset to the organization (Jugdev & Müller, 2005). Thus, the dimensions of project success transcend the benefits to the organization and become concerned with strategically "preparing for the future" through contributing to future innovations and development of core competencies (Jugdev & Müller, 2005).

Accordingly, several researchers enrich the existing views on project success with an additional strategic aspect that goes beyond the project's implementation phase. For instance, Shenhar and colleagues (2001) emphasize on the importance of projects as strategic assets in a company, labeling them as "powerful strategic weapons". In developing their multidimensional framework for assessing project success, they identify four dimensions to success, namely project efficiency, impact on the customer, direct business and organizational success, and preparing for the future. Shenhar and colleagues (2001) add that success dimensions vary according to time and technological uncertainty within the project. Project efficiency is concerned with meeting schedule and budget goals, which are two vertices of the Iron Triangle (Shenhar et al., 2001). The impact of the customer dimension includes meeting the needs of the customer, as in the technical and functional requirements. It is concerned with satisfying the customer, solving the customer's problem and having them use the final product. Business success regards the benefits of the performing organization which can range from additional profits to increasing market share and launching new product lines (Shenhar et al., 2001). Finally, preparing for the future addresses the question: "How does the current project help the organization face future challenges?" (Shenhar et al., 2001). Their framework illustrates how these dimensions vary according to time and technological uncertainty:

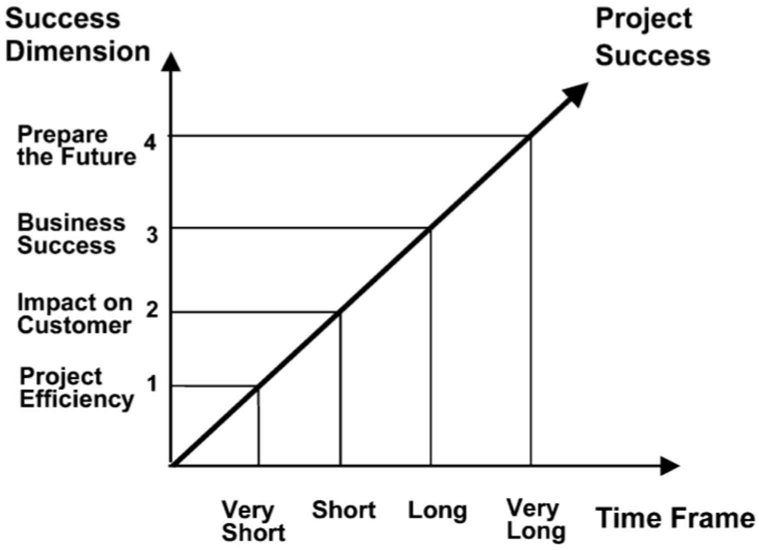


Figure 3 Time frame of success dimensions (Shenhar et al., 2001)

Their framework displays the relative importance of these success dimensions as contingent on time. Shenhar and colleagues (2001, p. 717) assert that "different dimensions are more important at different times with respect to the moment of project completion". For example, while the first dimension, project efficiency, is highly relevant in the very short term, meaning during the project's execution and right after it is done, the fourth dimension can only be assessed in the long run. Shenhar and colleagues (2001) estimate that to be two, three, or even five years after the project has been completed.

While their first framework presented relevant project success dimensions as time-dependent, Shenhar and colleagues (2001) presented another framework according to which the importance of project success dimensions varies depending on the type of project. To determine the typology of projects, Shenhar and colleagues (2001) use technological uncertainty as a way to differentiate between projects and demonstrate that technological uncertainty affects the relevance of the four project success dimensions previously discussed. Shenhar and colleagues (2001) thus argue that for projects with lower technological uncertainty, the efficiency dimension is highly relevant, since their expected benefits can be predicted in advance, and their immediate success relies on meeting the schedule and budget goals. For projects with higher technological uncertainty, on the other hand, long-term benefits make up for short-term poor performance (Shenhar et al., 2001).

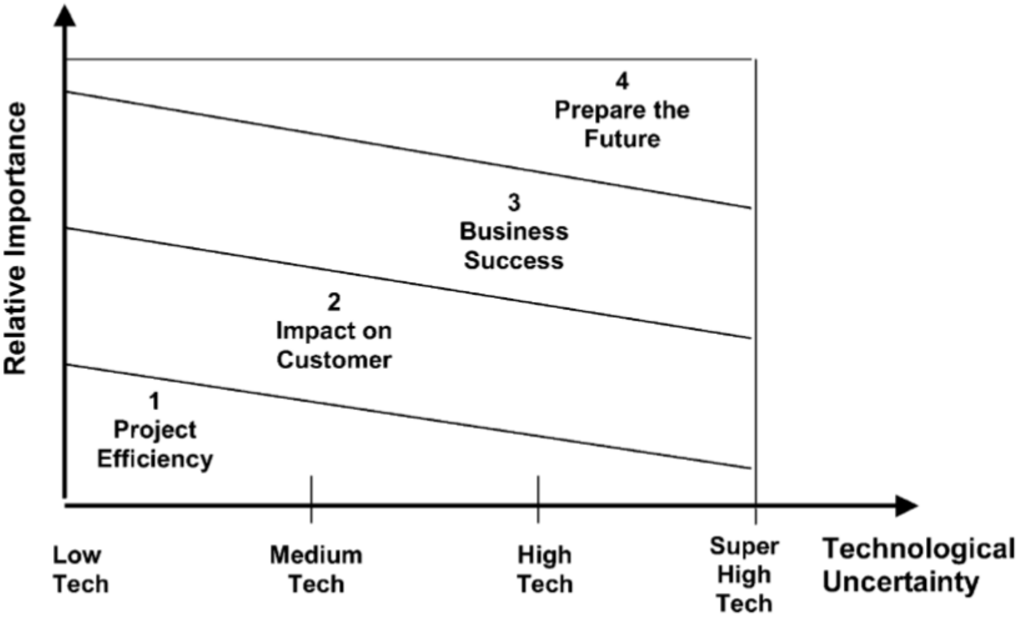


Figure 4 Relative importance of success dimensions according to project type (Shenhar et al., 2001)

Similarly, Nelson and colleagues (2014) also highlight the element of preparation for the future in their framework by encompassing groups of factors that go beyond the project management process and also assess the outcome of the project. Starting with the Iron Triangle as a way to measure process success, Nelson and colleagues (2014) add another triangle to assess the project outcome. These outcome criteria include the use of the system by the targeted users, the impact on stakeholder knowledge, the impact on the organization’s readiness for future challenges, and the impact on effectiveness and efficiency.

Recent project success literature is also characterized by the growing recognition of the importance and responsibility of other stakeholders, particularly project owners, in project success (Jugdev & Müller, 2005; Davis, 2014). It is worth mentioning that while the terms “project owner” and “project sponsor” are sometimes used interchangeably, Tuner and Zolin (2012) distinguish between the two roles: project owners are investors with whom the main point of contact is at the beginning of the project. The contact with project sponsors, on the other hand, is throughout the whole project, meaning before, during, and after the project. Moreover, given that senior management is responsible for linking organizational plans with project goals, their commitment is established as essential throughout the whole project and not just at the start (Jugdev & Müller, 2005; Davis, 2012).

This century has therefore seen the literature moving away from focusing entirely on the project manager, to focusing on stakeholders' satisfaction and how to reconcile different stakeholders' perspectives. This opens up a new area of research interest focusing on stakeholders and their different expectations and definitions of project success. Jugdev and Müller (2005) suggest this opens an area of research on stakeholders' perspectives. In line with Jugdev and Müller (2005), Davis (2018) presents a new model that comprises critical attributes in measuring project success across different stakeholder groups. His research supports the claim that stakeholders' views must be taken into account at different stages of the project. The objective of this stakeholders' approach to success is to establish that stakeholders' opinions are valued and taken into consideration, therefore enhancing motivation and creating a more productive environment, the ultimate objective being an agreed upon successful project delivery (Davis, 2018).

While there still exists no common definition of project success (Davis, 2014; Albert et al. 2017), Jugdev & Müller (2005) claim that most recent literature outlines four conditions for project success. While essential to project success, these conditions do not guarantee successful project outcome. These project success conditions are stated by Turner (2004, p. 350) as follows:

> 1. Success criteria should be agreed on with the stakeholders before the start of the project, and repeatedly at configuration review points throughout the project
> 2. A collaborative working relationship should be maintained between the project owner (or sponsor) and project manager, with both viewing the project as a partnership
> 3. The project manager should be empowered with flexibility to deal with unforeseen circumstances as they see best, and with the owner giving guidance

as to how they think the project should be best achieved
4. The owner should take an interest in the performance of the project.

These conditions reflect the liability of the project owner in the overall project success. Moreover, they also show a more flexible perspective on project success criteria, as these can be renegotiated with the project owner and other stakeholders throughout the project life cycle (Jugdev & Müller, 2005). These findings support the perspective-based view of project success of McLeod and colleagues (2012), according to which stakeholder perspectives impact the perceived outcome of a project and that project success is evaluated both according to stakeholders and across time. Furthermore, these findings also encourage the reconciliation of different stakeholders' perspectives as stressed by Davis (2018).

In conclusion, the concept of project success is considerably discussed in project management literature. While there is still no consensus on its definition, project success has gradually widened in scope as a result of several research contributions to the topic. While the Iron Triangle remains a fundamental basis for evaluating project success in both research and practice, additional criteria have become increasingly relevant to the understanding of project success. This latter has therefore evolved from a sum of three technical, process-oriented measures, to a more comprehensive and strategically relevant construct. The table below synthesizes the different aspects of project success that have been discussed:

Success Aspect	Elements
Project Management Success	Quality Budget Scope Schedule
Stakeholders Success	Customer satisfaction Senior Management/Sponsor satisfaction Team satisfaction
Direct Organizational Success	Benefits to the organization
Strategic Success/Value	Long-term advantages

Table 3 Synthesis of the different aspects of project success

2.3 Software development methodologies

Ur Rehman and Hussain (2007, p. 1) define software development methodologies as follows:

> A methodology provides a strategic level plan for managing and controlling IT project. It is a combination of interrelated processes which tells us "what should be done?" but not how it has to be done?" Therefore, it is the methodology that is adapted to work. It is serves as a template to initiate planning and development and implementation phases of projects.

In recent decades, new software development methodologies referred to as Agile have been put forth as an alternative to traditional waterfall methodologies (Thummadi et al., 2011). Although often considered opposites, the traditional approach and the Agile one share the same goal: delivering good quality software in an efficient way (Palmquist et al., 2013). Essentially, they are two different ways of doing the same things: defining, gathering, analyzing, designing, coding, testing, releasing, and maintaining (Palmquist et al., 2013). The following sections present the two approaches, their characteristics, and how they compare to each other.

2.3.1 Traditional software development

The traditional approach of developing software encompasses several methodologies such as the waterfall method, V-Model and RUP. Methodologies classified as traditional are based on a series of steps that are carried out in a sequential way, such as requirements definition, solution development, testing and deployment (Leau et al., 2012). Because of the heavy aspect of the plan-driven approach of these methodologies, they became known as “heavyweight methodologies” (Awad, 2005). This sequential lifecycle is commonly referred to as “waterfall”; however, several variations exist, and there are different traditional methodologies like the Spiral Model, Rational Unified Process, and the V-model (Deemer et al., 2010; Leau et al., 2012).

2.3.1.1 The heavyweight software development process

Heavyweight software methodologies rely on predefined processes and extensive documentation throughout the project (Leau et al., 2012). Such methodologies are managed in a phase-by-phase manner, as every predefined process needs to be fully completed before moving to the next one. The first step of any heavyweight methodology consists of a comprehensive requirements elicitation, in addition to the elaboration of a plan regarding the length of time of the project as a whole and that of each phase. In this first step, the team also tries to forecast any problems they might run into throughout the project. After the requirements are set up, the next step is the design and architecture. In this step, the team formulates a design and architecture plan by producing a technical infrastructure using models and/or diagrams. The design and architecture serves as a road map for the implementation phase. Next, the team moves into the development phase, in which the software is coded. The testing phase may overlap with the development in order to fix software issues early on and avoid future complications. Then, towards the end of the project, when the code produced is close to meeting the specified plan, the customer is included in the tests and prompted for feedback. Once the customer is satisfied with the tests, the project is deemed ready and can be delivered.

Heavyweight methodologies share four similar characteristics (Awad, 2005):

1) *Predictive Approach*

   Heavyweight methodologies tend to plan out significant portions of the software process at early stages. The plans are generally in great detail and cover a long period of time. Accordingly, this approach focuses on planning how to efficiently solve the needs of the system before moving to implementation.

2) *Comprehensive Documentation*

   Traditional software development gives importance to documentation and gathering requirements. It relies on the assumption that all customers' requirements can be gathered prior to the development phase.

3) *Process Oriented*

   Heavyweight methodologies rely on their process. It is therefore crucial for them to define well-working processes consisting of tasks to be performed by the different team members according to their role and responsibilities. Moreover, for each task, a detailed procedure is outlined.

4) *Tool Oriented*

   Heavyweight software development methodologies rely on different project management tools for the completion of tasks. Examples of these tools include Gantt charts and applications such as Microsoft Project (Deemer et al., 2010).

As previously mentioned, traditional software development methodologies include an umbrella of different methods. This review will, however, only focus on the waterfall methodology.

2.3.1.2 The Waterfall methodology

In the 1960s, the general approach to software development was described as “code and fix” (Awad, 2005). Accordingly, the process of software development would consist of only two steps: the first step would be coding, and the second one would consist of fixing all the bugs and problems. Nonetheless, as software grew in size and complexity, this method was no longer efficient or sustainable (Awad, 2005). In 1970, the computer scientist Winston Royce proposed the waterfall methodology in his paper called “Managing the Development of Large Software Systems” (Marchewka, 2014). The term “waterfall” is used as a metaphor to represent cascading activities moving from one phase to the other, where one phase should be completed before beginning the next one (Marchewka, 2014).

The waterfall model, like other heavyweight methodologies, is a sequential logical flow of software development activities comprised of phases (Marchewka, 2014). Each phase has a set of activities and deliverables that must be completed before moving to the next phase. For example, design activities can only begin once the requirements are elicited and defined. In practice, there may be some iteration in which the developers have to return to previous stages, but these iterations are neither wanted nor easy. The waterfall model, just like other traditional models, assigns a lot of time and effort to the early stages of requirements’ elicitation and design in order to get them right and avoid expensive modifications at future stages of the project (Marchewka, 2014). The waterfall model is still used nowadays, especially for large governments systems (Marchewka, 2014).

It is worth mentioning that Winston Royce, considered the father of the waterfall approach, asserts the need for iterations in his model, suggesting a more iterative version of the waterfall model (Larman & Basili, 2003; Palmquist et al., 2013). He even describes his model, if used as is, as being risky and an invitation for failure (Royce, 1970). Hence, Royce (1970) advises running the model at least twice before delivering the software to the customer, especially in terms of design and operations areas, to ensure success. Palmquist and colleagues (2013) explain that the first run of the model is meant for prototyping and understanding the customer's requirements and the technologies involved in order to deliver the most appropriate software to the customer.

2.3.2 Agile software development

Agile methods are a set of practices in software development created by experienced practitioners in an effort to make better, faster and cheaper software (Dybå & Dingsøyr, 2008). Indeed, as the market raised expectations for highly innovative software delivered in short time spans while embracing ever-changing environments, Agile methods came about as a response to these market expectations when traditional methods were not performing as required (Highsmith & Cockburn, 2001). As discussed in the previous section, traditional methods assume that rigorous work during the early stages is needed to anticipate all the requirements and reduce cost by developing a plan, sticking to it, and reducing changes. However, development teams realized that major changes in requirements, technology and scope could arise outside of their control (Highsmith & Cockburn, 2001). Moreover, as discussed in the section addressing success in project management literature, the emphasis shifted from conforming to plan, i.e. respecting scope time and budget, to satisfying the customer at the time of delivery. For these reasons, the need for methodologies that proactively embrace change while focusing on customer value emerged, resulting in Agile methodologies.

2.3.2.1 Agility

Before outlining Agile methodologies, it is important to clarify the concept of “Agility” in the context of software development. The Cambridge Dictionary defines agility as “the ability to move about quickly and easily” (Agility, n.d.). In project management, this ability to move quickly and easily has to do with implementing project management frameworks that allow for the embracing of change and constant readiness to face it during the project. The concept of agility in ISD has no unified or widely accepted definition (Abrahamsson et al., 2002; Hummel, 2014).

According to Hummel (2009), the most commonly used research-based definitions are those of Conboy (2009) and Abrahamsson and colleagues (2002). Abrahamsson and colleagues (2002, p. 19) review several definitions in literature before proposing this comprehensive definition:

> What makes a development method an agile one? This is the case when software development is incremental (small software releases, with rapid cycles), cooperative (customer and developers working constantly together with close communication), straightforward (the method itself is easy to learn and to modify, well documented), and adaptive (able to make last moment changes).

Conboy (2009), on the other hand, explores facets of agility based on definitions from related fields before proposing a definition. He describes the difference between agility, flexibility and leanness. While leanness is the about the perceived value provided to the customer, and flexibility is about being rapidly adaptive to change, Conboy (2009, p. 340) says that both flexibility and leanness are parts of agility, which he defines as:

> The continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment.

While Conboy’s (2009) definition presents more of the social and technical aspects of Agile methodologies, both definitions stress the importance of adaptability that comes with agility in project management. Other authors also draw on flexibility and reactivity to change in their definition of agility. Boehm and Turner (2004) claim that agility is the opposite of discipline. As with an artist that improvises during their performance, Boehm and Turner (2004, p. 1) put forth that agility lets engineers adjust to new technologies and changing needs. Accordingly, their definition of agility is as follows:

> Agility applies memory and history to adjust to new environments, react and adapt, take advantage of unexpected opportunities, and update the experience base for the future.

Finally, Qumer and Henderson-Sellers (2006, p. 505), like Boehm and Turner (2004), also note using past experience and learning to accommodate new challenges in their definition of agility:

> A persistent behavior or ability of a sensitive entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, uses economical, simple and quality instruments in a dynamic environment and applies updated prior knowledge and experience to learn from the internal and external environment.

In the four definitions presented, the concepts of cooperation, flexibility, speed, leanness, learning and responsiveness seem to emerge as constituent of agility. In an effort to capture the different dimensions that make up agility in ISD, Vial and Rivard (2015) review the literature and come up with four facets of agility in information system development: flexibility, cooperation, learning, and leanness. Responsiveness and speed are also sometimes mentioned as part of agility, although Vial and Rivard (2015) consider them to be included in the flexibility dimension.

2.3.2.2 The Agile Manifesto

In 2001, a group of software practitioners published the Agile Manifesto, a document comprising four core values and twelve principles that govern the Agile information systems development process. The objective was to move away from heavyweight, documentation-driven methodologies and come up with alternative ways to develop software (Beck et al., 2001). The group included representatives, who labeled themselves “The Agile Alliance” and stressed the importance of individuals as the most important asset (Beck et al., 2001). Moreover, they stated that Agile is not anti-methodology.

Rather, Agile ISD principles want to restore credibility to methodologies as they recognize the importance of documentation and planning, but also recognize their limits in turbulent environments (Beck et al., 2001). In essence, the Agile Manifesto invites one to think of methodology in a practical way. Therefore, steps are undertaken for their usefulness and are not overdone (Beck et al., 2001). For example, they state that documentation shall be embraced, but not “hundreds of pages of never-maintained and rarely-used tomes” (Beck et al., 2001, p. 5). The publication of the Agile Manifesto resulted in increasing popularity and interest regarding Agile methodologies and initiated unprecedented changes to the field of software development (Vial & Rivard, 2015; Dybå & Dingsøyr, 2008).

2.3.2.3 Agile values

There are four central values to Agile presented in the Manifesto (Beck et al., 2001, p.1):

1. **Individuals and interactions** over processes and tools.
2. **Working software** over comprehensive documentation.
3. **Customer collaboration** over contract negotiation.
4. **Responding to change** over following a plan.

The first value in the Manifesto for Agile Software Development is valuing people and interaction over processes and tools. Agile methods have been known to put great emphasis on the people factor. Cockburn and Highsmith (2001) affirm that putting great emphasis on the people factor, that is, amicability, talent, skill, and communication, is the most important implication for Agile managers. While processes and tools are important, Agile considers real value to come from individuals and how well they work together. In Agile methodologies, this emphasis is reflected in close team relationships, working environment arrangements, and other procedures boosting team spirit (Abrahamsson et al., 2017).

The second value in the Manifesto is to privilege working software over documentation, as the objective of Agile methodologies is to continuously deliver tested working software (Abrahamsson et al., 2017). While documentation is important, Agile’s main advantage is speed. Accordingly, there are frequent new releases and developers are urged to deliver working code while keeping the documentation to an appropriate level (Abrahamsson et al., 2017). One of the biggest misconceptions regarding Agile is that this second value means “no documentation” (Dybå & Dingsøyr, 2008). Rather, it means to only document what is useful and necessary. The aim is to minimize wasteful documentation, which is considered unnecessary work (Dybå & Dingsøyr, 2008). Moreover, the frequent interactions between people compensate for the smaller amount of documentation (Cockburn & Highsmith, 2001).

Next is valuing customer collaboration over contract negotiation. Here again, contracts are certainly important, and their importance is correlated with the size and complexity of the software (Abrahamsson et al., 2017). However, contracts do not replace communication. This Agile value is about ensuring delivery of value to the customer by continuously cooperating with them and making sure they guide the evolution of the end product (Dybå & Dingsøyr, 2008). This way, we significantly reduce the risks of non-fulfillment of the contract. For Cockburn and Highsmith (2001), it is about having all key players on the same team (the sponsor, customer, user, and developer) so they can go in the same direction.

Finally, Agile acknowledges changing realities and embraces uncertainty during IS development, which is why the fourth value is about responding to change over following a plan. Changes can come from various sources: changing customer priorities, business environment, technology evolution, and so on. That makes change almost inevitable during the process of software development. While some planning is essential in every project, most plans are out of date within a few days, which is why the team should be ready to make changes (Cockburn & Highsmith, 2001). Moreover, because Agile development teams are made of both competent software developers and well-informed customer representatives, they are better prepared to face changes. Furthermore, contracts in Agile development should also be designed to support and accommodate such changes (Abrahamsson et al., 2017).

Last but not least, it is important to note that the Agile Manifesto stresses that there is value in both items of the right and those on the left; however the Agile Alliance values the items on the right more (Beck et al., 2001).

2.3.2.4 Agile principles

The Agile Manifesto also comes with a set of 12 principles to follow (Beck & al, 2001, p.1):

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

2.3.2.5 Agile methods

With its core values and principles, Agile is considered a philosophy or a development approach (Palmquist, 2013; Lapham 2010). One definition of Agile by Lapham (2010, p. 5) is:

> An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework with “just enough” ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.

Agile encompasses several methods that act as frameworks in the management of software development. Agile methods are a modern project management family of development methods and processes known for their lightweight activities (ur Rehman & Hussain, 2007). Larman and Basili (2003) identify the Dynamic Systems Development Method (DSDM) to be the earliest developed Agile method, followed by Extreme Programming. Several other Agile methods have followed, such as Feature Driven Development, the Rational Unified Process, Crystal family of methodologies and Scrum.

Considered the most widely used Agile method in software development, Scrum is of particular interest to researchers and practitioners (West et al., 2010; Anwer et al., 2017; Versionone, 2019). The following section discusses this method in further details.

2.4 Scrum

2.4.1 Definition

Scrum is a term that initially referred to a strategy in rugby consisting of getting an out-of-play ball back in (Abrahamsson et al., 2017). Schwaber and Sutherland (2013, p.3) considered to be the creators of Scrum, define it as:

> A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Being a member of the Agile family, this iterative incremental process framework which is considered lightweight, simple to understand yet difficult to master, emerged in the early 1990s to help manage complex product development (Schwaber & Sutherland, 2013; Ashraf & Shabib, 2018). The framework is based on empirical process control theory, also known as empiricism, which claims that knowledge is derived from practical experience (Schwaber and Sutherland, 2013). Empiricism relies on three pillars: transparency, inspection, and adaptation (Schwaber and Sutherland, 2013). Transparency requires every aspect of the project to be visible to anyone considered responsible for the outcome. Observers should share a common understanding of what is being seen thanks to definitions by common standards. An illustration of the transparency pillar can be all participants sharing the same language when referring to the process (Schwaber and Sutherland, 2013). The inspection pillar promotes frequent inspection of Scrum artifacts in the progression toward a Sprint Goal. That being said, the inspection should not hamper the work process (Schwaber & Sutherland, 2013). The adaptation pillar supports the claim that the process or the material being processed may be changed or adjusted in case of deviations from the acceptable limits (Schwaber and Sutherland, 2013). Therefore, Scrum advocates for continuous, frequent inspection and adaptation through a disciplined management process (Ashraf & Shabib, 2018).

Scrum is not a technique or a process itself, but rather a framework (or method) within which techniques and processes can be used. The framework consists of components serving specific purposes and a set of rules binding them together and governing their relationships and interactions (Schwaber and Sutherland, 2013). The fact that Scrum lacks an explicit guide for product engineering is considered a challenge among practitioners (Ashraf & Shabib, 2018). The Scrum framework consists of a Scrum team and their related roles, events and artifacts. The table below summarizes the components of the Scrum framework:

Scrum Components	
Scrum Roles	The Product Owner
	The Development Team
	The Scrum Master
Scrum Events	The Sprint
	Sprint Planning
	Daily Scrum
	Sprint Review
	Sprint Retrospective
Scrum Artifacts	Product Backlog
	Sprint Backlog
	Increment

Table 4 Agile Scrum framework components (Schwaber & Sutherland., 2013)

2.4.2 Components

2.4.2.1 The Scrum Team

The Scrum team is made up of the Development Team, the Project Owner and the Scrum Master. Scrum teams are cross-functional and self-organizing. This means the team has all the competencies needed to work on the project, making them independent from others outside the team. Moreover, it is up to the team to choose how to accomplish their work. This team model is designed to maximize flexibility, creativity and productivity (Schwaber and Sutherland, 2013).

2.4.2.1.1 The product Owner

This role represents the interests of the stakeholders, mainly the customer's viewpoint. The Product Owner is responsible for pushing the team in the right direction to maximize ROI (Hayata & Han, 2011; Ashraf & Shabib, 2018). S/he therefore summarizes the user requirements and is the only one in charge of prioritizing in the Product Backlog (Schwaber and Sutherland, 2013). Product Owner tasks include: formulating Product Backlog items, ordering them in a way to best achieve goals and missions, making sure that the development team fully understands the items in the Product Backlog (Schwaber and Sutherland, 2013).

2.4.2.1.2 The Scrum Master

This role is responsible for making sure the Scrum is understood and applied according to the Scrum theory practices and rules. The Scrum Master is often considered to be a "servant leader" for the Scrum team (Schwaber and Sutherland, 2013). The Scrum Master maintains the process and makes sure to remove impediments in the way of delivering Sprint goals. On a practical level, the SM can be viewed as a facilitator working towards keeping the team focused on solving problems in the Sprint Backlog while enforcing the Scrum principles (Hayata & Han, 2011; Ashraf & Shabib, 2018). The Scrum Master responsibilities include: training and leading the organization in its Scrum adoption, coaching the Development Team in self-organization, and facilitating Scrum events (Schwaber and Sutherland, 2013).

2.4.2.1.3 The Development Team

Comprising between 3 and 9 highly skilled individuals for optimal interaction level and productivity, the Development Team is in charge of delivering “done” increments of the product at the end of each Sprint. The Development team is self-organized, meaning it is completely up to them to decide how to turn Product Backlog items into increments of releasable functionalities (Schwaber and Sutherland, 2013). As previously mentioned, the team is cross-functional, meaning it gathers people with all the skills needed to create a product increment. Even though the individuals of the team may have specialized skills of a specific area of study, accountability applies to the team as a whole. Moreover, no sub teams are allowed within the Development Team, with no exception to this rule. Furthermore, regardless of the nature of the work performed by the individual, Scrum recognizes no title other than Developer (Schwaber and Sutherland, 2013).

2.4.2.2 The Scrum events

Scrum prescribes a set of events to follow to ensure inspection and adaptation without wasting of time in the process. All the events are time-boxed, meaning they have a maximum duration. The Scrum guide claims that omission of any of these events would result in a reduction of transparency and opportunity to inspect and adapt.

2.4.2.2.1 The Sprint

At the heart of Scrum, Sprints are time-boxed iteration cycles of development during which a potentially releasable product increment is built. Accordingly, each Sprint has a definition of what should be built, its design, and a flexible plan to create it. A Sprint lasts no more than one month. This way, Scrum ensures adaptability and inspection, and limits risk to one calendar month of cost. Immediately after a Sprint is completed, another one starts. All the other events are comprised within the Sprint. A Sprint includes a Sprint Planning, Daily Scrums, Sprint Review, Sprint Retrospective, and the development work

(Schwaber & Sutherland, 2013). The following rules should be respected during the Sprint (Schwaber & Sutherland, 2013, p.9):

- No changes are made that would endanger the Sprint goal;
- Quality goals do not decrease; and,
- Scope may be clarified and re-negotiated between the Product Owner and Development as more is learned.

Although considered traumatic to the Scrum team and very uncommon, Sprint cancellations may occur. The Product Owner is the only one able to decide to cancel a Sprint. Once a Sprint is cancelled, all completed items of the Product Backlog are reviewed and can either be accepted by the Product Owner for being potentially releasable or re-estimated and put back in the Product Backlog (Schwaber and Sutherland, 2013).

2.4.2.2.2 Sprint Planning

During this event, the whole Scrum team collaborates to understand the Sprint. Accordingly, the meeting has for input the Product Backlog, the most recent product increment in addition to the projected and past performance of the Development Team. The Product Owner discusses the objectives that the Sprint should reach and the Product Backlog items that need to be worked on. The Development team decides how many items from the Product Backlog can go in the Sprint, as it the only one capable of assessing how much it can accomplish in the Sprint. The Scrum team then agrees on a Sprint goal, which is the objective to be met by the Sprint implementation. By the end of the meeting, the Development Team must be able to explain to the Product Owner and the Scrum Master how it intends to work in a self-organized manner towards the accomplishment of the Sprint goal. These are time-boxed to a maximum of eight hours, which the Scrum Master is responsible for enforcing (Schwaber and Sutherland, 2013).

2.4.2.2.3 Daily Scrum

Daily Scrums are daily meetings exclusive to the Development Team, time-boxed to a maximum of 15 minutes. This meeting serves the purpose of synchronizing activities and planning the next 24 hours of work, and is held at the same time and place each day.

During the meeting, the team members can help achieve the meeting's purpose by answering three questions: what they did the day before, what they are planning on doing this day, and if they faced any obstacles or challenges in their work (Schwaber & Sutherland, 2013).The objective behind Daily Scrums is to track progress towards the Sprint Goal and to help the team understand how it intends to work towards creating the anticipated increment as a self-organized team (Schwaber and Sutherland, 2013).

2.4.2.2.4 Sprint Review

At the end of each Sprint, a meeting is held to demonstrate the release to the stakeholders and management in order to capture their feedback (Hayata & Han, 2011; Ashraf & Shabib, 2018). These meetings are time-boxed to 4 hours and serve to adjust the Product Backlog if needed after inspecting the increment. During the meeting, the Product Owner is in charge of explaining which items of the Product Backlog were completed and which ones were not. The Development team discusses the things that went well and those with which they encountered challenges. The entire group is encouraged to collaborate on what needs to be done next, the changes occurring to the marketplace and/ or anticipated product use. The group may also discuss the next anticipated release of the product. The meeting ends with a revised Product Backlog.

2.4.2.2.5 Sprint Retrospective

While the Sprint Review's goal is to inspect and adapt with regard to the product, the Sprint Retrospective's goal is in regard to the process (Cervonne, 2011). Accordingly, the team assesses what worked and what did not, and gathers suggestions that are later reviewed and implemented to improve the process (Ashraf & Shabib, 2018) The meeting is time-boxed to three hours, and the team discusses how the last Sprint went with regards to people, process, tools and relationships. After prioritizing the main potential improvements, the team creates a plan to implement them in their next Sprint.

2.4.2.3 Artifacts

Scrum artifacts are used to ensure transparency of key information within the project team (Schwaber and Sutherland, 2013). The Scrum guide counts three artifacts, namely the Product Backlog, the Sprint Backlog, and the Increment. While some authors like Cervone (2011) might consider Burndown charts to also be an artifact, the Scrum guide considers it to be one way among others to track progress within the Product Backlog artifact.

2.4.2.3.1 The Product Backlog

Simply put, the Product Backlog represents a list of prioritized items (Ashraf & Shabib, 2018). It gathers everything that might be needed to produce the product and is the only source of requirements. The Product Backlog also gathers everything about the future changes to be implemented in future releases:

> The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate and value (Schwaber and Sutherland, 2013, p.13).

Concretely, several tools or software management tools can be used to create and manage a Product Backlog; Cervone (2011) mentions MS-Project and Spreadsheets as examples. The Product Owner is the only person responsible for managing the Product Backlog. S/he manages the content, order, and availability of the items in the Backlog. It is dynamic, which means that it is constantly changing to accommodate shifts in the environment and to make sure the product developed is appropriate, competitive, and useful. That also means that at the early stages of the project, the Product Backlog is not complete and includes only the best-known requirements. Throughout the releases and as the product is being developed and used, the Product Backlog is modified based on feedback from the marketplace, technological changes, and the evolution of business requirements. Because the Product Backlog keeps changing, expanding, and getting more exhaustive with time, it is called a living artifact. Product Backlog refinement is the action of adding details, estimates and order to the Product Backlog items. The Product Owner collaborates with the development team to decide on these details. Accordingly, the higher in the list, the more detailed the item.

2.4.2.3.2 The Sprint Backlog

The Sprint Backlog is a list of tasks to be completed in a particular Sprint (Ashraf & Shabib, 2018). It is created and managed solely by the development team and is a subset of Product Backlog items that are defined as part of the work for a particular Sprint. However, unlike the Project Backlog, the Sprint Backlog is created only by the Scrum team members. Ideally the Sprint Backlog is updated every day and contains no more than 300 tasks. The team may need to break down a task if it is determined that it will take more than 16 hours. Furthermore, the team may determine that items may need to be added or subtracted from the Sprint, but this is the team's decision and not something that is directed by the Product Owner (Cervone, 2011). All in all, the Backlog makes for a real-time picture of the work to be accomplished by the development team in the Sprint (Schwaber and Sutherland, 2013).

2.4.2.3.3 The increment

The Scrum guide defines an increment as the sum of the Product Backlog items accomplished in the Sprint above what was already produced in previous Sprints (Schwaber and Sutherland, 2013). An increment must be in useable condition, as it is a potentially shippable software or one that is of use to the stakeholders (Ashraf & Shabib, 2018).

It is worth mentioning that the Scrum guide warns that its artifacts, roles, events and rules are "immutables" and that changing or implementing only a few of these does not result in Scrum, as "Scrum exists only in its entirety" (Schwaber and Sutherland, 2013, p.19).

2.4.3 Scrum flow

The Scrum process consists of three phases: the pre-game phase, development phase and post-game phase (Abrahamsson et al., 2017):

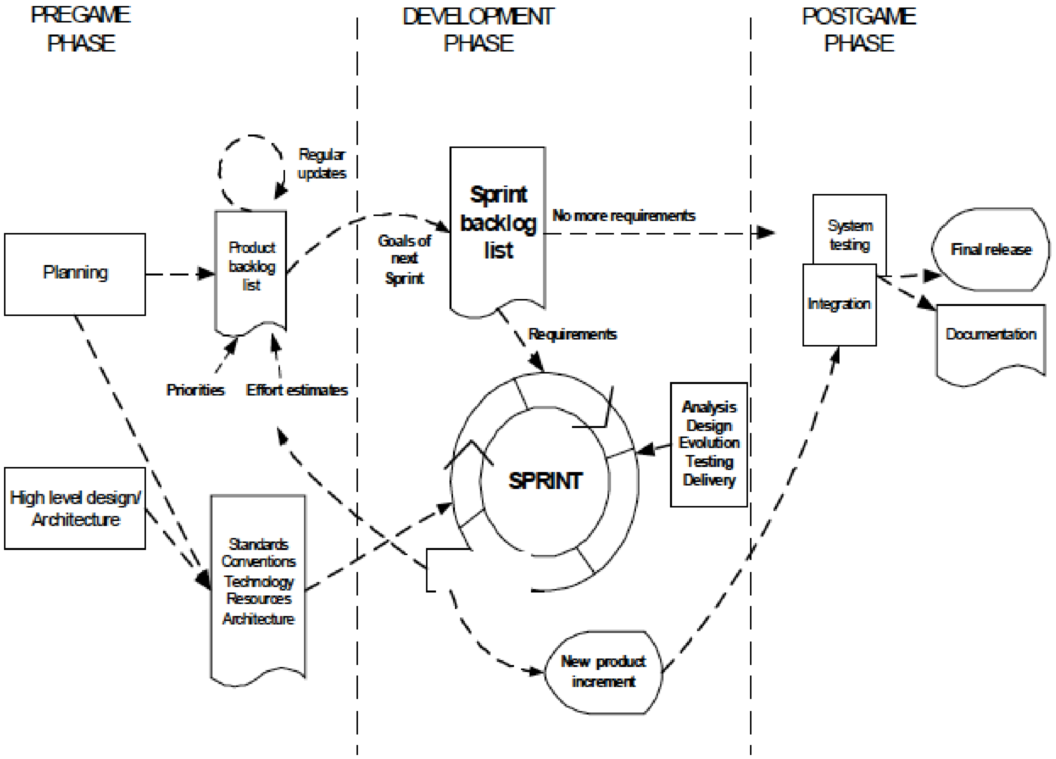


Figure 5 Scrum process representation (Abrahamsson et al., 2017)

2.4.3.1 The pregame phase

This phase includes two parts: The planning and the architecture of the system, also referred to as high-level design. In the planning phase, the team defines the system to be developed and sets the ground for the work to come. For that, a first version of the Product Backlog is created, combining all the requirements known at the time of creation. These requirements can be supplied from the customer and the developers but can also originate from other departments such as sales or marketing (Abrahamsson et al., 2017). Accordingly, the Product Backlog items can be customer features such as enabling premium users to benefit from free shipping, or functional/ engineering goals such as scalability or security matters. Some may also be about exploratory work, or bugs encountered later in the project (Deemer et al., 2010). This Product Backlog will constantly change through the iterations of the project as new items get added or initial elements become more detailed. It will also be updated according to what has been done, how ordering of priorities evolve, and will accommodate any changes in the environment of the project. Moreover, the planning phase also involves defining the team, tools and resources to be used in the project. Based on the elements identified in the Product Backlog, the architecture is designed for the system. A design review meeting is later held to discuss implementation decisions (Abrahamsson et al., 2017).

2.4.3.2 The development phase

Sometimes called the game phase, this phase is where functionalities are developed in Sprints. As described above, these are iterative cycles of less than 4 weeks in which functionalities are developed and the output is an increment of a potentially shippable product or product part. Abrahamsson and colleagues (2017) describe this phase as being the Agile part in the Scrum approach because the project is treated as a "blackbox," since many elements are unpredictable. That means that environmental and technical factors are taken into consideration and observed throughout all the sprints, not only at the beginning of the project. Indeed, quality, changing customer requirements, project resources, implementation tools and technologies, and development methods are constantly controlled through the various Scrum practices discussed earlier. Every sprint includes the usual software development phases of requirements analysis, design, evolution testing and delivery. As Sprints are carried out, the overall architecture and design of the system are reviewed and the Product Backlog is updated (Abrahamsson et al., 2017). Team members also update the estimates of remaining work in their Scrum Backlog on a daily basis. Following this update, the hours of work remaining for the whole team are computed and plotted on a Sprint Burndown Chart (Deemer et al., 2010). This chart is a graphical representation of how much work remains in terms of person hours until all the tasks are completed and should be downward sloping to reach zero by the last day of the Sprint if all goes as planned. Analogous to the Sprint Burndown Chart, the Release Burndown chart used is based on the same concept and shows progress towards the release date of the product. It is updated by the Product Owner, who is assisted by the Scrum Master (Deemer et al., 2010).

2.4.3.3 The post-game phase

This phase is reached when the items of the Product Backlog are finished, and no other items or issues can be found. This phase prepares the product release and includes tasks such as documentation, integration and system testing.

2.5 Comparison of Agile and Traditional software development

Drawing on the work of Boehm (2002), Awad (2005), Leau and colleagues (2012), and Stoica and colleagues (2013), the following table presents a comprehensive comparison of the main elements of the two software development approaches.

Element	Traditional	Agile
Fundamental hypothesis	Systems are fully predictable and specifiable. They can be developed through extensive and detailed planning (Stoica et al., 2013).	High-quality adaptive systems are developed by small teams, thanks to the principle of continuous improvement of design and testing based on fast feedback and change (Stoica et al., 2013).
Developers	Developers are plan-oriented, they have adequate skills and have access to external knowledge (Boehm, 2002; Stoica et al., 2013; Awad 2005).	Developers are agile. They are collaborative, knowledgeable, collocated, and collaborative (Boehm 2002; Stoica et al., 2013; Awad, 2005).
Additional Abilities Required from Developers	Nothing in particular (Stoica et al., 2013; Leau et al., 2012).	Interpersonal abilities and basic business knowledge (Stoica et al., 2013; Leau et al., 2012).
Customers	Customers are knowledgeable, cooperative, representative, and empowered (Boehm, 2002). Their level of involvement in the process is low (Leau et al., 2012). The interactions with customers are on a need basis, and are contract-focused (Awad, 2005).	Customers are dedicated, present onsite, knowledgeable and cooperative. They are representative, and empowered (Boehm, 2002). They are highly involved in the process (Leau et al., 2012).

Documentation	Heavy (Awad, 2005)	Low (Awad, 2005)
Requirements	Requirements are known early, and remain stable in general (Boehm, 2002; Awad, 2005). They are detailed and defined before coding and implementation (Stoica et al., 2013; Leau et al., 2012).	Requirements are mostly unknown, emergent, and change rapidly (Leau et al., 2012, Award, 2005). They are acquired iteratively (Stoica et al., 2013).
Architecture	Designed for current and foreseeable requirements (Boehm, 2002; Stoica et al., 2013; Awad, 2005).	Designed for current requirements (Boehm, 2002; Stoica et al., 2013; Awad, 2005).
Development Direction	Fixed (Leau et al., 2012 ; Stoica et al., 2013).	Easily changeable (Leau et al., 2012 ; Stoica et al., 2013).
Planning and Control	Documented plans, quantitative control (Awad, 2005).	Internalized plans, qualitative control (Awad, 2005).
Success Measurement	Conformation to plan (Awad, 2005).	Business Value (Awad, 2005).
Objective	High Assurance and safety (Boehm, 2002; Stoica et al., 2013; Awad, 2005).	Rapid value (Boehm, 2002; Stoica et al., 2013; Awad, 2005).
Size	Large teams and projects (Boehm, 2002; Stoica et al., 2013; Awad, 2005; Leau et al., 2012).	Small teams and projects (Boehm, 2002; Stoica et al., 2013; Awad, 2005; Leau et al., 2012).
Approach	Predictive (Awad, 2005).	Adaptive (Awad, 2005).
Refactoring	Expensive (Boehm, 2002 ; Stoica et al., 2013 ; Awad, 2005 ; Leau et al., 2012).	Inexpensive (Boehm, 2002 ; Stoica et al., 2013 ; Awad, 2005, Leau et al., 2012).
Testing	After coding phase is completed (Leau et al., 2012; Stoica et al., 2013).	On every iteration (Leau et al., 2012; Stoica et al., 2013).
Rework Cost	High (Stoica et al., 2013 ; Leau et al., 2012).	Low (Stoica et al., 2013 ; Leau et al., 2012).
Communication	Formal (Stoica et al., 2013).	Informal (Stoica et al., 2013).
Knowledge Management	Explicit (Stoica et al., 2013).	Tacit (Stoica et al., 2013).
Development Model	Life cycle model (waterfall, spiral or modified models) (Stoica et al., 2013).	Evolutionary-delivery model (Stoica et al., 2013).

Quality Control	Difficult planning and strict control. Difficult and late testing (Stoica et al., 2013).	Continuous control of requirements, design and solutions. Permanent testing (Stoica et al., 2013).
Management Style / Organizational Structure	Command and control Mechanic (bureaucratic, high formalization), targeting large organization (Stoica et al., 2013). Autocratic management style (Awad, 2005).	Leadership and collaboration Organic (flexible and participative, encourages social cooperation), targeting small and medium organizations (Stoica et al., 2013) Decentralized management style (Awad, 2005).
Cycles	Limited (Awad, 2005).	Numerous (Awad, 2005).
Domain	Predictable (Awad, 2005).	Unpredictable/Exploratory (Awad, 2005).
Emphasis	Process oriented (Awad, 2005).	People oriented (Awad, 2005).
Return on Investment	At the end of the project (Awad, 2005).	Early in the project (Awad, 2005).

Table 5 Comparing Agile and traditional software development

The Culture dimension presented by Awad (2005) was omitted from the list of elements presented in the table above. The Culture dimension of Awad (2005) described culture as “leadership-collaborative” in the Agile environment and “command-control” for the plan-driven approach. The reason behind this omission is the assumption that culture is included in the management style of the organization and that organizational structure is linked to the culture of the organization. Because these elements are aligned and overlap, the Culture component was omitted in an effort to reduce redundancy. Likewise, the elements Management Style and Organizational Structure were put together as they comprised overlapping aspects that were therefore redundant. Similarly, the Upfront Planning element by Awad (2005) is encompassed in planning and control. Furthermore, while Stoica and colleagues (2013) separated Client from Client involvement, these were merged in the above table. Finally, the Remodeling element by Stoica and colleagues (2013) was considered equivalent to the Refactoring element by the other authors.

Despite the fact that the two methodologies aim for the same end goal, that is, delivering good quality software in an efficient matter, their fundamental hypotheses are very distinct (Palmquist et al., 2013; Stoica et al., 2013). Indeed, while the traditional approach assumes a predictable system, Agile assumes a fast-paced and continuously changing environment in which systems ought to be adaptive (Stoica et al., 2013). From these two distinct hypotheses stem distinct assumptions and practices governing the two approaches. For instance, in traditional development methods, unlike Agile ones, extensive documentation is followed, and all requirements should be known early. This makes traditional approaches more suitable for larger, riskier projects. On the other hand, the minimum-documentation policy and emergent requirements of Agile methods are more suitable for exploratory and unpredictable development. Moreover, the collaborative nature of Agile methods requires developers with interpersonal skills and a collaborative culture.

2.6 Agile and project success

Given the limited number of studies focusing on the Scrum framework and project success and the fact that Agile methods derive from the same values and principles, this section will explore Agile methods in general (including Scrum) in relation to project success. Agile project management methodologies have been introduced to address the weaknesses of traditional methods (Serrador & Pinto, 2015). Indeed, traditional models for planning and execution are often no longer optimal for the dynamic environments in which software projects are being developed. The traditional methodology makes it hard to accommodate changes during the project because of their excessive planning at early stages. Serrador and Pinto (2015, p.1041) summarize the issues and criticism that made researchers and practitioners seek an alternative to these methods:

> Evidence continues to accumulate suggesting that a rigid development process can result in significant downstream pathologies, including excessive rework, lack of flexibility, customer dissatisfaction, and the potential for a project to be fully developed, only to discover that technological advances have eclipsed the need for it.

In their research, Serrador and Pinto (2015) demonstrate the efficacy of Agile methodologies by establishing a correlation between using Agile methodologies and higher reported success in three different categories namely: overall project success, efficiency, and stakeholder success. Accordingly, they show that the greater the reported use of agile and iterative methodologies the greater the project success reported. Their large-scale empirical analysis show that Agile project management is more likely to deliver successful projects regardless of the team experience or the perceived project's complexity. It is worth mentioning that the quality of the vision and goals for the project are significant moderators of the relationship between the use of Agile methodologies and project success (Serrador & Pinto, 2015). Furthermore, Lee and Xia (2010) also provide empirical evidence that software development agility contributes to better software development performance in terms of time, budget and functionality. In their model, Lee and Xia (2010) conceptualize software development agility as response extensiveness (scope and variety of changes) and response efficiency (time and cost associated with changes). Their study suggests that response efficiency, which is supported by team autonomy, positively affects the three aspects of software development performance, namely time, cost, and functionality. Response extensiveness, on the other hand, positively affects only the functionality aspect of software development performance and is positively affected by team diversity but negatively affected by team autonomy. Their study therefore suggests a trade-off relationship between the two fundamental elements of software development agility: response extensiveness and response efficiency. In their 2015 CHAOS report, the Standish Group illustrate the superiority of Agile methods in delivering successful projects. The report studies 50,000 projects from around the world, and results show that Agile approaches lead to more successful projects and fewer failures regardless of the project size. Subsequently, the group includes “Agile Process” in their list of CHAOS factors of success, a list of factors they have been analyzing and ranking for over 20 years.

Agile methodologies have therefore been introduced to improve project implementation as they contrast with traditional project management approaches for being incremental and iterative, and emphasize principles such as customer flexible scope, continuous design and embracing uncertainty (Serrador & Pinto, 2015). Ever since, Agile methodologies have been continuously spreading and gaining in popularity (Dybå and Dingsøyr, 2008). Following is a summary of the main contributions of the scientific literature to assessing how agile methods impacted different aspects of project success.

2.6.1 Agile and project management success

2.6.1.1 Scope

Agile is designed to allow cheap and easy changes to the scope, which is considered an opportunity in software projects. Accordingly, the scope is variable in order to handle any unpredictable requirements or continuously evolving ones. Changes to the scope are supported by mechanisms such as a flexible, continuously reviewed Product Backlog that is changed throughout the project (Walczak and Kuchta, 2013). Santos and colleagues (2013) have established that, thanks to their multifunctional teams led by experienced facilitators, Agile practices improve scope management. Lee and Xia (2010) support the same claim by establishing that diverse Agile teams positively affect response extensiveness in software development which in turn positively impacts software performance in terms of functionality. This setting, combined with continuous communication through practices such as stand-up meetings, helps determine the requirements along with their priority and make sense of deadlines. Other researchers warn that Agile methods present a risk of scope creep, meaning uncontrollably growing project scope (Aitken & Ilango, 2013).

2.6.1.2 Quality

Several studies recognize the effectiveness of Agile methodologies in achieving quality software. Masood and Farooqi (2017, p.26) attest that “Agile methods have stressed the need to define quality from the customers’ perspective”. Assuming that the customer is also the end user, agile methods focus on satisfying customers’ needs and preferences throughout the entire product development process (Masood & Farooqi, 2017). Bhasin (2012) also claims that Agile contributes to better quality software, but credits that to the quality assurance mechanisms that are built into Agile activities. Such mechanisms include daily meetings and test-driven development (Bhasin, 2012). More specifically in regard to the Scrum framework, Permana (2015) asserts that using Sprints helps the development team align the quality of the software developed to the determined business needs. Nonetheless, Masood and Farooqi (2017) warn that Agile methods may lack guidance in evaluating the team’s performance due to the lack of a complete project scope to be used as a reference. Hence, the lack of plans and predetermined specifications makes it harder to control quality and judge whether the project is going in the right direction.

2.6.1.3 Budget

Several researchers attest to the positive influence of Agile methods on budget. Santos and Colleagues (2013) consider Agile practices to have established their reputation as a better alternative for cost control in software development projects. Because of their ability to accommodate changes through iterations, Agile approaches avoid costly rework that results from changes in the project specifications. Moreover, Masoosd and Farooqi (2017) make the same claim, asserting that this approach provides time savings through the use of short-term scope, plans, and design. This in turn allows for better budget performance. Similarly, Kurup and Sidhardhan (2015) affirm that a significant benefit of Agile is that it reduces project cost by eliminating costly changes and rework. Futhermore, the research model by Lee and Xia (2010) establishes that response efficiency of Agile methods has a positive effect on the budget dimension of project success.

However, many researchers offer an opposing perspective. Cao and colleagues (2013) assert that determining the upfront cost of an Agile development project is a challenge, making Agile budgeting a major issue. Accordingly, there is a major need to develop a funding process for Agile projects (Cao et al., 2013). In this case, Cao and colleagues (2013) give recommendations such as making the Agile process more visible to customers, establishing the risk-management process and layering project scheduling. Furthermore, Cao and colleagues (2013) advise that funding decisions should be based on feedback within the team and negotiations based on changing customer values.

2.6.1.4 Schedule

Researchers' opinions on the schedule performance of Agile methods are divided as some believe in the positive contribution of Agile methods while other claim the opposite. Raith and colleagues (2013) believe that group estimations yield better estimates and that Agile processes such as Planning Poker result in more accurate estimates if the teams are experienced. Moreover, Lee and Xia (2010) demonstrate that response efficiency, an important element of software development Agility, has a positive effect on delivering according to schedule. In contrast, other researchers believe that scheduling is a major challenge in the Agile approach to software development. Because plans and estimates are specific to each cycle rather than to the project as a whole, Masood and Farooqi (2017) suggest Agile methods make it difficult for the team to make estimates for the entire project, leading to scheduling issues. Moreover, Masood and Farooqi (2017) argue that this lack of exactitude in scheduling hinders the team's visibility in the project, exposing the project to risk. Furthermore, Masood and Farooqi (2017) claim that scheduling in Agile methods makes it difficult for organizations to manage other projects as a result of this lack of schedule clarity. Consequently, Masood and Farooqi (2017) assert that agile methods are structured as though there were several smaller projects in one project, which can lead to increases in the overall project cost. Leybourne (2009) also criticizes the scheduling issues in Agile methods, declaring that such methods encourage excessive changes to the project requirements, which can cause significant delays.

2.6.2 Agile and stakeholders' success

Agile is known to be people-focused, and for that reason, many researchers associate it with stakeholders' success. According to Nerur and collegues (2013), the success of this project management approach applied to software development is heavily dependent on collaboration, communication, and cooperation between project teams and other stakeholders. Dybå and Dingsøyr (2008) attest to the greater focus on customers' perspective provided by Agile methods, allowing a greater sense of control, involvement and communication. In the context of the Scrum framework, practices such as daily Scrum meetings were demonstrated to make customers feel more up-dated on the project and gain a better understanding what is developed (Mann & Maurer, 2005). Additionally, developers experience greater job satisfaction when using Agile methods. Developers using Scrum report experiencing less overtime and recommend future use of the framework (Mann & Maurer, 2005). Furthermore, Dybå and Dingsøyr (2008) report that the planning game activity, one of the techniques used in Agile methods to generate estimates in group, was found to have a positive impact both on the company and with customers because it gave both groups insight into the development process.

2.6.3 Agile and organizational success

Agile methods are associated with positive organizational impact. For instance, Solinski and Petersen (2016) believe that such methods provide several internal benefits that can be categorized into knowledge, learning, and employee satisfaction benefits. Moreover, a study by Tripp and colleagues (2016) shows a link between Agile practices and positive impact on the organization, especially in terms of job satisfaction. Nonetheless, Tripp and colleagues (2016) suggest that the organization's management should be supportive and encouraging of Agile practices.

2.6.4 Agile and strategic success

Little research has examined the impact of Agile project management on strategic success. Denning (2017) urges for future research on Agile and strategic management, considering it a new and exciting approach to innovation. In the view of Denning (2017), Agile methods can impact organizations' innovation and regenerate mature companies by uncovering new customer experiences and creating new products and services to fulfill unmet needs. Furthermore, Denning (2017) highlights the importance of strategic agility in organizations as a response to current market circumstances. In the words of Denning (2017, p.17):

> Given that industry borders are dissolving and competition is more dynamic than ever, the need for Strategic Agility - speedy, customer focused innovation that aims to make markets- is becoming increasingly obvious.

Building on the summary table of project success aspects presented earlier in this chapter, the following table synthesizes Agile methods' impact on multiple aspects of project success:

Success Aspects	Literature
Project Management Success	
Quality	Focus on customer's needs and preferences (Masood & Farooqi, 2017) Built-in mechanisms to ensure quality (Bhasin, 2012) Quality control (Permana, 2015)
Scope	Flexible scope (Walczak and Kuchta, 2013) Multifunctional teams and experienced facilitators, continuous communication and transparency (Santos et al., 2013; Lee & Xia., 2010)
Budget	Accommodate changes through iterations and avoid costly rework from project specifications changes (Santo et al., 2013; Kurup & Sidhardhan, 2015; Lee & Xia., 2010)
Schedule	Contributes to on-time delivery (Lee & Xia., 2010)

	Lacks clarity and definition (Masood & Farooqi, 2017) Excessive changes and delays (Leybourne, 2009)
Stakeholders’ Success	
Customer Satisfaction **Team Satisfaction**	People-focused (Nerur et al., 2013) Collaboration, communication, and cooperation between the team and the stakeholders (Nerur et al., 2013; Dybå and Dingsøyr, 2008) Greater sense of control, involvement (Dybå and Dingsøyr, 2008; Mann & Maurer, 2005) Job satisfaction (Mann & Maurer, 2005; Tripp et al., 2016)
Management/Sponsor satisfaction	NA
Direct Organizational Success	
Benefits to the organization	Promote knowledge and learning (Solinski & Petersen, 2016) Employee satisfaction (Tripp & al, 2016; Solinski & Petersen, 2016; Mann & Maurer, 2005)
Strategic Success/Value	
Long-term advantages	The confluence of Agile and strategic management promotes innovation and creates strategic opportunities (Denning, 2017)

Table 6 Agile project management and success aspects

2.7 Agile in practice

Thanks to the flexibility they provide, Agile methods have been increasingly adopted in the software development industry over traditional methods (Campanelli & Parreiras, 2015; Kiv et al., 2018). Nonetheless, Agile adoption remains a complex process requiring significant efforts from the organizations and teams involved (Campanelli & Parreiras; 2015, Kiv et al., 2018). Indeed, after several failed projects, the software development community concluded that method tailoring is important to match the software development context both at the project and organizational level (Kalus & Kuhrmann, 2013). In fact, many researchers believe that software methods ought to be tailored for them to be optimal (Joslin & Müller, 2005; Fitzgerald et al., 2006; Conboy & Fitzgerald, 2010). According to Joslin and Müller (2005), it is not the use of a project management methodology that brings about project success, but rather the ability to tailor the methodology to the organizational context. Several researchers examine the tailoring of Agile methodologies. While some look at ways to mix traditional and Agile methods to find the right balance, others focus on ways to tailor Agile principles to their context.

A small stream of research looks at hybrid software development, or how Agile methods can coexist with traditional ones. In their study, Kuhrmann and colleagues (2017) conclude that hybrid approaches are widely used in practice for organizations of various sizes and from multiple industries. Their findings suggest that most hybrid approaches follow the pattern of having a traditional framework in which several selected Agile practices were plugged (Kuhrmann et al., 2017). These results corroborate those of Theocharis and colleagues (2015) who also claim that companies combine traditional and Agile approaches to create context-specific hybrid ones referred to as "Water-Scrum-Fall" (West, 2011). In line with these findings, a case study by Batra and colleagues (2010) concludes that hybrid development methods are not only a reality but also a necessity in managing large distributed software projects. According to Batra and colleagues (2010), both agility and control are needed to overcome the challenges of large distributed software projects, and agility alone can lead to chaos.

Even when not formally mixed with a traditional method, it is rare that an Agile method is adopted as prescribed in the literature. Ken Schwaber himself, co-founder of the Scrum framework, believes that 75% of companies that claim to be using Scrum are not actually using it adequately. He also claims that while Scrum is a framework designed to expose organizational dysfunctions, several organizations change the framework to accommodate these dysfunctions instead of removing them (Mattcallanan, 2010). Method tailoring, therefore, seems to be the reality of Agile adoption in practice, yet is also a challenge to the benefits of these same methods. Researchers have therefore been studying software method tailoring and will continue to do so (Conboy & Fitzgerald, 2010). Campanelli and Parreiras (2015) define tailoring in the software development context as “the adaptation of the method to the aspects, culture, objectives, environment and reality of the organization adopting it” (p. 87). Software method tailoring can be classified into two categories: Contingency factors and method engineering (Conboy and Fitzgerald, 2010; Fitzgerald et al., 2000):

Contingency factors

This approach to tailoring entails having a portfolio of methods predefined within an organization and selecting which method to use, depending on the context of the development. This requires the teams to learn about different methods and be able to select the most suitable one based on contextual features such as structure and uncertainty level (Fitzgerald et al., 2000; 2003).

Method engineering

This approach uses existing method fragments to create a new method that would be suitable for the context of development. Instead of acquiring already existing methods, this approach proposes creating organization-specific methods that are tailored to the reality of the projects (Henderson-Sellers & Ralyté, 2010).

The method engineering approach is the primary choice when it comes to software method tailoring in practice (Campanelli & Parreiras, 2015). Several researchers have been looking at this process and how to ensure it is done successfully, but no consensus has been reached on the matter (Campanelli & Parreiras, 2015). Accordingly, there is no clear path for how the Agile practice selection should proceed (Campanelli & Parreiras, 2015).

Nonetheless, research has been conducted to determine which practices are the most widely used and how to best adapt Agile methodologies. In order to investigate which Agile practices are being used in the software development industry on both an organizational and project level, Kurapati and Colleagues (2012) conducted a survey mapping use of some Agile practices specific to the Scrum and XP frameworks. Tripp and Armstrong (2018), on the other hand, focuses on the twelve most popular Agile practices and investigate how organizational motives for adopting Agile impact the practices adopted. They also provide four guidelines for tailoring Agile methodologies regarding the organization and the teams involved. Cao and colleagues (2009) also provide prescriptions on how to adapt Agile methodologies based on a multisite case study. Also based on a case study, Rasnacisa and Berzisaa (2016) propose a method for the adoption and implementation of Agile project management methodologies based on best practices in change management, methodology adaptation, and implementation.

A few researchers address Agile tailoring in the specific context of global software development (Jalali & Wohlin, 2010; Hossain et al., 2011). Their results verify that, even in a global software development context, Agile practices are subject to change in response to situational factors (Jalali & Wohlin, 2010; Hossain et al., 2011).

Finally, three important literature reviews examine Agile in practice. First, the mapping study by Diebold and Dahlem (2014) gathers empirical studies on Agile application in practice. Based on universal Agile practices, the study presents a systematic map of these Agile practices in relation to industry domains and different Agile processes, among which was the Scrum process. The study concludes that most Agile practices are not used completely. Moreover, Campanelli and Parreiras (2015) present a systematic literature review on Agile methods tailoring. The study concludes that method engineering is the base for tailoring and that this latter is mostly dependent on internal organizational environment and project objectives. An interesting conclusion from this same study is that most papers are not specific to any Agile methodology (Campanelli and Parreiras, 2015). Furthermore, Kiv and colleagues (2018) provide a systematic literature review on the Agile Manifesto and practice selection for tailoring software development. The study investigates the relation between the Agile Manifesto and Agile practices selection, mainly how the Manifesto is discussed in the tailoring process and whether the practices selected can be related to the Manifesto’s principles. Their results suggest that the Manifesto is often neglected by practitioners. Moreover, they conclude that Agile value, principles and practices are linked and that practitioners ought to be empowered to select practices while also knowing their impact. The study therefore stresses the importance of understanding and following the Agile Manifesto. Kiv and colleagues (2018) also note that Agile practitioners follow the rules of specific Agile methodologies like Scrum and XP rather than the Agile Manifesto itself.

In summary, while organizations in practice focus more on specific method adoption, research focuses mainly on studying universal Agile practices, especially when it comes to tailoring. Nonetheless, a few contributions to the literature on Scrum method tailoring are identified and presented in the next section.

2.8 Scrum in practice

Despite the fact that Scrum is the most popular Agile framework, only a small number of studies address how it is actually tailored in practice. Diebold and colleagues (2015) look at what practices German practitioners vary in Scrum in terms of Team, Sprint, Meetings, Requirements, and Quality assurance. They find that none of the companies conformed entirely to the framework and that there is at least one company deviating in each aspect. The reasons behind such deviations vary; while some changes are made to improve efficiency, others are found to be legacies of previous non-Agile processes (Diebold et al., 2015). Other relevant research is work by Eloranta and Colleagues (2016) which investigates how deviating from Scrum and adopting ScrumButs ("we do use Scrum, but …") can lead to practices that could be risky to the project. Accordingly, the research identifies 14 anti-patterns, a commonly-occurring deviation from Scrum that has potentially unfavorable consequences. However, the fourteen anti-patterns and their relationship to project success are not verified empirically.

Two literature review studies are pertinent to Scrum tailoring. The first work by Ashraf and Aftab (2017) consists of a systematic review to uncover the latest transformations to the Scrum process model. The review concludes that practitioners often alter Scrum in its adoption in an effort to optimize it. However, while Scrum has been subject to many changes in order to improve efficiency and effectiveness, more empirical research is needed to evaluate the consequences of these changes (Ashraf & Aftab, 2017). Similarly, Hron and Obwegeser (2018) also investigate Scrum adaptations through a systematic literature review. Their objective, however, is to understand the motivations behind the deviations. The seven motivations extracted are: distributed settings, combination with other methods, increased requirements for UX and usability, vertical scaling, size scaling, tools, and adaption to different contexts. Hron and Obwegeser (2018) also acknowledge the fact that Agile methodologies ought to be tailored according to the context and needs of organizations. However, they also warn that if not orchestrated carefully, changes to the method might dilute the benefits of the Agile adoption.

While the literature acknowledges that Agile methods are tailored in practices, most of the research concerned Agile practices in general. Accordingly, little is known about specific changes made to Scrum guidelines. This research aims to further investigate the deviations from the Scrum guidelines, uncovering what practitioners choose to adopt, change, and disregard on a day-to-day operational level. This research also aims to contribute to the gap discussed by Hron and Obwegeser (2018) and Ashraf and Aftab (2017) by informing how deviations in Scrum guidelines in practice link to project success, which has not been done so far to the knowledge of the research.

CHAPTER 3 : METHODOLOGY

The definition of research varies according to experts and fields, but ultimately, it is a rigorous and systematic process aiming to answer questions (Kumar, 2011). The aim of this paper is to uncover whether there is a gap between theory and practice in how organizations adopt and work with the Scrum framework in a software development context. Moreover, this study also aims to investigate the links between Scrum guideline deviations and project success. This chapter presents the methodology followed to ensure the research question is answered following a scientific and rigorous approach. Accordingly, this section sets forth the methodological choices adopted in terms of data collection and analysis, as well as the justifications behind these choices.

3.1 Qualitative research

Because of the exploratory nature of this research, a qualitative approach seemed to be the most appropriate choice of methodology. Similarly, a positivistic approach was undertaken because this research considers social facts that already exist and seeks to observe, describe and provide explanations in a neutral and objective way (Silverman, 2015). The choice of a qualitative method was hence motivated by two main reasons.

The first reason is the research problem addressed. This paper attempts to investigate the "gap" between theory and practice. Consequently, there is a need to understand the way software projects are run in different companies while leaving room for insight regarding roles, practices, events, and success perception that may stray away from the literature on the topic. Unlike quantitative research, qualitative research allows one to "seek understanding" and "interpret meanings" (Silverman, 2015). The main differences between qualitative and quantitative approaches, as presented by Silverman (2015), are highlighted in the table below. Moreover, qualitative research allows for a closer look into the reality of things, as Creswell and colleagues (2017, p.7) put it, "qualitative researchers studies things in their natural settings, attempting to make sense of, or interpret, phenomena in terms of the meaning people bring to them." Adopting a qualitative approach is thus more suitable to addressing the above-mentioned research questions.

Quantitative	Qualitative
Generates data that allow numerical analysis	Describes a phenomena in a context
Uses statistical calculations	Interprets processes or meanings
Uses statistical software and pre-tested scales	Uses theoretically based concepts
Seeks explanations and correlations	Seeks "understanding"

Table 7 Comparison between qualitative and quantitative research (Silverman 2005)

Another reason for adopting a qualitative approach is the state of research around the application of Scrum in industry and its link to project success. As previously discussed, only a small stream of research has examined the topic of Scrum tailoring in industry, and no previous work regarding Scrum guidelines tailoring and project success was found. Indeed, while research has established that Agile methodologies are often modified, it is still unclear what specific changes are made to all the elements of the Scrum framework, and how these modified guidelines link to the project's outcome. Having said that, qualitative methods are more appropriate when confronted with a high level of uncertainty regarding the subject researched (Trauth, 2001). A qualitative method is therefore more adapted to the explorative nature of this research. More precisely, the method used for data collection is semi-structured interviews.

3.2 Data collection

3.2.1 Semi-structured interviews

Various definitions of interviews exist. Ultimately, interviews are person-to-person interactions between two or more subjects (Kumar, 2011). Often conducted in a face-to-face manner, interviews allow one to collect facts and gain insights into attitudes, beliefs, opinions, experiences, and behaviors (Rowley, 2012). Based on how much control the interviewer has over the process, Harell and Bradley (2009) place interviews on a continuum of structure, going from unstructured interviews to highly structured ones. Semi-structured interviews allow the researcher to have a guide with themes and questions while leaving him/her with the flexibility to order the questions and carry out the interview in a conversational style (Harell and Bradley, 2009). This method of data collection, considered the most common one in qualitative IS research, is also the one undertaken for this research for the following reasons:

First, while providing enough structure to make sure all the necessary themes are discussed, semi-structured interviews allow for flexibility and spontaneity in the interviewee’s answers, which allows one to gain more depth and insight on the topics discussed (Harell and Bradley, 2009). In the case of this research, it is essential to go over all the themes of the Scrum framework, such as the roles, events, and artifacts mentioned in the Scrum Guide, as well as all the aspects of success in IT projects. However, interviewees should be left with enough room to discuss other matters around their use of the Scrum framework and the specificities of their projects and experiences.

Moreover, semi-structured interviews are designed in a way to elicit the interviewee’s opinion and ideas on the subject, rather than the preconceived ones the researcher has (Zorn, 2008). This allows the for the collection of more reliable and comparable data. Furthermore, because such semi-structured interviews rely on the interviewer’s probes regarding the interviewee’s input, they allow for more depth and understanding (Zorn, 2008). Consequently, this puts the researcher in a favorable position to be able to take into account the interviewee’s ideas and to understand his/ her perspective with regard to the phenomenon under study during the interview (Anadon & Savoie-Zajc, 2009). In the case of this research, little is known about the way the Scrum framework is applied for different projects and/or the perception of IT project success for each respondent, which is why semi-structured interviews come naturally as the most appropriate option to collect data.

3.2.1.1 Interview guide

An interview guide (Appendix A) was developed based on the literature review findings, comprising a grid of forty-four specific Scrum guidelines (Appendix B). These guidelines were derived by the researcher by summarizing the explicitly stated instructions in the Scrum Guide regarding the roles, artifacts, and events of the method as well as the rules that govern their relationships. The unit of analysis in this research is IT projects; interviewees are therefore asked to talk about software projects they are involved in, and answer questions regarding specific aspects of these projects. The interview process is based on Galetta's (2013) recommendations that advise starting with open-end questions about the participant before moving gradually to discussing the topic at hand in more depth. Galetta (2013) also recommends revisiting the interviewee's narratives for theoretical connections at the last stage of the interview. Galetta's (2013) recommendations can be summarized as follows:

Segment of the interview process	Recommendations
Preliminary steps	• Stating the purpose of the research • Expressing gratitude for the participant's involvement • Signing the consent form for the participant to understand his/her rights including the consent for audiotape
First segment	Start with open-ended questions about the participant's experience before moving to more theoretically shaped questions, leaving the participant with time and space to narrate his/her experience
Middle segment	Pursue the topic in more depth with questions drawn from the participant's earlier input
Closing segment	Revisiting the participant's narratives for theoretical connections and closure

Table 8 The interview process (Galetta, 2013)

The interview guide consists of four portions. The first part of the interview guide includes general open-ended questions about the interviewee, their background and experience, their organization, as well as the project to be discussed in the following sections of the interview. The second section involves a grid gathering 44 statements about the Scrum methodology as explained in the Scrum guides (Appendix B). In this section, participants are asked to rate the 44 statements on a numerical scale ranging from 1 to 5, one being No and 5 being Yes, regarding their own application of the Scrum guideline in their Scrum project. Interviewees are also asked to verbally justify their ratings as they proceeded, explaining the reasons behind the rating of each of the statements. At the end of this section, interviewees are prompted about the other roles, practices and events in their projects that may have not been mentioned in the statements. The third section comprises questions about the different aspects of project success as raised by the literature review. Finally, in the fourth section, interviewees are invited to share their perceptions about specific adjustments they made to the methodology and how these contributed to the project's output. The interview ends with an open question to share anything the interviewees consider relevant.

3.2.1.2 Interview process

The interviews were mostly arranged in a face-to-face format, in the greater region of Montreal, except for two of them. The first one involved a Participant that was working in Brazil, while the second one was with a project manager on a business trip. Two of the interviews were followed up by a message exchange on LinkedIn and by text. In the first case, the contact was established in order to clarify certain aspects of the interview that were unclear to the researcher. The second case, however, concerned a question that was left unanswered during the interview. In both cases, participants welcomed the inquiries and provided the necessary information.

During the interviews, although not followed strictly, the interview guide was used to direct and nourish the interview. The respondents were given freedom to talk about elements that were outside of the scope of the interview guide material but were nonetheless relevant to the research question. As claimed by Harell and Bradley (2009), this structure proved effective in gathering new insights on the topics discussed, as well as unveiling new directions the researcher was not initially familiar with. Similarly, the order of questions and topics in the interview guide were not followed in a strict order, as sometimes certain topics would be brought up before or after others. Furthermore, different interviewees brought different topics to the discussion, which led to different probing questions by the interviewer from one interview to the other as they related to the context and direction the discussion was taking with each participant.

For the purpose of transcription and later analysis, all interviews were recorded, with the consent of the participants. Parallel to conducting the interviews, the researcher took notes of relevant contextual information and details to bring up in following questions. While useful, note taking can also be distracting and interfere with the interview (Doody & Noonan, 2013). It is therefore important that the appearance of note taking does not influence the participants' future answers (Turner III, 2010). With that in mind, the researcher used their sensitivity and interpersonal skills to the best of their ability to ensure the note taking process was inconspicuous and did not interfere with the interview.

3.2.1.3 Sample

The target population of this research is professionals who are or were very recently involved in IT projects carried out using Scrum or some variation of Scrum. Respondents had to be able to answer specific questions about roles, practices, and events during their involvement in the project. Respondents were first recruited based on the researcher's network and that of their supervisor. Then a "Snowball" approach was used to get in touch with more respondents, who were first reached out to by email or through LinkedIn messaging (Appendix C). Once contact was established, other means of communication were used such, as phone communication.

In qualitative research, variation sampling is a way to provide evidence that findings are more likely to be a generalizable phenomenon (Robinson, 2014). Indeed, when a pattern is common across a heterogenous group of observations, it is more likely to be widely generalizable than one found across a homogenous group. This helps establish that the findings are not tied to a particular time, place, group, or context (Robinson, 2014). With that being said, the sample chosen for this research encompasses respondents with different characteristics such as backgrounds, industries, positions and roles in the project.

One of the most significant distinctions between qualitative and quantitative research is sampling. While quantitative inquiry focuses mostly on large samples selected randomly, qualitative research, on the other hand, focuses on small samples that are studied in depth (Patton, 1990). In selecting a small sample with high diversity, not only does it allow for significant patterns to be uncovered, but it also allows for the collection of high-quality and detailed data (Patton, 1990). Such samples allow for depth, rather than breadth, which is ideal for the exploratory purpose of this research (Patton, 1990). Subsequently, given the considerable workload associated with qualitative research and the time limitations of the researcher, the sample studied consisted of 11 interviews (Zyzanski & al., 1992).

3.3 Data Analysis

The data analysis process was conducted in four steps. The first step consists of a detailed transcription of the interviews conducted, followed by an iterative coding process that led to an inter-respondent analysis, and finally, a pattern finding procedure.

3.3.1.1 Transcription

Often described as a chore (Agar, 1996), transcription is also deemed “a pivotal aspect of qualitative inquiry” (Oliver & al., 2005). Indeed, transforming the audio tapes into written text is an essential first step to the analysis. In doing so, all the verbatims were kept, without any sorting. The naturalness of the transcript structure was preserved, and the transcript was made to be an exact reproduction of the interview. These principles drawn on the guidelines of Mergenthaler & Stinson (1992) are considered ideals that can be challenging to achieve in practice. They do provide, however, a direction for the researcher to follow to ensure a certain quality in producing the transcripts.

Interviews lasted between 50 and 90 minutes, averaging 68 minutes each. Every interview conducted was therefore transcribed, totaling 341 pages of verbatim and 31 pages on average.

Project_Respondent	Length (minutes)	Transcription (pages)
INTERNET_Andrew	68	35
MEDICARE_Matt	85	29
ERP1_Yanny	52	32
ERP2_Natacha	57	18
REGISTRATION_Albert	56	31
FINANCE_Eamon	76	51
SPORTRETAIL_Dexter	84	30
STREAMING_Anna	73	23
AUTHENTICATION_Bert	64	27
VIDEOGAME_Fabrice	66	39
FOODRETAIL_Patricia	72	26

Table 9 Length and number of pages per interview

3.3.1.2 Coding

Following the transcription of the interviews, these were meticulously coded. Coding is often considered the core of qualitative analysis and consists of dissecting and organizing different segments of the interviews using codes (Miles & Huberman, 1994). Sandaña (2015) defines code in qualitative research as a piece of text that "symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute for a portion of language-based or visual data" (p. 3). Coding is thus a way to systematically organize data by putting units of data representing a same idea or phenomenon together in order for the researcher to better make sense of it and analyze it.

Sandaña (2015) explains that coding can be inductive, deductive, or hybrid. Given the research question this paper is trying to answer, a hybrid method was deemed the most appropriate. Accordingly, the first step in the process consisted of deductive coding, meaning that the researcher developed preliminary codes based on the findings of the literature review and the research question. With the help of NVivo software, the researcher elaborated a node map that reflected the findings of the literature review for the Scrum roles, practices, and artifacts as well as the various aspects of success. The node map was iteratively improved in order to enhance the granularity and comprehensiveness of the nodes of code. Following this step, a node structure was generated for the first cycles of coding. Annexe C presents a hierarchy chart of the first four levels of nodes.

The next step in the process involved magnitude coding both for the elements of the grid section of the interview and the section of success. Sandaña (2015, p. 69) describes this coding as follows: “Magnitude Coding applies alphanumeric or symbolic codes and/or sub-codes to data, when needed, to describe their variable characteristics such as intensity or frequency.” In the case of this research, magnitude coding was used in the form of words, such as “Yes”, “No”, “Sometimes”, “NA”, to suggest the intensity and frequency of the practices, roles, and artifacts of the methodology, as well as the perception of success. It was also used to analyze the ratings that participants gave to different statements when answering the grid section of the interview. This was useful to interpret participants’ answers, as some did not fill out the rating grid but did orally comment on it, and others were inconsistent in their statements and ratings. Accordingly, Excel tables were developed classifying elements of the grid as well as various aspects of success. This step was significant in the analysis as it allowed to highlight similarities, differences and patterns across different projects.

The third step in the coding process involved creating new coding nodes based on the results of the magnitude coding and extending the level of granularity and detail of the coding nodes. Finally, inductive coding was also used based on an open, data-driven approach to identify any new themes brought by the respondents. Each of the steps was conducted iteratively, and project-based cases and data queries were used in NVivo to assemble themes together and identify patterns.

3.3.1.3 Pattern Finding

In order to assess the possible patterns between the application of Scrum guidelines and project success, matrix tables were developed to map the Scrum guidelines against success aspects. The first step consisted of selecting the Scrum principles to be mapped from the interview table. In order to identify patterns, there needs to be variability in the data mapped on the matrix tables. Due to their lack of variability in terms of adoption, Scrum guidelines that were always followed or never followed were not taken into consideration for the mapping, leaving seven guidelines. Naturally, most of these guidelines were also discussed earlier in the section “Scrum framework guidelines that are not followed” in this same chapter. They can be summarized in the following table:

Guideline	Followed	Partially / Not Followed
The development team is the only one to select the number of elements from the Product Backlog to include in a Sprint.	4	7
The team is not disrupted by other (non-)project related work during the Sprint.	3	8
Daily Scrums are internal to the development team.	6	4
The Product Backlog is used as the only source of requirement.	6	5
The Product Owner is the only person making changes to the Product Backlog.	6	5
Sprint Retrospectives are carried out by the Scrum team to inspect itself and create a plan for improvements to be enacted during the next sprint.	7	3
A Sprint Backlog is made by the development team to forecast the work needed to be done to meet the Sprint goal.	6	3

Table 10 Guidelines with most variability in terms of adoption

These seven guidelines were compared to the two main aspects of success, namely project management success (Iron Triangle) and customer satisfaction[1]:

[1] The remaining aspects of success were not taken into consideration for several reasons. First, the projects studied were either ongoing at the moment of the interview or recently completed. While this method of selecting respondents helped minimize retrospective bias, it was less useful in assessing organizational success. Organizational success is a long-term criterion that could

3.3.1.3.1 Project Management Success (Iron Triangle)

Although the literature review established the fact that the Iron Triangle (budget, schedule, technical performance) is an overly simplistic way of assessing project success, according to Byrde (2008), the Iron Triangle remains the most cited success measure among managers of different industries. Indeed, its simplistic criteria make it easier to assess whether or not a project is successful (Jugdev and Müller, 2005). Because of its long-standing establishment and its measurability, the Iron Triangle, with its three constraints, remains a core measure of project success and a golden standard in the project management field (Papke-Shields & al., 2010).

3.3.1.3.2 Customer satisfaction

The customer satisfaction aspect of success is considered the very end for which Agile methods were developed, along with increasing the quality of the product developed (Williams & Cockburn, 2003). Indeed, customer collaboration is one of the four core principles of Agile. Thanks to the collaborative environment provided by Agile frameworks, customers give and receive feedback throughout the project, which contributes to their satisfaction (Dyba & Dingsoyr, 2009).

3.4 Ethical considerations

The present study was conducted in accordance with the rules and policies of the Research Ethics Board of HEC Montreal, which in turn adheres to the requirements of Canada's three granting agencies: The Social Sciences and Humanities Research Council

not be assessed at the moment of the interview as it is often measured after project closure. Moreover, in most projects, clients were either internal or both internal and external, making senior management and project sponsor satisfaction equivalent to client satisfaction. Product owner satisfaction and team satisfaction were also two aspects of success that overlapped, as most Product Owners were also part of the development team. This category was also disregarded because it is not as significant, as development teams were full-time temporary resources in the organizations.

(SSHRC), the Canadian Institutes of Health Research (CIHR), and the Natural Sciences and Engineering Research Council (NSERC).

Consequently, a Certificate of Ethics Approval for this research was granted on the 4th of September 2018. The aforesaid certificate ensures that human subjects involved in this study do so voluntarily, in addition to being given all the necessary information to make an informed consent. To guarantee the compliance to these rules, all respondents were presented with a consent form that guaranteed their anonymity and explained the steps undertaken by the researcher ensure data confidentiality throughout the entire research process.

CHAPTER 4 : RESULTS

This section presents the results following our data analysis. It is worth mentioning that the unit of analysis in this study is software projects using the Scrum framework. Accordingly, this chapter can be sectioned into four main parts: first, an overview of the context of the eleven projects, mainly describing the organizations, the projects themselves and the respondents who talked about them. Second, the results concerning the adoption of the Scrum Framework guidelines in the project studied. Third, the emergent practices identified during the analysis stage, and finally, the main patterns identified in relating Scrum guidelines to project success.

4.1 Projects and respondents

For every project studied, one respondent was interviewed. A grid was used to summarize the characteristics of projects and respondents:

Projects	Team size	Project type	Company Size	Company's industry	Status & sprints completed	Res-pondent	Role
MEDICARE	>9	Independent project	Medium	Medical Device	Ongoing: Approx. 22	Matt	Developer
FINANCE	>9	Core business	Large	Financial services	Ongoing: 4	Eamon	PMO
SPORTRETAIL	4-9	Independent project	Large	Sport clothing and equipment	Completed: 32	Dexter	Agile coach
REGISTRATION	4-9	Core business	Medium	Online registration services	Ongoing: Approx. 200	Albert	VP
STREAMING	4-9	Core business	Medium	Online streaming services	Ongoing: Approx. 5	Anna	Product Owner
ERP1	4-9	Core business	Medium	ERP and software development	Ongoing: 14	Yanny	Architect & Developer
INTERNET	>9	Core business	Large	Internet services	Ongoing: Approx. 85	Andrew	Developer
AUTHENTICATION	4-9	Core business	Medium	Security and authentication systems	Ongoing: Approx. 60	Bert	Scrum Master
ERP2	0-3	Core business	Medium	ERP	Ongoing: 4	Natacha	Developer
VIDEOGAME	>9	Core business	Large	Video Games	Completed: 30-60	Fabrice	Developer
FOODRETAIL	NA	Independent project	Large	Food Retail	Ongoing: 9	Patricia	Project Manager

Table 11 Overview of projects and respondents

4.1.1 MEDICARE

MEDICARE’s organization is a private medical device company. The organization has branches in different North American cities, and that of Montreal comprises of around 200 people. MEDICARE’s first attempt at Agile development was in 2011, and today, the organization claims to be completely agile since 2012.

Project MEDICARE involves the development of a robotic-assisted surgery system. The project, which started in 2016, represents an independent initiative in the IT department of the organization. MEDICARE is divided in two big sub-projects. The first sub-project, which lasted a year from 2016 to 2017, consisted in a “Proof of Concept”. During this first part, no paperwork was produced, and the focus was put on making sense of the project itself and its requirements. This was mainly done through iterative discussions and feedback between the development team and “surgeon developers” who represented the customer side. The second phase consists of the actual development of the solution through Sprints of three weeks. At the time of the interview, MEDICARE’s team had completed 22 Sprints and was in the second phase of the project.

For this project, Matt, senior developer with a focus on testing, is interviewed. Matt’s educational and professional background revolves around software engineering, as he had earned a bachelor in the same field and worked in several positions involving coding and testing.

4.1.2 FINANCE

FINANCE’s organization is a multinational company operating in investment banking and financial services. Project FINANCE takes place in the enterprise cybersecurity department of the organization. Until January 2018, the department had been managing its projects following the traditional waterfall approach. January 2018 marked the beginning of the shift to Agile methods, when Agile became the main “buzzword” inside the company. Nonetheless, at the time of the interview, only a few teams within the enterprise cybersecurity department had adopted Scrum. Some teams were in the process of adopting the framework, while others were still not on board. The team of project FINANCE belongs to the group that had recently adopted Scrum.

Project FINANCE concerns a hygiene security infrastructure work involving a computer network authentication protocol. The project, which was 4 Sprints old at the time of the interview (17th January 2019), was carried out in Sprints of two weeks and was expected to be completed by April 2019. The team working on the project was made of 12 full-time employees, most of whom were subject matter specialists.

The respondent for project FINANCE is Eamon, head of the Project Management Office (PMO). Before assuming his current position, Eamon studied information system management and earned an MBA. He worked in web development and managed e-content and digital projects before moving to PMO. Although not yet official, Eamon mentioned that he was in process of earning a Scrum Master certification.

4.1.3 SPORTRETAIL

SPORTRETAIL's company is an important sport clothes reseller based in Germany. The company envisions expanding its e-commerce activities to several new countries as part of its commercial strategy. Prior to this project, the company had no previous experience working with Agile.

Project SPORTRETAIL represents a major initiative in support of the company's strategy, as it involves the development of different websites for different countries to support the company's e-commerce activities. In total, fifteen Scrum teams were mobilized to create these different websites, totaling 220 people involved in the project.
Project SPORTRETAIL, which started in 2016, is the first attempt at Agile for the company. Although eager to shift to Agile, the company counts very few Agile experts. In fact, only two or three of the 220 people involved in the project have an expertise in Agile. Before hiring the expertise of an Agile coach, SPORTRETAIL was a challenged project that suffered internal dependencies and conflicts between its different Scrum

teams. Moreover, SPORTRETAIL was performing poorly in terms of time, budget and quality.

The respondent for this project is the Agile coach who joined the project for 10 months starting in April 2017. Dexter, Agile coach, studied administration and business communication prior to working in project management. He assumed different roles in project management, including that of a Scrum Master, before becoming an Agile coach. Throughout his career, Dexter has earned several certifications from the Scrum alliance, the Management and Strategy Institute, and other Agile educational and training organizations.

4.1.4 REGISTRATION

REGISTRATION is the core software product of an online registration company. The company, which counts around 150 employees, takes pride in being Agile as a whole. Not only that, the company also takes initiatives such as speaking engagements to educate its own market to shift the way the market does business to a way that matches that of REGISTRATION company. In the company, the term "project" is switched for "initiative," and while initiatives also have start dates, they do not have set end dates, unlike projects. Instead, the teams are assumed to have motivated individuals that are doing as much as they can. During the interview, the respondent reported that the company is undergoing internal growth and scaling its developers' team from 29 to 42 employees.

Because project REGISTRATION concerns the core product of the company, it has no end date, as the software is constantly being improved and updated. Accordingly, the company reports an average of two updates per day. In the project, five different Scrum teams work together on the same code base. They have different Product Owners, Sprint goals and problems to solve, but are working towards the same bigger initiative. The project is carried out in Sprints of two weeks, and over 200 Sprints had already been completed at the time of the interview.

The respondent for this project is Albert, VP product of the company, also known as the Product Owner of Product Owners. In addition to being responsible for coordinating and harmonizing the different teams, Albert also assume the role of Scrum Master at the early stages of the project. Albert has a technical background as he earned a bachelor's degree in software engineering and worked as a developer for several years before switching to Agile project management. Albert also holds a Scrum Master and a Product Owner certification from Scrum.org. In addition to his professional responsibilities, Albert also takes part in speaking engagements and Agile speaking tours, as well as educational presentations on Agile.

4.1.5 STREAMING

With its 50 employees, project STREAMING's company is a small media services company that is a newcomer to Agile. Indeed, the company has only shifted from traditional to Agile project management Scrum in its last year. In initiating the shift, the company hired four Product Owners, three Agile coaches and two Scrum Masters (one for every two teams).

STREAMING is a project directly related to the core product of the company. Accordingly, the workload was split into four different teams. The first team was in charge of the User Experience, while the second team handled artificial intelligence and content generation. The third team was in charge of customer lifetime value. Finally, the fourth team took care of back office work. Project STREAMING is carried out in Sprints of three weeks, five of which were completed at the time of the interview.

The respondent for this project, Anna, is the Product Owner in charge of one of the four Scrum teams of the project, namely the back-office. The team handles essential elements of the core product of the company such as billing, login, and customer support. Anna earned a bachelor's degree in international business. She then worked in the CRM industry before switching to her first role as a Product Owner in a communication agency. Soon after, she joined STREAMING to work as a Product Owner.

4.1.6 ERP1

ERP1 was carried out in a middle-sized software development company specialized in ERP solutions. The company has been using informal Agile practices for the past eight years and has formally introduced Scrum in the last two years.

Project ERP1 aimed at developing new features to the current core ERP software product that the company is commercializing. The new features allow the penetration of new markets for the company. The project started in October 2018 and was completed in May 2019. The team working on the project included between ten and fifteen senior developers. Because of the team members' expertise and their years of experience working together, the project had no formal Backlog. Instead, most technical matters were discussed orally or expressed as comments in the code itself.

The respondent for project ERP1, Yanny, is a senior architect. He comes from a technical background, having earned a bachelor's and a master's degree in software engineering. In addition to designing solutions' architectures, Yanny is frequently involved in project management and is one of the firm's associates.

4.1.7 ERP2

Project ERP2 takes place in a middle-sized ERP solutions provider. The company, which has around 300 employees, offers integral ERP solutions paired with a managerial learning system that teaches businesses them how to best leverage the IT solutions they offer. Indeed, in addition to developing in-house ERP solutions, the company offers its own "University" in the form of online classes, lectures, exams, and professor meetups. The company is relatively new to Agile, having shifted its project management practices in the last year.

Project ERP2 takes place in the Innovation department of the company and consists of a new add-on feature to the initial ERP product. The feature, referred to as Business Intelligence Analyst, is an intelligent system that serves as a personal assistant to the users. The assistant can be accessed on several devices, such as tablets and mobile phones, and offers summarized and simplified explanations of the business data, metrics and indicators in the ERP. The project started in January 2018, and although it does not have an official release date, was expected to be completed in 2019. At the time of the interview, the team had been strictly using the Scrum framework for two months and was on their fourth Sprint. Prior to that, the team was using an informal Agile project management approach that did not involve a Backlog or strict meetings.

The respondent in the case of ERP2 is a developer named Natacha. Natacha recently earned a bachelor's degree in computer science and is starting her career in software development.

4.1.8 INTERNET

Project INTERNET was initially started in a small technology start-up that was later acquired by a multinational company specializing in internet-related services. The company has been using Agile methodologies for the past three years, which is also the length of project INTERNET.

The project revolves around one software application product aimed at supporting human resources processes in businesses. Hence, it is mostly concerned with features updates and technical enhancements. For instance, the team focused on customer onboarding, and their current objective is to develop public APIs for the application to integrate with other software products. The project is carried out in Sprints of two weeks with no specific end date. The main pressure in terms of schedule comes from the company's sales department, which pushes for new features to be developed faster in order to be put out in the market. Seven Scrum teams consisting of between three to nine members are involved in the development of the application.

The respondent for this project is Andrew, developer and one the first employees of the company. Andrew's education is mostly focused on engineering, but his computer science skills were mostly self-taught or learned in job internships.

4.1.9 AUTHENTICATION

AUTHENTICATION’s company is a start-up in expansion, operating in Real Estate Access-control Management. Accordingly, the company develops software solutions adapted to customers’ needs to manage security and access to buildings. The company has been trying to instill agility since its very beginnings three years ago. However, it has only started a stricter application of the approach after hiring an Agile coach a year and a half after, and a full-time Scrum Master six months later.

Project AUTHENTICATION has been ongoing for three years, as it is the core software product of the company. The length of Sprints vary from one week to one month, and around 60 Sprints have been completed. The development is shared between three teams: the mobile team, the development team, and the embedded team. The focus of the interview is on the imbedded team.

The respondent of this project, Bob, is the Scrum Master who has joined the company the previous year and was in charge of the three development teams. Bob studied management information system and started his career as a business system analyst. He then developed an interest in Scrum and held a first Scrum Master position before joining project AUTHENTICATION.

4.1.10 VIDEOGAME

VIDEOGAME’s company is a major player in the industry of video games. While the company is not entirely agile, most of the development work is based on Agile methodologies. It is not known, however, when the development teams started adopting the methodology, as it varies depending on the team.

Project VIDEOGAME was about launching a new version of an already commercialized video game. The project was carried out in Sprints that lasted between two and four weeks. The project was lasted approximatively three years and was completed in February 2018, although the exact number of Sprints completed was not known to the respondent. Around 1,000 employees were involved in the development of this videogame, which led to adopting a hierarchical structure with several layers of sub-teams.

The respondent for this project, Fabrice, was one of the developers involved in the animation section of VIDEOGAME. Fabrice is a recent graduate with a Bachelor of Computer Science who joined the company as a junior developer. Project VIDEOGAME was his first experience working with the Scrum framework.

4.1.11 FOODRETAIL

Project FOODRETAIL took place in a big food retail company. The company still ran most of its projects with a traditional waterfall approach, fixing the scope at the beginning of its projects.

This project, which aimed to merge the different wholesale websites of three different regions, was considered a concretization of the company's global strategy to harmonize its processes and unify its practices. The project had kicked off in November 2018 and was planned to be completed by April 2019. The project was comprised of different phases that were handled as sub-projects. The first phase, also known as the front-end phase, was the only sub-project in development at the time of the interview and was therefore the focus of the discussion. For the front-end portion of the project, the company worked with external resources, outsourcing the development to an external Scrum team. That resulted in the development team having a customer/service provider relationship with the business analysts and the project managers. Nevertheless, the first phase was considered a success in terms of time, technical performance and customer satisfaction. Although it was a considerable expense for the company, the project manager considers that outsourcing development is a cost-efficient decision on the long term, especially for strategic projects such as FOODRETAIL.

The respondent for project FOODRETAIL was a newly appointed project manager who studied business and IT management. She had worked as a business analyst before joining FOODRETAIL as a project manager.

4.2 Consistently followed Scrum guidelines

4.2.1 The Product Owner as one person and not a committee

In 10 of the 11 projects studied, respondents reported that the Product Owner was one person rather than a committee. This guideline, which is explicitly stated in several versions of the Scrum Guide, stipulates that the role is filled by one person in spite of the fact that this Product Owner may represent the desires of a committee (Schwaber & Sutherland, 2017).

While participants reported that this guideline was consistently followed at any given time in the project, over time the guideline could be challenging to implement because of employee absences and significant workload. Accordingly, an interesting finding was that “Product Owner” was considered a role rather than a job specifically assigned to one person. Accordingly, it could be transferred and even passed back and forth, as was the case for Project ERP1 and AUTHENTICATION respectively.

> The project has one Product Owner that is a person and not a committee. Yes. Well, yes, except in particular situations where there are two people […] It happens because sometimes there are holidays and other things, so there is a takeover knowledge transfer, and a return to normal after they’re back (Yanny, ERP1).

> It's actually a bit peculiar because I would say yes, we have a Product Owner...We have more than one Product Owner, one as a junior Product Owner and one as a senior Product Owner, as you have seen. [Employee] was the Product Owner, except that with the high number of requests, we have a Product Owner help him make user stories, his name is [Employee2], he is in New York. He often meets customers and also collects data and talks about it with the Product Owner to decipher it (Bert, AUTHENTICATION).

Moreover, as it is a role, it can be added to previously held responsibilities, as was the case for project ERP2 and VIDEOGAME. Natacha, developer in the project ERP2, claims: “Then there is my Product Owner, my boss, who is also a developer, so it's not clearly divided.” Similarly, Fabrice, developer in project VIDEOGAME, explains:

> We have what is called a creative director, who is the one who makes the creative decisions but, […] I do not think it's totally true [that the PO is one person and not a committee] because I think that he works together with other people, but have one creative director.

4.2.2 The Scrum Master in charge of enforcing the Scrum process

The Scrum Master is described in the Scrum Guide as a “servant-leader”. Indeed, the Scrum Master is in charge of promoting and supporting the application and understanding of the Scrum framework throughout the project (Schwaber & Sutherland, 2017). All of the subjects interviewed asserted that their Scrum Master was “in charge of enforcing the process and avoiding distractions”.

Nevertheless, similarly to the role of Product Owner, that of Scrum Master was sometimes combined with other responsibilities rather than carried out as a separate and independent position. For Project REGISTRATION, Project SPORTRETAIL and project FINANCE, Scrum Master was a hat worn by someone with other responsibilities in the project. For project AUTHENTICATION, the role of Scrum Master was enacted by the VP product, who also happens to be the Product Owner of Product Owners in a scaled organization of Scrum containing several Scrum teams. In project FINANCE, the PMO took on the responsibility of facilitating events and introducing the framework to his teams. In project SPORTRETAIL, an employee was also taking care of facilitating Scrum ceremonies without being themselves a Scrum Master:

> No, we have zero Scrum Master. I will put a two because the role is enacted by me at this point.[…] For the first eight months of the application of agility, I was spending, I was doing 100% of the job of a Scrum Master with every single team because I wanted them to pick it up in their own proper manner, if you would, so that at some point I could step out and they would share the role of the Scrum Master […]So in one team a developer plays the role of Scrum Master, and another team it’s the Product Owner, and another team it’s me, so it kind of varies, but the role is still here (Albert, REGISTRATION).

> My role is not called Scrum Master, but you can say I run the Sprints, yeah, call me Scrum Master, but it's not official at all.[…]I am not Scrum Master. But I do it. That's the why, in theory it's something, but in reality no one looks at you like "Oh, he's a Scrum Master," like, "Wow" (Eamon, FINANCE).

> [On having a Scrum Master] Not really. There was a person who was facilitating the ceremonies, but who was not really a Scrum Master. (Dexter, SPORTRETAIL).

4.2.3 Sprints are four weeks or less, carried out one after the other

Across all the projects investigated in this study, participants reported conducting Sprints of four weeks or less. The length of the Sprints varied from a project to another within the same company or department. Shorter Sprints came across as more demanding and requiring development teams with higher levels of expertise, as the delivery pace is faster. Respondents of project STREAMING and MEDICARE illustrated that:

> Les Sprints durent 4 semaines ou moins. Oui nous on avait 3 semaines. J'ai déjà fait deux semaines, c'est vraiment beaucoup de rituel, à un moment donné. Des fois tu n'as pas le choix, si ton projet dure 2 mois bien tu n'as pas le choix, j'ai même vu des Sprints d'une semaine. Ça, c'était trop intense (Anna, STREAMING).

> Nous on faisait des Sprints de 3 semaines. On a toujours fait 3 semaines dans ce projet ci. D'autres projets on avait fait 2 semaines, mais 2 semaines on était, c'était du monde plus 'cranké' Sprint là, tu sais, ça roulait (Matt, MEDICARE).

In addition to respecting the one-month time box indicated by the Scrum Guide, 10 out the 11 projects declared starting a new Sprint immediately after one is completed. The only exception, project FOODRETAIL, justified the delay between Sprints by the fact that the team did not always have all the necessary indications regarding the requirements to be able to start the next Sprint. Indeed, the confirmation of the requirements, responsibility of the Project Manager, the Business Analyst and the Product Owner delayed the start of the Sprint by one or two days.

4.2.4 Sprint Reviews are carried out at the end of each Sprint

Another key guideline that the projects studied seem to adhere to is holding a Sprint Review meeting at the end of each Sprint. As previously explained in the literature review section, a Sprint Review is a meeting carried out to inspect the increment produced during the Sprint in the presence of the development team, the Product Owner, and key stakeholders (Schwaber & Sutherland, 2017). The main goal behind such meetings is to foster feedback and collaboration and adapt the Product Backlog if needed.

10 out of the 11 projects studied confirmed carrying out a Sprint Review meeting at the end of their Sprints. The remaining project -INTERNET- did not carry them out at all. It is important to note, however, that out of the 10 projects that did have Sprint Reviews after each Sprint, only three of them asserted the presence of key stakeholders, including customer and end-user representatives, on the regular. These projects were FINANCE, STREAMING, and FOODRETAIL. For MEDICARE and SPORTRETAIL, key stakeholders were updated and prompted for feedback about the product, but with bigger time intervals. For MEDICARE, a demo was set up every quarter to gather insight and feedback. SPORTRETAIL, on the other hand, had its key stakeholders attend Reviews on an irregular basis. Project REGISTRATION deemed that because the product developed was meant to be generic, internal stakeholders that were not part of the team were enough to gather valuable feedback:

> The reason why this is happening at REGISTRATION is a reflection of our business model. Because we are building a single generic product for thousands of businesses and hundreds of thousands of users, stakeholders aren't external customers of REGISTRATION; they usually will be executive sponsors, or members of the product team. I don't have the exact stat, but it's safe to assume over 95% of our Sprint Reviews are joined by internal stakeholders and guests that didn't contribute to the development but will benefit from the product increment.

ERP1, ERP2, and AUTHENTICATION had some stakeholders attend, but had no customer/end-user insight throughout the development phases. The end-users for these projects would only be able to see the final product once the development is completed. The Scrum Master and respondent from AUTHENTICATION expressed an interest in initiating customer demos in the upcoming Sprints, saying it could provide a significant improvement to their product. According to him, this feedback would increase greatly the team's motivation and would even help feature-wise (Bert, 2019). Moreover, he also explained that the fact that the requirements travel from the customer to the Product Owner to the development team creates a gap between how each party understands and expects the work to be done. This extract from Bert's interview summarizes his idea:

> En termes de motivation. En termes de feedback, en termes de genre euh... feature-wise et everything. C'est très différent. Interne et externe c'est deux affaires différentes, parce que qu'est-ce que moi je dis c'est que les PO sont quand ils créent un produit c'est toujours leur bébé c'est sûr que c'est le perfect design. Mais la réalité souvent c'est qu'un client regarde et dis « ah bien c'est pas exactement ça que je voulais et ah j'aimerais mieux ça » et par la suite bien… Genre si le Product Owner avait juste meet avec le client, bien souvent tu vas faire que tu protèges encore un peu, so tu fais un peu changer les concepts et donner au dév et le dév pense ah « OK c'est ça ». Mais si tu vas vraiment au complet le dév va avoir un sentiment et va dire au PO « ah tu tu te rapelles le client? avait vraiment dit ça, je pense que c'est ça qu'il voulait dire » et le PO va dire « bon OK ». So il y a un peu de sentiment, mais oui. Euh. Mais il y a beaucoup de choses très avantageux de montrer aux clients, que les dév soient là. La contribution, c'est ça que je dis, c'est très important.

4.3 Rarely followed Scrum guidelines

4.3.1 Deciding the number of elements to include in a Sprint

According to the Scrum Guide, the number of elements to include in a Sprint shall be "solely up to" the development team (Schwaber & Sutherland, 2017, p.10). However, for over half of the projects studied (7 out 11), this rule was challenged. With the exception of project REGISTRATION, which did not have a formal selection of elements to include in the Sprint, six projects had other parties included in the selection process.

For project AUTHENTICATION, the Scrum Master reported being involved, mainly to validate the choices of the development team in terms of velocity. His role was therefore to ensure that the development team was going to work at optimal capacity by selecting the right number of elements. This process is done by breaking down stories to sub-tasks and estimating their difficulty in T-shirt sizes. Each T-shirt size correlates with the number of hours of work the sub-tasks demanded. For instance, an XS would require around an hour of work, while an XL would require 16 hours. A consensus on the number of hours needed to complete the sub-tasks was reached through “Poker Voting” and discussions. This means that the developers vote all at once on the number of hours needed for a specific sub-task using cards on which they write down a number of hours. Following the vote, discussions may be initiated to clarify the thoughts and expectations of voters with different numbers in order to reach a consensus.

In project ERP1, the decision regarding the elements to be included in the Sprint was made in the presence of the entire Scrum team. However, each team member would decide for themselves since they were often involved in different projects and did not have a formal and unified Backlog for any of them. Rather, each team member would have their own notes of things to do and would share them with the rest of the team.

In project VIDEOGAME, the development team was not the only one deciding on the number of elements to work on during a Sprint, mainly because of the hierarchical structure that involved several people between the Creative Director (who was also Product Owner) and the development team. The developer interviewed for this project used to receive the work to complete in a Sprint by email. Finally, projects MEDICARE, FOODRETAIL, and STREAMING reported a constant back and forth with the development team on one side and the Product Owner on the other when deciding how many elements to work on in the Sprint. The Product Owner, representing the client needs, would generally push for more work to be done in the least amount of time possible, while the development team would push back to avoid delays and skipping tests. Matt, developer at MEDICARE, summarizes this idea very well:

> L’équipe de développent est la seule à sélectionner le nombre d’éléments du projet Backlog à inclure dans un Sprint. Nombre d’éléments… ça, ça a été difficile. Ça a été difficile parce que le PO veut toujours en pousser plus, puis c’est comme « let’s go les gars, on est à 110 % de capacité mais on va l’avoir ». On n’a jamais réussi à remplir dans-, cette équipe-là, on a le record absolu, mais c’est pas vrai parce qu’il y a plein d’équipes qui font ça, mais pour moi c’est inacceptable. Mais moi je m’attends, dans d’autres équipes, si on avait, mettons on décidait qu’on avait 30 points, on avait une vélocité moyenne de 30 points mettons, c’est un chiffre abstrait, puis que là on planifiait 30 points, on livrait 30

> points 90 % du temps ou 95 % du temps. Tous les Sprints, on planifiait 30 points et on livrait 30 points et on avait une journée à la fin du Sprint, dans le meilleur des mondes, on avait une journée à la fin du Sprint pour planifier là, tu sais pour faire des tests informels euh, tu sais de s'assurer que tout est beau là, puis de rajouter un petit peu de, tu sais de cadrer nos affaires. Pas dans cette équipe-là. Dans cette équipe-là, c'était le concept proche, on finissait 2 jours après tu sais ? Tout était plus ou moins fini à la fin. Fait que là qu'est-ce qui arrive quand tout est plus ou moins fini ? On finit l'essentiel, mais la couverture de test n'est pas adéquate, les specs sont pas ajustés de façon adéquate, il y a plein d'affaires qui ne sont pas adéquats qui deviennent une sorte de dette. Puis à mon avis ça vient d'une pression aussi, un déséquilibre entre le PO qui veut toujours que ce soit livré le plus vite possible et qui ne comprend pas le concept de dette technique puis euh, puis d'équipe qui fait « ah c'est bon, on va serrer les coudes puis cette fois ci on va réussir ». Ça marche jamais. Fait que ça n'a pas été la pire catastrophe, mais je suis vraiment obligé de coter négatif.

In the case of FOODRETAIL, this balance of power was even more relevant, as the development team was external to the company. FOODRETAIL was therefore on the client side, which led to the business analyst and the Product Owner at FOODRETAIL being heavily involved in these decisions with the development team. The remaining four projects studied did report that the development team was the only one involved in the choosing the number of items to be included in each Sprint.

4.3.2 Distractions during a Sprint

One of the responsibilities of the Scrum Master, according to the Scrum Guide, is to identify and remove any impediments that prevent the development team from progressing towards the Sprint goal. Moreover, the guide also provides directions on how to handle new work that emerges during a Sprint. Accordingly, the guide suggests that the development team adds any new work to the Sprint Backlog and to update the estimated remaining work consequently. As per the Scrum Guide, only the development team may change its Sprint Backlog during a Sprint (Schwaber & Sutherland, 2017, p.16). Having said that, the projects studied evidenced another reality. In fact, of the 11 projects studied, only three projects confirmed that their development teams were not disturbed by other work, whether it be related or not related to the Sprint. These projects were MEDICARE, REGISTRATION, and ERP2. In compliance with the Scrum Guide recommendations, Matt from project MEDICARE explained why it was important to avoid distractions and adjust for new work in the planning in order to deliver a successful Sprint:

> L’équipe de développement n’est pas perturbée par d’autres tâches liées au projet lors du Sprint ? [Reading the guideline] Ça, le PO n’était pas très bon là-dessus. Il pouvait avoir des urgences et ce qui est poche dans les urgences c’est que ça enlevait pas le focus, on ne changeait pas, si on rentre « hey là, là, il y a telle affaire super critique », bien ok là, mais il faut revoir notre planning parce que si tu me mets là-dessus, moi je ne travaille plus sur ce que je travaillais puis là on ne livre pas ce qu’on nous avait demandé de livrer.

Projects AUTHENTICATION, SPORTRETAIL, INTERNET, FINANCE, and ERP1 were on the other side of the spectrum, reporting frequent disturbance and repeated adjustments to the scope of the Sprint. In these projects, development teams were disturbed by both Sprint-related and non-Sprint-related work. For project AUTHENTICATION, for example, Bert the Scrum Master attributed the disturbances to the fact that the company was a start-up with a small team. SPORTRETAIL, on the other hand, concerned a big company, but experienced similar issues related to the development team disturbance. Dexter, the Agile coach interviewed within this project, blamed this disturbance for the team’s lack of productivity:

> C’était un des gros sujets sur lesquels on a travaillé c’était ça. C’était lié aussi au fait que le Scrum Master, il n’y avait pas de Scrum Master donc pas forcément de personne pour protéger un petit peu l’équipe. C’était lié aussi au fait qu’il y avait beaucoup de dépendances entre les équipes, enfin avec une autre équipe en particulier. Euh, que ces dépendances créent des situations où un développeur était « réquisitionné » pour aller aider l’autre équipe. Et c’était aussi lié au fait qu’il y avait une personne dans l’équipe qui avait une maîtrise assez forte du projet dans sa globalité, une très bonne connaissance, une très bonne expérience dans la société, donc une expertise très large, ce qui fait que quand d’autres équipes avaient des problèmes elles n’arrivaient pas à trouver de solutions, elles allaient lui poser des questions. « Qu’est-ce que tu ferais… ? », « est-ce que tu peux nous aider ? », etc. Donc l’équipe était, c’était un des éléments qui expliquaient pourquoi l’équipe avait du mal à produire euh, avoir à trouver une capacité de production assez intéressante c’est que ils étaient beaucoup perturbés.

Finally, the development teams of projects VIDEOGAME and STREAMING were only disturbed by project-related work. This means that the new work fell within the scope of the project at hand and within their responsibilities.

4.3.3 Daily Scrums that are not internal to the development team

As previously established, Daily Scrums are 15 minute time-boxed daily meetings, internal to the development team, that aim to discuss progress, increase visibility, and avoid any impediments (Schwaber & Sutherland, 2017). Among the respondents interviewed, only five projects were found to adhere to this format. These projects were REGISTRATION, ERP2, SPORTRETAIL, AUTHENTICATION, and MEDICARE. In project FINANCE, Daily Scrums were not carried out at all. Eamon, the respondent from project FINANCE shared that he considers it a rather improbable scenario at his workplace: "No daily Scrums… Yeah, there is no way they'll join daily." In Project VIDEOGAME, the project had several sub-Scrum teams because of the large number of contributors to the project. While some sub-Scrum team would have Daily Scrums, other would do them on a less regular basis. That was the case of Fabrice, a programmer in one of the sub-Scrum teams who said:

> Il y a de daily Scrum effectués mais pas dans toutes les sous équipes. Comme moi, c'était pas chaque jour. Mais mon ami lui c'était chaque jour.

He explained that Daily Scrums were more useful and common in larger teams, which joins the idea expressed by Natacha, a developer in the ERP2 project:

> I'm not a huge fan.[...] Just because, as I told you, in our case, we're already a small team so instead of cutting 15 minutes of what, as long as I'm concerned, is reiterating of what I've been telling my boss since the last hour he asked, because he checks in frequently. It just feels like an overhead that is not necessarily fit for our situation.

In projects INTERNET and STREAMING, the Product Owner would also attend the Daily Scrums, whereas in project ERP1 all the departments of the company are present during the Daily Scrum, except the accounting one. In project FOODRETAIL, although the development team is external, the business analyst from FOODRETAIL attends the Daily Scrums of the external team. Having Daily Scrums that are internal to the development team appears to be a guideline that is often changed in industry.

4.3.4 Product Backlog as the only source of requirements

In the Scrum framework, the Product Backlog is meant to be the only source of requirements for the product developed (Schwaber & Sutherland, 2017). Accordingly, any changes to be made to the product should necessarily go through the Product Backlog. Five of the interviewed respondents confirmed it was the case for their projects. These projects were REGISTRATION, STREAMING, SPORTRETAIL, FINANCE, VIDEOGAME, and ERP2. Anna, the Product Owner from project STREAMING, attested that it was not always easy, as new things would sometimes pop up in the middle of the project. She explained that the team did its best to respect this guideline:

> Après ça, le Product Backlog utilisé comme seule source de besoin. En théorie oui…Dans le sens que oui n'importe quoi qui entrait je le faisais entrer par là. Mais tu sais en théorie, il peut y avoir des petites exceptions des fois c'est sûr. Quelqu’un qui vient voir quelqu'un des trucs comme ça. Le but c'est de s'en aller vers ça, oui.

For projects INTERNET, AUTHENTICATION, and MEDICARE, not all requirements went through the Product Backlog, although the major portion did. In the case of INTERNET, functional requirements related to technical enhancements of the product in development did not go through the Product Backlog. On the other hand, for project AUTHENTICATION, most of the requirements did go through the Product Backlog, except for the ones related to the support provided to customers. As a start-up, the team in charge of the development of the AUTHENTICATION project was also responsible for client support. Bert, Scrum Master at AUTHENTICATION, explains why the team still struggles to adhere to this guideline:

> Pas toujours le cas, de temps en temps. Parce que souvent aussi on doit faire du support et le support, on fait à l'externe avec un client. Mais ce n'est pas toujours été loggé dans le Product Backlog. Oui, parce que puisque nous sommes une petite équipe souvent il y a des supports, d'habitude QA est la frontière, so QA déjà est impacté. Parce qu'on n’a pas de channel de support comme un call center, ou quoi que ce soit. C'est vraiment nous là. So oui il y a le Product Owner qui va essayer de les supporter. S'il n'est pas capable, ça tombe à nous puis c'est le côté de développement qui va faire le support…C'est souvent ça peut être sur Slack, par email, quoi que ce soit.

On projects FOODRETAIL and ERP1, the Product Backlog was not the primary source of requirements. In project ERP1, there were several sources of requirements from one project to the other, and the team did not have a formal Product Backlog. Instead, they would rely on contract documents, comments from colleagues, and changes required by the project manager. In the case of FOODRETAIL, the Product Backlog was mainly a restatement of a document named the Detailed Requirement Analysis. This was a document signed by the key stakeholders at the start of the project and was considered as the main source of truth during the project, as the Product Backlog's scope could not stray from the official document:

> A cause des politiques internes de l'entreprise, on doit également avoir un document de requis. Ça s'appelle Detailed Requirement Analysis, du « requirement analysis », puis ça c'est encore plus la source des besoins que le Product Backlog… C'est pas plus détaillé mais c'est plus la source de vérité. C'est plus officiel, par ce que c'est la chose que les utilisateurs puis les clients externes qui sont le VP wholesale, comme le project sponsor là, c'est ça qu'eux ils ont signés. …C'est comme un contrat que nous les TI on a avec eux… Nous les TI on va leur expliquer : « voici le projet, voici ce qui entre dans le scope du projet, voici une charte de projet, signez ça », puis là ils signent ça. Puis nous on prend ça comme la source de vérité principale… Ça c'est moi qui le fais, avec la Product Owner, moi je suis responsable de ça… Parce qu'on est pas une organisation agile, fait que on est obligé de faire ça. Puis ce document est utilisé - c'est le même, puis le même prototype pour les projets waterfall (Patricia, FOODRETAIL).

4.3.5 The Product Owner's responsibility over the Product Backlog

One of the most fundamental guidelines presented in the Scrum Guide was often found to be modified in no less than five of the projects studied. As it happens, the Scrum Guide highlights several times the responsibility that the Product Owner has over the Product Backlog. As a matter of fact, the Product Owner is said to be the only one responsible over the content, ordering, and availability of Product Backlog (Schwaber & Sutherland, 2017). And while other team members may influence or work on the content of the Product Backlog, the Product Owner remains the person accountable for it. Projects REGISTRATION, INTERNET, STREAMING, AUTHENTICATION, ERP2, and MEDICARE were run according to this principle. Nevertheless, the case was different for projects FOODRETAIL, SPORTRETAIL, FINANCE, VIDEOGAME, and ERP1.

Interestingly enough, the five projects that did not adhere to the principle had different reasons not to. In the case of SPORTRETAIL, the Product Owner was a beginner at his work. He therefore requested help from different team members. However, he did not take care of filtering and homogenizing the content entered by different team members, which resulted in a disorganized Backlog, as Dexter, Agile coach explains:

> Euh, non, j'explique pourquoi, c'est que le Product Owner était débutant et euh, il demandait de l'aide un peu de nombreuses personnes, ce qui fait qu'il y a plusieurs personnes qui modifiaient son produit de Backlog ce qui emmenait beaucoup de confusion… Il y avait de tout. Des personnes qui oralement disaient « hey, on peut faire ci, on pourrait faire ça », qui est logique, ce qui est très bien même et des personnes qui modifiaient le produit du Backlog, euh, en direct, ce qui est aussi possible, c'est pas interdit par Scrum, mais le problème c'est que ça a été fait de manière un petit peu euh, enfin ce n'était pas organisé ce qui fait qu'à la fin on avait un produit de Backlog qui allait un peu dans tous les sens. Il n'y a pas d'homogénéité en fait dans l'expression des besoins parce que c'était pas, le Product Owner ne faisait pas son rôle de filtrer, homogénéiser tout ça.

In the case of FOODRETAIL, the business analyst was heavily involved in the management of the Backlog. He shared this responsibility equally with the Product Owner. As for the FINANCE project, it was neither the Product Owner nor the Business Analyst, but rather the PMO, and in this case the Scrum Master, who took full responsibility of managing the Product Backlog through a tool called JIRA. Project VIDEOGAME, on the other hand, had a different person called the "closer" managing the Product Backlog. The main role of the "closer" was to make sure that the team was meeting its deadlines and estimates. To finish, project ERP1 had its own particular way of managing requirements, as the team did not have an actual Backlog. Consequently, the technical requirements were not written down in a formal way, given the "lack of time". Rather, they would be added as comments in the code for other developers to see. The business requirements were also under note format and were communicated and jotted down during Daily Scrums as the project evolved.

4.3.6 Monitoring and sharing progress

The Scrum Guide includes a section that highlights the importance of monitoring progress towards a goal and suggests that the Product Owner tracks the total remaining work at least at every Sprint Review, which means at the very least once a month (Schwaber & Sutherland, 2017). Moreover, the Guide urges making the progress transparent to all stakeholders. Of the 11 projects researched, only 5 confirmed monitoring success and sharing it with the team. These projects were: STREAMING, FINANCE, MEDICARE, AUTHENTICATION, and ERP1. In the STREAMING project, the team would look at a Burndown chart every morning before work. The MEDICARE and FINANCE projects also resorted to Burndown charts that they shared less regularly. Project ERP1, on the other hand, used Gantt charts to track the progress of their projects, while AUTHENTICATION relied on Sprint Progress graphs in the JIRA tool. The INTERNET project used both JIRA tracking tools and a Burndown chart, but restricted their visibility to the team as claimed by Andrew, developer:

> On la regarde jamais [The Burndown chart]. Je crois un petit peu le team lead.

Finally, respondents from projects FOODRETAIL, VIDEOGAME, ERP2 and SPORTRETAIL did not give account of any formal way of to track and share progress. For instance, in project ERP2, Natacha, developer, declared when asked about how the team tracks progress:

> We talked about it, haha. I don't know, to be honest. I'm not entirely sure if there's a graphical way of tracing the progress. I just know we have a Backlog and then we define what should be done in every Sprint, but I'm not sure who's keeping track of the bigger image, you know, of who is looking at how our things are done. I don't know, I am not aware.

4.4 Industry-driven practices

This section presents an overview of the practices that have emerged during interviews without being part of the initial chain of coding of the researcher. Resulting from an inductive coding analysis, these practices represent common topics that were brought up by respondents when talking about their respective projects. Accordingly, the emergent practices identified were: holding regular Grooming meetings, scaled Scrum teams, and continuous Scrums in Software-based companies.

4.4.1 Other events: Grooming and Triage

Formally, the Scrum Guide prescribes three main events when applying the Scrum framework: Sprint Planning, Sprint Review and Sprint Retrospective (Schwaber & Sutherland, 2017). The ongoing process of Product Backlog refinement is mentioned in the Product Backlog section of the Guide. It is described as “the act of adding detail, estimates, and order to items in the Product Backlog” (Schwaber & Sutherland, 2017, p.15). The guide leaves freedom to the Scrum team to decide how to conduct refinement. Although not directly asked about refinement practices, five out the 11 respondents reported having a regular refinement meeting often referred to as “Grooming”. Accordingly, the respondents from projects STREAMING, AUTHENTICATION, SPORTRETAIL, INTERNET, and MEDICARE all reported regular refinement meetings. The Product Owner from project STREAMING asserted that refinement was an essential part of their work:

> En fait le Grooming je trouve que c'est la rencontre la plus importante […] C'est super important parce que c'est là que le Product Owner présente les stories à l'équipe et qu’il va chercher toute l'information qu’il a besoin (Anna).

For projects STREAMING, AUTHENTICATION, SPORTRETAIL, and MEDICARE, Grooming meetings are conducted prior to Sprint Review meetings. The purpose of Grooming meetings, in this case, is to better prepare for the Sprint Planning meetings by better breaking down the complexity of the requirements to come. This is done through clarifying discussions and estimates made by the PO and the development team. Indeed, projects STREAMING, AUTHENTICATION, SPORTRETAIL and MEDICARE reported that Grooming meetings helped clarify and better understand the requirements to come. Dexter, Agile coach in project SPORTRETAIL, explained why he added the Grooming meeting:

> Moi je la [Cérémonie de Grooming] mets en place parce que je trouve qu’elle permet de bien travailler le besoin en amont et elle permet de réduire le temps dans la mise en place-, enfin, le temps de la cérémonie de planification. Fait que t’as une planification qui est beaucoup plus courte et tu es en planification avec une meilleure compréhension des besoins, et avec un besoin qui a été vraiment travaillé en amont.

Estimates are also a key part of Grooming meetings, as it was reflected in the statements of respondents of projects STREAMING, AUTHENTICATION, and MEDICARE. For the three projects, estimates were the result of a vote and discussion among the team members. Voting consisted in assigning story points that would represent the level of complexity and/or time needed to work on a specific story. Once all the team members voted, a discussion would be started to reach consensus. Bert, Scrum Master at AUTHENTICATION, illustrates the process with an example of Poker Planning:

> Le poker planning je peux donner un exemple, […]le PO disons il présente un story et on doit faire un bouton en différente couleur, après les gars ils lisent le user story et les use cases et les acceptances criteria et ils posent des questions pour que tout se comprend et moi je demande « alright sounds good ? » […] « are you ready to vote ? 1,2,3 ! » Et everyone chooses at the same time the card number So what's interesting, the concept is tout le monde va avoir souvent des différentes cartes bien que la plupart du temps c'est la même chose, mais souvent c'est peut-être que moi j'ai voté 3 et l'autre a voté 5. Après je dis bien « ok why did you vote 5 ? » Bien là la personne explique « ah je vois que les changements l'API et on peut avoir du refactoring à faire », et l'autre personne dis ok, je donne la chance au numéro 3 est-ce qu'il a la même idée puis il dit « ah I didn't think about the refactoring, you were right », after everyone agrees on the same point, we go « OK so 5 », then we go again or else, everyone agrees then it's a 5 and then we go with 5.

Project INTERNET had a different approach to Grooming. Indeed, the team calls it a “Triage Meeting”, conducting it twice a week instead of having regular Daily Scrums. The Triage Meeting for project INTERNET consists of organizing the elements present in the different columns of the board the team uses to keep track of user stories. The team uses the board to organize its user stories into different categories such as: Need Spec (needs additional details on specification), To Do, In Development, In Review, Tested, Merged, and Done. During their Triage meeting, the team goes over the elements that are stuck in certain categories and makes Triage notes as to why the elements did not move and what to do about it. Project AUTHENTICATION also reported conducting a Triage meeting every two weeks, specifically to inspect the bugs in the Backlog and control the quality of the elements developed. The meeting’s purpose, in this case, is to avoid any problems related to the software production. The Triage meeting is conducted just before the team’s regular Grooming meeting.

4.4.2 Scaled Scrum Teams

Whereas the Scrum Guide and most of the literature found assumes one Scrum team, the interviews conducted mirrored a different reality; that of multiple Scrum teams making up one scaled Scrum team. Although the interview guide did not directly address the number or scale of the Scrum teams involved in the projects, seven out of the 11 respondents reported having some kind of scaled organization of Scrum.

For some projects, such as SPORTRETAIL and REGISTRATION, scaling Scrum was done following a formal scaled organization of Scrum called Nexus. Created by Ken Schwaber, co-creator of Scrum, Nexus was introduced by Scrum.org as a framework to scale Scrum based on the initial structure of Scrum (Scrum.org, 2019). Nexus is therefore organized like a Scrum structure of sub-Scrums, or as Albert, respondents for project REGISTRATION, explained:

> People kept asking year after year after year, "Tell us what you mean," "How do you scale agility?" So they got tired of it and they've basically built a new framework that's called Nexus, that basically is Scrum within Scrum. It scales up to nine Scrum teams.[…]If we have 50 developers, what are we doing, you have a team of nine, and they're going to spread, so you're building multiple teams, all working on their own Sprints, they have their own retrospectives, they are all self-organized in their own ways, they have their Product Owners, and their own Scrum Masters, and the way that they work is independent, but towards the same vision, so in my role, here at [Company A], what I do is align these Product Owners working with each of teams, so the initiatives they are working on contribute to a broader initiative.

For other projects, scaling was less methodical. This was the case with both project MEDICARE and project INTERNET, who reported having different teams without specifying any formal structure. For instance, in project MEDICARE, the team split into two when the number of developers increased. Soon after, as the number of developers continued to increase, the team was again split, and three teams were formed. After a few Sprints, the development team expressed in their retrospective that having three sub-teams hindered visibility, as they all worked in silos. As a result, the project went back to having two sub-teams, although both teams have more than nine members.

A third way to scale projects was that of projects STREAMING, AUTHENTICATION, and VIDEOGAME, who broke their teams down by functions. In project STREAMING, the teams were broken down into User Experience, Artificial Intelligence and Content Generation, Client Loyalty, and Back Office, although the teams had fewer than nine members. For project AUTHENTICATION, Bert, Scrum Master, explained that development was shared between three teams: Mobile, Embedded, and Backend, and that they each had their different Sprints:

> Chaque équipe ont leurs différents Sprints, comme mobile ont leurs propres Sprints, leur propre cycle, Backend eux aussi.

4.4.3 Continuous Scrum in software companies

A particular characteristic that was found in all of the software-based companies studied was that of a continuous Scrum with no specific scope. Indeed, in companies REGISTRATION, ERP1, ERP2, INTERNET, AUTHENTICATION, and STREAMING, continuous delivery led to a continuous Scrum with no specific scope. Albert, VP product and Product Owner of Product Owners in project REGISTRATION, clearly stated, “We don’t have a finish or a scope for the end of the Sprint.”

When continuously delivering in Scrum, deadlines are more flexible and progress tracking cannot be done with tools such as Burndown charts, because there are no specific days assigned to software development initiatives. Respondents have reported that this lack of deadlines created a few tensions with customers that are not familiar with the process, whether these are external end-users (REGISTRATION) or internal customers such as the sales department (INTERNET, STREAMING). In explaining the continuous delivery in his company, Albert, VP product and Product Owner of Product Owners in project REGISTRATION, made a parallel with the popular social media giant Facebook:

> I like to explain this with common pop culture stuff, if you think about Facebook for instance, this is a product that never ended, every single day you get approximatively a thousand changes on Facebook that most of us don’t notice. […] This is basically what we got [Company A] to do for our customers, when you get the software, it’s basically a web page, just like Facebook. At [Company A], we apply changes to all our customers at once, approximately twice a day.

> Facebook is in the thousands, so it really scales with the amount of engineers that you have. So we have 20-something, maybe 29 developers right now, again we are scaling up to 42. Facebook has 8,000 engineers? And we are scaling up to 40, so that's why we do two changes a day, they do thousands of changes a day.

Unlike the other companies studied, software-based ones reported looser practices related to progress tracking, scope definition and schedule as a result of their continuous delivery in Scrum.

4.5 Participants' perceptions of success

As discussed in the success portion of the literature review chapter, project success has different facets which can be grouped in four main categories: project management success (also known as the Iron Triangle: budget, schedule, scope/quality), stakeholders' success, organizational success, and strategic success. This section will present the findings in terms of perceived success by the respondents:

4.5.1 Project Management success (Iron Triangle)

All respondents were asked specific questions about their perceptions of project management success in their respective projects. More specifically, they were asked about the budget, schedule, and quality of the projects. The answers regarding the quality portion of success unveiled different angles of viewing and assessing quality, as the criteria taken into consideration to define quality were inconsistent across respondents. The respondents from projects VIDEOGAME and FOODRETAIL considered that respecting deadlines made up for the quality of the projects. Hence, the scopes in these projects were directly tied to the schedules and deadlines they had to respect. In the case of the FOODRETAIL project, a new requirement emerged in the midst of project and was voiced by the client. Although the requirement represented a significant improvement to the system and the process it was supporting, it had to be rejected. Similarly, Fabrice, developer in the VIDEOGAME project, mentioned that in their project, the development team had to cut several functionalities to meet schedule constraints:

> Encore là, c’est du temps la qualité … Mais souvent ce qui va arriver c’est qu’on va souvent couper des features pour assurer que la qualité soit pas coupée.

On the other hand, projects SPORTRETAIL and AUTHENTICATION consider quality in terms of how many bugs the product has during its production. Accordingly, the less bugs the better. Project SPORTRETAIL also mentioned client satisfaction as a way to assess quality. When explaining their approach to quality, Bert, Scrum Master at AUTHENTICATION, explained how they carried “Triage Meetings” to ensure that bugs are kept to a minimum:

> […] Pour moi je trouve que c'est bien, parce qu'on a d'autre, on a des cérémonies qui s’appellent le bug triage, le bug triage sur les bugs, les problèmes. […] Oui ça c'est ça, c'est moi qui l'ai introduit. C'est euh, en effet la cérémonie, on le fait à chaque 2 semaines juste avant le Grooming et on regarde tous les bugs qui sont dans le Backlog et on dit « Est-ce que c'est important, pas important ? Est-ce qu'on doit le fixer bientôt ou… », etc. Puis on décide en équipe et on le met dans le prochain Sprint pour qu'on le fait. So ça c'est un peu la qualité qu'on contrôle et on peut dire qu'on contrôle la qualité. Quand j'ai commencé on avait une centaine de bug et là on est rendu à comme 10 bugs environ dans le Backlog […] C'est juste pour vraiment éliminer tous les problèmes de production.

Another perspective on quality was that of Natacha, developer in the project ERP 2, who viewed quality in terms of features and system functionalities and how she felt the product as a whole could be improved because “some parts of it could definitely be better”. In project REGISTRATION, however, quality meant conforming to the “definition of done” that was previously decided and communicated to all the teams. This ensured that consistent acceptance criteria were applied to all the stories developed. The Product Owner at project STREAMING explained that quality was guaranteed by testing the product at every stage. However, she warned that having the tasks broken down into Sprints creates deadlines for delivery that can make the developers rush to deliver the increment at the expense of the quality. In her opinion, it is the responsibility of the Product Owner in this case to determine whether parts of a specific Sprint should be put back in the following Sprint to be improved, as she explains:

> Ça dépend, ça dépend des équipes, ça dépend... Souvent un des problèmes c'est quel les gens vont rusher à la fin du Sprint c'est comme tu te mets un genre de limite invisible que dans la vraie vie, si tu avais pris une semaine de plus pour le

> faire ça aurait été encore mieux. Fait que tu sais des fois les gars me demandaient « OK pourquoi on a la limite le vendredi ? ». Parce que le Sprint finit, mais dans la vraie vie pour une compagnie on aurait pu très bien décider de retravailler. Là, rendu là c'est la responsabilité du Product Owner de dire « ça ne passe pas on écrit une story et on va le finir bien au prochain Sprint ». C'est ça le problème dans le fond. Fait que oui la qualité devrait être bonne parce qu'en effet tu teste à chaque niveau, mais cette histoire-là de deadline toujours, c'est ça faut que tu sois vraiment vigilante de ne pas accepter parfois une story et la faire passer à l'autre Sprint parce qu'ils l'ont rushé puis ça parait.

Moving on to the budget portion of project management, six out of the eleven respondents declared that a budget was not applicable to their situation, when asked about the state of their project in terms of budget. The six respondents justified not taking budget into consideration with the fact that only full-time resources, in the form of salaries, were involved in these projects. This was the case of projects ERP1, ERP2, STREAMING, REGISTRATION, FINANCE, and AUTHENTICATION. Moreover, in all these projects except FINANCE, the project carried out was also the core business of the company rather than an independent one, as they were software-based companies. This means that the projects had no specific end date, making a budget estimate rather challenging. Albert, VP in charge of the AUTHENTICATION project, explains this situation:

> And see, that's the point. Like, what is the budget to build a Facebook, for instance? There is no budget because there is no end. […] So, I don't have a budget because there is no end. But if you really want to look at it in a project way, the end date is at the end of the Sprint, I know exactly how much it's going to cost me in salary at the end of the Sprint.

As a result, budget does not seem to be a major concern to most respondents. A good illustration of that is project ERP1, in which the budget is calculated after finishing the work on different modules of the project, as Yanny, architect in the project, explains:

> Bien il n'y a pas de budget. D'où le fait qu'à la fin de chaque module du projet il y a toute une analyse pour le coût de reviens, pour le coup de vente à avoir. Mais comme je te dis c'est euh... c'est, les gens vont pas juste trainer pour trainer. Tout le monde travaille. Pour la bonne simple raison que tu ne peux pas trainer.

Similarly, in project VIDEOGAME, Fabrice, developer, considered budget and time to be the same thing, because any time added translates into costs in terms of salary. Finally, Patricia, project manager in project FOODRETAIL, focused on the time aspect of project success as a driver to budget success. She explained how although it seems expensive to outsource their development to an external team, major financial gains are made when looking at the big picture as a result of the time saved by eliminating dependencies in their internal project portfolio:

> Et encore là, c'est plus cher sur le coup, mais quand tu calcules le coût des retards des projets cumulés, puis le temps des ressources qu'il mettent à gérer les retards, puis à gérer les impacts. Tu sais admettons ma job là, pas ma job, ma patronne, ma job à temps plein c'est quasiment ça, fait qu'une, deux, trois, quatre personnes à 90000/ année juste pour gérer ça. C'est sur qu'au bout du compte, ça coute plus cher, même si le taux horaire est plus élevé, les dommages collatéraux, les frais collatéraux, c'est beaucoup plus élevé. J'ai pas de chiffres à l'appui, mais tout le monde consent à dire que…

Schedule was the aspect of project management success which respondents stressed the most. As the two previous sections on quality and budget show, most respondents link the overall project management success, the budget and sometimes even the quality to the schedule of the project. When asked about the performance of their projects in terms of schedule, five projects out eleven reported that they were doing well. These projects were MEDICARE, INTERNET, FINANCE, STREAMING, and FOODRETAIL. Patricia, project manager in the FOODRETAIL project, highlighted how opting for an Agile method with an external team helped the company, and more specifically the IT department, save time and meet deadlines:

> Oui, puis si je peux commenter, comme je t'ai dit je fais d'autres projets en waterfall, puis c'est tout à fait le contraire, les autres c'est tout le temps en retard là. […] Mais celui-là c'est on time, on a même pas eu un jour de retard pour le go live. C'est parfait, ça va vraiment bien. […] Puis dans le fond, l'impasse c'est ça, comme je te disais, dès qu'il y a un problème avec un autre projet, tu sais, t'as même pas d'autres ressources. C'est ces ressources là qui travaillent pour cinq projets. S'il y en a un des cinq qui va mal, puis que ce projet-là requiert 15% plus de leur temps, ben ça va affecter les 4 autres projets, il faut que tu ailles chercher ce temps là quelques part. On ne peut pas inventer du temps, puis on ne peut pas engager quelqu'un. Tu peux pas engager un développeur en deux semaines ou en une semaine. Si t'engage des consultants, même avec des consultants ça prend comme un mois. Fait que l'affaire à l'interne c'est que tu es vraiment dépendant de la performance du portfolio, de la performance des autres projets. Tandis qu'à l'externe, le fournisseurs peut pas te dire « eh j'ai un autre projet qui va mal, fait que le tiens je le livrerai pas à temps» Tu sais t'es un client, il peut pas te dire « excuse j'ai un autre client, j'ai un autre clients qui à de la priorité sur toi.

Respondents of projects ERP2 and REGISTRATION, stated that, just like budget, schedule did not apply to their situation. Albert, VP at REGISTRATION explains why:

> What I basically say, and it's part of, it comes from the Agile principles. One of them is build software around motivated individuals, something along these lines. From my perspective, what this basically means is don't ever try to figure out how much you're going to achieve, just assume that your team is going to do as much as they can because they're motivated about what they're building. […] I really never care about "was it only one line of code" if they learned a dozen things, if they tried new technology, if they failed and learned from it, I don't care if they wrote a single line of code. That they finish the entire feature that we assume would be finishes in months, I don't find that more impressive than writing a single line of code. Um, so they never select what they're going to do.

While this deadline-free work environment may seem convenient from a Scrum team perspective, it remains frustrating for the customers, as Albert, VP in project REGISTRATION describes:

> Most customers are not satisfied with the pace at which we are going, and I think it's fair to say that this stems from changing their mind from a project or a more traditional way of doing things towards agility cannot be done overnight without any coaching or extreme curiosity. […]

In contrast, projects ERP1, SPORTRETAIL, and VIDEOGAME reported being challenged and doing worse in terms of schedule. In project SPORTRETAIL, Dexter, Agile coach, noted that the reason behind that was poor planning. In VIDEOGAME and ERP1, however, the reason was the nature of the job itself, and not conforming to the schedule was accepted as a fatality in every project. Yanny, architect at ERP1 illustrate that in his statement:

> Non. Je peux te dire que pour les projets internes ça ne va jamais 'as expected' […] Pourquoi ? Parce qu'il y a la priorité toujours mise pour les projets clients […] C'est-à-dire qu'en terme de temps alloué aux projets ça grosso modo c'est ça. En tant que délais, de livraison, ce n'est jamais ça. Mais si, j'ai euh, si on a... Si le grand boss, grand pote à moi il arrive avec un projet client, puis il n'y pas de ressources, bien c'est « qui est sur quelque chose d'interne ? », « Bien, moi. », « Est-ce que tu peux ? » Ouais. Donc tu prends ton équipe tu leur dis, faite un shelf de votre travail et euh... c'est ça.

4.5.2 Stakeholders' Success

Within stakeholders' success, client success is the main focus of Agile methodologies and one of the fundamental reasons behind their creation (Williams & Cockburn, 2003). Nonetheless, only four respondents affirmed their customers were satisfied with the progress of the projects. These projects were ERP1, MEDICARE, FINANCE, and AUTHENTICATION. While MEDICARE, FINANCE, and AUTHENTICATION relied on direct feedback from their main customers, ERP1 reported drawing conclusions on customers' satisfaction thanks to a set of metrics from their marketing department. However, this is done after the development of the product, as Yanny, architect, explains:

> **Yanny :** Les clients externes tu veux dire ? Oui bien sûr, toutes les métriques t'as tout tout tout en marketing il y a plein de métriques pour ça.
> **Intervieweur :** Mais durant le développement du projet ?
> **Yanny :** Non à la fin. […] Pas durant c'est après.

Similarly, Natacha, developer at ERP2, could not give her impression on customer satisfaction because customers would only be able to see the final product. Indeed, customer feedback was not solicited throughout the development phase, but customers were informed about the new version and "showed enthusiasm", according to Natacha.

Projects VIDEOGAME, STREAMING, and FOODRETAIL reported varying levels of customer satisfaction. In the case of VIDEOGAME, variability was from a Sprint to the other, depending on the increment produced. For STREAMING, variability was due to the fact that many different customers were involved, namely senior management, end users, support agents, and other departments such as marketing. Because the project impacted all of these stakeholders, it was challenging to ensure that all parties were satisfied. Anna, Product Owner at STREAMING, acknowledged that sometimes she is put in a position where she has to refuse or delay clients' demands, which naturally frustrates them. The internal clients in the FOODRETAIL project, namely the Wholesales department and more specifically the VP Wholesales and Customer Relationship Manager Wholesales were satisfied with the deliverables, but the lack of flexibility in the scope of the project created some tensions, as Patricia, project manager, describes:

> Parce que comme je t’ai dit l’organisation n’est pas agile, puis nous ça revient à ce que je disais tantôt c’est un peu à la logique du contrat. Nous on a un project scope, puis c’est un très bon exemple parce que c’est arrivé, la semaine passé […] Oui on s’est rendu compte qu’il y avait quelque chose dont les utilisateurs avaient fréquemment besoin et qu’on pouvait traiter au projet, c'est des fonctionnalités pour gérer, genre admettons changer le mot de passe d’utilisateur, parce qu’en ce moment, dans le site web, pour changer ton mot de passe si t’es un client et que tu as un compte, pour changer ton mot de passe il faut que tu appelles. […] Tu peux pas le changer en ligne, ouais. […] Maintenant on s’est rendu compte « Oh mais on pourrait intégrer ça au projet » tu sais développer pour que l’utilisateur puisses changer lui même son password. […] Mais ça ne faisait pas partie du project scope! Fait qu’on [IT department] a refusé. […] Parce qu’on aurait été over budget, on aurait peut-être pas respecté notre date de fin de projet, fait qu’on à refusé. […] Fait que dans ces cas là, les utilisateurs sont un peu négligés si on veut. Mais c’est comme un contrat fait que le scope change pas. […] Ça a causé tout un tas de frustrations.

Finally, projects SPORTRETAIL and REGISTRATION reported unsatisfied customers, but for different reasons. While the customer was disappointed at the increments produced in project SPORTRETAIL, customers of REGISTRATION were fond of the product developed. However, they disliked the pace at which functionalities were developed and the lack of rigid deadlines for production and delivery. Albert, VP at REGISTRATION, illustrates that in his interview:

> So they love our product, there is no going out of it, because it does everything for them, and nothing else does it, but they don’t like the pace at which we’re going. […] “When are we going to have this feature?” “We don’t know!” “Ah, we need to know!” “That’s not true.” (laughs) We never tell them, we never lie to our customers in that sense. […] And that’s infuriating for them.

On the other hand, Product Owners came across as generally satisfied stakeholders in eight of the eleven interviews. Only project SPORTRETAIL reported a dissatisfied Product Owner due to internal conflicts between him and another Product Owner in a scaled Scrum project, in addition to conflicts between the same Product Owner and his own development team. In project STREAMING, the Product Owner said her satisfaction depended on the team members’ engagement and performance:

> Ça dépendait des personnes, parce que chaque story était assigné à des développeurs, […] fait que tu sais des fois je voyais que j'en avais un qui tirait vraiment beaucoup de la patte, qui avait de la misère à livrer ses stories […] ça dépend de l'engagement des gens, pour d'autres c'était des supers stars et ils

> arrivaient en fin de Sprint et me demandaient d'autres choses pour rentrer dans le Sprint parce qu'ils avaient fini leurs trucs.

As far as senior management and project sponsors were concerned, most of them were satisfied and supportive of their project. Indeed, eight respondents reported content senior management. One noteworthy case was that of MEDICARE, where senior management was so pleased with the way the team worked that they decided to shift the whole company to Agile. Matt, developer at the MEDICARE project, shared:

> Tu sais, oui, le boss chez MEDICARE en fait tout le monde, avec ce projet-là, on a donné une visibilité à notre compagnie pour dire « eux, ils… vous leur donnez un projet complexe, risqué, et ils vont vous le livrer ». Puis là la compagnie MEDICARE est en train de dire « on passe, toute la compagnie passe à Agile ; on est une compagnie Agile, on va travailler en Agile », puis l'exemple c'est Montréal.

It is worth mentioning that although several projects claimed that their senior management was satisfied and supportive, projects STREAMING and AUTHENTICATION both underlined the need to continuously explain and communicate the way framework works to high-level management. These talks were necessary to set the needed boundaries to work in Scrum, and to clarify what sometimes looked like "too many meetings".

> Mais je pense qu'il y a encore un-, personnellement j'ai le sentiment qu'il y a un comme un côté euh que c'est pas super clair pour eux et demain ils vont venir en effet et je pense je vais lui parler un peu, one on one juste pour vraiment expliquer. Un des feedback c'est-, que j'entends d'autres mondes c'est qu'on fait trop de meeting. Mais ce n'est pas vraiment ça. Ce n'est pas trop de meeting, tu dis « ah tu as le Grooming, tu as le bug triage, tu as le daily standup tu as le planning, tu as le Scrum of Scrum tu as plein de choses comme ça ». Mais c'est aussi très important pour avoir des buts très importants. […] (Bert)

Communication is therefore crucial to overcoming tensions and conflicts with senior management, especially when the framework is newly introduced in a non-Agile work environment.

The respondents from projects VIDEOGAME and INTERNET could not answer questions about top management satisfaction. On one hand, the respondent from project

INTERNET did not know how to define satisfaction and was not sure how to assess it. On the other hand, the respondent from project VIDEOGAME expressed how it was difficult to tell, given the large size of the project:

> **Fabrice:** Je ne lui ai pas parlé [Creative Director/Product Owner], personnellement fait que c'est difficile à dire. Comme je dis, il y avait 1 000 employés fait que c'est rare que tu as accès aux gens qui sont en haut. Mais dans les meetings, il avait l'aire de dire qu'il était satisfait de où on avançait, mais moi je suis parti avant que le projet sorte.
> **Intervieweur :** Puis est-ce que la haute direction était satisfaite ?
> **Fabrice:** Hey [Expletive]… la haute direction est en France là, je n'y ai vraiment pas accès.

In regards to the team's own satisfaction, all respondents gave account of good levels of team satisfaction, except for project SPORTRETAIL, where the team suffered internal conflicts and a lack of expertise that translated into a toxic work environment. Project FOODRETAIL could not comment of the topic as the development team was external. While project STREAMING's respondent also shared that the team was satisfied, she did mention that frustrations did occur as a result of the team being disturbed during their Scrum, which sometimes caused developers to deliver less. In addition, she also explained how the fact that the team was newly introduced to the methodology took some effort on her behalf to keep the members on track:

> Je pense qu'en grande majorité oui [the team is satisfied]. Mais il avait quand même des frustrations à chaque fois qu'on déviait le focus sur des demandes autres qui arrivaient d'en haut. C'était vraiment ça. Là ça devient une excuse pour ne pas livrer qu'est-ce qu'ils sont en train de faire parce qu'ils sont dérangés mais, tu sais, des fois c'est vrai, des fois... Combien de temps tu peux vraiment attribuer au fait qu'ils ont été dérangés ? Ça ne devrait pas être une excuse pour ne pas livrer aussi tu sais ? Fait que c'est difficile de quantifier ça quand tu fais deux choses en parallèle aussi, comme je te disais, un 20 % ou 10 % de leur temps qui était sur d'autres demandes bien, ils ne sont pas en train de clocker exactement combien d'heures. C'était difficile de les rendre tranquillement au bout de 3 mois je n'ai pas été capable mais tu sais de les rendre à vraiment bien estimer les heures dans leurs tâches. Ça ne leur tente pas de faire ça. Ça ne leur tente pas de dire exactement combien de temps ils ont passé là-dessus par c'était devient... fallait leur expliquer que ce n'était pas pour les checker eux, mais c'est vraiment pour les protéger justement et pour… fait que, tu sais, la théorie est vraiment magnifique tout est parfait là, dans la vraie vie de mon expérience c'est vraiment difficile honnêtement de bien le faire. Ça prend vraiment une équipe qui s'entend bien ensemble qui s'entraide, qui reste tard les derniers soirs du Sprint pour bien

> livrer. Ça prend, oui, ça prend du temps de s'habituer et pas tout le temps changer l'équipe et le scope (Anna).

4.5.3 Organizational and Strategic Success

Despite the fact that respondents were not asked about organizational benefits of their projects, respondents from projects MEDICARE and AUTHENTICATION did bring them up during their interviews. Bert, Scrum Master at AUTHENTICATION, described how their project enables them to compete against bigger IT companies by improving their business side, their teamwork, and their motivation. Concerning MEDICARE, Matt, developer in the project, talked about how using Scrum improved the information technology department’s visibility in the company, delivery speed, and set a new standard to follow for the organization at a global level.

As most projects were currently ongoing at the time of the interviews, respondents were not in a position to comment on their strategic success. Nonetheless, all eleven respondents agreed that their projects carried significant strategic importance in their respective companies. Projects were therefore set for strategic objectives such as market development and international expansion (ERP1, ERP2, SPORTRETAIL). Yanny, architect at project ERP1, asserted: « Oui, c'est l'idée même [strategic contribution of the project], en termes de marché, c'est parce qu'on s'ouvre sur un nouveau marché ». Likewise, Dexter, coach at SPORTRETAIL, stressed the strategic significance of the project for the company’s international development:

> Oui, très stratégique parce que euh logique de développement international qui était très, très fort et c’était leur projet de, c’était, si ce n’était pas leur projet le plus important ou l’un des projets les plus importants et le plus stratégique de l’entreprise ce qui expliquait qu’ils avaient mis euh, aussi euh, des moyens sur le fait d’embaucher des consultants extérieurs, des coaches, etc., parce qu’il fallait à tout prix que ça réussisse.

Other projects such as VIDEOGAME and INTERNET highlighted sales and increase in revenues as the main strategic goals for their projects (VIEOGAME, INTERNET). For project FOODRETAIL, the project was one of the main strategic goals for the year, as Patricia, project manager reports:

> Oui c’est un projet comme je t’ai dit qui à vraiment une importance stratégique pour eux. Puis d’un point de vue client relation, leur site était affreux fait qu’eux ils croyaient que ça leur faisait perdre des ventes. Puis à des fins d’harmonisation, c’est vraiment dans la stratégie de l’entreprise d’harmoniser tous les processus à travers le Canada, fait que tout était vraiment aligné avec la stratégie de l’entreprise. C’est comme un premiers des projets stratégiques de l’année au complet là, fait que, fait que oui il y avait beaucoup support étant donné qu’il avait cette importance là, et pour vrai j’ai d’autres projets où c’est pas le cas là.

Finally, in the case of project FINANCE, the central strategic element was security, as the team worked to avoid exposure on the company’s infrastructure to major risks and vulnerabilities.

4.5.4 Success: a subjective evaluation

The interviews conducted reflected that across and within projects, the definition and evaluation of success varied considerably. When it comes to project management success, respondents expressed different views of evaluating the same three aspects, namely budget, schedule, and quality. With that being said, schedule came across as the most important of the three aspects, being stressed by several respondents.

Concerning stakeholders’ success, levels of satisfaction were not constant among the different stakeholders of a same project. Thus, within the same project, different stakeholders would have different satisfaction levels, as it was the case in projects SPORTRETAIL, INTERNET, STREAMING, FOODRETAIL, REGISTRATION, AUTHENTICATION, and VIDEOGAME.

Finally, organizational and strategic success focused mainly on competitive advantage and profit generation. Only project MEDICARE expressed internal organizational benefits linked to using the Scrum framework such as improving delivery speed, department visibility, and organizational work processes.

4.5.5 Patterns in Scrum guidelines application and success

In an attempt to uncover potential relationships between Agile Scrum guidelines application in practice and project success, matrix tables mapping the guidelines with the most significant variability in terms of adoption and project success were developed. Accordingly, 14 matrix tables were produced. This section presents the four patterns observed in the study.

4.6 Scrum deviations' link to project success

As explained in the methodology chapter, 14 matrix tables were developed mapping seven Scrum guidelines to two aspects of project success: project management success (Iron Triangle) and customer satisfaction.

Among the 14 tables, four displayed potentially interesting patterns emerged:

- A positive relationship between the development team's control over items to include in a Sprint and project management success, more specifically the aspect of quality
- A positive relationship between the Product Owner's sole responsibility for the Product Backlog and project management success
- A Positive relationship between conducting retrospective meetings with the objective of inspection and improvement and both project management success and customer success
- A positive relationship between the team creating a Sprint Backlog to forecast the work needed to achieve the Sprint goal and project management success.

4.6.1 The development team’s control over Sprint workload and project management success

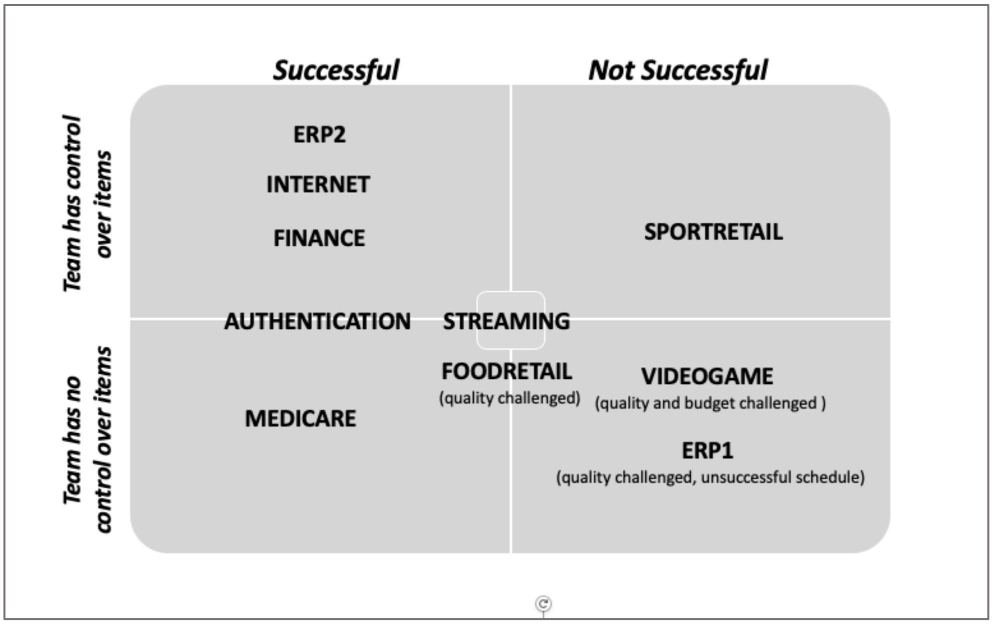

Successful
Not Successful
Team has control over items
Team has no control over items
ERP2
INTERNET
FINANCE
SPORTRETAIL
AUTHENTICATION
STREAMING
FOODRETAIL
(quality challenged)
VIDEOGAME
(quality and budget challenged)
MEDICARE
ERP1
(quality challenged, unsuccessful schedule)

Figure 6 The development team’s control over items to include in a Sprint vs project management success

The above matrix representation suggests there may be a relationship between the development team’s control over how many items from the Backlog to work on in a Sprint and project management success. More specifically, it suggests that the quality aspect of project management success may be affected by the team’s power to independently make that decision. This relationship can be explained by the pressure the development team experiences when an external party such as the Product Owner or the Scrum Master decides to include additional items in a Sprint. Matt, developer at MEDICARE, touched on the tension between his development team and his Product Owner over the number of elements to include in a Sprint. He explained that because the Product Owner represents the customer/client side in the project, he thus pushes for features to be developed faster. While this could motivate the team to be more effective and productive, it can also contribute to technical debt and employee stress, which in turns impacts the quality of the increment. Matt explained that while the pressure from the Product Owner helped them finish the essential parts that made up an increment, a lot of work surrounding that increment such as specifications adjustment and testing was not done appropriately resulting in a build-up of technical debt. He referred to this situation as “la pire catastrophe”.

Similarly, Anna acknowledged that, as a Product Owner in project STREAMING, she would push her team to deliver more, although she was not supposed to:

> L'équipe de développement est la seule à sélectionner le nombre d'éléments du Product Backlog à inclure dans un Sprint. Oui, je poussais un peu, même si je ne suis pas supposée, mais mon Scrum [Master] était là pour me calmer.

In project AUTHENTICATION, it was the Scrum Master, Bert, who got involved with the team in their selection of items to tackle in the Sprint. This being said, Bert only got involved at the validation stage. This means that he validated what has already been chosen by the team based on his estimates of their work capacity and the work required for the development of the items (voted and agreed upon in the Grooming ceremony).

In project VIDEOGAME, over a thousand people were involved, which made the project management very hierarchical. Accordingly, Fabrice, developer, reported receiving the items to work on in the Sprint by email. He underlined how the workload could fluctuate from one Sprint to the other:

> La plupart du temps ils étaient bons pour estimer mais ça fait que les fois où ils n’estimaient pas bien, soit tu étais pogné pour être super stressé parce que t’avais plein de trucs à faire ou bien tu te tournais les pouces, [ce] qui n’était pas vraiment plus cool.

The Scrum Guide recommends that development teams decide the work estimates in Sprints autonomously. However, the development team can lack control over Sprint work estimates due to authoritative Product Owners/Project Managers/Scrum Masters and hierarchical work practices. The lack of control can result in a lack of commitment from the team and inaccurate estimates, which may translate into poor performance and a defective schedule (Eloranta & al., 2016).

4.6.2 Product Owner’s responsibility for the Product Backlog and project management success

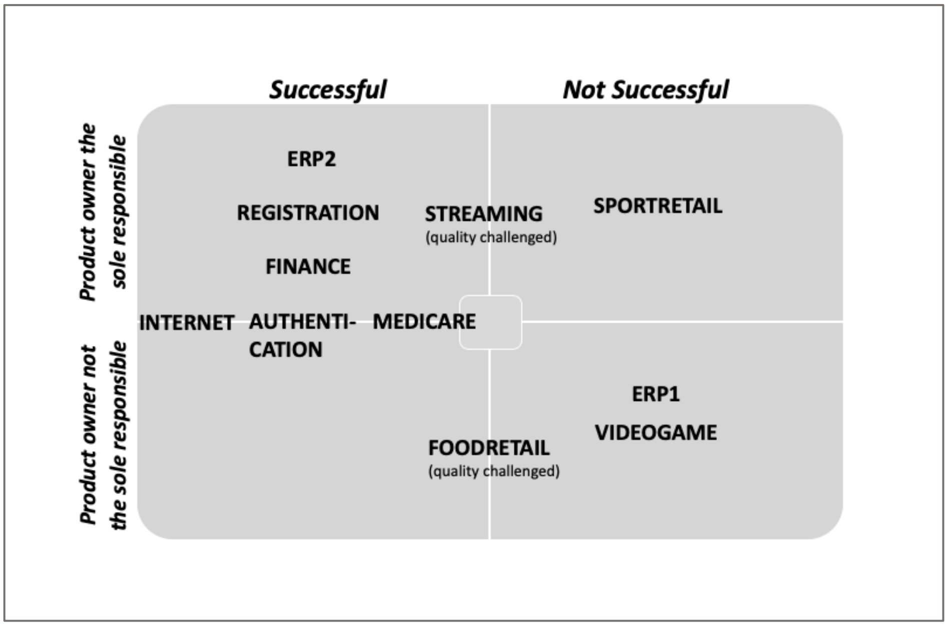


Figure 7 Product Owner’s sole responsibility for the Product Backlog vs project management success

While the Scrum Guide invites discussion and input from the entire Scrum team about the Product Backlog’s items, it does emphasize the responsibility of the Product Owner for the Product Backlog (Schwaber & Sutherland, 2017). The interviews conducted revealed that Product Owners do not always take on full responsibility for the Product Backlog, with only six respondents confirming that their Product Owners did.

The reason Product Owners would not be responsible for managing the Backlog was specific to each project and stem, for the most part, from the ways the teams chose to work together. For instance, in project ERP1, the team did not have a technical Backlog. Instead, they relied on Daily Scrum notes and comments in the code to keep track of technical details. Consequently, each team member was responsible for his/her own work. Contrastingly, in project FINANCE, it was Eamon, PMO, who managed the Backlog for all the different teams under his supervision. In Eamon's view, this work structure benefited the department better as teams were made of subject matter experts that prefered focusing on the technical aspects of work rather than the managerial ones. In project FOODRETAIL, having an external development team required the cooperation on the Backlog of an internal business analyst. Project VIDEOGAME was characterized by its significant size, which made its work structure hierarchical. As a result, the respondent from project VIDEOGAME could not confirm who was involved in the management of Product Backlog with certainty. However, he did assert it was not only left to the Product Owner alone.

For projects FOODRETAIL, FINANCE, ERP1, and VIDEOGAME, the shared responsibility over the Product Backlog items came across as a choice made by the team given the circumstances of their projects. Moreover, the respondents from these projects did not link the shared responsibility for the Product Backlog to any negative aspects in the project.Contrastingly, the respondent from project SPORTRETAIL had a different narrative. In the case of project SPORTRETAIL, the Product Owner was a novice to his work responsibilities. To compensate for his lack of experience, he would ask other team members to help with writing in the Product Backlog. In this case, the different contributions resulted in general confusion as the Product Backlog lost its homogeneity. Dexter, Agile coach at project SPORTRETAIL, explained why, in this case, shared responsibility over the Product was harmful to the project:

> Il y avait de tout. Des personnes qui oralement disaient « hey, on peut faire ci, on pourrait faire ça », qui est logique, ce qui est très bien même et des personnes qui modifiaient le produit du Backlog en direct, ce qui est aussi possible, c'est pas interdit par Scrum, mais le problème c'est que ça a été fait de manière un petit peu, enfin ce n'était pas organisé, ce qui fait qu'à la fin on avait un produit de Backlog qui allait un peu dans tous les sens. Il n'y a pas d'homogénéité en fait dans l'expression des besoins parce que c'était pas- le Product Owner ne faisait pas son rôle de filtrer, homogénéiser tout ça.

Several teams have decided to share the responsibility of the Product Backlog differently than prescribed by the Scrum guide. While this practice was reported to be convenient for some teams, it did create confusion in the Product Backlog content for others. An disorganized Product Backlog can hinder the team’s visibility and create complications in late stages of the project, which may explain the pattern observed (Eloranta & al., 2016).

4.6.3 Conducting retrospective meetings and both project management success and customer success

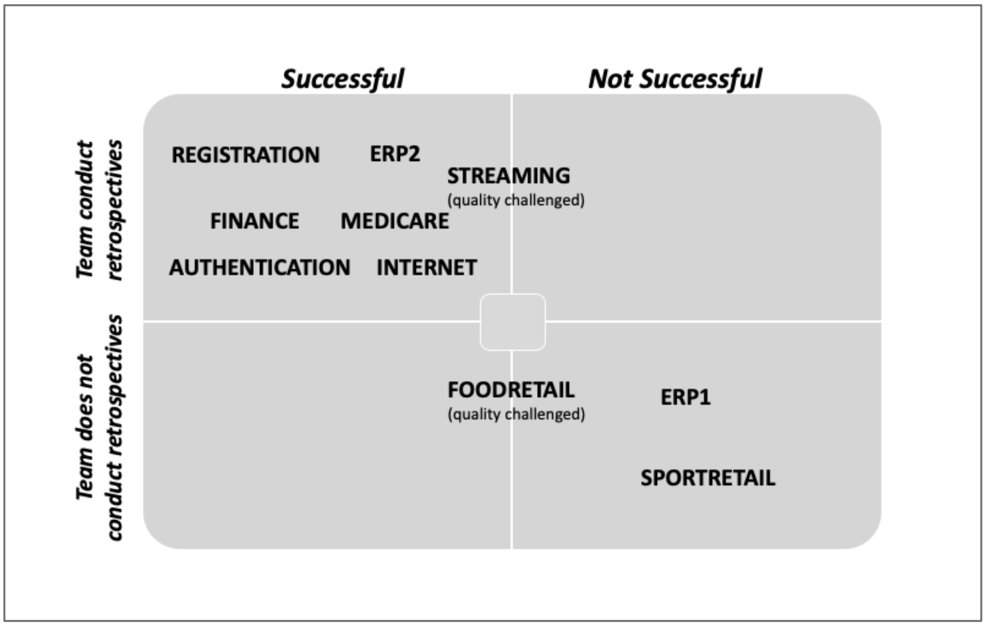


Figure 8 Conducting retrospective meetings with the objective of inspection and improvement vs project management success

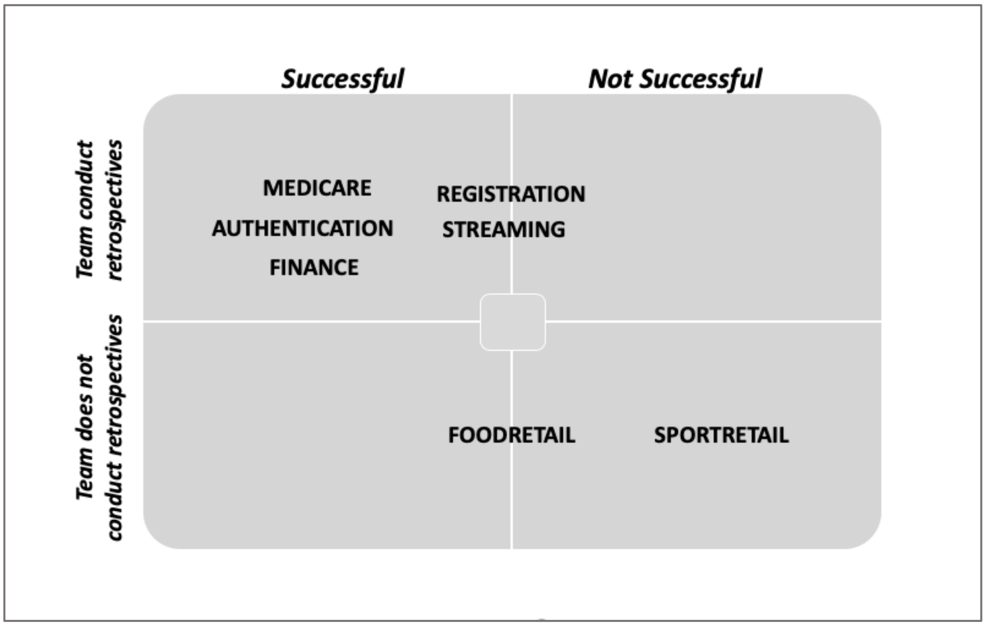


Figure 9 Conducting retrospective meetings with the objective of inspection and improvement vs customer success

Sprint retrospectives are meetings dedicated to improving the way the team works as a whole. Unlike Sprint Reviews, which focus on the increment being developed, Sprint Retrospectives focus on how the team members work together through the development process and how to improve it (Schwaber & Sutherland, 2017). With that being said, only six out of the eleven respondents confirmed having a Sprint retrospective meeting that actually serves to inspect the team and addresses work process improvement. Among the respondents who affirmed carrying out Sprint retrospectives with an objective of continuous improvements was Anna, Product Owner in project STREAMING, who expressed the importance of such meetings:

> Oui à 100 % […] Ça, ça se fait honnêtement super mieux un vendredi après-midi avec des chips. C'est une bonne place pour relaxer et juste se dire les vraies choses. Puis, faut, c'est super important aussi […] Il y a plein d'exercices qui peuvent être faits par la Scrum Master, ça l'air niaiseux, mais ça vaut quand même la peine, mais ce qui est plus important de la rétro c'est d'avoir des actions items clairs avec des personnes assignées et que pour le prochain rétro on soit capable de démontrer que ça a avancé. Si quelque chose frustre les développeurs et c'est quelque chose qui peut être réglé par X personnes. Souvent c'est le Scrum Master qui va en prendre la responsabilité mais ça peut être le PO aussi qui n'a pas fait quelque chose de comme il faut. Ça peut être des processus de la compagnie au

> complet qui ne seront pas réglé avant la prochaine rétro, ça se peut, mais il faut vraiment que chaque item soit assigné à quelqu'un et que cette personne-là démontre qu'il a de l'avancement dans cette chose-là après.

On the other hand, while projects FOODRETAIL and SPORTRETAIL did carry out retrospective meetings, those did not focus on improving work processes. For instance, Sprint retrospective meetings in project SPORTRETAIL involved mainly team members airing grievances and expressing frustrations. This resulted in a waste of time and higher levels of frustration within the team. In line with Anna’s view of Sprint Retrospective meetings, Dexter, Agile coach in project SPORTRETAIL, described why the lack of focus on improvement was highly problematic in his team:

> Pour moi c’était extrêmement négatif. […] Certes ça libère la parole, ça c’est chouette. […] Mais vous n’en faites rien. […] Je suis sûr que je vais voir dans la prochaine itération, je reviens dans 3 semaines, on va dire exactement la même chose plus des nouvelles choses qui arrivent. […] Comme vous n’avez jamais fait de réflexion en termes de solution, vous allez vous décourager et vous allez vous dire « on a une montagne de problèmes, ça ne sert à rien’ ». […] Il faut vraiment éviter ça. Il faut prioriser la mise en place d’actions d’amélioration et c’est pas tout de se dire « on peut mettre en place ces actions », il faut faire ce suivi d’actions. Tous les jours, « où est-ce qu’on en est par rapport à ces actions ? Est-ce qu’elles ont été mises en place ? Est-ce que ça n’a pas été mis en place ? Est-ce que, si ça a été mis en place, est-ce que ça a emmené ce qu’on souhaitait ? Si ça n’a pas emmené ce qu’on souhaitait, est-ce qu’il faut mettre une autre action ?

Yanny, the respondent from project ERP1, reported not conducting Sprint retrospective meetings. In the viewpoint of the senior software architect and firm associate, Sprint retrospective meetings can be a waste of time. Dexter, Agile coach at SPORTRETAIL, expressed a contrasting opinion, asserting that agility’s purpose in delivery, communication, reflecting, and improvement. Sprint retrospectives, according to Dexter, are what allow for this cycle to be:

> C’est pour moi l’une des raisons pour-, ce qui expliquait que l’équipe n’avançait pas c’était le, elle n’inspectait pas ce qu’elle faisait. Elle ne réfléchissait pas à ce qu’elle faisait et elle ne s’améliorait pas. Donc c’est impossible parce que quand on essaie de résumé l’Agilité, si tu veux résumé l’Agilité en 4 verbes, qu’est-ce que c’est ? Bon c’est de livrer, quand tu livres un travail mais une fois qu’il est livré qu’est-ce que tu dois faire ? Tu dois communiquer sur cette livraison, tu dois réfléchir à ce qui s’est fait, à ce que tu as livré et cette réflexion doit permettre l’amélioration. Et donc la communication, la réflexion et

> l'amélioration ils ne le faisaient pas. Ils faisaient que livrer et en plus ça ne se passait pas bien. […] Et c'est la rétrospective qui permet de travailler sur ces 3 verbes.

Skipping retrospectives or holding ones that are not centered around improvement may therefore result in higher levels of frustration within the team. Moreover, it may also hinder the team's ability to communicate, reflect, and progress in their work. This can lead to the team stagnating in terms of efficiency and productivity, which may explain the pattern identified (Eloranta & al., 2016).

4.6.4 Sprint Backlogs for work forecasting and project management success

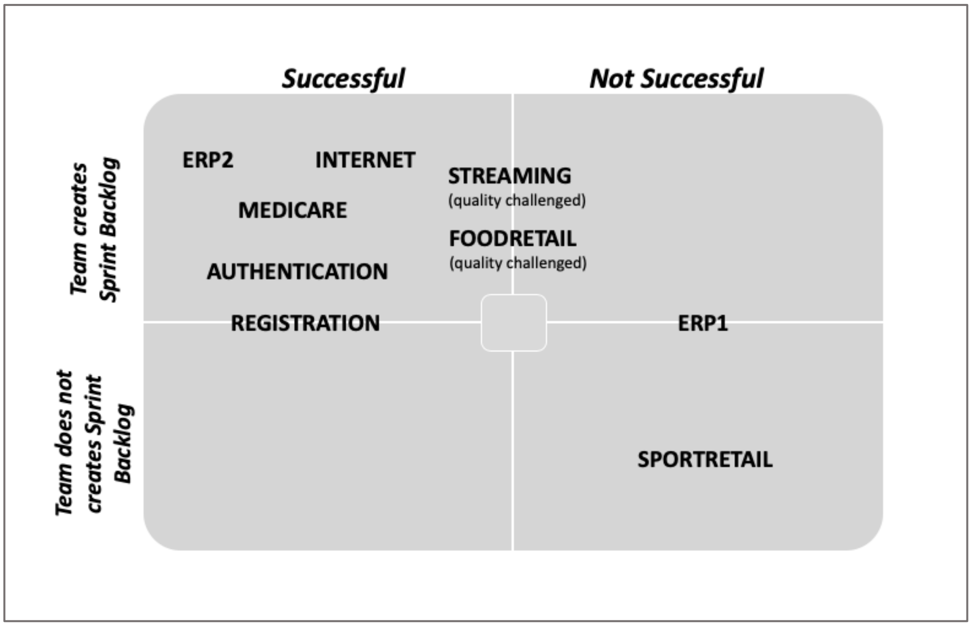


Figure 10 Team creating a Sprint Backlog to forecast the work needed to achieve the Sprint goal vs project management success

A Sprint Backlog is defined as the Product Backlog elements picked for a specific Sprint, as well as the development team's plan to develop them (Schwaber & Sutherland, 2017). For projects FINANCE and ERP1, Sprint Backlogs were not applicable because of the team's organization and nature of work. Indeed, in project FINANCE, the PMO was in charge of breaking the Backlog elements down into sub-tasks that were specifically assigned to subject matter experts. In project ERP1, on the other hand, the team did not use a formal Backlog to begin with. Consequently, each developer was free to develop, if they wished, their own version of a Sprint Backlog specific to their work. Even though he was himself a software developer involved in project VIDEOGAME, Fabrice did not know who was in charge of the Sprint Backlogs. As reported earlier, Fabrice would get specific elements to deliver in the Sprint by email.

Although working with an external development team, the project manager at project FOODRETAIL reported using Sprint Backlogs thanks to an online tool called Teamwork. The development teams of projects ERP2, MEDICARE, REGISTRATION, SPORTRETAIL and STREAMING also produced Sprint Backlogs. With that being said, two distinctions characterized project REGISTRATION and project SPORTRETAIL. For project SPORTRETAIL, a Sprint Backlog was developed without a specific goal, while for project REGISTRATION a Sprint Backlog was produced without a specific scope. Albert, VP Product at project REGISTRATION, explains that not having a scope was a purposeful choice:

> […] The outcome of estimates and the selection of Sprint planning is the Sprint Backlog. The Sprint Backlog will also include the continuous improvement plan that comes out of to the retrospective any chores, any definitions of what done items, so I would say that there is a Sprint Backlog that is created, but it doesn't have an end […] I was explaining earlier, every Backlog item from the top, or most, were groomed, inspected, plan is done, and we'll add a definition of done items in there. We'll add outcomes of the retrospective, so we also intend to work in a different way, it'll involve this and this and this. […] But it needs to have a scope, and ours doesn't. It's a sliding window of items that expand our Backlog as we go, or stick to a small window if we are not going fast enough.

This pattern can be explained by the fact that Sprint Backlogs constitute the team's immediate plan of what and how to deliver in a specific Sprint period. Omitting them or restraining them to a shortened version of the Product Backlog with no vision of the Sprint goal within the project could therefore make the development team's work harder. Indeed, Sprint Backlogs are meant to help the teams forecast their work and keep track of their progress in the Backlog (Schwaber & Sutherland, 2017). Producing a Sprint Backlog also provides the opportunity for the team to discuss the project’s scope with their Product Owner, who represents the customer’s viewpoint, and receive any industry or technical advice (Schwaber & Sutherland, 2017). Moreover, the additional level of detail of Sprint Backlogs breaks down the complexity of the tasks and provides greater visibility to the team as a whole. Sprint Backlogs are also the only way provided by the Scrum Guide to track Sprint progress, by tracking the remaining work in their Daily Scrums (Schwaber & Sutherland, 2017). Because Sprint Backlogs are meant to be the team’s operational plan to meet schedule, budget and performance constraints, omitting them may explain the pattern observed.

CHAPTER 5 : DISCUSSION AND CONCLUSION

This last chapter provides a summary of the main findings of this research, as well as the discussions that stem from its literature review, methodology, and analysis. Accordingly, this section begins with a reminder of the research objectives and methodology adopted. Subsequently, this chapter outlines the theoretical and practical implications of this research. Lastly, an overview of the research limitations and opportunities for future research are presented.

5.1 Reminder of research objectives and methodology

This research aims to better understand the application in industry of one of the most revolutionary frameworks in software project management. By answering the two research questions- “Is there a gap between theory and practice in Agile Scrum application?” and, “How are variations of Scrum in industry affecting project success?”- this study attempts to bring the reality of Agile Scrum in industry to academic research. Moreover, the aim of this research is also to optimize the use of Agile Scrum methodology by identifying which adjustments could be harmful to a project’s outcome. With this understanding, the methodology can be better adjusted in its application while maximizing chances for project success. Finally, this research provides insight for practitioners on how tailoring the methodology may impact their projects. The overall ambition is to leverage the full potential of the Agile Scrum framework.

To reach these objectives, a qualitative exploratory approach was undertaken. The first step in the research process consisted of an in-depth literature review to better understand the Agile Scrum framework. The literature review also explored the concept of project success and its evolution through time. The literature review helped shape an interview guide and a grid comprising the guidelines of the Agile Scrum framework that were later used in interviewing eleven Scrum team members. The interviews consisted of a comprehensive evaluation of the application of Agile Scrum in projects, as well as questions about the respondents’ perceptions of the different aspects of success in the projects. Furthermore, respondents were invited to share their impressions on the effect of Scrum tailored practices on their personal satisfaction and on project success. These interviews were later transcribed and thoroughly coded for analysis. The analysis unveiled that while a few guidelines were systematically followed, others were rarely followed consistently. Further, the analysis also uncovered emergent guidelines in the Scrum teams such as formal Grooming meetings and different scaled organizations of Scrum. Finally, the mapping of Scrum method deviations to project success dimensions was used to identify potential relationships between Scrum guidelines deviations and project success.

5.2 Major results

5.2.1 Variability in Agile Scrum guidelines application

The results of this research have served to answer the research question, “Is there a gap between theory and practice of Agile Scrum in industry?” The results have unveiled that while some guidelines of the framework are systematically followed, others present more variability in their adoption. On that account, four guidelines have been found to be consistently followed across software projects:

- Having one Product owner that is a person and not a committee;
- Having a Scrum Master in charge of enforcing the Scrum process throughout the project;
- Conducting back-to-back Sprints of four weeks or less;
- Conducting Sprint Reviews at the end of each Sprint.

On the other hand, six guidelines have shown lower rates of adoption and adherence by the teams, namely:

- Giving the development team control over how many items to include in a Sprint;
- Protecting the team from distractions during the Sprint;
- Keeping Daily Sprints internal to the development team;
- Using the Product Backlog as the only source of requirements;
- Having the Product Owner be the only person responsible for managing the Product Backlog;

- Monitoring and sharing progress.

Importantly, the results of this research also unveiled that agile Scrum guidelines adoption in practice is a gray area, as most teams fell in between a Yes and a No when reporting on their adoption practices. Actually, most teams had tailored the practices to their contextual factors.

Furthermore, a salient point uncovered by this research is that the direct application of guidelines does not necessarily increase the benefits of the practice. Rather, it is the understanding of the *purpose* behind the guidelines and how they contribute to Agile principles that make their adoption relevant in the project. For instance, most teams reported conducting retrospective meetings. However, not all of them conducted them with the objective of inspecting the team and working on a plan of improvement. The teams who did not work on that aspect exhibited negative effects on both customer success and project management success (Iron Triangle). It is therefore important for teams to understand the reasons behind Agile Scrum guidelines rather than simply adopting them without understanding. These findings lend support to the perspective of Kiv and colleagues (2017; 2018), according to which Agile principles contributes to Agile value and Agile practices are applied to achieve the principles. Agile practices are therefore a means to help development teams reach Agile value and principles (Kiv et al., 2017; 2018). Consequently, Kiv and colleagues (2018) stress that development teams should acquire a deep knowledge of the Agile Manifesto in order to make better tailoring decisions and maximize Agile benefits. A deep understanding of the Scrum method principles could lead to a better implementation and tailoring of practices.

5.2.2 Emergent industry-driven practices

The three main data-driven practices uncovered by inductive coding concerned the scaled organizations of the Scrum teams, the standardization of Grooming meetings and the Continuous Scrum in software-based companies. Although most of the literature assumes one Scrum team, our interviews unveiled that the reality of Scrum in practice often involves multiple scaled Scrum teams. Scaled teams structured themselves mainly by scaling the Agile Scrum framework itself. This scaled Agile Scrum implies scaling all the already existing roles and practices of the framework. Accordingly, the teams would have a Product Owner of Product Owners, a Scrum Master of Scrum Masters, a scaled global planning, review, and retrospective. In certain organizations, however, different Scrum teams were created according to software features or functions. This resulted in separate teams with separate Sprints and different cycles working on the same software product. This type of scaled organization of Scrum seems to go against one of the core principles of Agile, that is, continuous cooperation throughout a project, as it contributes to working in functional silos (Beck et al., 2001).

Moreover, although not an official Scrum ceremony like planning, reviews, and retrospectives, the Grooming meetings are an emergent practice that provides great value to the team, as they have been reported to be a critical phase of the Scrum process. Grooming meetings were therefore conscientiously followed by most teams, but for different proposes. While some teams carried out Grooming meetings to break down requirements' complexity in preparation for the Sprint Planning, other teams used Grooming meetings to generate estimates, fix bugs, or discuss item progression.

Finally, in all the software-based companies studied, Scrums were continuous, just like the project's delivery. The Product Backlog would change based on the company's environment, goals, customer feedback, as well as any arising technical problems with the current version of the software. This resulted in Scrum projects with no end date, scope, or budget. This lack of milestones and structure contributed to tensions with both internal and external customers.

5.2.3 Success as a subjective element

The literature review section presented the different aspects of project success existing in the scientific literature. These aspects can be classified into four categories: project management success, stakeholders' success, direct organizational success, and strategic success.

The results of this study evidenced that when it comes to project management success (Iron Triangle), the most significant element put forward by respondents was schedule. This result contradicts the claim of Tukel and Rom (2001) that quality is the most important aspect to project management practitioners.

Another key result reflected in this investigation is the subjectivity in the evaluation of project success, with different respondents stressing different criteria in their own evaluation of project success. Accordingly, these findings corroborate that success was a subjective element to stakeholders, as different respondents emphasized different elements of success concerning the process, the product developed, and the organizational context. These findings align with the findings of Davis (2018) and McLeod and colleagues (2012), according to which project success varies across stakeholder groups and phases of the project. The criteria used to evaluate a project's success are therefore subjective to each stakeholder's perspective in addition to being context-dependent (McLeod et al., 2012).

5.2.4 Scrum deviations and project success

This study also aimed to link the application of the Scrum Framework Guidelines to project success. In doing so, four patterns were identified based on a comparison between the application of the guidelines and the two main aspects of success, namely project management success and customer satisfaction:

- A positive relationship between the development team's control over items to include in a Sprint and project management success, more specifically the aspect of quality.
- A positive relationship between the Product Owner's sole responsibility for the Product Backlog and project management success.

- A positive relationship between conducting retrospective meetings with the objective of inspection and improvement and both project management success and customer success.
- A positive relationship between the team creating a Sprint Backlog to forecast the work needed to achieve the Sprint goal and project management success.

5.3 Research contributions

5.3.1 Contributions to knowledge

This study contributes to the current scientific literature on tailored Agile Scrum in industry. Indeed, this research provides insight into which guidelines of the framework are followed by Agile Scrum teams in industry and which are followed less often. Moreover, this research unveils other realities of Agile Scrum that stem from practice adoption of the framework, such as informal organizations of scaled teams, continuous Scrums in software-based companies, and official Grooming meetings. This work extends that of Eloranta & colleagues (2016), who attempted to identify empirically potential harmful uses of the main Scrum and software engineering concepts. Their investigation focused on: Sprint, testing, specifications, Product Owner, Product Backlog, estimates, Burndown charts, team disruptions, and team organization. This research adds a layer of depth by investigating the different guidelines relating to each one of the concepts studied by Eloranta & colleagues (2016), as well as other key concepts of the Scrum framework such as the role of the Scrum Master and the different Scrum ceremonies.

In accordance with the work of Diebold and colleagues (2015), our research confirmed that no team was in perfect conformity with the Scrum Guide. Aligning with the same study, the results showed that Daily Scrums are carried out regularly in most teams, Sprints are generally time-boxed to four weeks, and that the roles of Product Owner and Scrum Master are often adapted. In contrast with the study of Diebold and colleagues (2015), however, retrospective meetings came across as a common practice in most of teams investigated whereas Sprint Backlog adoption came across as a highly variable guideline.

The second portion of this research involved establishing a link between the deviations of Agile Scrum guidelines in industry and project success through the identification of patterns between the two. This is considered the first empirical study to link between project success and deviations of Agile Scrum in industry. In their study, Eloranta and colleagues (2016) presented 14 potentially harmful practices labelled as anti-patterns of Scrums that resulted from deviations of the framework in practice. The 14 anti-patterns focused on the Sprints, the Product Owner's role and Product Backlog, and the Team. In their study, Eloranta and colleagues (2016) considered a practice an anti-pattern if it was a deviation, was observed at least three times (in 18 teams), and had unfavorable consequences expressed either by the interviewee or by the literature. The 14 anti-patterns were not empirically verified. Interestingly, three of the four patterns discovered in this study were also anti-patterns in the study of Eloranta and colleagues (2016). These concerned the Product Owner's responsibility over managing the Backlog, the team's ability to estimate its work autonomously, and carrying out retrospective to inspect and enhance work processes. Although investigated as a potential pattern in this study, team disruption was not found to be linked to success, unlike the anti-pattern identified in the study of Eloranta and colleagues (2016). Similarly, long or non-existent feedback loops from stakeholders came across as the reality in most teams, and did not seem to interfere with success, unlike the findings of Eloranta and colleagues (2016). This research contributes by addressing a gap in the literature by empirically linking Scrum practice adoption to project success. It also contributes by offering a detailed view of the reality of Agile Scrum adoption in industry.

Furthermore, the results of this study support the perspective-based framework for evaluating success by McLeod and colleagues (2012), as project success evaluation varied significantly across respondents. Accordingly, there can be an advantage in reconciling the different project success views from different stakeholders, in accordance with suggestions from Davis (2018), in order to achieve a greater understanding of project success.

5.3.2 Contributions to practice

Given the practical topic of this research, as well as its empirical basis, it also has implications for practice. As stated in the literature review section, Scrum relies on the three pillars of empiricism: transparency, inspection, and adaptation. This research suggests that not all modifications to the Scrum guidelines are aligned with these pillars. First, ad-hoc scaling of Scrum teams can recreate hierarchical work structures and product function silos, hindering the Scrum process visibility and going against the transparency pillar of Scrum. Moreover, failing to include external stakeholders in Sprint reviews impedes the collection of feedback in order to inspect and adapt the product developed. This modification therefore contradicts these fundamental pillars. Finally, in the absence of retrospectives focused on process improvement, the development teams fail to inspect and adapt their work processes, contradicting again the Scrum pillars. Practitioners should therefore be wary that these deviations may clash with the fundamental values of Scrum and impact project outcomes.

Another significant contribution to project management practitioners are the patterns identified that link deviations of the Agile Scrum framework guidelines to key aspects of project success. Although a perfect Agile Scrum application seems idealistic, these patterns invite practitioners to reflect on the meaning and the effect of their deviations from the framework. With this knowledge, practitioners can make better informed decisions when tailoring their own projects.

Furthermore, as previously mentioned in the major results section, this research unveils that it is not the literal and superficial guidelines adoption that ensure the Agile value for which the framework was initially implemented. Accordingly, ensuring that the project has all the roles, practices, and artifacts of an Agile Scrum framework does not guarantee an optimal adoption or Agile benefits. Rather, it is knowledge and understanding of the objectives and principles behind framework components that contributes to achieving agility. This research therefore invites practitioners to educate their teams on what the roles, practices, and artifacts mean and how they contribute to having the most advantageous development environment.

The results of this study highlight that success, in addition to having multiple facets, is a subjective element that varies according to stakeholders and context. This research brings a new perspective of success for management to contemplate. Indeed, the results of this research invite a new outlook on project success and a more flexible and integrative evaluation of the latter through the consideration of different stakeholders' input.

5.4 Limitations and future research

Despite the rigorous methodology adopted in this research, there are some limitations. The first limitation identified is a common challenge and source of confusion in qualitative research (Marshall, 1996), namely the sample size and type used. Indeed, given the time and budget constraints in addition to the significant workload associated with qualitative inquiry, the sample was restricted to 11 participants. Given the smaller size of this research, the findings are less generalizable. This leaves room for future research to expand the sample size and carry out similar investigations on a bigger scale in order to confirm (or refute) the findings and patterns identified.

Another limit of this research, also regarding the sample used, is the fact that only one respondent was interviewed for each project. It would thus enrich the research and its results to interview different people from the same project. Moreover, having different perspectives would also draw a more complete picture of the different aspects investigated in the project. This is especially true of the perception of project success, which was based on the respondent's view. Another potential solution would be to assess project success based on objective measures, such as the use of performance indicators, rather than respondents' perceptions.

Another limitation is that respondents had different roles. Because the roles of the respondents may have affected their opinions on the project, the points of view regarding a given project could vary according to the role of the respondent. For instance, an Agile coach may be inclined to be more critical of Agile Scrum practices and project success, especially when hired for the purpose of improving Agile Scrum adoption. Inversely, a Scrum Master may be more likely to positively convey the practices adopted, as it is their responsibility to ensure the Scrum process is followed appropriately. Having multiple respondents from different roles on each project would therefore draw a more accurate picture of the project's reality, lessening bias and delivering more valid results.

As Agile method tailoring is a relatively new area of research, several opportunities for future research were identified. First, it would be interesting to explore the Scrum guidelines that presented the most variability. Future research can look at how these guidelines are transformed under different project or organizational settings. This could in turn open several other research tracks on improving Agile adoption in practice. For example, while not directly examined in the present study some differences emerged between Scrum teams serving internal versus external clients, and these differences may influence adoption of Scrum framework guidelines and project success. Indeed, the client and provider contract that resulted from a Scrum team for an external client appeared to affect the autonomy of the development team which is "obligated" to adhere to the clients' requests. Furthermore, other roles were needed as a bridge between the Scrum development team and the external client. Future studies should examine how external versus internal client could drive differences in the Agile Scrum application and the resulting impact on project success.

Moreover, this research identified four patterns between Agile Scrum guidelines adoption and project success. These patterns ought to be investigated more rigorously to establish any potential relationships between Agile Scrum guidelines adoption and project success. We believe this could be a promising area of research to explore.

Finally, it is apparent that research focuses predominantly on Scrum as one entity of one Scrum unit, meaning one Scrum team. As most of the Scrum teams observed were scaled, it would be relevant for future research to target scaled Scrums. Because scaled Scrums come with their additional layers of roles and practices, future work should consider a scaled Scrum team in its entirety to further analyze how these added elements interfere with the basic guidelines of Scrum. Examining scaled Scrum teams could therefore bring academic research even closer to the reality of Scrum in practice.

5.5 Conclusion

This research investigates the tailoring in practice of the most used Agile method in the software development industry. Following a literature review, an exploratory approach was undertaken using semi-structured interviews to assess the adoption of the method in practice.

The findings uncovered that all the projects studied tailor the guidelines to their organizational context, with a few guidelines being followed systematically and others less consistently. Scaled organizations of Scrum teams, formal Grooming meetings, and continuous delivery in Scrum emerged as guidelines implemented in practice. Moreover, this study provides considerable insight into potential relationships between Scrum guidelines and project success. One of the most striking observations that emerged from our analysis is that Agile adoption goes beyond the simple implementation of methods guidelines. Rather, it is the understanding of the purpose of these guidelines and how they contribute to Agile principles that make their adoption relevant in the project.

The results of this study are promising but need to be validated by a larger sample size drawing on information from multiple roles per project. Further studies are needed to examine how the understanding of the role of Agile practices impacts their adoption. Moreover, further work needs to be done to validate the patterns identified between specific guidelines and project success dimensions.

APPENDICES

Appendix A : Interview Guide

Part I: Interviewee

- Age
- What is your educational background?
- What is your work history?

Part II: Organization

- What is the size of the organization?
- How long has the company been using agile? [Experience with Agile in years]
- Which departments use Agile-Scrum?

Part III: Project

- What is the project about?
- When did the project start?
- When is the project expected to be completed?
- Who is the client of the project? Is it an internal or external client?
- How many sprints were completed so far?

Part IV: Agile-Scrum

On a scale of 1-5 (1 being No and 5 being Yes), please rate the following statements

(refer to the guidelines in Appendix B) **concerning your Agile-Scrum project**?

⇒ Note, do not answer the questions as it should be done, but **as it is currently done in your project.**

1	2	3	4	5

No sometimes/it depends Yes

- For each item above that you have rated 4 or less, can you please explain the adjustments made to [specific item] of your Agile-Scrum project?
- Did you have other roles not mentioned in the items above?
- Did you have other practices/ events not mentioned above?

Part V: Success

- Is the project going as expected?
    - Budget
    - Schedule
    - Quality/Scope
- Is the product owner satisfied with the project's progress?
- Is the client satisfied with the deliverables?
- Is the team satisfied with the project's progress?
- Are the project sponsor and senior management satisfied with the project so far?
- Is senior management supportive of the project? If so, how?
- Do you think the project will have a strategic impact for the company?
    - Market
    - Competitors
    - Business expansion/Development

Part VI: Understanding

- Did the adjustments to [specific element] deliver different results?
- How satisfied were you with the adjustments made to [specific element]?
- Would you say that adjusting [specific element] contributed to the project outcome?

Part VII: General & Closing

Is there anything else you would like to share?

Appendix B : 44 Scrum Guidelines

1. The process is visible and understood by all the participants
2. All participants share a common definition of “done” increments
3. The team is open about their work and the challenges they are facing.
4. The project is carried out in an incremental and iterative manner.
5. The project has one Product Owner that is a person and not a committee.
6. The project has one Scrum Master.
7. The project has a cross-functional development team (architects, testers, designers…).
8. The Scrum Master is in charge of enforcing the Scrum process and avoiding distractions/barriers.
9. The Product Owner is responsible for managing the Product Backlog.
10. The Product Owner is responsible for the Product Backlog items priority.
11. The development team executes sprints in a self-organized manner.
12. The development team has sub-teams.
13. The development team is accountable as a whole for the work delivered.
14. The development team has between 3 and 9 members.
15. Only minimal initial planning was done before the first Sprint.
16. A Sprint Planning meeting is conducted before each Sprint.
17. The entire Scrum team takes part in the Sprint Planning meeting.
18. Sprint Planning meetings last a maximum of 8 hours.
19. A Sprint goal is decided within the Sprint Planning.
20. The development team is the only one to select the number of elements from the Product Backlog to include in a Sprint.
21. The development team is responsible for all estimates.
22. The Sprints are 4 weeks or less.
23. The software is tested within the Sprint
24. A new Sprint starts immediately after a Sprint in concluded.
25. The team is not disrupted by other project- related work during the Sprint.
26. The team is not disrupted by other non-project work during the Sprint.
27. Daily Scrums are carried out.
28. Daily Scrums are internal to the development team
29. Daily Scrums are 15 minutes.
30. During Daily Scrums the development team discusses what has been done in the last 24h, what will be done in the next 24h, and any problems encountered.
31. Sprint reviews are carried out at the end of each sprint.
32. Sprint reviews are carried out in the presence of the Product Owner and key stakeholders.
33. Sprint reviews last 4 hours or less.
34. During sprint reviews the increment is inspected and the Product Backlog adapted if needed.

35. Sprint Retrospective meetings are carried out after the Scrum Review and before the next Sprint Planning.
36. Sprint Retrospectives are carried out by the Scrum team to inspect itself and create a plan for improvements to be enacted during the next sprint
37. Sprint Retrospectives last 3 hours or less
38. The Product Backlog is used as the only source of requirement.
39. The Product Backlog evolves and changes according to the environment.
40. The Product Owner is the only person making changes to the Product Backlog.
41. A Sprint Backlog is made by the development team to forecast the work needed to be done to meet the sprint goal.
42. Requirements are documented as user stories or use cases
43. A Burn down chart is used to track the project’s progress
44. The Burn down chart is shared with all the team

Appendix C : Sample Recruitment Email

Greetings,

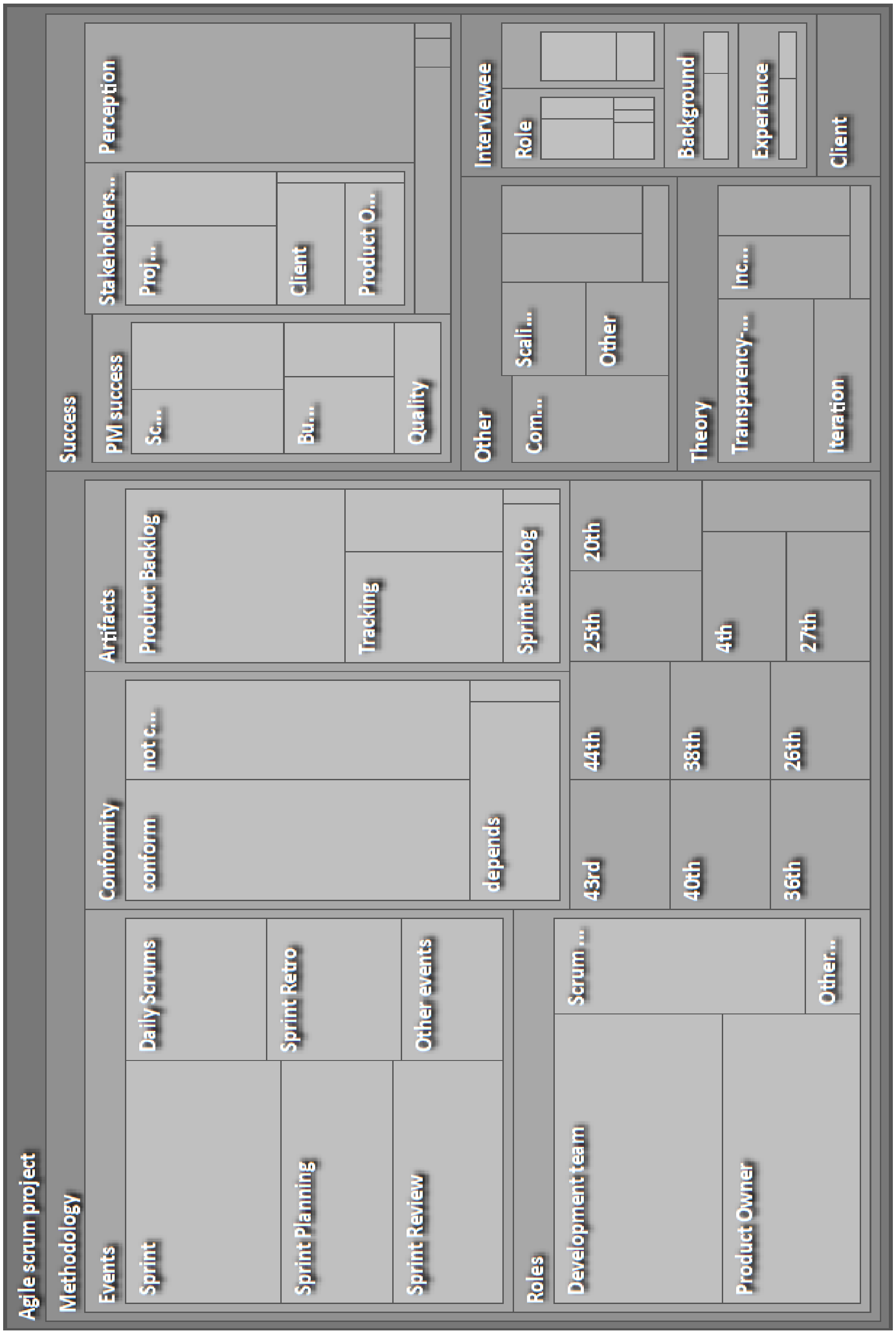
My name is Salma Hassani Alaoui, Master's student at HEC Montreal under the supervision of Dr. Ann-Frances Cameron. I am currently working on a research study about the Scrum project management methodology in software development, and conducting interviews to understand how this methodology is adopted in the organizational context. I am thus seeking professionals who are involved in IT projects using Scrum or some variation of Scrum.

If you are interested in taking part in this research, we would highly appreciate hearing from you. The interviews (approximately one hour) can be conducted in person, by phone, or via Skype at the time and location of your convenience. You will be asked your experiences on your Scrum project(s).

If you are interested or have questions, please contact me at : salma.hassani-alaoui@hec.ca

Thank you.

Appendix D : Hierarchy Chart of Nodes (4 levels)



REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.

Agar, M. H. (1996). The professional stranger: An informal introduction to ethnography.

Agility. (n.d) In *the Cambridge English Dictionary*. Retrieved from https://dictionary.cambridge.org/dictionary/english/agility

Ahimbisibwe, A., Cavana, R. Y., & Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management*, *28*(1), 7-33.

Aitken, A., & Ilango, V. (2013). *A comparative analysis of traditional software engineering and agile software development.* Paper presented at the System Sciences (HICSS), 2013 46th Hawaii International Conference on.

Alami, A. (2016). Why do information technology projects fail? *Procedia Computer Science, 100*, 62-71.

Albert, M., Balve, P., & Spang, K. (2017). Evaluation of project success: a structured literature review. *International Journal of Managing Projects in Business, 10*(4), 796-821.

Anadón, M., & Savoie-Zajc, L. (2009). Recherches qualitatives. *L'analyse qualitative des données, 28*(1), 1-7.

Anwer, F., Aftab, S., Shah, S. M., & Waheed, U. (2017). Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum. International Journal of Computer Science and Telecommunications, 8(2), 1-7.

Art Gowan Jr, J., & Mathieu, R. G. (2005). The importance of management practices in IS project performance: An empirical study. *Journal of Enterprise Information Management*, *18*(2), 235-255.

Ashraf, S., & Aftab, S. (2017). Latest Transformations in Scrum: A State of the Art Review. *International Journal of Modern Education and Computer Science, 9*(7), 12.

Ashraf, S., & Aftab, S. (2018). Pragmatic Evaluation of IScrum & Scrum. *International Journal of Modern Education and Computer Science, 10*(1), 24.

Atkinson, R. (1999). Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International journal of project management*, *17*(6), 337-342.

Awad, M. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*.

Badewi, A. (2016). The impact of project management (PM) and benefits management (BM) practices on project success: Towards developing a project benefits governance framework. *International journal of project management, 34*(4), 761-778.

Bakker, R. M. (2010). Taking stock of temporary organizational forms: A systematic review and research agenda. *International Journal of Management Reviews, 12*(4), 466-486.

Ballantine, J., Bonner, M., Levy, M., Martin, A., Munro, I., & Powell, P. L. (1996). The 3-D model of information systems success: the search for the dependent variable continues. *Information Resources Management Journal (IRMJ), 9*(4), 5-15.

Batra, D., Xia, W., VanderMeer, D. E., & Dutta, K. (2010). Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry. *CAIS*, *27*, 21.

Beck, K. (1999). Embracing change with extreme programming. *Computer, 32*(10), 70-77.

Beck, K., & Gamma, E. (2000). *Extreme programming explained: embrace change*: addison-wesley professional.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001). Manifesto for agile software development.

Belassi, W., & Tukel, O. I. (1996). A new framework for determining critical success/failure factors in projects. *International journal of project management, 14*(3), 141-151.

Belout, A. (1998). Effects of human resource management on project effectiveness and success: toward a new conceptual framework. *International journal of project management, 16*(1), 21-26.

Berczuk, S. (2007). *Back to basics: The role of agile principles in success with an distributed Scrum team.* Paper presented at the Agile Conference (AGILE), 2007.

Bhasin, S. (2012). *Quality assurance in agile: A study towards achieving excellence.* Paper presented at the 2012 Agile India.

Birkinshaw, J. (2018). What to expect from agile. *MIT Sloan Management Review, 59*(2), 39-42.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer, 35*(1), 64-69.

Boehm, B. (2006). *A view of 20th and 21st century software engineering.* Paper presented at the Proceedings of the 28th international conference on Software engineering.

Boehm, B., & Turner, R. (2004). *Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods.* Paper presented at the Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on.

Bryde, D. (2008). Perceptions of the impact of project sponsorship practices on project success. *International journal of project management, 26*(8), 800-809.

Brynjolfsson, E. (2010). The four ways IT is revolutionizing innovation. *MIT Sloan Management Review, 51*(3), 51-56.

Callanan, M. (2010). Ken schwaber on Scrum.

Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring–A systematic literature review. *Journal of Systems and Software, 110*, 85-100.

Cao, L., Mohan, K., Ramesh, B., & Sarkar, S. (2013). Adapting funding processes for agile IT projects: an empirical investigation. *European Journal of Information Systems, 22*(2), 191-205.

Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International digital library perspectives, 27*(1), 18-22.

Charette, R. N. (2005). Why software fails [software failure]. *IEEE spectrum, 42*(9), 42-49.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer, 34*(11), 131-133.

Collyer, S., Warren, C., Hemsley, B., & Stevens, C. (2010). Aim, fire, aim—Project planning styles in dynamic environments. *Project management journal, 41*(4), 108-121.

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research, 20*(3), 329-354.

Conboy, K., & Fitzgerald, B. (2010). Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. *ACM Transactions on Software Engineering and Methodology (TOSEM), 20*(1), 2.

Cooke-Davies, T. (2002). The "real" success factors on projects. *International journal of project management, 20*(3), 185-190.

Cram, W. A. (2019). Agile Development in Practice: Lessons from the Trenches. *Information Systems Management, 36*(1), 2-14.

Creswell, J. W., & Poth, C. N. (2017). *Qualitative inquiry and research design: Choosing among five approaches*: Sage publications.

Davis, K. (2014). Different stakeholder groups and their perceptions of project success. *International journal of project management, 32*(2), 189-201.

Davis, K. (2018). Reconciling the Views of Project Success: A Multiple Stakeholder Model. *Project management journal, 49*(5), 38-47.

De Wit, A. (1988). Measurement of project success. *International journal of project management, 6*(3), 164-170.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). The Scrum primer. *Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: http://assets. Scrumtraininginstitute. com/downloads/1/Scrumprimer121. pdf, 1285931497*, 15.

Deloitte (2017). «Technology budgets: From value preservation to value creation», *CIO Insider*. Retrieved from https://www2.deloitte.com/insights/us/en/focus/cio-insider-business-insights/technology-investments-value-creation.html

Denning, S. (2017). The next frontier for Agile: strategic management. *Strategy & Leadership, 45*(2), 12-18.

Diebold, P., & Dahlem, M. (2014). *Agile practices in practice: a mapping study*. Paper presented at the Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213-1221. doi:https://doi.org/10.1016/j.jss.2012.02.033

Doody, O., & Noonan, M. (2013). Preparing and conducting interviews to collect data. *Nurse researcher, 20*(5).

Driver, M. & Klinect, T. (2018). « IT Market Clock for Programming Languages, 2018», *Gartner*, document no G00343347. Retrieved from https://www.gartner.com/document/3872986

Drury-Grogan, M. L. (2014). Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors. *Information and software technology, 56*(5), 506-515.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology, 50*(9-10), 833-859.

Dyba, T., & Dingsoyr, T. (2009). What do we know about agile software development? *IEEE software, 26*(5), 6-9.

Eloranta, V.-P., Koskimies, K., & Mikkonen, T. (2016). Exploring ScrumBut—An empirical study of Scrum anti-patterns. Information and software technology, 74, 194-203.

Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: the state of research. *Journal of Database Management (JDM), 16*(4), 88-100.

Fernandez, D. J., & Fernandez, J. D. (2008). Agile project management—agilism versus traditional approaches. *Journal of Computer Information Systems, 49*(2), 10-17.

Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems, 15*(2), 200-213.

Fitzgerald, B., Russo, N. L., & O'Kane, T. (2003). Software development method tailoring at Motorola. *Communications of the ACM, 46*(4), 64-70.

Fitzgerald, B., Russo, N., & O'Kane, T. (2000). An empirical study of system development method tailoring in practice. *ECIS 2000 Proceedings*, 4.

Galletta, A. (2013). *Mastering the semi-structured interview and beyond: From research design to analysis and publication*: NYU press.

Garfinkel, J. (2018). «Gartner Says Global IT Spending to Grow 3.2 Percent in 2019», *Press Releases.* Retrieved from https://www.gartner.com/en/newsroom/press-releases/2018-10-17-gartner-says-global-it-spending-to-grow-3-2-percent-in-2019

Gill, A., & Henderson-Sellers, B. (2006). *Measuring agility and adaptibility of agile methods: A 4 dimensional analytical tool.* Paper presented at the The IADIS international conference on applied computing 2006.

Harrell, M. C., & Bradley, M. A. (2009). *Data collection methods. Semi-structured interviews and focus groups*. Retrieved from

Hartman, F., & Ashrafi, R. A. (2002). Project management in the information systems and information technologies industries. *Project management journal, 33*(3), 5-15.

Hastie, S., & Wojewoda, S. (2015). Standish group 2015 chaos report-q&a with jennifer lynch. *Retrieved, 1*(15), 2016.

Hayata, T., & Han, J. (2011). *A hybrid model for IT project with Scrum.* Paper presented at the Service Operations, Logistics, and Informatics (SOLI), 2011 IEEE International Conference on.

Henderson-Sellers, B., & Ralyté, J. (2010). Situational method engineering: state-of-the-art review. *Journal of Universal Computer Science*.

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer, 34*(9), 120-127.

Hossain, E., Bannerman, P. L., & Jeffery, R. (2011). *Towards an understanding of tailoring Scrum in global software development: a multi-case study*. Paper presented at the Proceedings of the 2011 International Conference on Software and Systems Process.

Hummel, M. (2014). *State-of-the-art: A systematic literature review on agile information systems development.* Paper presented at the System Sciences (HICSS), 2014 47th Hawaii International Conference on.

Icmeli Tukel, O., & Rom, W. O. (2001). An empirical investigation of project evaluation criteria. *International Journal of Operations & Production Management, 21*(3), 400-416.

Ika, L. A. (2009). Project Success as a Topic in Project Management Journals. *Project management journal, 40*(4), 6-19. doi:10.1002/pmj.20137

Jacob, S. A., & Furgerson, S. P. (2012). Writing interview protocols and conducting interviews: Tips for students new to the field of qualitative research. *The Qualitative Report, 17*(42), 1-10.

Jalali, S., & Wohlin, C. (2010). *Agile practices in global software engineering-A systematic map.* Paper presented at the 2010 5th IEEE International Conference on Global Software Engineering.

Jugdev, K., & Müller, R. (2005). A retrospective look at our evolving understanding of project success. *Project management journal, 36*(4), 19-31.

Jugdev, K., Thomas, J., & Delisle, C. (2001). *Rethinking project management–Old truths and new insights*.

Kalus, G., & Kuhrmann, M. (2013). *Criteria for software process tailoring: a systematic review.* Paper presented at the Proceedings of the 2013 International Conference on Software and System Process.

Kiv, S., Heng, S., Kolp, M., & Wautelet, Y. (2017). *An intentional perspective on partial agile adoption*. Retrieved from

Kiv, S., Heng, S., Kolp, M., & Wautelet, Y. (2018). *Agile Manifesto and Practices Selection for Tailoring Software Development: A Systematic Literature Review.* Paper presented at the International Conference on Product-Focused Software Process Improvement.

Kloppenborg, T. J., & Opfer, W. A. (2002). The current state of project management research: trends, interpretations, and predictions. *Project management journal, 33*(2), 5-18.

Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., . . . Hanser, E. (2017). *Hybrid software and system development in practice: waterfall, Scrum, and beyond.* Paper presented at the Proceedings of the 2017 International Conference on Software and System Process.

Kumar, R. (2014). *Research Methodology: A Step-by-Step Guide for Beginners*: SAGE.

Kwak, Y.-H. (2005). A brief history of project management: Greenwood Publishing Group.

Lapham, M. A., Williams, R., Hammons, C., Burton, D., & Schenker, A. (2010). *Considerations for using agile in DoD acquisition*. Retrieved from

Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer, 36*(6), 47-56.

Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). *Software development life cycle AGILE vs traditional approaches.* Paper presented at the International Conference on Information and Network Technology.

Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *Mis Quarterly,* 34(1).

Lenfle, S., & Loch, C. (2010). Lost roots: how project management came to emphasize control over flexibility and novelty. *California Management Review, 53*(1), 32-55.

Leybourne, S. A. (2009). Improvisation and agile project management: a comparative consideration. *International Journal of Managing Projects in Business, 2*(4), 519-535.

Malcolm, D. G., Roseboom, J. H., Clark, C. E., & Fazar, W. (1959). Application of a technique for research and development program evaluation. *Operations research, 7*(5), 646-669.

Mann, C., & Maurer, F. (2005). *A case study on the impact of Scrum on overtime and customer satisfaction.* Paper presented at the null.

Marchewka, J. T. (2014). *Information technology project management*: John Wiley & Sons.

Marshall, M. N. (1996). Sampling for qualitative research. *Family practice, 13*(6), 522-526.

Martin, R. C. (2002). *Agile software development: principles, patterns, and practices*: Prentice Hall.

Masood, Z., & Farooqi, S. (2017). Benefits and key challenges of agile project management under recent research opportunities. *International Research Journal of Management Sciences, 5*, 20-28.

McCOY, F. A. (1986). *Measuring success: Establishing and maintaining a baseline.* Paper presented at the PMI Annual Seminar & Symposium, Montreal.

McKinsey & Company (2015). «Beyond agile: Reorganizing IT for faster software delivery», Retrieved from https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/beyond-agile-reorganizing-it-for-faster-software-delivery

McKinsey & Company (2015). «The perils of ignoring software development». *McKinsey Quarterly,* Retrieved from https://www.mckinsey.com/industries/high-tech/our-insights/the-perils-of-ignoring-software-development

McLeod, L., Doolin, B., & MacDonell, S. G. (2012). A Perspective-Based Understanding of Project Success. *Project management journal, 43*(5), 68-86. doi:10.1002/pmj.21290

Mergenthaler, E., & Stinson, C. (1992). Psychotherapy transcription standards. *Psychotherapy research, 2*(2), 125-142.

Miles, M. B., Huberman, A. M., Huberman, M. A., & Huberman, M. (1994). *Qualitative data analysis: An expanded sourcebook*: sage.

Mir, F. A., & Pinnington, A. H. (2014). Exploring the value of project management: Linking Project Management Performance and Project Success. *International journal of project management, 32*(2), 202-217. doi:https://doi.org/10.1016/j.ijproman.2013.05.012

Morris, P. W., & Hough, G. H. (1987). The anatomy of major projects: A study of the reality of project management.

Munns, A., & Bjeirmi, B. F. (1996). The role of project management in achieving project success. *International journal of project management, 14*(2), 81-87.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization, 17*(1), 2-26.

Nelson, R. R., & Morris, M. G. (2014). IT Project Estimation: Contemporary Practices and Management Guidelines. *MIS Quarterly Executive, 13*(1).

O'Shaughnessy, W. (1992). *La faisabilité de projet: une démarche vers l'efficience et l'efficacité*: Trois-Rivières [Québec]: Éditions SMG.

Oliver, D. G., Serovich, J. M., & Mason, T. L. (2005). Constraints and opportunities with interview transcription: Towards reflection in qualitative research. *Social forces, 84*(2), 1273-1289.

Olsen, R. P. (1971). *Can project management be defined?*

Palmquist, M. S., Lapham, M. A., Miller, S., Chick, T., & Ozkaya, I. (2013). *Parallel worlds: Agile and waterfall differences and similarities*. Retrieved from

Papke-Shields, K. E., Beise, C., & Quan, J. (2010). Do project managers practice what they preach, and does it matter to project success? *International journal of project management, 28*(7), 650-662.

Patton, M. Q. (1990). *Qualitative evaluation and research methods*: SAGE Publications, inc.

Permana, P. A. G. (2015). Scrum method implementation in a software development project management. *International Journal of Advanced Computer Science and Applications, 6*(9), 198-204.

Pinto, J. K., & Prescott, J. E. (1988). Variations in critical success factors over the stages in the project life cycle. *Journal of management, 14*(1), 5-18.

Pinto, J. K., & Slevin, D. P. (1987). Critical factors in successful project implementation. *IEEE transactions on engineering management*(1), 22-27.

Pollack, J., Helm, J., & Adler, D. (2018). What is the Iron Triangle, and how has it changed? *International Journal of Managing Projects in Business, 11*(2), 527-547. doi:http://dx.doi.org/10.1108/IJMPB-09-2017-0107

Raith, F., Richter, I., Lindermeier, R., & Klinker, G. (2013). *Identification of inaccurate effort estimates in agile software development.* Paper presented at the 2013 20th Asia-Pacific Software Engineering Conference (APSEC).

Rasnacis, A., & Berzisa, S. (2017). Method for adaptation and implementation of agile project management methodology. *Procedia Computer Science, 104*, 43-50.

Raymond, L., & Bergeron, F. (2008). Project management information systems: An empirical study of their impact on project managers and project success. *International journal of project management*, *26*(2), 213-220.

Recker, J., Holten, R., Hummel, M., & Rosenkranz, C. (2017). How agile practices impact customer responsiveness and development success: A field study. *Project management journal*, *48*(2), 99-121.

Reifer, J. Donald (2017). «Quantitative Analysis of Agile Methods Study (2017): Twelve Major Findings», *InfoQ*. Retrieved from https://www.infoq.com/articles/reifer-agile-study-2017/

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

Robinson, O. C. (2014). Sampling in interview-based qualitative research: A theoretical and practical guide. *Qualitative research in psychology, 11*(1), 25-41.

Rowley, J. (2012). Conducting research interviews. *Management Research Review, 35*(3/4), 260-271.

Royce, W. W. (1987). *Managing the development of large software systems: concepts and techniques.* Paper presented at the Proceedings of the 9th international conference on Software Engineering.

Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular Agile process*: Addison-Wesley.

Saarinen, T. (1990). System development methodology and project success: An assessment of situational approaches. *Information & Management*, *19*(3), 183-193.

Saldaña, J. (2015). *The coding manual for qualitative researchers*: Sage.

Santos, M. d. A., Bermejo, P. H. d. S., de Oliveira, M. S., Tonelli, A. O., & Seidel, E. J. (2013). Improving the management of cost and scope in software projects using agile practices. *arXiv preprint arXiv:1303.1971*.

Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum* (Vol. 1): Prentice Hall Upper Saddle River.

Schwaber, K., & Sutherland, J. (2013). The Scrum guide-the definitive guide to Scrum: The rules of the game. *SCRUM. org, Jul-2013*.

Scrum.org (2019). *Scaling Scrum with Nexus*. Retrieved from https://www.Scrum.org/resources/scaling-Scrum

Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. *International journal of project management, 33*(5), 1040-1051.

Shenhar, A. J., Dvir, D., Levy, O., & Maltz, A. C. (2001). Project success: a multidimensional strategic concept. *Long range planning, 34*(6), 699-725.

Silverman, D. (2015). *Interpreting qualitative data*: Sage.

Smith, M. & Proctor, P. E. (2013**)**. « Implement Business Outcome Monetization as a Process for Increasing Project Success », *Gartner*, document no G00249950. Retrieved from https://www.gartner.com/document/2492715

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica, 17*(4).

Theocharis, G., Kuhrmann, M., Münch, J., & Diebold, P. (2015). *Is water-Scrum-fall reality? on the use of agile and traditional development practices.* Paper presented at the International Conference on Product-Focused Software Process Improvement.

Thummadi, B. V., Shiv, O., Berente, N., & Lyytinen, K. (2011). *Enacted Software Development Routines Based on Waterfall and Agile Software Methods: Socio-Technical Event Sequence Study*, Berlin, Heidelberg.

Trauth, E. M. (2001). The choice of qualitative methods in IS research *Qualitative research in IS: Issues and trends* (pp. 1-19): IGI Global.

Tripp, J. F., Riemenschneider, C., & Thatcher, J. B. (2016). Job satisfaction in agile development teams: Agile development as work redesign. *Journal of the Association for Information Systems, 17*(4), 267.

Turner III, D. W. (2010). Qualitative interview design: A practical guide for novice investigators. *The Qualitative Report, 15*(3), 754-760.

Turner, J. R. (1996). Editorial: International Project Management Association global qualification, certification and accreditation. *International journal of project management, 14*(1), 1-6.

Turner, J. R. (1999). *The handbook of project-based management* (Vol. 2): London: McGraw-Hill.

Turner, J. R. (2004). Five necessary conditions for project success. *International journal of project management, 5*(22), 349-350.

Turner, R. J., Huemann, M., Anbari, F. T., & Bredillet, C. N. (2010). *Perspectives on projects*: Routledge.

Turner, R., & Zolin, R. (2012). Forecasting success on large projects: developing reliable scales to predict multiple perspectives by multiple stakeholders over multiple time frames. *Project management journal, 43*(5), 87-99.

Ullah, A., & Lai, R. (2013). A systematic review of business and information technology alignment. *ACM Transactions on Management Information Systems (TMIS), 4*(1), 4.

ur Rehman, A., & Hussain, R. (2007). *Software Project Management Methodologies/Frameworks Dynamics" A Comparative Approach".* Paper presented at the 2007 International Conference on Information and Emerging Technologies.

van der Hoorn, B., & Whitty, S. J. (2015). Signs to dogma: A Heideggerian view of how artefacts distort the project world. *International journal of project management, 33*(6), 1206-1219.

Versionone (2019). «13th Annual State of Agile Survey». Retrieved from https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508

Vial, G., & Rivard, S. (2015). Understanding Agility in ISD Projects.

Vijayasarathy, L. R., & Butler, C. W. (2016). Choice of software development methodologies: Do organizational, project, and team characteristics matter? *IEEE software, 33*(5), 86-94.

Walczak, W., & Kuchta, D. (2013). Risks characteristic to Agile project management methodologies and responses to them. *Operations Research and Decisions, 23*.

Wateridge, J. (1998). How can IS/IT projects be measured for success? *International journal of project management, 16*(1), 59-63.

Weaver, P. (2007). *The origins of modern project management.* Paper presented at the Fourth annual PMI college of scheduling conference.

West, D., Gilpin, M., Grant, T., & Anderson, A. (2011). Water-Scrum-fall is the reality of agile for most organizations today. *Forrester Research, 26*.

West, D., Grant, T., Gerush, M., & D'Silva, D. (2010). Agile development: Mainstream adoption has changed agility. *Forrester Research, 2*(1), 41.

Williams, L., & Cockburn, A. (2003). Agile software development: it's about feedback and change. *IEEE Computer, 36*(6), 39-43.

Wright, G. P. (2013). *Success rates by software development methodology in information technology project management: A quantitative analysis.* Capella University.

Wufka, M. (2013). *Domain understanding in agile information system development: the role of conceptual modeling.* University of British Columbia.

Yabuuchi, Y., Kocaoglu, D., & Watada, J. (2006). *Analysis of project management in software development.* Paper presented at the Technology Management for the Global Future, 2006. PICMET 2006.

Zorn, T. (2008). Designing and conducting semi-structured interviews for research. *Waikato: Waikato Management School.*