Challenges related to the adoption of Scrum

Case study of a financial IT company

Aahd Kanane
Abstract

Software development projects took an important turn after the emergence of agile software development methods. Indeed, this kind of projects has always been managed in a traditional way that requires an elaborate effort of planning. A way considered as inflexible and that goes against the constantly changing and hardly predictable nature of software development projects. Agility offers more flexibility as far as requirements and change management are concerned. In this study, the focus is placed on one of the most popular agile software development methods, namely Scrum. It investigates the challenges related to the adoption of Scrum, and for this the case of a company operating in the financial IT sector and that has been using Scrum for 5 years is examined. This case displays that the nature of Scrum makes that there is no single way of adopting it. Adopting Scrum is more a process of continuous adaptation and improvement, therefore facing challenges is an inseparable part of this process.

Keywords: Agile, Scrum, Challenges, adoption, software development methods.

1. Introduction

The success of IT-based projects is increasingly important to the survival of businesses. However, these kinds of projects are the ones with the highest rate of failure or abandonment (Hardy-Vallee, 2012). This failure is the result of considerable difficulties and issues continued to be observed despite the use of more structured management practices (Wateridge, 1995). Therefore, the 20th century had observed the appearance of a vast number of methods, approaches, tools and techniques included under the umbrella of agility (Dingsøyr, Nerur, Balijepally, and Moe, 2012).

The emergence of agile approaches to software development projects is motivated by their specificity as IT projects. Indeed they are characterized by a high level of uncertainty, which makes it really hard to define clear and precise objectives from the beginning of the project; it takes time to build a concrete understanding of the clients’ expectations and what it will take to meet these expectations (Parnas and Clements, 1986). Another characteristic of IT projects is that they require an important user involvement to succeed (Wateridge, 1995).

This may seem obvious because it is the client who will judge the quality of the product and forge the success of the project, but the fact is, that extant research has highlighted the lack of users’ involvement as one of the main reasons related to the failure of IT development projects (Wateridge, 1995).

The particularities of IT projects restrain the use of the traditional methods for project management and this is exactly where agility offers more options. Agility allows always being
ready to manage changes (Vokurka and Fliendner, 1998) and it makes it possible to manage changes with flexibility and rapidity as a routine activity (Conboy, 2009).

Being agile means being more attentive to the people and freeing them from the constraints of the process. Moreover, the documentation is kept to a minimum level or even to naught always in the direction of releasing the project’s team to better focus on the project’s development. Parallel to this liberation, agility implies active client involvement in all stages of the project development. Finally, being agile is acknowledging that change requests are quite natural and being open and ready to manage them (Dingsøyr et al, 2012).

The flexibility and the responsiveness that agility offers has caused a noticeable regression of the use of the traditional methods (Glaiel, Moulton and Madnick, 2013), and companies are increasingly heading to methods, for example, eXtreme Programming (XP), lean software development, feature-driven development, crystal clear method, and Scrum.

All those methods share an iterative and incremental approach, but the most popular remain the Scrum method (Hossain, Barbar and Paik 2009). The term Scrum comes from the rugby game (Cho, 2008), where the players have to work together to take the ball and pass it to each other and through the different rows to win. In addition to the name, many strategies from rugby were adapted to be used in the Scrum project development process, for instance, the team interaction strategies and keeping the same core team members during the project (Cho, 2008).

Actually, rugby players are trained to play, they know how to play. They have a coach, sponsors, and supporters. They set a strategy, a plan before each game. But once the game starts they will have to adapt to the other team, to the weather, and to different parameters that are hard to plan, in order to win the game. This is exactly what happens in the companies using Scrum. Indeed, their teams are prepared to accomplish the tasks they will be assigned to, they have plans and strategies but once the project starts they will have to adapt to deliver, on time, a software that works and that the client is happy with.

Similarly to the other agile methods, Scrum advocates that individuals and interactions are more valuable than processes and tools. Also, that an operational product is more valuable than an exhaustive documentation. Besides that a successful cooperation with the client is more valuable than official contracts. Finally, that managing changes is more valuable than following the initial plans (Fowler, and Highsmith, 2001).

The main motivation for using Scrum is being able to adapt when one is not able to predict the requirements and how the product will be built (Vlaanderen, Jansen, Brinkkemper and Jaspers, 2011). It relies on ceremonies and meetings with the client and within the development team such as the daily Scrum meetings and sprint reviews, and it uses techniques that facilitate the communication and the visualization of the workload such as the virtual wall (Berczuk, 2007). The process itself allows building the product in a progressive way by planning small parts that fit on previous parts and get them approved by the client (Rising, and Janoff, 2000) before moving to the next step.

In addition to all the openness and flexibility that Scrum offers, it presents a certain number of challenges while adopting it. Challenges in terms of changing the organization’s culture, organization, and structure (Nerur, Mahapatra and Mangalaraj, 2005); changing the management style and the role of the project manager and the team members (Amiri, 2012);
and adapt to the new work procedures, tools and techniques that the Scrum requires (Nerur et al, 2005). Scrum presents also some challenges related to the knowledge management and to the communication channels and style (Nerur et al, 2005) in addition to the ones related to the customers’ involvement (Cho, 2008).

This paper focus on the challenges related to the transition governance while adopting Scrum, it investigates the challenges connected to the execution of the process, the work routines, and tools and techniques used to develop IT based projects with the Scrum method. It can contribute to increase the understanding around the challenges faced during and after the transition to Scrum and may allow to work on defined and effective transition models. The research question addressed is:

What kind of challenges do the organizations face while adopting Scrum?

This research relies on a single case study of a firm operating in financial technology sector that has been using Scrum for five years and still find difficulties to complete their transition to Scrum. The study looks closely at how this firm is using the Scrum method and at the challenges they are facing on two levels: the challenges faced by the team members as individuals and the ones related to the management of the team as a whole.

2. Related research

In this section the literature related to project management approaches and methods regarding software development are examined. Thus, relevant research pertaining to the management of IT projects with a special focus on agile methods and challenges faced when adopting agility is presented.

During this literature review, the focus is to identify and understand the challenges around using Scrum and build a definite view on what previous research has addressed so far. This work allows gradually identifying a gap and expressing its aspects. Therefore, this section provides also a framework and a base for this research to build on.

2.1 Software development projects

Software development is a label that refers to the process going from the design to the production and the introduction of the software even though it can be used to designate programming and coding basically (Kaur and Garg, 2013). From a wider perspective “software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products” (Kaur et al., 2013).

A considerable amount of literature has been published on the particular characteristics of software development projects. These studies agree that software development projects are very complex, and they need a high level of integration to succeed since they are composed of many units built on each other (Kraut and Streeter, 1995).

Another flagship feature of software development projects is their considerable level of uncertainty (Jurison, 1999; Kaur et al, 1995). “The unpredictability of both the software and
the tasks that software engineers perform” (Kaur et al., 2013: 70) makes that the necessary resources and the duration of the project are hard to estimate (Jurison, 1999; Kaurt et al, 1995).

This uncertainty is also valid when it comes to defining the requirements. It is hard to identify and foresee complete requirements and that is why software development projects require a lot of flexibility to manage the changes and adjust the project while developing the requirements as the project is going (Jurison, 1999; Kaurt et al, 1995; Stepanek, 2012).

2.2 Software development methods

Despite all the specificities of software development projects, they remain projects before all and it is natural to deal with them as such, especially that they encounter the same constraints as any other kind of project as for instance cost, quality, and time related issues (Schwalbe, 2011; Spasibenko and Alite, 2009). In this regard, the literature acknowledges two branches of project management methods: the traditional and the agile methods (Jurison, 1999; Cicmil, Williams, Thomas and Hodgson, 2006).

2.2.1 Traditional system development methods

The traditional system development methods are based on a plan-driven approach using a standard development process (McConnel, 2010; Dybå, 2000) mainly built around the waterfall method (McConnel, 2010). That necessitate an extended, exhaustive, and fully documented assortment of requirements (Cohen, Lindvall and Costa, 2004; Khalifa and Verner, 2000; Bohem, 2002), and focus on an elaborated planning phase (Glass, 2001).

The traditional methods state that the development should be managed by following a number of well-defined steps (Thomsett, 2002; Cohen et al., 2004) going from a complete study of feasibility to the implementation and maintenance passing through software plans and requirements, product design, coding, and integration (Thomsett, 2002; Cantor, 1998).

The purpose of those methods is to focus on the process and take the steps one after the other with paying a special attention to the intended milestones (Somerville, 1996; Bohem, 2002). This naturally brings a strong rigidity to the development process. A rigidity that goes against the rapidly changing environment that software development belongs to (Thomsett, 2002).

Those methods are highly effective when it comes to well defined projects with fixed requirements (Bechtold, 2007). However, as they are based on predictable goals and are not suitable for changing requirements they are becoming less and less appreciated by software developers (Jag, 2000).

Furthermore, if we add to it the complexity and lack of certitude around IT projects, a new branch of project management methods had to come out. And that is what was called in late 1990 agile system development methods (Bohem, 2006).

2.2.2 Agile system development methods

Agile system development methods are intended to guide the projects in a volatile environment and suffering from the tenacity of a very competitive market (Cho, 2008). Indeed, they offer a more open working environment, they are attentive to the project
environment and stakeholders, and most importantly they offer a much more flexible framework.

The first actual reference for agility dates back to 2001, when a group of managers in different fields gathered and set a number of principles that would become the benchmark of any person or organization that identify themselves as agile. They created the agile manifesto one of the first references for agile system development methods (The agile manifesto, 2001).

In this manifesto they agreed on 12 core principles that form the spirit of agility and shape the essence of all the agile system development methods, which without agility will not occur. Those principles are as following:

“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” (The agile manifesto, 2001).
In this manifesto, they also agreed to give a high importance and when relevant give priority to individuals and interactions, creating working software, fully collaborate with the customer, and respond to change. And this, over the attention dedicated to the processes and tools, developing a comprehensive documentation, negotiating the contracts, and following the initial plans (The agile manifesto, 2001).

Regarding software development being agile can be actually expressed through the use of different methods, for instance eXtreme Programming (XP), lean software development, feature-driven development, crystal clear method, and Scrum (Cho, 2008). In this paper, the emphasis is put on the challenges related to the adoption of Scrum and therefore it seems appropriate to look at it more closely.

### 2.3 Description of Scrum

Scrum is an agile development method that has emerged in late 1990’s (Cho, 2008). It offers a “process for incrementally building software in complex environments” (Rising et al., 2000: 27). The development is based on iterations called sprints that usually last from 1 to three weeks (Rising, 2000; Cho, 2008; Glaiel et al., 2013), for each sprint a sprint backlog is defined.

The sprint backlog is basically a set of tasks that were selected to be executed during the sprint. It is built according to the product backlog, which includes all the features that the product must have (Cho, 2008; Glaiel et al., 2013).

As “Scrum is defined not so much by its process as by the practices that compromise it” (Spasibenko, et al., 2008 citing Koch, 2005: 257), The process includes many ceremonies, for instance the sprint planning meeting where the objectives and tasks to be performed during the sprint are presented and defined. Then the daily Scrum meetings where the development team members can share their work, their problems and their advices. Finally, the sprint reviews where the development team meets the client and talk about the outcome of the sprint (Rising, 2000; Cho, 2008; Glaiel et al., 2013).

By the end of each sprint the deliverables are tested by the customer and a decision is made: they are either validated or have to be re-worked in an upcoming sprint.
A Scrum team usually includes between seven and ten people (Cohen et al., 2004), including besides the development team a product owner who has a complete view of the product and how it should operate, and a Scrum master who is in charge of ensuring the best conditions for the execution of daily work (Cho, 2008; Glaiel et al., 2013).

To promote the success of this process a number of success criteria have to be taken in consideration. The size of the team for instance is important, indeed, working in small teams is highly recommended when it comes to using Scrum because people in small teams have tendency to communicate better and easily using informal channels (Boehm et al., 2005).

Besides the size of the team, the distribution of work in each sprint on team members has to be clear and well defined and the length of the sprint has to be short this will encourage to keep focused on the objectives (Boehm and Turner, 2005; Schwaber, 2002). Not to forget, of course to take in consideration the evolution of the market and be sensitive to change in order to always offer the best to the customer (Boehm, et al., 2005).

Scrum has a lot of positive sides as the numerous meetings that favors team building that impact largely the effectiveness and rapidity of individual and team problem solving, it counts simple practices that can be easily adapted to the capacity of the team, and the most important is the evolutionary definition of the requirements that helps to meet the customers’ expectations as best as possible (Larman, 2003).
2.4 Identified challenges in relation to agility and to the Scrum method

Since 2001, the agility brought a lot to software development methods. In 2012, the Project Management Institute (PMI) published a study about, inter alia, the effectiveness of companies in change and risk management. This study was conducted in among 1,239 practitioners involved in project management around the world. The study demonstrated that: 92% of the organizations that were assessed as highly effective at change management and 90% of the organizations that were assessed as effective at risk management are highly to moderately agile (PMI, 2012).

The attractiveness of agile software development methods is undeniable, but being agile is not an easy task. It presents many challenges in terms of team management and the perception of the roles (Nerur et al., 2005; Vinekar, Slinkman and Nerur, 2006; Chow and Cao, 2008; Cockburn and Highsmith, 2001; Glaiel et al., 2013), challenges due to adopting a new culture, attitude and practices (Nerur et al., 2005; Vinekar et al., 2006; Cho, 2008; Greene, 2004).

In addition to challenges related to new communication arrangements within the team and with the customer (Cho, 2008; Paasivaara and Lassenius, 2006; Nerur et al., 2005; Chow et al., 2008; Conboy, Coyle, Wang and Pikkarainen, 2011; Cockburn et al., 2001; Glaiel et al., 2013) as well as the ones related to the customers’ involvement (Cho, J., 2008; Nerur, S. et al., 2005). Then, the challenges related to the new decision making procedures (Conboy et al., 2011 Dingsøyr et al., 2012).

A large and growing body of literature has investigated the challenges related to agility and adopting Scrum in particular. Indeed, using Scrum in particular brings its own challenges related to the process, the ceremonies and the practices around the method. It presents challenges related to defining the product and sprint backlogs, to the estimation of the sprints in terms of size and length. Also, challenges in managing the daily Scrum meetings, the sprint planning, and the sprint review meetings. In addition to the challenges related to the expectations of the client and his involvement in the whole process (Cho et al., 2008; Glaiel et al., 2013; Amiri, 2012).

There is a large volume of published studies describing what agility and Scrum bring to software development projects in terms of flexibility and reactivity and a considerable amount of literature has been published on the challenges related to Scrum but very few of them pointed the challenges issued from the transition to Scrum and may compromise completing this transition. Whereas the use of Scrum keeps growing, more is needed to be known about the challenges that may arise during the transition and that can last for years after adopting this method.
3. Research design

In this section all the elements specific to the case and to the research are presented. Namely, the description of the case, how the research was conducted to obtain the results, and the conclusion reached at the end of the paper. First the case and the context where the study has been conducted are described, then how the data was collected and analyzed is presented, and finally, the limitations that the study presents in terms of the methodology used and reliability of the results are discussed.

This research was conducted in cooperation with a software development company in the financial technology field. A company that define itself as always having been agile but have never adhered to any particular method before Scrum.

The case of this company is not only interesting because they have been using Scrum for five years and still face challenges to fully adopt it, but also because they work with customers in different countries and work with development teams of more than 20 people located in two different cities.

In order to know more about how this company managed to use Scrum in those circumstances and what challenges they are facing a qualitative research based on the case study method was conducted.

3.1 The case study method

The case study method is a detailed study of a particular case to explain actual circumstances that require a comprehensive description (Yin, 2009).

It “allows investigators to retain the holistic and meaningful characteristics of real-life events, such as individual life cycles, small groups behavior, organizational and management process, neighborhood change school performance, international relations, and the maturation of industries” (Yin, 2009: 4). This method helps as well understanding complex phenomenon that are in deep relation with their context (Yin, 2009).

Besides this using the case study allows a “multiplicity of perspectives which are rooted in a specific context” (Ritchie and Lewis, 2003: 52) which is rewarding when it comes to understanding the complexity and the different aspects of a problem (Ritchie et al., 2003).

Therefore, this method is considered as relevant and appropriate for this research, keeping in mind that this is an exploratory research where we are trying not only to understand how a software development company managed to adopt an agile project management method in its particular conditions but also to understand the kind of challenges they are facing.

For this purpose data was collected from the team development members working in different projects and having different roles. The process of data collection is developed in an upcoming section.

3.2 Case description

The company used as a basis to develop this study, operates in the financial IT sector and develop products for partners in different countries. It is a company that counts around 250 employees who work in teams of 20 to 30 members taking part from two different locations. They manage large projects with international stakeholders and that last for several years.
In order to communicate with the part of the team or with the stakeholders in a different location they use an in-house developed platform to share all the information about the product, the backlogs, the tasks, the progress, and any problem they are facing. They also use instant messaging or video conferences on a daily basis to lead the meetings and to help these simple and informal communication channels which is representative of the adoption of Scrum.

This company has been defined as agile since its beginnings, but never adhered to any particular method before Scrum. Today, it has been five years since they started using Scrum. However, they are still not satisfied with the way in which they use it and consider that they are still facing several challenges to adopt it.

3.3 Data collection and ethical considerations
The data was collected in cooperation with another master student in the IT management master program. We worked together during this step; we elaborated the interview guide (appendix A), and conducted the interviews together. Therefore, we share the same case and the same data for our respective papers. It is worth noting that no analysis of any kind has been done in common.

The data was collected using semi-structured and face to face interviews with eight employees in the company including developers, a project manager, a requirement analyst, a tester, a Scrum master, and a technical account manager. We did not set any particular criteria choosing the participant except that we tried to have as many different roles as we could.

We used the semi-structured interviews because they were assessed as the most sufficient way when we can interview the respondent only one time and also because they provide data that can be compared among all the participants (Cohen and Crabtree, 2006)

The interviews were structured around four main themes: the respondents’ background, their work routine, how they use the Scrum method on a daily basis, and the challenges they are facing. We interviewed eight members from the company, who are all working in the same location and are in charge of different roles and are involved in different projects.

All interviews were conducted individually in the company’s office, and were recorded with the consent of the company and the participants. Information such as the name, gender, and age were not collected. The following table summarizes the roles of the participants, how long they have been using Scrum, and the length of the interviews.

<table>
<thead>
<tr>
<th>Role</th>
<th>Years using Scrum</th>
<th>Length of the interview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer 1</td>
<td>5 years</td>
<td>77min</td>
</tr>
<tr>
<td>Tester 1</td>
<td>8 months</td>
<td>33min</td>
</tr>
<tr>
<td>Developer 2</td>
<td>2-3 years</td>
<td>40min</td>
</tr>
</tbody>
</table>
## Table 1: An overview of the respondents and the length of the interviews

<table>
<thead>
<tr>
<th>Role</th>
<th>Experience</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project manager 1</td>
<td>2 years</td>
<td>31min</td>
</tr>
<tr>
<td>Requirements analyst 1</td>
<td>8 months</td>
<td>32 min</td>
</tr>
<tr>
<td>Tester 2</td>
<td>2 years</td>
<td>28 min</td>
</tr>
<tr>
<td>Scrum master 1</td>
<td>2 years</td>
<td>43 min</td>
</tr>
<tr>
<td>Technical account manager 1</td>
<td>2 years</td>
<td>32 min</td>
</tr>
</tbody>
</table>

Even though the interviews were organized in different ways, and the questions were adapted and built upon the respondents’ answers, all the interviews had the same core questions. In other words, all the interviews had the same guideline leading the conversation and were focused on understanding how Scrum is used in this company and what kind of challenges they are facing.

During the collection of this data a particular attention was given to the ethical considerations that basically “involve ensuring that potential participants have a clear understanding of the purpose of the study, the funder, the organization or individuals conducting it, how the data will be used, and what participation will mean for them” (Ritchie et al., 2003: 76). Therefore before starting the interview we presented to each participant the topic and the goal of the research, how the data will be used, and we asked them if they agree that we record the interview.

The interviews were transcribed later and coded for the analysis. The analysis procedure is described in the next section.

### 3.3 Data analysis

This section presents how the data was examined to understand how Scrum is used in the company and to identify the challenges that go with adopting it. First the method used for the analysis is defined and motivated, and then the process used to execute the analysis is described.

In order to analyze the data, the grounded theory approach was used as a method for the analysis. This takes the form of an inductive process for generating theories (Corbin and Strauss, 1990; Myers, 1997; Ritchie et al., 2003). This approach is “extremely useful in developing context-based, process-oriented descriptions and explanations of the phenomenon” (Myers, 1997: 8).

Applying this approach consists basically of: firstly, an open coding of raw data, this is an iterative step that ends up with grouped codes. Secondly, axial coding is made by identifying the similarities between the codes and similar codes are grouped in labelled categories. Thirdly, selective coding is made upon the categories identified in the previous step. The selective coding focuses on the central phenomenon that emerges from the axial coding and
that will structure the theory. Finally, and when all those steps are achieved we end up with ex-post results that have to be compared with previous research (Joannides and Berland, 2008).

Figure 2: An overview of the coding process

The analysis of the data collected started when all the interviews were made and was built on three levels. Those levels are described in the following paragraphs but the outcome of this process is developed in the upcoming section.

In the first level of the analysis, all the interviews were transcribed into text, coded, and grouped in two types of categories. This level resulted to categories for challenges and categories of potential causes as perceived by the respondents.

In the second level, the goal was to identify the challenges that were faced by some roles more than others and to understand why. The following table gives an example of how those steps were done in a concrete way.

<table>
<thead>
<tr>
<th>Role</th>
<th>Transcription</th>
<th>codes</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer 2</td>
<td>“It’s difficult to get a big picture of what the system really does”</td>
<td>Full view of the product</td>
<td>Prerequisites to execute the work</td>
</tr>
</tbody>
</table>
Tester 1  “It’s hard to know who I can ask and even if you find him it’s still hard to understand how it’s supposed to work”  Hard to get help  Prerequisites to execute the work

Tester 1  “Teams are too big for a short meeting”  Team size  Meetings management

Developer 1  “defining the backlog is challenging because you never get it right and it affects the rest of the work”  Backlog definition  Sprint management

Table 2: An overview of the codification process

Finally, in the third level the identified challenges were compared to the previous research and put in two categories: The challenges that are more or less common and already addressed by the literature and the ones that are specific to the case of the company. This third part can be found in the discussion section.

3.4 Limitations

The main weaknesses and limitations that this research presents are basically related to the results significance. Indeed, this research is based on a single case study which means that it wasn’t possible to compare the results with any other references than the literature. Moreover, only one source of data has been used, the interviews weren’t combined with observation for instance.

The interviews involved only the team members of one location. No data was gathered about how the other part of the team perceives the challenges they are facing. Besides this, the analysis was not done from the beginning of the data collection as recommended. In fact, the “analysis is necessary from the start because it is used to direct the next interview and observations” (Corbin et al., 1990: 6). The analysis started when the data collection was finished, so the interview guide was not improved during the data collection phase.

4. Results

The purpose of this research is to increase the understanding around the challenges related to the adoption of Scrum as an agile software development method. We have conducted semi-structured interviews with development team members of a company that has been using Scrum for five years. This section presents the results of the analysis made on the data collected.

The first thing that caught my attention during the data analysis is that all the respondents are talking positively about their experience with Scrum and believe that it is an appropriate way to manage the projects they are involved in, the second thing is that many of them never
worked with Scrum before but most of them were working in a very agile way that does not really belong to any software development method. Finally, they all have different perceptions about the method, how it should be used, and the purpose of the activities related to it.

The different respondents describe challenges that can be classified in three main categories and are presented by order of relevance from the described as most problematic to the described as less problematic by the respondents. The categories are as following: challenges related to the perquisites to execute the work, challenges related to the sprint management, challenges related to the team dynamic.

Some other secondary challenges were identified related to the time gap and the interaction with the customers but are qualified as occasional by the respondents and do not have a significant impact on their work routine or on the success of the project.

Also team members find that understanding the area of expertise of their clients very challenging, and it is hard to develop a product without understanding in which context they will have to be implemented. However, this challenge has been considered as not related to the adoption of Scrum and was therefore rejected during the analysis.

4.1 Prerequisites to execute the work

The development team members face a number of challenges that adversely affect their ability to execute properly the tasks assigned to them. The first challenge in this regard is the access to a full view of the product.

Even though the whole product is developed by the same company it is hard for the development teams to get a picture of what the product is supposed to look like and what features it will contain.

“It’s difficult to get a big picture of what the system really does” Developer 2

“The systems are very complex and it’s hard to see how it’s supposed to work” Tester 1

Therefore, the team members have usually to spend a lot of time asking different stakeholders and do crosschecks to understand what the product is supposed to do or how it is supposed to work.

“There isn’t a person who knows all the parts of the projects, so you to go to different people to get the information needed” Developer 1

Because of this, the team members do not only spend a lot time to build this full view of the project but also, as they are doing it separately and using different sources, they are exposed to many misunderstandings and leaving a large margin for personal interpretations. All this of course, has a considerable impact on performance of the development team.

“In order to develop fast and effectively this part should be managed more smoothly” Developer 1
The second challenge they face in terms of preconditions for the execution of their daily tasks is to seek and get the help and guidance when it is needed.

Indeed, five years after adopting Scrum, the development team members still find considerable difficulties to get help or advices. This situation is usually described as following:

“It’s hard to get help” Tester 1

“We have situation when things go terribly wrong and you don’t have anyone to ask” Developer 1

They have the feeling that they are left to themselves, and have very often to cope with the difficulties they are facing alone.

“It’s hard when you pick a task and you don’t know how to do it, or you don’t know who can help you to do it or when you don’t understand how it’s supposed to work or who knows” Tester 2

“It’s hard to know who I can ask and even if you find him it’s still hard to understand how it’s supposed to work” Developer 2

This adds more stress to their work days and endangers their ability to realize the work properly. Also, this makes them miss many opportunities for learning and improving.

In addition to the difficulties in finding help and figuring out how to execute the work, many team members have troubles to understand what is expected from them. It is usually because the requirements are not clear, or not exhaustive.

“Sometimes the requirements are missing or are not clear so you never know when you are done or when it’s good enough” Developer 2

“The most challenging is to understand what I have to do. It’s not really difficult to execute, the real problem is to find out what to do” Tester 1

“The requirements are often not well defined and sometimes they don’t have the requirements for specific parts” Developer 2

“Understanding the backlog is the most challenging; they are not always very well explained” Tester 1

Now, that the first category of challenges is defined. To summarize this category counts as major challenges the access to the full view of the product, getting help and guidance, and understanding the requirements and the expectations. We move to the second category of challenges related to the sprint management.
4.2 Sprint management

In this category, the challenges are mainly related to keeping the sprints under control. The main challenge that the respondents face in this category is the definition of the sprint backlog.

“Defining the sprint backlog is challenging because you never get it right and it affects the rest of the project” Project manager 1

“Defining the backlog and the functionalities to be delivered all the time are always very challenging” Requirements analyst 1

Facing difficulties to define the backlogs affects deeply the ability to estimate the size and the length of the sprint backlog. Normally, the company uses a bottom up approach to build the estimation.

“The client prioritizes the functionalities he wants for each sprint, the team is involved in the sprint backlog estimation; each member says how long they will need according to the functionalities planned for the sprint” Project manager 1

But it is still very problematic to estimate how much time the tasks will take and decide how the sprint will be loaded.

“When you start up a new team it’s very hard to estimate how much time they will need. But after a while you can expect how much they can deliver, it takes some starting time” Project manager 1

“The hardest part is to do the estimation, the estimation line by line is always a bit tricky and it’s always hard to know how different types of the lines or of the tasks or of the stories will affect each other” Project manager 1

Finding the balance between the client’s requirements and deadlines and what the team is able to deliver is a negotiation process that has to take its time to operate properly. This makes the team members struggle very often with overloaded sprints hard to survive for the team members.

“They are usually pushing more than what we can chew and pulling out stuff afterward, I think they should do the opposite so it will be much easier for everybody to relax and do a good job” Developer 2

And this also creates problems in terms of sprint length management or time management when parts of the development take more time than expected and the rest of the team have to work in conditions even more tense.

“It’s hard to keep the sprint length around three weeks to 1 month because we have complexes releases” Developer 1
“Usually the sprints are three weeks development and 1 week testing and if the sprint gets shorter or the development takes more time, usually we cut on the testing time” Developer 2

The company also faces difficulties from time to time to freeze the sprint backlogs once the sprint starts, as their clients expect them to be able to carry out changes whenever this seems appropriate.

“sometimes we receive change request after the sprint starts and it is allowed to request changes once the sprint starts, if the changes are small we can do them otherwise they will have to write a change request for it” Tester 1

“Most of the time if those changes don’t take a lot of time and are quite easy we just do them” Developer 2

This can be seen as very positive and proactive as far as they are dealing with minor changes, but this carries the risk of making quick decisions that will not be properly documented and may damage parts of the product previously delivered. And once no special attention is paid to the documentation it is challenging to trace the process and understand where the problem comes from.

“It’s hard to see where the problem comes from when a bug occurs” Developer 2

“The problems are not well written, it’s not about the method it’s about how the people write” Technical account manager

“We can’t see what the problem is or was because of the way it was recorded” Technical account manager

To summarize this section includes the challenges related to the sprint management which are derived from the difficulty of defining and estimating the backlogs, managing the sprint length and load, freezing the sprint backlogs during the sprints, and documenting the changes.

4.3 Team dynamic

In this section, the challenges that prevent the team members to work as a team are described. Those challenges can basically be summarized around the meetings management and the perception of how the process should run.

The first challenge in this category is managing the daily Scrum meetings, many respondents think that the daily Scrum meetings are valuable even if they are not always effective and still need to be improved.

The first difficulty that pops out when it comes to manage such meeting is the number of participants. Indeed, the size of the teams is one of the biggest obstacles that they have to manage as far as the meetings are concerned.
“The teams are too big for a short meeting” Tester 1

“The daily Scrum meetings are not very productive when we are too many, but I don’t see another way of doing it. It takes a lot of time but it’s also difficult to split up” Developer 2

The number of participants influence the duration of the meetings as well. Indeed, Scrum meetings are considered as time consuming especially when they are associated with other meetings that the team members have to attend and when their effectiveness is largely disputed.

“In the beginnings all of us, everyone was talking and if you answer the people and everyone takes two minutes it’s an hour!” Tester 2

“Scrum meetings can be a bit long because we are around 30 people and people don’t really adhere to the method so they have tendency to talk a lot” Tester 2

Even if they try to keep the length of the daily Scrum meetings around 15 minutes, it is very challenging because of the number of participants and how much they want to share or they believe they have to share. Moreover, some team members started to consider respecting the 15 minutes as a criteria in the effectiveness of the meeting, therefore it became also hard to have those team members satisfied with the meetings.

“we try to set it to 15 minutes occasionally we can finish in ten minutes and have an extremely good meeting everybody has gone through what they have done, what s on the plan what they are going to do what are the obstacles and ... and all in ten minutes and that s good but the next time it can take 25 minutes because it’s important to discuss something specific and it can be a good thing to have it in a larger set that between two person so effective meetings is always a huge problem” Scrum master 1

“Well it’s always a struggle some people say that a meeting that is longer than ten minutes is a waste of time I can understand that from one point but then again it’s all about emotions also you need to have people satisfied from a meeting” Project manager 1

The effectiveness of the Scrum meetings is also related to the way with which they are managed and organized. Depending on the projects, many meetings are not organized as they should be and the agenda is not always interesting for everyone

“I don’t like when people start talking about a specific problem that can be taken after the Scrum meeting and loose everyone’s time” developer 2

“when we started most of us were very new to Scrum so then the meetings weren’t effective at all, because everyone just sort of starts talking about
anecdotes and stuff that is not really relevant but we are working on getting more and more efficient so I think the last two months or so we have been pretty effective I think that there is a sort of limit of 15 min but we are quite a lot of people on the developers Scrum it’s usually more than 20 people so ... then it becomes troublesome to keep the time limit” Tester 2

In order to provide a solution to this problem, in many projects they decided to split the team in two or three groups and have separate meetings for each group and one big meeting where only some members are expected to talk. But despite this, the meetings still time consuming and needs to be managed more effectively especially when the team members will have to attend more meetings.

“On the developers Scrum most people go but I’m usually a bit late to that because they talk a lot I usually just go to my Scrum” Tester 2

“if I go to both the developers meeting and my own Scrum it takes 35 to 40 minutes and if I wanna have one meeting that’s alright but then it could be like some serious issues come up and then there is another meeting and those meetings don’t really have a time frame It’s just until we figure out what to do so that could take also take more time so two to three meeting per day that’s like one hour and a half to two hours, that’s a lot of meetings” Tester 2

But then another challenge appears, having different types of daily Scrum meetings running at the same time creates a problem of communication between the different groups.

“what we have seen is that when you have a too big group it doesn’t work well but on the other hand if we split it up in three different Scrum meetings we got the issue of communicating between each group making sure that the decisions taken in one Scrum group is distribute to all the others and etc.” Scrum master 1

There is also another difficulty that should be taken seriously: the team members do not have the same perception of the purpose of the Scrum meetings.

“The hardest part is to get the team to work together and understand purpose of the meetings” Scrum master 1

Some of the respondents think that it is a method for reporting or for sharing with the rest of the team and with the Scrum master what they are doing. Others perceive it as a good way to get help so it should be used to report problems and discuss solutions. Some are just focused on how long it should take regardless of what is discussed during those meetings. Finally some other respondents believe that it is a tool for continuous improvement above all.

“The Scrum meeting is a good way to get help” Tester 1
“The meetings are important to get the synergy and the information flow it’s a very effective way of spreading information” Developer 2

“It become more a method to report to the project manager what you are doing than a method that makes you perform better” Scrum master 1

“Some people have tendency to say this is a bad Scrum meeting because it took 17 minutes” Scrum master 1

“Getting people to understand that Scrum isn’t about just reporting what you are doing but there is actually a purpose why we are doing the meeting and make sure they understand it” Scrum master 1

Several respondents think that as development team members they are not educated to the use of Scrum, especially when it comes to the daily Scrum meetings.

“It could be possible to educate the ones who are holding the meetings so they will be able to stop some discussions and the meetings don’t get longer” Developer 2

They state also that they work in a very stressful environment and in addition to all the other challenges that we have described so far, they have to deal with a tense sequencing of the sprints and that they are always overloaded.

“I want to become better on this to find a way of having a recreation between the sprints because we are not good at that we are usually running one sprint and when it’s ended we start the requirements for the next sprint and we will continue so it’s a constant movement of effectiveness so if you run really hard you have the chance to burn out people” Project manager 1

“It’s very competitive always struggling on the effectiveness and it’s very easy to get into the process where you do not take any time to go back having some sort of formal education or something. So planning on some sort of those breaks with vacation for the team or education is extremely important” Project manager 1

The questioned development team members have all described more or less the same challenges regardless of their background, role, or the project they are involved in. The challenges that can be related to a particular role more than others could not be identified. Moreover, we have interviewed only one person from several roles which makes it ambiguous to attribute a certain challenge to a specific position or role.
5. Discussion

The objective of this study is to identify the challenges related to the adoption of Scrum, therefore the case of a company that have been using Scrum for five years has been studied and the results are structured around three categories of challenges: challenges in relation with the prerequisites for executing the work, with the sprint management, and with the team dynamic.

Despite the specificity of this company as it is handling large and complex projects, with teams of 20 to 30 people in two different locations, and with international stakeholders, the findings of this research confirm the results of previous research regarding the challenges related to Scrum.

Moreover, the results extend the research reflecting upon the suitability of Scrum for large teams or in an international context, as this company does not face more challenges than the others.

In this section we first come back to the challenges identified and confront them with the results of previous research. Then, a reflection upon the implication of the results for research and for practice is made.

5.1 Reflection on the results

The data analysis has identified challenges related to the prerequisites that the team members require to execute their tasks properly, such as getting the full view of the product, getting help, and understanding the requirements. Also, challenges related to the sprint management, for instance defining the sprint backlog, estimating the length of the sprint, and freezing the sprint backlog after the sprint starts. Besides the challenges related to the team dynamic for example the managing the meetings and dealing with a stressful environment. In this section the challenges will be presented in the same order of the result section.

5.1.1 Prerequisites to execute the work

The team members are facing difficulties getting access to a full view of the product. Indeed, breaking down the product to defined backlogs and to tasks helps to stay focused during the sprints, but sometimes it is also important to have a big image of how the whole product should work. This is even more problematic because most of the times the user’s stories are not entirely written, they are built as parts of the product are delivered.

The team members can work closer with the product owners in this direction to get a better view of what has been done so far and the main expectations for what is coming, but it will be always hard to get clear view of the functionalities of the product, what it will do and how it is supposed to work.

Breaking down the product backlogs to defined tasks may have another side effect: the company may end up with team members focusing only on the tasks they have been assigned to without taking in consideration that they will have to cooperate with other team members to build other parts of the product.

Scrum, and similarly to other agile methods, is about focusing on the individuals and interactions over tools and processes, therefore the interaction between individuals take
precedence over the formality of written procedures. Knowledge transfer is accomplished through spontaneous and informal communication channels (Luz, Gazineu and Teófilo, 2009).

Team members should be more involved in taking decisions in relation with the communication tools and have to learn to ask for help as soon as they start facing a problem (Berczuk, 2007), and the work environment has to be favourable to ask for help and get it.

This is also true when it comes to understanding the requirements and the expectations. As Scrum is mainly based spontaneous and informal communication channels and little attention is given to documentation, this leaves a large space for personal interpretation and misunderstandings.

It is important to spend time in the beginning of each sprint to be sure that everyone in the team has the same understanding of the requirements, and it is true that it will take even more time when this team involves 30 members. However, on the longer term it will reduce the mistakes and rework rate and will save the team members time and frustrations as they will have to spend less time figuring out what to do.

5.1.2 Sprint management
Defining the sprint backlog is definitely one of the hardest parts in the Scrum process. It is the equivalent to the planning phase in the traditional methods. In the case of this company, the customer defines the order with which he wants to receive the releases but they still have to organize it in sprints and make sure that those sprints do not exceed one month. Therefore they are in a constant negotiation with the customer and with the team members to agree on what will be delivered and when.

Having the right size of sprint backlogs that the customer is satisfied with and that the team is able to deliver is problematic. It takes time to adjust, learn, and apprehend how much the team is able to handle and deliver within a specified period (Glaiel et al., 2013), and this regardless of how big is the team and how complex is the product.

Getting right estimates is also hard when the team keeps receiving requests from the client once the sprint starts. Normally “no new requirements can be introduced during these sprints” (Vlaanderen et al., 2011: 59), but this is hard to achieve in reality. The customer is expecting to be able to request changes whenever this seems appropriate, and the company has to define what can be considered as minor change and integrated in the sprint or what can be considered as big change and has to be reconsidered for upcoming sprints.

What can happen is that the team does not really realize the impact of this change and accept to do it without enough analysis. This will of course affect their ability to deliver on time so they will just start cutting in the testing part to be able to respect the deadline taking the risk to have to deal with bugs and rework later.

As we have discussed earlier, when using the Scrum method producing extensive documentation is not a priority anymore. Therefore, agile methods in general consider that “the code itself should be a document” (Cho, 2008: 192), so more comments should be written to explain the difficult parts in the code or the changes they have made. However, even by doing this, the problem of what to write and how to write it is always imminent. Also, if these comments had to be add to the work of the developers they will certainly consume more time but on the other hand the rest of team members will spend less time asking
questions and trying to figure out what is happening when a problem occurs or when a new member is joining the team (Cho, 2008).

5.1.3 Team dynamic

The specificity of this company is that they are working with big teams, teams that count more than 20 members in two different locations, but surprisingly the team size does not seem a problem for them except when it comes to meetings management. However, due to the complexity of releases it is impossible to do it differently. It would be almost impossible to deliver the same releases with a reduced team.

To face the problem of overcrowded meetings, in some projects they decided to split the teams into two to three groups, each group each group makes its own daily Scrum meeting and then they meet in a bigger one with all the groups but not everyone will be talking. This procedure performed in an instinctive way is very similar to another practice already described by the literature and known as “The daily Scrum meeting of Scrum meetings”, it is used when the project requires several teams. The objective of this meeting is to coordinate and integrate the work of those different teams (Cho, 2008). This practice is highly recommended for large projects (Paasivaraa et al., 2006) and can be borrowed for large teams.

One challenge that the company is facing is that the team members do not perceive the purpose of the Scrum meetings in the same way and this can be even more serious when the team is split in two locations.

Also, the daily Scrum meetings are not only intended for discussing what has been done “this session is primarily meant to improve the productivity and the effectiveness” (Vlaanderen et al., 2011: 60) of the team and to discuss a range of potential improvements. The team’s members are expected to share rather than to report during the daily Scrum meetings (Berczuck, 2007).

The main challenge for this company will remain to change the culture around the meetings and to educate both the Scrum masters and the team members to share common understanding and common values around those meetings.

It is essential that the team members embrace the same values around Scrum and a common understanding of how to work with it to reduce the confusion especially when working with separated teams (Berczuck, 2007).

Working in an environment that changes all the time can be frustrating, sharing common values and knowing that we have principles in common and around which we can come together if a problem occurs can be very comforting. Moreover, educating the team members to the method can be very constructive, it will support them in their process to approach the method and tame it.

Also, the teams work in a very tense flow of workload and series of sprints. Getting into this spiral can be fatal, it is important to take the time to rest and think about the executed work and how it was done, and to take the time to go for trainings or take some courses as well.
5.2 Implications for research

This research can be considered as an extension to all the previous studies reflecting upon the challenges related to Scrum. It confirms previous studies that support the possibility of using Scrum in large teams or in an international context.

Taking the case of the studied company, large teams is a problem only when it comes to meetings management it shows that by making the necessary means and resources available, the size of the team becomes one detail among many others.

The main result of this study shows that adopting Scrum is an ongoing process that never ends. It is a method easy to understand but still very broad and flexible enough to leave a large space for interpretation. Moreover, there are no standards or guidelines discussing how to adopt Scrum successfully (Boehm et al., 2005). This can be positive as it allows the companies to adapt the method to better fit their needs and the expectations of their clients but on the other hand they will have to determine how to use Scrum by their own.

Adopting Scrum is also problematic because it requires new roles with new competency profiles freshly and not sufficiently addressed. How to put the right people in the roles corresponding to them best? How to select a Scrum master for instance, or what makes a Scrum master better than another one?

This is perhaps where the greatest challenge in adopting Scrum lies. It is a method that has been largely discussed for years now. However it remains very general and has always zones of shades that still require answers.

5.3 Implications for the practice

Adopting Scrum is very attractive to companies because it offers simple practices, self-managed teams, evolutionary requirements management, and a high capacity of adaptation. However, many companies get stuck in the transition process. They never succeed adopting the method.

The way in which Scrum is described by the literature plays a significant role in the fact that it seems easy to adopt. A large and growing body of literature praises the easiness of the practices without highlighting the difficulties related to have these practices implemented and how facing challenges is natural and constant when adopting Scrum.

Therefore, managers do not have all the elements to consider how they want to implement Scrum with their teams, and to reflect upon the goals and the principles they want to carry out during the transition process.

Changing the traditional practices, believes and creating new roles is primary to adopt Scrum, but it is not enough. The fact that this method has many unexplored areas and that it is very large getting closer to a set of guidelines than to an actual set of procedures to follow in a more or less systematic way makes that the concern is not just in the decision to adopt Scrum in a particular way. It is about building a common understanding of the use of Scrum at every level of the company.

Educating team members to the method and make sure that they share common principles around it. Having principles that everyone understands and adheres to is an effective way to be prepared to face the challenges related to the adoption of Scrum on a daily basis.
6. Conclusion

Scrum is a project management method above all which means that there is no line that can be crossed to fully adopt it. Facing challenges is a normal part of adopting it, at least because it is about managing and dealing with humans. What can be done is only to reduce the impact of these challenges on the success of the projects.

The main challenge in adopting Scrum is how to use Scrum. So far there is only a definition of the process and the practices around it but there is no description about how it should be applied concretely in the companies or any recognized process or standards reference for best practices.

Or maybe because the philosophy behind adopting Scrum advocates that it must be used to best serve the customer and to develop a working software and that is all what matters. There are no effective and non-effective ways of adopting Scrum so far, and if it had to become a standard method applied in a regulated way it may lose all the freedom and the lightness it brings to software development projects.

The issue of projects’ suitability to Scrum is largely addressed, and even though there is a growing body of literature talking about adopting Scrum in international projects and in larger teams the rule remain that Scrum is suitable for small local projects with teams that contain between seven and ten people and that all work in the same place.

The case of the studied company is completely at the opposite of this rule. They manage large projects with a considerable complexity. Projects that last for years with stakeholders in different parts of the world and with teams including between 20 and 30 people and despite all this they are still able to deliver with quality and on time.

They do not face more or different challenges than the companies that use Scrum for local projects involving small teams as described by the literature. This case is a great example to start thinking about what if adopting Scrum or not is more a matter of resources than a matter of kind of projects.
7. References


Appendix A: Interview guide

This study aims to increase the understanding around the challenges related to the projects’ development process and technology while using the Scrum method. The objective is to define the challenges you are facing and possibly identify ways to deal with them in the future.

We will talk to you about your work routine and the problems you may be facing. This discussion will be organized in 4 themes: your personal background, your daily work, how do you use the Scrum method, and the challenges you face during the process or with the technology used. If you feel uncomfortable with any question we can rephrase it or skip it.

All the interviews will be conducted anonymously and all the information collected will be used for research purposes only. The interviews will be recorded, if you don’t feel comfortable with this please let us know.

Questions:

Theme 1: Personal background.

1. What is your role in the company?
2. How long have you been working with this company?
4. What kind of background do you have?
5. What are your previous work experiences?
6. How long have you been working with the scrum method?
7. Which method have you been working with before?

Theme 2: Daily work.

1. Can you describe a typical working day? (Working time, day activity)
2. Do you usually work alone or in pair/team? (Programming)
3. How big are the teams you are working in/with?
4. Is your work in contact with the clients or within the company exclusively?
5. The projects you are working with are for the customers or are internal projects?
6. How many meetings you have to attend, how long do they take, how productive you think they are?
7. What do you find more challenging in your work day? And what can be improved?
8. Can you describe something that you find exciting about your role in the project?
Theme 3: Specific using of scrum as a method.

1. Do you think scrum is suitable for all the projects? Why?
2. Is it difficult for you to find clients for the scrum projects?
3. What does the Scrum method adds to your work/projects?
4. Who are involved in a Scrum team? Can you describe the roles?
5. Which members of the scrum team are you most in contact with? [Why? What do you do together?]
6. What kind of projects you are handling? Are they short or long term projects?
7. Are you usually able to deliver the project on time? If not, how long is the average delay?

Theme 4: Technology and process.

1. Can you describe the current project development process?
2. How satisfied are you with this process?
3. In your opinion how your clients are satisfied with this process? Why?
4. Which of those phases you find more challenging and why?
   - Defining product backlog/ sprint backlog;
   - Execution of sprints;
   - Testing and delivery or rework;
   - Organizing effective meetings;
   - The technology used.
5. What challenges do you face, in relation to the process (procedure) of scrum in your projects?
6. What challenges do you face, in relation to the technology (tools and techniques) of scrum in your projects?
7. Have you ever faced issues with the client in relation to the process and technology? If yes, what are they and how you dealt with them?
8. How the client’s involvement in the process does affect your work?
9. As you said you are facing challenges related process and technology, besides all these challenges why do you think scrum is suitable for these projects?
10. If you can change one thing in this process what would it be? Why?

Research findings:
If you want us to share with you the outcome of this research please leave your e-mail (it will be used for this purpose only).