Blending Agile Scrum & Offshore Outsourcing
Software Development

Adolfo Sanabria Laporte
4116488
Master Thesis

Management of Technology
Master Thesis

Master Program: Management of Technology

ICT Section

Graduation Committee
Chairman: Dr. Yao-Hua Tan
First Supervisor: Dr. Harry Bouwman
Second Supervisor: Dr. Ir. Wolter Lemstra
External Supervisor: Dr. Albert Plugge

Author:
Adolfo Sanabria Laporte
Student ID: 4116488

September 7, 2012
Acknowledgement

To God, my family and friends, without all your support I would not be here.

Furthermore, I would like to express my sincere and deep gratitude to Dr. Harry Bouwman, Dr. Ir. Wolter Lemstra and Dr. Albert Plugge. Thank you for sharing your valuable experience with me.

Additionally, I would like to acknowledge MICIT-CONICIT – Costa Rican institutions awarding my scholarship – and all the Costa Ricans who through their taxes made this lifetime experience come true. Thank you all for believing that education that is the only tool to bring progress in a country.

Adolfo Sanabria Laporte
September, 2012.
Abstract

In today’s highly competitive environment, it becomes increasingly common for many companies to face internal and external pressure for delivering their products and services with the lowest cost and/or fastest time possible. Information Technology Outsourcing (ITO) has helped to release part of such pressure by enabling companies to gain cost advantage and obtain access to qualified labor pools (Manning, Massini, & Lewin, 2008). Not too far from this topic is Agile Software Methodologies, which came into place by promoting continuous releases and close customer participation along the software development process. Agile Scrum has been defined as the most mature and widely adopted method from all the Agile methodologies (Hossain, Babar, & Paik, 2009). Further investigations suggest that Agile Scrum, in collocated development, can increase productivity up to 5-10 times higher than the industry average and empirical evidence proposes that non-collocated teams can reach the same performance (Sutherland, Schoonheim, & Rijk, 2008).

Blending Offshoring Outsourcing projects with Agile Scrum, seem to be the perfect combination for many companies. However, companies continue facing difficulties in reaching such promised results and the mechanisms behind the ability of Agile Offshoring Outsourcing to shorten time-to-market remain hazy. Investigations explaining why certain projects cannot reach such promising results are scarce. This problem context led us to the main research question: What criteria hinder the reduction of time-to-market in software development projects that combine Agile Scrum with Offshoring Outsourcing? To answer it, we first researched what is Agile Scrum and how software development takes place under this methodology. Our research revealed that Agile Scrum is conformed by three main actors: ScrumMaster, Product Owner, Development Team. Additionally, literature suggests that Scrum demands high levels of interaction, openness and flexibility among all its participants. The second step in our research was to understand on how Offshore Outsource project are executed through their most common problems. Literatures addressed us into five main topics: 1) Culture Differences – subdivided into National Culture and Organizational Culture, 2) Coordination, 3) Trust, 4) Time Zone Differences and 5) Effective Communication. Based on these theoretical concepts we built a Conceptual Model containing twelve propositions describing what criteria influence the reduction of time-to-market. Each proposition is associated to Scrum concepts and how they could stimulate or hinder the reduction of time-to-market based on the context where the project takes place.

Following, we conducted a multiple case study considering a Belgium telecommunication company (the Client) and an Indian outsourcing service provider (the Vendor). The unit of analysis: projects blending Outsourcing Offshoring and Agile Scrum. The Case Study analyzed two projects: 1) Red Project. The project suffered major delays on the original schedule, thus the Vendor decided to introduce Agile Scrum to reduce such project gap. The project delivered part of its functionality and later it was halted. It is considered a major
failure in their Client-Vendor relationship. 2) Blue Project. Second project in timeline. Agile Scrum practices were also introduced after the project was already started through a Vendor initiative. The project experiences neither major delays nor exceptional performance, however Client-Vendor frictions were found in the meantime.

The refined Conceptual Model is described in Figure 6-7, page 78. Our findings can be summarized as: 1) The Identification of criteria influencing the reduction of time-to-market: Literature deductions supported by empirical evidence suggest the existence of at six criteria influencing the reduction of time-to-market. Such criteria is expressed in six prepositions the mentioned Conceptual Model (P1,P3,P5,P7,P9,11). 2) Implicit evolution of the Agile Scrum methodology. No scientific evidence was found tracking the implicit simplification of Scrum as a methodology, confirming the gap between practitioners and scientists with regards Agile practices.

Additionally, based on the theoretical findings and empirical experience from the Case Study, we derived a set of recommendations to Clients and Vendors aiming to reduce time-to-market while conducting this type of projects. For further research, we recommend the following topics: 1) Complement the current research quantitative research techniques and also conduct case studies in other industries besides telecommunication 2) Investigate the most suitable contract agreements in Agile Scrum Offshore Outsource Projects. 3) The execution of Agile Scrum Offshore Outsource Projects involving multiple organizations and 4) Research on why Agile Scrum certifications are largely concentrated in western countries.

Keywords: Agile Scrum, Offshoring Outsourcing Software, time-to-market, qualitative research, multiple case-study
# Table of Contents

Acknowledgement .................................................................................................................................. 4  
Abstract....................................................................................................................................................... 5  
Table of Contents ..................................................................................................................................... 7  
List of Illustrations.................................................................................................................................. 9  

## 1 Introduction.................................................................................................................................... 10  
### 1.1 Problem Context ................................................................................................................... 11  
#### 1.1.1 Agile Methodologies ................................................................................................................... 11  
#### 1.1.2 Offshore Outsourcing Software Development ................................................................. 13  
### 1.2 Problem Definition ............................................................................................................... 15  
### 1.3 Scope and Objective............................................................................................................. 16  
#### 1.3.1 Scope ........................................................................................................................................... 16  
#### 1.3.2 Research Objective ............................................................................................................. 18  
### 1.4 Research Questions.............................................................................................................. 18  
### 1.5 Research Relevance............................................................................................................. 19  
#### 1.5.1 Scientific Relevance .................................................................................................................... 19  
#### 1.5.2 Practitioner Relevance .............................................................................................................. 19  
### 1.6 Research Approach.............................................................................................................. 19  

## 2 Agile Scrum...................................................................................................................................... 21  
### 2.1 Initial Framework .................................................................................................................... 21  
### 2.2 Updated Methodology ......................................................................................................... 22  
#### 2.2.1 Roles and Responsibilities ....................................................................................................... 23  
#### 2.2.2 Artifacts ............................................................................................................................................ 25  
#### 2.2.3 Events ............................................................................................................................................... 27  
### 2.3 Conclusions............................................................................................................................. 29  

## 3 Offshore Outsourcing Software Development.................................................................... 32  
### 3.1 Cultural Differences............................................................................................................. 32  
### 3.2 Coordination........................................................................................................................... 35  
### 3.3 Time zone differences......................................................................................................... 37  
### 3.4 Trust .......................................................................................................................................... 38  
### 3.5 Effective Communication................................................................................................... 40  
### 3.6 Conclusions............................................................................................................................. 41  

## 4 Conceptual Model.......................................................................................................................... 45  
### 4.1.1 Agile Scrum and the reduction of time-to-market ................................................................. 45  
### 4.1.2 Offshore Outsourcing projects and the reduction of time-to-market .............................. 46
5 Case Study Outline .......................................................................................................................... 53
5.1 Case Study Research .................................................................................................................. 53
5.2 Case Study Selection .................................................................................................................. 53
5.3 Case Study Protocol ................................................................................................................... 54
  5.3.1 Case Study overview ............................................................................................................ 54
  5.3.1.1 The Client ..................................................................................................................... 54
  5.3.1.2 The Vendor .................................................................................................................. 55
  5.3.1.3 Project Description ........................................................................................................ 56
  5.3.2 Field Procedure ................................................................................................................... 58
  5.3.3 Interview Format .................................................................................................................. 59
5.4 Data Collection Process ............................................................................................................. 60
6 Case Study Findings ......................................................................................................................... 62
  6.1 Exploration and Specification .................................................................................................... 62
  6.2 Reduction .................................................................................................................................. 63
  6.3 Integration .................................................................................................................................. 67
    6.3.1 Culture Differences .......................................................................................................... 67
    6.3.2 Coordination .................................................................................................................... 69
    6.3.3 Time Zone Differences ...................................................................................................... 71
    6.3.4 Trust ................................................................................................................................... 72
    6.3.5 Effective Communication .................................................................................................. 74
  6.4 Conceptual Model Evaluation .................................................................................................... 76
6.5 Conclusion .................................................................................................................................... 79
7 Conclusions ...................................................................................................................................... 80
  7.1 Findings ..................................................................................................................................... 81
  7.2 Recommendations ..................................................................................................................... 82
  7.3 Reflection ................................................................................................................................... 84
  7.4 Limitations and Further research .............................................................................................. 84
    7.4.1 Research Limitations ......................................................................................................... 85
    7.4.2 Further Research ............................................................................................................... 85
8 References ....................................................................................................................................... 87
9 Appendices ...................................................................................................................................... 93
  9.1 Agile Principles ......................................................................................................................... 93
  9.2 Letter for Interview Request ..................................................................................................... 94
  9.3 Interview Reviews ...................................................................................................................... 95
    9.3.1 Red Project ....................................................................................................................... 95
    9.3.2 Blue Project ...................................................................................................................... 96
  9.4 Transcript Sample ...................................................................................................................... 97
  9.5 Specification Coding Phase ....................................................................................................... 104
    9.5.1 Codes from Vendor's Employees ....................................................................................... 104
    9.5.2 Codes from Client's Employees ......................................................................................... 106
List of Illustrations

List of Figures

Figure 1-1. Top 5 country distribution of Agile Scrum Certifications (Alliance, 2012) ........................................ 13
Figure 1-2. DSD models. Originally from (Robinson & Kalakota, 2004) ............................................................ 14
Figure 1-3. Areas of research (Prikладники, Audy, Damian, & de Oliveira, 2007) ........................................... 16
Figure 1-4. The Information Systems outsourcing life cycle. Source: (Chou & Chou, 2009) ................ 17
Figure 1-5. Research approach ...................................................................................................................... 20
Figure 2-1. Scrum Development Process. (Schwaber, 1995) ........................................................................... 22
Figure 2-2. The Scrum Roles (Wiener, 2006) .................................................................................................... 25
Figure 2-3. Relations between roles and artifacts ......................................................................................... 26
Figure 2-4. Joint view of roles, events and artifacts (Clinton, 2010) .............................................................. 29
Figure 2-5. Agile Scrum evolution ................................................................................................................... 30
Figure 3-1. Classification of coordination mechanisms and its typology (Sabherwal, 2003) ...................... 36
Figure 3-2. Time zone effects on speed and accuracy .................................................................................. 38
Figure 3-3. Trust, structure, performance virtuous-vicious cycle. Source (Sabherwal, 1999) .................... 40
Figure 3-4. Allen curve (Thomas & Günter, 2007) .......................................................................................... 41
Figure 4-1. Influence of Agile Scrum over reduction of time-to-market ......................................................... 46
Figure 4-2. Offshore outsource criteria influencing time-to-market .............................................................. 49
Figure 4-3. Conceptual model of criteria influencing reduction of time-to-market .................................. 50
Figure 5-1. Red project evolution .................................................................................................................. 57
Figure 6-1. Concepts related to culture differences ..................................................................................... 64
Figure 6-2. Concepts related to coordination ................................................................................................. 65
Figure 6-3. Concepts related to time zone difference .................................................................................... 65
Figure 6-4. Concepts related to trust ............................................................................................................. 66
Figure 6-5. Concepts related to effective communication ............................................................................ 67
Figure 6-6. Client and vendor’s ruling coordination mechanisms, based on coordination typology from Sabherwal (2003) and Scrum’s concepts ....................................................................................... 70
Figure 6-7. Refined conceptual model of criteria hindering reduction of time-to-market ......................... 78

List of Tables

Table 1-1. Main Agile development methods. Modified from: (Dyba & Dingsøyr, 2008) .................... 12
Table 4-1. Propositions describing the reduction of time-to-market ......................................................... 52
Table 5-1. Interview topic list ....................................................................................................................... 60
Chapter 1. Introduction

Corporations, in their perpetual journey pursuing competitive advantage and cost reduction, have crossed the frontiers of their organizations, aiming to find more efficient and effective paths to develop their Information Systems. The first Information Technology Outsource agreement was registered in 1989, when Kodak decided to subcontract IBM and DCE to manage its data centre (Loh & Venkatraman, 1992). Initially, the main reason behind subcontracting companies used to be labour cost reduction, but this has changed and new drivers have emerged. Surveys to company leaders have revealed that organizations are also attracted by the option to access qualified personnel, enhance efficiency, respond to market pressures (Lewin, Perm-Ajchariyawong, Sappenfield, & Aird, 2010), as well as improve their time-to-market (Moore & Barnett, 2004).

Information Technology Outsourcing (ITO) has evolved through time, and two well-defined periods have been registered. The first one (1990-2000) characterized by technology solution providers and the second (2000-2010) with more mature suppliers and service-oriented agreements (Plugge, 2011). Recent studies have predicted that the Worldwide spending for IT Outsourcing services in 2012 will be 2.1 percent higher than in 2011, reaching an annual expenditure of $251.7 billion (Gartner, 2012).

Similarly to ITO, Software development frameworks have been under constant transformation. In the last fifty years, different software development methodologies have emerged aiming to improve how Information Systems are built. Plan-driven methodologies like Waterfall, Spiral and Prototyping; have been the most used techniques to develop software. However, a set of methodologies self-called Agile, have arrived claiming to be more suitable for highly changing environments. The current thesis brings together two relevant topics for clients and vendors that are aiming to join a cross-nation voyage while using Offshore Outsourcing Software Development and Agile methodologies.
1.1 Problem Context
The context of this thesis takes place in the convergence of Agile Methodologies with Offshore Outsourcing Software Development. The first step to take is to present the landscape where this research will occur and analyse the main topics discussed between both areas.

1.1.1 Agile Methodologies
The initial traces of the Agile Methodologies dates from 1950s when projects from the U.S. Department of Defence and NASA used Iterative and Incremental Design and Development (IIDD) in internal projects (Sliger & Broderick, 2008 p. 11). Later, in 1986 Takeuchi and Nonaka, published a paper called 'The New New Product Development Game' that provided the foundations of Scrum, one of the most diffused Agile approaches. They concluded that companies were considering two additional elements besides high quality, low cost and differentiation. Those elements were 1) speed and 2) flexibility. They used the analogy of a strategy used in rugby scrum teams, where the ball is controlled and passed by different players but the team acts as a single unit or block (Sliger & Broderick, 2008 p. 11). Their approach suggested the inclusion of six elements: built-in instability, self-organizing project teams, overlapping development phases, ‘multilearning’, subtle control, and organizational transfer of learning (Takeuchi & Nonaka, 1986). Despite these investigations, the industry mainly followed Traditional Methodologies (e.g. Waterfall and Spiral). Such techniques are based on assembly line production processes and other areas of knowledge outside software development; characterized by highly structured phases with strong emphasis towards documentation (Bednar & Robertson, 2012; Danait, 2005). Decades after the establishment of Traditional Methodologies, experts are still facing problems to foresee changes and understand requirements in long-term projects. Larman (2004 p. 3) explained that software development, in contrast with predictable manufacturing processes, rarely registers up-front unchanging requirements.

Agile Methodologies have captured the attention of practitioners and researchers because, among other returns, they seem to be suitable to cope with the perpetual problem of software development: changing requirements. Agile Methodologies promote continuous releases and customer participation (Fritzsche & Keil, 2007). International corporations like Yahoo! and Google, have included Agile Methodologies into their software projects, registering positive results in their projects (Benefield, 2008; Striebeck, 2006). Even though each Agile Methodology differs in mechanisms and strategies to develop software, their support towards the Agile principles remains.

Table 1-1 provides an overview the most important Agile Methodologies.

---

1 Approach to build systems based on several iterations, resulting on multiple deliverables (Sliger & Broderick, 2008 p. 11).

**Blending Agile Scrum & Offshore Outsourcing Software Development**
<table>
<thead>
<tr>
<th>Agile Method</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal Method</td>
<td>Teams are organized based on their sizes and task importance using the following classification: Clear, Yellow, Orange, Red, Blue. The Clear team will be focused on non-life critical tasks.</td>
</tr>
<tr>
<td>Dynamic Software Development Method (DSDM)</td>
<td>The project is divided in three phases: pre-project, project life cycle, and post project. It incorporates nine principles towards team empowerment, testing and communication with the client, among others.</td>
</tr>
<tr>
<td>Feature-Driven Development</td>
<td>It combines model-driven and Agile software development. One iteration consists of 2 phases: design and development.</td>
</tr>
<tr>
<td>Lean Software Development</td>
<td>It is built around seven core principles: Eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole</td>
</tr>
<tr>
<td>Scrum</td>
<td>It consists on Empirical process control mechanisms using the iterative cycles –called Sprints – and a Backlog (list of items to develop every Sprint). There is a Scrum Master that coordinates the team’s progress and backlog.</td>
</tr>
<tr>
<td>Extreme Programming (XP)</td>
<td>It promotes practices such as: Continuous system integration, pair programming and refactoring.</td>
</tr>
</tbody>
</table>

Table 1-1. Main Agile Development Methods. Modified from: (Dyba & Dingsøyr, 2008)

From the mentioned methodologies, Scrum has been recognized as one of the most promising and widely supported Agile Methodology (Hossain, et al., 2009). During the last decade, different organizations have emerged aiming to secure and certify Scrum practitioners. Some of the most important organizations are: the Scrum Alliance, Scrum.org and Scrum Foundation. Just between 2008-2011, the Scrum Alliance registered a steep increase of approximately 400% in one of their certification programs, reaching almost in 160000 worldwide certified members and confirming the increasing interest of practitioners to know more about this methodology (Scrum Alliance, 2011). At a country level, top5 countries with the largest number of Agile Scrum Certifications issued by the Scrum Alliance are: U.S.A, India, U.K, Germany and Finland (See Figure 1-1).
Nevertheless, Agile Methodologies are not free of problems and some of most cited critics are:

- Currently, there is limited scientific evidence supporting the Agile benefits claimed by practitioners. (Dyba & Dingsøyr, 2008).
- Agile Methodologies has little emphasis towards documentation and this can cause a company memory loss effect (Turk, France, & Rumpe, 2002).
- Possible incompatibility with other IT frameworks (e.g. Capability Maturity Model Integration framework) (Fritzsche & Keil, 2007).

### 1.1.2 Offshore Outsourcing Software Development

Sourcing and other related terms are usually confused and misused (Manning, et al., 2008). Therefore it is important to clarify the meaning of Offshore Outsourcing that will be used along this document. Oshri et al (2009, p 2) defines it as:

> “Sourcing is the act through which work is contracted or delegated to an external or internal entity that could be physically located anywhere. Sourcing encompasses various in-sourcing and outsourcing arrangements such as offshore outsourcing, captive offshoring, nearshoring and onshoring”.

**Offshoring** refers to tasks and business functions, which are executed across national borders. **Outsourcing** is aligned to deliver products or services through external providers of the firm placed either domestically (onshore) or abroad (offshore) (Manning, et al., 2008). Figure 1-2 supports this definition by illustrating the different configurations of Distributed Software Development (DSD) modes.
Prompted by a single objective: to cut labour costs, Information Technology (IT) Outsourcing has evolved through time revealing many other benefits and challenges for all its parties. Nowadays, service suppliers from all over the world are providing highly skilled talent to Clients that are more appealed to cross the frontiers of their homelands. Countries like India and China have become technology poles, giving the option to outsource not only standardized IT processes, but also more complex tasks such as: Research and Development (R&D), Product Development and Product Design (Manning, et al., 2008).

Some positive elements related to Offshore Outsourcing Software Development are: the option to have access to qualified labor pools and gain cost advantage (Manning, et al., 2008). The increment of cultural and geographical distance can also be beneficial to organizations because companies could access new know-how ideas and decrease the time-to-market (Carmel & Agarwal, 2001).

On the other hand, some of the most cited risks and drawbacks Outsourcing Offshore Development presents are: cultural barriers, time zone mismatches (Krishna, Sundeep, & Geoff, 2004), conflict of interests, opportunistic behaviour (Gopal, Sivaramakrishnan, Krishnan, & Mukhopadhyay, 2003) and poor service quality (Lewin & Peeters, 2006). Non-collocated development settings demand high levels of trust because presence controls are reduced; forcing team leaders and managers to adopt new management mechanisms. Also, physical absence forces teams to rely more on technological means to maintain their communication. Aspects like: knowledge transfer, codification and documentation between the different locations; become vital for all the parties (Dedrick, Carmel, & Kraemer, 2009).
Contract development and contract compliance are also highly discussed topics in Outsource Offshore contexts. Contracts are used as control mechanisms to avoid strategic behaviour and define the actors’ responsibilities, however they are not faultless mechanisms. Incomplete Contract Theory (Grossman & Hart, 1986) explains that asymmetric and limited information do not allow to create all-inclusive contracts. In Outsource Software Development projects, this weakness can be amplified if Agile methodologies are implemented. Outsourcing requires detailed contracts, however Agile Methodologies fundamentally embrace change making requirements a moving target hard to capture in contracts (Shahir, Daneshpajouh, & Ramsin, 2008).

1.2 Problem Definition

Merging Agile Methodologies and Offshore Outsourcing Software Development seem incompatible (Paasivaara & Lassenius, 2006). On one hand Agile Methodologies encourage face-to-face communication, close interaction with the stakeholders and flexible agreements (Agile Alliance, 2001b). On the other hand, Offshore Outsourcing Software Development is typically characterized by detailed documentation, geographic distance and formal agreements. Not being discouraged by this initial mismatch of principles, clients and vendors have started blending both practices aiming to maximize benefits and minimize undesired effects.

Agile Methodologies are presented to many companies as the ‘secret sauce’ to excel. Mah (2008), conducted a research comparing 26 Agile projects from 5 different companies to a database of 7500 traditional projects. He concluded that Agile projects had a 37% faster time-to-market compared to traditional projects. Furthermore, research supporting non-collocated Agile projects, suggests that Agile Methodologies can help to solve problems usually found in offshoring settings like: cultural incompatibilities, leadership struggle and the lack of trust (Phalnikar, Deshpande, & Joshi, 2008). Investigations suggest that constant and open communication, typically present in Agile methodologies, can help Offshored Software development projects to build trustful environments (Paasivaara & Lassenius, 2006). In the specific case of Agile Scrum, investigations suggest that Scrum collocated development can increase productivity up to 5-10 times higher than the industry average and empirical evidence suggest that non-collocated teams can reach the same performance (Sutherland, et al., 2008). All these arguments can largely encourage companies to transform their software development processes and pursue a time-to-market reduction together with low development costs.

Regardless such prominent scenario, practitioners and researchers continue registering difficulties to fully reach all benefits associated with Agile Methodologies. Projects involving multiple organizations can become very complex to manage because every party can have a different perception of the events and actions to follow. Customers can see occurrences as problems while Vendors can see them as normal events (Michell & Fitzgerald, 1997). This can lead to severe misunderstandings and incompatibilities between organizations. The complexity of these environments can increase when one of the parties has no experience using Agile Methodologies.
The mechanisms behind Agile Offshoring Outsourcing leading to short time-to-market remain hazy. Sutherland et al. (2008), concluded that teams can reach outstanding results in terms of quality and velocity in fully distributed Agile Scrum teams and become “Hyperproductive Offshored Development Teams”. Investigations explaining why certain projects cannot reach such promising results are scarce. Companies, aiming to reach such promising can make large investments while training their employees, improving their Information Technology infrastructures and even staring new partnerships with Agile ‘experts’. If Clients and Vendors do not succeed blending Agile Scrum with Offshore Outsourcing settings, their client-vendor relationships and the project status can be compromised; leading to a revenue decline and a plausible lost of customers.

Large part of the existing literature related to Agile Global Software Engineering is presented as experience reports and few scientific research is available (Jalali & Wohlin, 2010). Previous investigations have highlighted the need and importance to continue researching Agile practices in non-collocated environments (Paasivaara & Lassenius, 2006; Ramesh, Cao, Mohan, & Xu, 2006).

1.3 Scope and Objective
As we have previously shown, Offshore Outsourcing and Agile Methodologies are very complex and wide topics. Therefore, we need to clearly define the scope of this thesis, as well as the objective we want to achieve.

1.3.1 Scope
IT Outsourcing projects can cover many activities, therefore we need to narrow the specific configuration of IT Outsourcing projects we are aiming to research. On Figure 1-3, demarked with light blue, we disclose the specific configuration this will cover.

Figure 1-3. Areas of research (Prikladnicki, Audy, Damian, & de Oliveira, 2007).
Regarding the specific Agile Methodology we aim to research, we must say that time constrains, workload and resource availability have forced us to focus only on one Agile Methodology. We have chosen Agile Scrum based on the following criteria:

- With an increasing number of managers willing to use Agile Scrum, it is considered the most widely adopted agile software methodology (Hossain, et al., 2009).
- Agile Scrum Certification programs issued by Scrum Alliance, provide us a relatively stable framework to manoeuvre and make deductions. In most of the remaining Agile methodologies there is no official body of knowledge that could guide researchers or practitioners while conducting investigations.
- Agile Scrum has been designed to rapidly drive new product to market (Sutherland, 2005).

Chou et al (2009), researched the life cycle of Outsourced Information System projects and they identified three main phases that characterize this type of projects. Such phases are: 1) Pre-Contract Phase, 2) Contract Phase and 3) Post-Contract Phase. This research will focus on the second phase, specifically when the project is executed. Agile Scrum is mainly focused on the development phase itself, rather than pre or post phases of the outsourcing projects. Figure 1-4 illustrates the Outsourcing Life Cycle described by Chou and collaborators, and highlighted with light blue is our main interest.

![Figure 1-4. The Information Systems outsourcing life cycle. Source: (Chou & Chou, 2009)](image-url)
1.3.2 Research Objective
Based on previous context, the research objective that will drive this master thesis is:

To understand the mechanisms that may influence the reduction of time-to-market in software developing projects combining Agile Scrum and Offshoring Outsourcing.

1.4 Research Questions
Considering the research context and an initial literature review, the following research question emerged:

Q.1: What criteria influence the reduction of time-to-market in software development projects that combine Agile Scrum with Offshoring Outsourcing?

To answer our main question, first we should describe the Agile Scrum Methodology and have a clear understanding on how software development takes place. Due to Agile's preference towards individuals and customers (Agile Alliance, 2001a), we should look at the different actors involved in Scrum and understand how they collaborate. Furthermore, we also need to research Scrum's main processes and principles in order to comprehend the implications of conducting an Agile Scrum project. The previous aspects addressed us to state the first sub-question as follows:

Q.1.1. What are the core principles, actors, responsibilities and processes behind Agile Scrum?

The second sub question aims to cover the topic of Offshoring Outsourcing Software Development. In order to reduce time-to-market, we need to detect and understand the issues that mostly affect projects in these settings. This led us to the second sub question:

Q.1.2. What are the main challenges faced in Offshoring Outsourcing Software Development?

Based on the findings from sub questions Q.1.1 and Q.1.2, we will build a Conceptual Model revealing which elements could affect time-to-market and map them to the Agile Scrum methodology. Subsequently, we will conduct a Case Study to refine our initial findings.

After building a Conceptual Model from theoretical grounds and refine it with empirical evidence, the concluding question of our research refers to recommendations towards clients and vendors on how to improve reduction of time-to-market in this type of projects.

Q.1.3. What recommendations can be made to clients and vendors aiming to reduce time-to-market while conducting Agile Scrum Offshore Outsource Projects?
1.5 Research Relevance

1.5.1 Scientific Relevance
Several researchers have stressed the need and opportunity to conduct further investigations related to projects Outsourcing Offshoring (da Silva, Costa, Franca, & Prikladinicki, 2010; Paasivaara & Lassenius, 2006; Ramesh, et al., 2006). Jalali et al. (2010) conducted a systematic literature review including Agile researches from 1999 up to 2009. They concluded that most of the literature is formed by industrial experience reports and they encouraged researchers to develop further investigations combining scientific literature with empirical evidence. The benefits of collocated and non-collocated Agile software development have been topics led by practitioners and not by scientific researchers. The academic community has much to investigate and learn from the supposed benefits that these methodologies can bring to its adopters (Smite, Moe, & Ågerfalk, 2010 p.7).

Confronting the core values behind Agile Scrum with the main challenges present in Offshoring Outsourcing settings can bring a new perspective on how the project synergy can be improved; reducing time-to-market and creating better Client-Vendor relationships.

1.5.2 Practitioner Relevance
Agile Scrum Offshore projects are settings that present many issues and having scientific evidence to clarify how to deal with these situations can be very useful for company leaders. Nowadays, clients have a large set of options to coordinate and implement their software solutions. Vendors and clients would benefit by having a deeper understanding of the implications and configurations needed to successfully implement Agile Scrum in Offshore Outsourced environments. Coordinating distributed teams is not a trivial task, and if they are not effectively managed transaction costs, development time and quality can be compromised (Sakthivel, 2005). One of the main benefits company leaders (clients and vendors) can obtain from this research is to have scientific evidence supporting or discarding actions from practitioners attempting to guide them in this type of settings.

1.6 Research Approach
The objective of this section is to provide a schematic view of how this research has been planned. Figure 1-5 fulfills this objective and it can be read as follows: The research starts in Chapter 1 with the problem definition, research scope and research questions. The Research question Q.1.1. (Related to Scrum Concepts) is investigated in Chapter 2. The research question Q.1.2, which is associated to Offshore Outsource challenges) will be covered in Chapter 3. Combining the results of Chapter 2 and 3 will result into a Conceptual Model built with theoretical grounds disclosing the criteria influencing time-to-market in Offshore Outsource software development projects using Agile Scrum.

The next step is to conduct a Case Study to confront empirical evidence with our Conceptual Model. In Chapter 5 we will describe the case Study selection process and the Case Study Protocol. The Case Study will consider both perspectives (Client and Vendor), and its unit of
analysis will be: projects blending Outsourcing Offshoring and Agile Scrum. In Chapter 6 we will analyze the data collected during the Case Study. Such analysis will help us to evaluate the Conceptual Model built in Chapter 4. The last step of this research is takes place in Chapter 7. This Chapter will contain a set of Conclusions and Recommendations based on the theoretical and empirical findings of this research.

Figure 1-5. Research Approach.
Chapter 2. Agile Scrum

This chapter will reveal the central concepts and processes in Agile Scrum. The main goal to achieve is to answer the first sub-question:

**Q.1.1. What are the core principles, actors, responsibilities and processes behind Agile Scrum?**

In 1995, Jeff Sutherland and Ken Schwaber created Scrum, founding their work in two publications: "The New New Product Development Game" from Takeuchi and Nonaka, and a report from Borland Corporation describing a hyper-productive project (Larman, 2004 p. 135). The convergence of the Scrum framework with the Agile practices occurred in 2001 when a group of representatives of different software practices, including Schwaber and Sutherland, joined forces to build the Agile Manifesto. This declaration is considered the backbone of all Agile Methodologies and it is conformed by twelve principles (see Appendix 9.1) reviewed in four core values (Agile Alliance, 2001a):

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 Agile Scrum framework is founded in an empirical process control theory, which promotes to acquire knowledge through experience and making decisions based on what is known. Its conceptual pillars are (Schwaber & Sutherland, 2011):

- **Transparency.** Promote visible processes through a common language for all the observers
- **Inspection.** Frequent inspections on the framework’s processes and artefacts to minimize undesired outcomes. Skilled team members are requested to perform these inspections
- **Adaption.** The inspections, used as evaluation tools, will establish whether the project’s results are acceptable or not and decide any possible adjustment to fulfil the goals

### 2.1 Initial Framework

Agile Scrum grounded its processes in the argument that the Analysis, Design and Development phases in software development are unpredictable. In response, the initial Scrum methodology suggested three phases: Pregame, Game and Postgame (see Figure 2-1). These phases were constructed to deliver better results in terms of flexibility, responsiveness, and reliability (Schwaber, 1995).
Pre-Game: It consists of two sub phases: a) Planning and b) Architecture/High Level Design. The first phase establishes a shared vision and sets expectations among the stakeholders (e.g. initial cost and schedule estimations). Also, in this phase are defined the infrastructure and development tools needed for the development of the project. An initial Product Backlog is built and product packages are mapped to backlog items. At phase b), the Product Backlog\(^2\) is reviewed to define the changes required to implement it. Moreover, an initial view of the systems architecture emerges and the Scrum Team is focused to refine requirements and identify problems that will block the Development phase.

Game: The Development phase is an iterative cycle consisting of one or more Sprints (development cycles). Every Sprint is aimed to last no more than one month and it contains a Sprint planning meeting, daily Scrum meetings and a Sprint Review meeting. All these activities are designed to result in a working deliverable (Larman, 2004).

Post-Game: This is the closing phase of the project. It is reserved for operational deployment activities, trainings and development of formal documentation. (Larman, 2004).

### 2.2 Updated Methodology

In early 2000’s, Agile Scrum gained popularity, but at the same time there was a need to formalize the body of knowledge because practitioners were showing misunderstandings in the core concepts behind the methodology. In 2004, Ken Schwaber, Esther Derby and Mike Cohn founded the Scrum Alliance; making the first step towards the standardization and certification of the Agile Scrum Methodology. However in 2009, due to differences on the

---

\(^2\) Product Backlog is a prioritized items list of tasks to develop related to the project. Further in this chapter it is explained in more detail.
organization’s goal (knowledge driven vs. money driven), Schwaber decided to resign the Scrum Alliance and start a new organization called Scrum.org (Schwaber, 2010).

Since the creation of the Scrum Alliance and later Scrum.org, the Agile Scrum Methodology has been under inspection and refinement. The most relevant change lies on the redefinition of the three phases described on the previous section (Pregame, Game, Postgame). Both organizations deprecated these concepts even in the latest edition of the Agile Scrum guidebook these concepts are absent. (Schwaber & Sutherland, 2011; Scrum Alliance, 2012). The framework was simplified and currently it emphasizes the concept of Sprint and the elements that surrounds it: Roles, Events and Artifacts. Following we will explain each of these elements.

2.2.1 Roles and Responsibilities
In the Agile Scrum methodology, the members with direct responsibility and ownership on the Scrum process are referred as Pigs (the Product Owner, ScrumMaster and Development Team). Outsiders of the project are called Chickens (e.g. Marketing representatives, clients, etc.). In the following section each of these concepts are described:

**Product Owner:** The Product Owner is a role that bonds authority and responsibility elements from roles that were traditionally spread in other positions (e.g. Project Manager, Product Manager) (Pichler, 2010). He is responsible for the following aspects (Cohn, 2010; Deemer, Benefield, Larman, & Vodde, 2010):

- Define the correct goal that the team should pursue
- Provide a common project vision
- Interact with the external stakeholders
- Define and prioritize the features in the backlog
- Maximize the products’ Return of Investment (ROI) while selecting the highest-business-value items with lowest-cost
- Define boundaries (e.g. cost, time)

The role of Product Owner demands constant and effective communication with the stakeholders to follow market trends and guarantee the product’s quality. He must be the bridging element between the rest of the Scrum Team and the project external actors (Chickens). Pichler (2010) underlined that finding the correct Product Owner is essential for any Scrum project and he must have the following traits:

- Visionary and Doer
- Leader and Team Player
- Communicator and Negotiator
- Empowered and Committed
- Available and Qualified

Additionally, the Product Owner must be aware of its role as mediator with project outsiders. Advisors and committees could influence on the product’s features, but it must be clear that no change should occur without the Product Owner’s approval.
**ScrumMaster**: He is a facilitator and tutor for the Development Team and Product Owner. His responsibilities are summarized into (Woodward & Surdek, 2010 p. 2):

- Guide the team towards the goal established by the Product Owner
- Guard the Scrum’s values, processes and rules
- Avoid any interference from the project outsiders
- Chair the Scrum Events

It is important to stress that the ScrumMaster is not a Project Manager (Deemer, et al., 2010). His authority is confined within the project’s processes. If the team fails reaching a goal (e.g. delivering a software component not meeting the expected quality), the ScrumMaster can suggest tactics to avoid this problem in the future (e.g. code revisions before the deployment), but the decision whether this solution will be executed or not, will completely rely on the Development Team. The team’s task execution is out of the ScrumMaster responsibilities and imposing solutions crosses his jurisdiction (Cohn, 2010 p. 118). LaFasto and Larson (2001) have suggested that leaders with technical knowledge have more potential to help teams to find suitable solutions for the project’s problems. However not always locating technical leader as ScrumMasters is the best decision because regularly technical people lack of interpersonal skills. Also, ScrumMasters could be tempted to cross their responsibilities and impose directions and teams could create a dependency towards such behavior (Cohn, 2010 p. 120 - 122).

**Development Team**: It consists of a group of cross-functional professionals committed to achieve the goal set by the Product Owner and secured by the ScrumMaster. Every member will be entitled as ‘developer’ and no specific titles are allowed. Developers can have areas of expertise, but the development’s accountability corresponds to the team as a whole. Between three to nine members are recommended to form the team, thus small productivity returns or coordination overheads are avoided (Schwaber & Sutherland, 2011).

Figure 2-2 illustrates the relationships and roles previously mentioned.
2.2.2 Artifacts

The methodology provides a set of artifacts to its participants to coordinate actions inside and outside of the Scrum team.

**Product Backlog:** It is important to stress that the Product Backlog is not a requirement list. A requirement list expresses a fixed vision of what the client wants. In contrast, the Product Backlog is a prioritized dynamic list affected by the constant feedback generated from every development cycle. In Agile Scrum, seeing the requirements as the exact need from the client is considered a mistake, thus every backlog item is the initial point to further discussion and refinement (Cohn, 2010 p. 236). The Backlog items can represent system defects (bugs), enhancements, new features and tasks to perform; and it is the Product Owner’s responsibility to keep it updated and prioritized.

**Sprint Backlog:** It is a subset of the Product Backlog and it is the main input for the Sprint (see Events). During the Sprint period the ScrumMaster must guard the Sprint Backlog from any outsider that could alter its content. The Development Team is the only entity allowed to sort and change how tasks will be performed. The Sprint Backlog is conformed by User Stories describing the desired feature to perform next to a delivery plan. Cohn suggested that User Stories are typically kept in sticky notes or software tools to make them visual to the all the team members and he recommended to follow two simple formats (2010 p. 239):

- **As a <type of user>, I want <some goal> so that <some reason>**
- **In order to <achieve value>, as <type of user>, I want <some goal>**
By using any of these formats, it is expected that Pigs will develop a sense of ownership over the task to develop and the team members will better understand the impact of the request over the project (Cohn, 2008).

**Increment**: An Increment is the sum of all the Sprint Backlog items completed and with a ‘Done’ status. The Scrum Team decides when an item is marked as ‘Done’ but the Product Owner chooses whether to release the Increment or not. Defining what does ‘Done’ mean, is very important because it will influence the quality of items delivered, as well as the common understanding and transparency of the team (Schwaber & Sutherland, 2011).

![Figure 2-3. Relations between Roles and Artifacts](image)

Figure 2-3 illustrates how the Agile Scrum actors use the mentioned artifacts. Starting from the Product Backlog, which it was build by the Product Owner, the Scrum Team agrees on the elements conforming the Scrum Backlog that will be guarded by the ScrumMaster. Following, the Development Team works on every User Story, finishing on an Increment that will be deployed if the Product Owner agrees. The Sprint Definition process will be explained in the following section.
2.2.3 Events

**Sprint**: It is a time-boxed cycle lasting no more than four weeks. During this period changes on the sprint elements are not allowed and the ScrumMaster is responsible to guard them. It should be designed to meet one of the Agile Manifesto principles: ‘working software over documentation’ (Agile Alliance, 2001a); meaning that the end of this period an Increment is achieved and functional software can be delivered. Concluding every Sprint with functional components is very important because: a) it creates feedback to the team b) it helps on the time estimations and c) it enables early shipments to the Product Owner (Cohn, 2010 p. 258). The Sprint is conformed by the Sprint Planning Meeting, Daily Scrum, the Development Work, the Sprint Review and the Sprint Retrospective (Schwaber & Sutherland, 2011).

**Sprint Planning Meeting**: This meeting lasts no more than eight hours for a four weeks Sprint and it occurs just before the start of the next iteration (Sprint). In the first half of the meeting, the Product Owner explains to the rest of the Scrum Team the prioritized items that conform the current Product Backlog. The team will be focused on answering *What* will be done this Sprint, having a preference towards high-priority tasks. To reach consensus four assets are used: a) Prioritized Product Backlog b) Latest product Increment c) Projected Development Team capacity during the Sprint d) Previous development performance (Larman, 2004; Schwaber & Sutherland, 2011).

The second half of the meeting is focused to answer *How* the Sprint goal will be achieved and which actions must be taken to deploy the Increment (e.g. deployment plan). The Development Team can decompose every task to perform in traceable time units of days or hours and facilitate self-organization (Schwaber & Sutherland, 2011).

The Sprint Planning Meeting encourages commitment and ownership towards the project because based on the Development Team’s deep knowledge, they decide how to tackle the tasks on the Product Backlog instead of just following commands. The Product Owner can lobby to include small tasks in the Sprint Backlog if the workload allows it, but in general creating the Sprint Backlog is an agreement between the Development Team, making this meeting one of the key practices in Agile Scrum (Deemer, et al., 2010).

Agile Scrum does not promote any specific tactic to prioritize nor estimate tasks development (Deemer, et al., 2010) however a common technique suggested by practitioners (Cohn, 2010 p. 297) is to create Story Points to each Backlog item and represent the level of difficulty or hours needed to conclude the task. Therefore the Scrum Team can assign a ‘Velocity’ to their development capacity and tune-up its estimations after every Sprint. The Story Points can be assigned through a Agile XP technique to plan and weigh tasks: Planning Poker. It consists on first a round of discussion explaining the User Story and later every team member would vote using cards marked numbers (typically 1,2,3,5,7,10). If the team agrees on the number, then it will be the User Story’s weight, if not, more discussions will follow until agreement is reached (Grenning, 2002).
**Daily Scrum:** The Daily Scrum meeting or also known as *Stand-Up meeting*, should not last more than 15 minutes and it occurs every working day. It is designed to answer three fundamental questions: *What we accomplished since the last meeting? What we will do today? What obstacle do we have?* The ScrumMaster can influence the course of the meeting, the Development Team is the one directly responsible. Every Development member should update the rest of the Scrum Team on the progress accomplished (if any) and transfer it to the Sprint Backlog.

Literature (Deemer, et al., 2010; Larman, 2004 p. 125) suggests the usage of graphs and Burndown charts to track the progress and remaining work. Schewaber *et al.* (2011) stresses that these techniques can be useful to forecast progress in known activities, but they should not replace the importance empiricism, because specially in complex and unknown environments what will occur is uncertain. Additionally, when multiple Scrum projects need to be synchronized, a *Scrum of scrum meeting* is commonly used. In this meeting, representatives from each Scrum team gather in order to discuss the overlapping areas and integration processes. The Scrum of Scrum meeting has three differences compared a daily scrum meetings: 1) It should not occur every day. 2) It should not be time-boxed to 15 minutes and 3) They are meant to solve problems (Cohn, 2010 p. 340-342).

**Sprint Review:** It occurs at the end of the Sprint and it lasts four hours for a one-month Sprint. The Development team presents to the ScrumMaster and the Product Owner the Sprint Backlog elements with ‘Done’ status and clarifies any relevant issue that affected the process and how they were tackled. In response the Product Owner provides an overview of the Product Backlog and expected dates of completion for the remaining tasks. (Schwaber & Sutherland, 2011).

**Sprint Retrospective:** It is a three-hours meeting for one-month Sprints and it takes place after the Sprint Review. During the meeting, the Scrum team analyzes the completed Sprint discussing what went well and what went wrong, in terms of: people, relationships, processes and tools. It is a learning moment for the whole team and the final output is an improvement plan to be applied for the coming Sprints.

Figure 2-4 describes the general overview of how a Sprint cycle takes place including Artifacts, Events and Roles.
During the first Sprint iterations, Scrum Teams typically register high failure rates meeting their initial commitments taken at the beginning of the Sprint. Nevertheless, after three or four cycles, teams start tuning up their Velocity and assessments become more accurate. Having a fixed Sprint length is recommended in order to refine the team's working pace (Deemer, et al., 2010).

Cancelling a Sprint is uncommon, but it can occur if the intended goal turns invalid. As a consequence, Scrum Team must verify the status of the Sprint Backlog items and evaluate which of the ‘Done’ elements can be shipped and which ones should be moved back to the Product Backlog. Team members must be aware that aborting a Sprint consumes resources and it can be traumatic for the team (Schwaber & Sutherland, 2011).

2.3 Conclusions
Agile Scrum is not a recent software methodology, but in the last decade it has become a trend in the software industry (Larman, 2004 p. 135; Scrum Alliance, 2011). Since its introduction, the framework has remained relatively stable, however a simplification of the original methodology (Schwaber, 1995) resulted in the current version thought by the Scrum Alliance and Scrum.org (organizations lead by Scrum’s authors and initial promoters). We consider relevant to warn future researchers and record such simplification because we encountered incongruities between scientific publications (Dullemond, van Gameren, & van Solingen, 2009) and what nowadays practitioners are thought to be ‘Scrum’
(Scrum Alliance, 2012). Figure 2-5 summarizes the changes from the original framework and the current emphasis towards the Sprint.

Having said that, we can return to the question that drove this chapter: *What are the core principles, actors, responsibilities and processes behind Agile Scrum?* Following we summarize our findings:

**Principles.** Agile Scrum’s core principles are formed by the general Agile Principles and the Scrum’s Conceptual Pillars. The first group are meta-principles shared by any methodology claiming to be ‘Agile’ and they are summarized into: (P1) Individuals over processes, (P2) Working software over comprehensive documentation, (P3) Customer collaboration over contract negotiation, (P4) Responding to change over following a plan. The second group of principles promotes (P5) Transparency, (P6) Inspection and (P7) Adaption.

**Actors and Responsibilities.** The framework consists on three main actors. A1) The Product Owner, who is the bridging element between the project outsiders and the rest of the Scrum Team. A2) The ScrumMaster. He is a process manager not a project manager, meaning that he is not responsible whether the project goals were reached or not; instead he is accountable for the Scrum processes and its execution. A3) The Development Team is a multidisciplinary team in charge of assessing and developing the Product Backlog items.

Connecting these findings with our interest towards the reduction of time-to-market, we conclude that Agile Scrum has the potential to help organizations to reach the market in a shorter time. Environments with highly changing requirements can be managed by Scrum’s frequent iterations containing valuable increments. However, organizations must be also aware that Agile Scrum is not free of problems and the reduction of time-to-market is not guaranteed. Project members can resist adopting Scrum’s software practices, because such practices can be contrasting with more classical software development methodologies (e.g. Waterfall). Organizations must be aware that Scrum's potential to improve time-to-market largely depends on an intensive and effective collaboration between Pigs (Scrum Team members) and Chickens (Project outsiders). Conducting an Agile Scrum project requires more than just defining frequent delivery dates and creating new roles, but it truly demands a mindset change in large part of the organization.
Chapter 3. Offshore Outsource Software Development

The purpose of this chapter is to answer the second sub question of this research:

**Q.1.2. What are the main challenges faced in Offshoring Outsourcing Software Development projects?**

Offshore Outsourcing Software Development is a trend that has registered a rapid adoption. Lewin *et al.* (2006) identified three factors that have boosted this phenomenon: 1) An increase in the demand of services to fix faulty software and a cost reduction trend, triggered by the Millennium Bug (Y2K) and the burst of the dotcom bubble. 2) Investments in education and trainings from Governments and companies in developing countries. 3) The improvements in Information and Communication Technology (ICT) and infrastructure in the host countries. However, building team forming teams cross-nation teams has presented important challenges for the service providers and clients.

Da Silva, *et al.* (2010), conducted a systematic literature review searching for publications on Distributed Software Development (DSD) published between 1999 and 2009. They concluded that previous researches have been mainly focused on five issues that emerge while software is developed with non-collocated teams. They identified five main challenges: 1) Cultural Differences, 2) Coordination, 4) Time Zone Differences, 4) Trust and 5) Effective Communication. Also, they highlighted that previous researches have been mainly focused on overcoming communication issues, but the remaining four challenges equally demand attention from researchers. Rather than focusing in one specific challenge, we decided to focus on the group, because we want to provide a joint perspective where this type of projects lay and not a challenge-specific view as it has been previously done in literature.

Following we will describe each of these challenges, but before doing this, it is important to clarify that we aim to disclose the core concepts behind each challenge and why they are considered challenges in Offshore Outsourcing projects. More in-depth knowledge about every topic (e.g. tactics to overcome each challenge) is out of the scope of this chapter. The decomposition of every challenge was done through a compilation of multiple academic sources mainly collected from Journal publications, Conference publications and books related to the research field.

### 3.1 Cultural Differences

Cross-culture Software Development is not trouble free (Krishna, *et al.*, 2004). Environments formed by people with different values, attitudes, and goals are complex to manage and project members must be aware of these implications. Hofstede *et al.* (2005 p.4) describe ‘Culture’ as a collective phenomenon formed by unwritten rules that
distinguishes one group from another. Every person belongs to multiple (sub) cultures due to personal experiences and social interactions. Carmel and Argawal (2001) suggested that Culture, in a Global Software Development context, can be observed with two lenses: the national culture and the organizational culture.

**National Culture:** It refers to values, norms and language associated to a group of people within the boundaries of a nation. It is transferred from generation to generation, shaping aspects ranging from body language and gestures to decision-making norms and business etiquette (Carmel & Tjia, 2005 p. 176). A research conducted by Yoshii and Higa (2010) can be used to illustrate how National Culture can affect software processes. They researched how software implementation in Japan, presents a mismatch between the original software methodology aimed in projects and the actual practices followed. Projects claiming to follow the waterfall methodology were actually using an agile-like implementation. Their explanation of such phenomenon was that Japan’s successful history in the manufacturing industry and a strong market-driven stance permeated into the software industry, creating a mirage of projects trying to follow waterfall methodologies, but actually implementing Agile practices.

In practice, National Cultures in cross-nation teams can bring incompatibilities and frictions between team members due to a clash of norms and values. Hofstede’s framework (1980), can be used to disclose the large spectrum of culture configurations that employees can be part of. For example, building a cross-nation software team with a young Dutch female leading the project and the rest of the team conformed by Chinese males can bring conflicts related to communication and power hierarchy. Dutch culture typically accepts direct communication, and gender is considered irrelevant in decision-making processes. On the other hand, in Chinese culture direct confrontation is traditionally avoided and gender could play a role in job positions that are usually held by male employees. Many of these conflicts can be explained through Hofstede’s framework, where dimensions like *Masculinity versus Femininity* and *Power Distance* plot The Netherlands and China in opposite positions (Robbins & Judge, 2010 p. 53). Furthermore, uncertainty (a property usually present in software products) can influence on how requirements are managed. Societies with high Uncertainty Avoidance favor detailed procedures and highly structured work environments (Ang, Van Dyne, & Begley, 2003); contrasting with societies with low-uncertainty avoidance that prefer loose specifications to increase innovation and creativity (Ang & Inkpen, 2008).

Next to values and norms, language is another important elements to consider within National Culture. Challenges faced by cross-nation teams go beyond just prescribing a common language. In many situations, social interaction demands the understanding of body language and language connotation, which is hardly supported by technological means. For example, Indians are generally reluctant to say ‘no’ and tell bad news. Instead they prefer to give a ‘wobbly’ yes, creating misunderstandings on agreements (Carmel & Tjia, 2005 p. 181).
Furthermore, setting a common language (e.g. English) does not necessarily imply mutual understanding. Many misinterpretations have been traced to verbal and nonverbal communications where each party have a different understanding of the same statement (Carmel & Tjia, 2005 p. 182). A Spanish speaking person can create confusion while stating the following phrase: “I have a doubt about your implementation”; meaning that he or she has a question about the implementation. In Spanish ‘doubt’ and ‘question’ are synonymous (duda and pregunta). Nevertheless, a native English speaker could infer a feeling of hesitation or even mistrust.

**Organizational Culture:** Cabrera et al (2001) describes Organizational Culture as a set of social norms that implicitly determine the appropriate and inappropriate behaviours in an organization. Hofstede et al (1990) researched Organizational Cultures in Dutch and Danish companies using qualitative and quantities research methods, identifying six dimensions used to define the culture of an organization. Following each dimension is described (Cabrera, et al., 2001; Hofstede & Hofstede, 2005):

- **Process oriented vs. Results oriented.** It refers when organizations are more focused on either the procedures that must take place or in the final outputs. Process oriented organizations are typically bureaucratic organizations rich in procedures and rules. In contrast, Results oriented organizations are more risk-taker and error-tolerant organizations.

- **Employee oriented vs. Job oriented.** This dimension describes when an organization is more focused on the employee’s well-being or if it has a higher interest to have the work concluded.

- **Parochial vs. Professional.** Members from parochial cultures consider that their personal and professional background were considered while they were hired and that the organization’s norms are aligned with their personal’s norms. Professional cultures rely merely on professional capabilities.

- **Open system vs. Closed system.** This dimension explains how information flows in the organization. In open system cultures, organizations are more open to share information and in closed systems cultures tend to remain more hermetic to outsiders.

- **Loose control vs. Tight control.** In this case, the level of internal structuring is analysed. ‘Loose control’ organizations are flexible about personal treatments (e.g. jokes are culturally common). On the other hand, tight control organizations are strict with respect concepts such as: punctuality, meetings, budgets, etc.

- **Normative vs. Pragmatic.** The last dimension refers how the organization perceives and acts upon the customer. Normative organizations relate its actions towards the outside world, as the ‘correct’ procedures and unbreakable rules. Pragmatic organizations are open to ad-hoc solutions and highly market-driven.

Nevertheless, why is Organizational Culture relevant to Offshore Outsourcing Software Development projects? Managers must be aware that bringing two organizations next to each other to develop a software product can bring incompatibilities and culture clashes.
due to intrinsic values in the organizations. For example, a client with strong process documentation policies (Tight Control) and reluctant to collaborate with external bodies (Closed System) can encounter difficulties to work with outsourced teams. In a development phase like Requirement Definition, Organizational Culture can even have a stronger influence in functional groups than National Culture because every company can infer and translate requirements differently (Damian, 2007). Organizations with domestic mindsets (conservative cultures with little experience managing remote teams) present mistrust to foreigners and problems to work with offshore vendors (Carmel & Agarwal, 2002). It should be clear that Organizational Culture can be shaped (Robbins & Judge, 2010 p. 264), but it requires effort (time and capital). “Organizational Culture is developed through the years and it is deeply grounded in the employee’s values” (Robbins & Judge, 2010 p. 255).

3.2 Coordination

Coordination has become a vital interest for companies aspiring to remain competitive in today’s business environment. Ranging from daily meetings with team members, up to defining requirements to long-term projects; companies are obliged to synchronize their activities performed by their employees and external parties. Coordination was defined by Malone et al. (1994), as the act to manage dependencies among activities involving multiple actors. In a software development context, Coordination involves individuals working in a common project, having a shared vision of the expected output and sharing information. In practice, team members are requested to coordinate their tasks to avoid work redundancy and pursue expedite deliveries (Kraut & Streeter, 1995), and usually these can be achieved by sharing design specifications and progress updates (Kraut & Streeter, 1995).

In Offshore Outsource Software Development environments Coordination turns into a two-dimension phenomenon due to the need of having client (external) coordination and vendor’s team (internal) coordination. Gopal, et al (2011) researched how Coordination and Performance take place in Global Software Development, and they concluded that external and internal Coordination positively affects software quality, but also overheads can emerge and reduce the development speed.

In addition, Coordination is also challenged by the physical absence that characterizes outsourced projects because distance can negatively affect task synchronization (Carmel & Agarwal, 2001). This can be explained through the fact that software development projects demand ‘frequent small adjustments’ that can emerge from informal and spontaneous communication, situations that are limited in non-collocated settings (Carmel & Tjia, 2005 p. 152). Therefore, achieving Coordination over distance is fundamental in this type of settings (Herbsleb, 2007). Sabherwal (2003) researched the Coordination phenomenon in outsourced Software Development projects and based on previous researches, he deducted four types of Coordination.

- **Coordination by standards:** It is planned in advance (a priori) and it relies on explicit rules. Some of the coordination mechanisms used are: compatibility
standards, data dictionaries, design rules, error tracking procedures and modification request procedures.

- **Coordination by plans**: It is also planned in an early stage of the project, but it is focused on specific goals to achieve. Milestones, delivery schedules, sign-offs and test plans, are its main coordination mechanisms.

- **Coordination by formal mutual adjustment**: Based on conjoint consensus, this type of coordination takes place during the project execution. Typical mechanisms to coordinate are: code inspections, coordination committees, design review meetings, hierarchies, liaison roles, reporting requirements and status review meetings.

- **Coordination by informal mutual adjustment**: Informal agreements are the main difference with previous types of Coordination. Some of the main mechanisms used are: co-location, spontaneous communications, informal meetings, joint development, and transition teams.

The first two types of Coordination (standards and plans) are expected to have a high impact on the project’s fixed costs. Both of them demand extensive planning and deep problem understanding to coordinate every aspect a priori. As a side effect, project’s variable costs are expected to be low. On the other hand, the last two Coordination types (formal and informal) are expected to have the opposite effect: low effect on project’s fixed cost, but high effect on the variable costs. Figure 3-1 depicts the mentioned concepts.

At first glance selecting a type of Coordination could seem very straightforward. Actors could define their main interests and analyze the project’s characteristics and choose the most suitable type of coordination for a specific circumstance, however in practice this is more complex. In Client-Vendor relationships, each party can have different perspectives of what should be the result of the coordination events. Clients are typically interested on
effectively manage the arrangement with the vendor, securing that the project’s business and technical requirements are met. On the other hand, Vendors aim to develop software efficiently (quality and time) thus that the vendor’s payoff is also protected (Gopal, et al., 2011). Clients and vendors must be aware that such mechanisms can be deeply intrinsic in the organization’s DNA. Aligning both organizations demands mutual agreements from both organizations, and it can lead to resistance to change among the team members. Fabriek et al (2008) investigated the relation of coordination mechanisms in successful software projects (achieving expected scope, time, budget and quality). They concluded that informal and formal mutual adjustment are the most important coordination mechanisms for successful projects; and the least important were A priori coordination mechanisms (Planning and Following Standards).

3.3 Time zone differences
Time zones are regions that define the time standard based on the meridians. The Earth is divided into 24 standard time zones of one-hour difference each and subsequently divided into portions of 15, 30 or 45 minutes. Countries are not homogeneously distributed along these regions (e.g. USA crosses 4 time zones considering its continental territory and 10 time zones including the Antarctic bases and Hawaii; but China is related to only one time zone). These time zones are used as time references but it does not imply that every society perceives time in the same way. Defining meal schedules, holidays, working days and religious celebrations can vary significantly between cultures or countries. For example: in Israel the weekend is on Friday and Saturday; the Chinese New Year is between January and February; and in Spain, the workday usually starts in the late morning finishing after 18:00. Humans are social creatures used to interact during the day and sleep at night, therefore switching to a single time zone is hardly possible.

In Offshoring Software Development projects, having time zone differences between teams is almost a given. Espinosa et al. (2003) explained that time separation is more than just time zone differences. They argue that distance separation is symmetric – e.g. distance (X, Y) = distance (Y, X) – but time separation is asymmetric and it directly shapes team interactions. In fully overlapped time zone projects, vendors and clients can clarify requirements, resolve problems and communicate in the most efficient and effective way avoiding rework, but in environments with multiple time zone this become a challenge to overcome.

Time separation is a phenomenon that cannot be removed from outsourced operations. Teams must find the most suitable configurations and practices to reduce negative effects and maximize benefits from their collaborations. Tradeoffs seem inevitable in this type of projects and actors must learn how to deal with this phenomenon. Espinosa et al (2007) conducted a laboratory study including forty-two dyadic teams researching the impact of time zone differences on team’s performance. They researched how quality (e.g. accuracy) and production speed were affected by different time separation configurations (full, 2/3, 1/3 time overlap and no overlap at all). Figure 3-2 depicts their findings.
They registered high production speed rates when teams were in both extremes of the graph (full time overlap, no time overlap). In the intermediate configurations (2/3 and 1/3) speed was considerably diminished. They relate this behavior to coordination tasks needed to synchronize teams. When teams are in both extremes of the curve, coordination tasks are scarcer than in the other two situations, allowing teams to work on their tasks and make their own interpretations. The other tradeoff refers to the output quality, specifically accuracy, suggesting that with small time differences accuracy is hardly affected, however while time overlap is reduced, accuracy is negatively affected. They think that synchronization windows used to reduce misunderstandings can explain this effect. When the overlap time is reduced, opportunities to clarify misunderstandings also reduced and therefore accuracy is reduced.

Previous researches have been focused on overcoming physical distance, and with today’s technology it can be said that distance is dead, but time is not (Carmel & Espinosa, 2012). Teams must be aware of the implications and complexity of working within multiple time zone environments. Every project must be revised and understand its priorities because teams can confront severe tradeoffs affecting the project’s goals and team interactions (Espinosa, et al., 2007).

3.4 Trust
Trust is considered a critical element to build and maintain client-vendor relationships in outsourced projects (Alibabar, Verner, & Nguyen, 2007; Oza, Hall, Rainer, & Grey, 2004), although it is a complex phenomenon to explain. Carmel et al (2005 p. 153) defined Trust as: “the comfort to make yourself vulnerable ... while others act in your place”. Furthermore, a more elaborated definition was given by Oza et al. (2006): ‘Clients and vendors having positive expectations of each other’s actions, while having a rational interest in maintaining
that relationship in the awareness of the risk in those expectations'. Both perspectives imply a two-actor phenomenon, but the latter highlights the existence of interests that connects them. Sabherwal (1999) explained that in Outsource Software projects, there are two types of agreements compressing each actor’s interests:

- **Written Contracts**: These contracts depend on explicit structural controls to guarantee their execution. Legal agreements (e.g. Non-Disclosure Agreements and Service Level Agreements) are common mechanisms to define the actors’ responsibilities and expected outputs.

- **Psychological Contracts**: These are unwritten and informal contracts describing the perceived obligations that parties should meet.

In settings with offshore parties, both types of contracts are needed. Psychological contracts complement the explicit obligations stated in written contracts with implicit obligations perceived, helping to shape the actors’ behavior (Ang & Inkpen, 2008). However (written) contracts are never complete (Macneil, 1980) nor faultless and they can be a double-edged sword. From one side they can create the basis for trustful environments where mutual obligations are defined and reduce opportunistic behavior from any of the parties involved. On the other side, vendors can perceive them as unnecessary control mechanisms that would secure the client’s interests while increasing expenditures (Sabherwal, 2003). In the literature four types of trust in outsourced Information System development projects has been identified (Sabherwal, 1999):

- **Calculus-based**: In this case trust is built around control mechanisms to reduce opportunistic behavior. Rewards and penalties are incentives to collaborate and trust each other. Written contracts are one way to gain this type of trust.

- **Knowledge-based**: Trust is reached through mutual credibility built by previous client-vendor experiences (e.g. former projects, ability demonstrations, etc.). If both parties are collaborating for the first time, both can agree to conduct small scope projects to demonstrate each ones capabilities.

- **Identification-based**: This type of trust is built when both parties identify a common goal and they are boned around such goal (e.g. have common corporate or project objectives).

- **Performance-based**: It is based on reaching tangible goals while the project is conducted (e.g. meet deadlines, reach quality standards, etc.). This is considered one of the most important types of trust in non-collocated settings, because clients have a reduced perception of processes and task evolution, mainly measuring completeness and quality.

Building trustful relationships takes time, but project stakeholders must be aware of its importance. Mistrust is a risk for offshoring projects, because it compromises common goals and it can create frictions among the actors. “Trust comes on foot and leaves by horse” (Carmel & Tjia, 2005 p. 235). Building Trust and retain it represent a challenge for both parties. Human beings are social creatures and our sense of trust is strongly linked to face-to-face communication and proximity. In traditional collocated projects, supervisors had
the option to rely on presence control mechanisms to create and maintain trust. However, in non-collocated environments, distance (geographic and cultural) next to ineffective communication can create the perfect environment to develop mistrust between the actors. Mistrust on Outsource software projects have been associated to a decrease in productivity, quality and information exchange (Smite & Moe, 2007).

Sabherwal (1999), while researching outsource software development projects, registered a virtuous-vicious cycle involving Trust, structural controls and performance (see Figure 3-3). Even though, his research was limited to only the Vendor’s perspective, he encouraged both parties (clients and vendors) to find the correct balance between structure and trust. Any excessive focus in either structure or trust, can hamper performance and compromise the future of the project.

### 3.5 Effective Communication

Communication is associated to exchange or share information between different actors and it is considered successful when “it has been transmitted, received acknowledged, understood and acted upon” (Carmel & Tjia, 2005 p. 152). In non-collocated settings effective communication becomes a vital issue to overcome because geographical distance disturbs fluent internal (within the team) and external (with the client) communication. In the last decade this phenomenon has been extensively researched resulting in a large variety of ‘best practices’, tools and models to avoid problems and maximize the benefits of this type of settings (da Silva, et al., 2010). In collocated environments, internal communication takes place through formal and informal communication channels with fewer difficulties than in non-collocated environments. When people are physically close to each other, many agreements, tacit knowledge transfer and decision-making take place. The lack of proximity limits informal and spontaneous communication because people lose the possibility to join unplanned situations where they can exchange information and increase their project awareness (Carmel & Tjia, 2005 p. 150).

Having this in mind, investigations have concluded that (geographic) distance can be the main challenge for non-collocated development because it negatively impacts team

---

3 Also known as ‘water cooler’ communication
communication and as a consequence it reduces effective coordination (Carmel & Agarwal, 2001). Nevertheless, *When geographic distance can start playing a role in team interactions?* Thomas, *et al* (2007) researched the relationship between physical separation and communication frequency in innovative engineering organizations. They concluded that after 30 meters distance, team members would experience the same probability to communicate as if they would be separated by several kilometers. This phenomenon is referred as ‘Allen Curve’ (see Figure 3-4).

![Figure 3-4. Allen Curve (Thomas & Günter, 2007)](image)

Poor communication and miscommunication should be avoided because it can bring undesired effects to the clients and vendors compromising shared goals. Miscommunication can increase monitoring costs, complicate contracts and increase transaction costs (Gefen & Carmel, 2008).

### 3.6 Conclusions

This chapter has focused on answering the research question: *What are the main challenges faced in Offshoring Outsourcing Software Development projects?* To answer this question we used as a starting point a systematic literature review conducted by da Silva *et al.* (2010) that studied challenges present in Offshoring settings. They concluded that the top-five challenges traced between 1999-2009 were: Cultural Differences, Coordination, Time Zones differences, Trust and Effective Communication. Using a compilation of multiple academic resources, we focused on disclosing why these elements are considered issues for Offshore Outsource projects. Following we summarize the main findings and link them to the reduction of time-to-market:

**Cultural Differences.** Literature (Carmel & Agarwal, 2001) suggests that Culture problems emerge from two sources: 1) National Culture and 2) Organizational Culture. The first source refers to clashes or incompatibilities related to language, values and norms,
originated after forming teams with people from different countries or regions. The second source (Organizational Culture) explains that each organization has intrinsic social norms defining the appropriate and inappropriate behaviours within each organization (Cabrera, et al., 2001). Bringing team members from different organizations can create personal and organizational incompatibilities, resulting into team misunderstandings and compromising the reduction of time-to-market.

**Coordination.** In Offshore Outsource projects, coordination is an important problem to overcome because team members from each organization (Client and Vendor) must synchronize their activities in an efficient and effective manner, expecting not to hamper the software development quality nor limit the development speed (Gopal, et al., 2011). Literature (Sabherwal, 2003) has identified 4 types of coordination mechanisms from which clients and vendors can orchestrate their tasks while conducting a software development project: 1) Coordination by plans, 2) Coordination by standards 3) Coordination by formal mutual adjustment and 4) Coordination by informal mutual adjustment (mechanism 1 and 2 are achieved before the project initiation; and mechanism 3 and 4 after its initiation). Literature suggests that between *A priori* and *Ongoing* coordination mechanisms, the second group (informal and formal mutual adjustment) are the most important mechanisms for successful outsourced software development projects (Fabriek, et al., 2008).

**Time zone differences.** Literature researching the impact of time zone differences in non-collocated teams has registered trade-offs between speed and accuracy (e.g. quality) (Espinosa, et al., 2007). Considering four time separation configurations (full, 2/3, 1/3 time overlap and no overlap at all), teams have shown the best development speed when they are in either the first or last time configuration (*full time overlap* and *no overlap*). However, if high accuracy levels are desired, the best configuration appears to be *full time overlap*. This suggests that Client and Vendors aiming to reduce time-to-market are encouraged to work under *full time overlap* or *no overlap* configurations, because their development processes can be concluded in less time.

**Trust.** In non-collocated environments Trustful relationships are hindered by the lack of physical and social interaction. Literature researching how Trust is built and maintained in outsourced software development projects have identified four types of trust (Sabherwal, 1999): 1) Calculus-based (trust established through punishment and reward mechanisms), 2) Knowledge-based (actors relying on previous experiences), 3) Identification-based (trust is sealed by shared interests) and 4) Performance-based (trust is built around tangible goals such as milestones or quality standards). Building and maintaining trustful relationships, is very important in Outsource software projects, especially if actors aim to reduce time-to-market. Mistrust has been associated to a decrease in productivity, quality and information exchange (Smite & Moe, 2007). Clients and Vendors must find the correct balance between structural controls and trust (Sabherwal, 1999). Excessive focus on any of the two elements can hamper performance and compromise reaching the market on the expected time.

**Effective Communication.** Improving communication between non-collocated team members has been the most explored challenge in the last decade, resulting in a large
variety of 'best practices', models and tools (da Silva, et al., 2010). Synchronous and asynchronous tools have emerged to overcome geographic distance between team members, however the lack of proximity limits informal and spontaneous communication and knowledge sharing. Research (Thomas & Günter, 2007) has shown that physical separation further than 30 meters dramatically reduces communication frequency between team members. Effective communication in offshore outsource projects is essential in order to facilitate transparency, avoid misunderstandings and promote knowledge sharing. If Client and vendor aim to reduce time-to-market, they must guarantee frequent and effective communication between all the team members.

Through our research of Software Development projects with non-collocated members, it was clear the importance of Computer-Mediated Communication tools as key enabler for remote collaboration and its essential contribution to Outsourcing Offshoring projects. Computer-Mediated Communication (CMC) is defined as: “the electronic exchange of information using computer terminals joined together via communication links” (Sproull & Kiesler, 1986). Decades of research on this field have contributed to the development of theories (e.g. Media Richness Theory) and collaboration tools (e.g. emails, chats, videoconference). Besides being the bridging communication component between remote teams, other benefits have been discovered. Nowadays, team members have the possibility to access and store information in shared repositories, allowing them to create historical records for more precise analysis (possibility hardly existent in Face-To-Face communications) (Kerr & Murthy, 2009). Also it has been found that CMC tools can leverage hierarchies between participants and improve the project’s synergy (McQuail, 2000). More recently, Triana et al. (2011), researched if the order in which Face-to-Face (FTF) and CMC could affect the participation of team minorities. They concluded that in team interactions, using first CMC and followed by FTF communication, could help minority members to improve their (perceived) inclusion and participation. These results disclose the idea that CMC can be more than just the carrier to send and receive information, but also a positive team catalyst.

Nevertheless, CMC is not free of limitations and recurrent debates around its claimed benefits have been the center of discussion. Kerr et al. (2009) researched the response of FTF teams and Computer-Mediated teams while solving hidden-profile problems4. They concluded that comparing FTF teams with computer-mediated teams while using chat tools; the first group could be more effective exchanging and processing information and consequently resolving problems. Also they warned that group memory could be affected by the large amount of information generated in CMC systems. Participants can lose focus on other participants’ information by just focusing on recording their own information.

Furthermore, research has disclosed that the effect of CMC tools can be sensitive to the cultural settings. Tan et al (1998) studied the effect of majority influence in decision making processes, comparing team members from U.S.A. (Individualistic Culture5) and Singapore

---

4 Situation where relevant information is initially know only by one member
5 (Hofstede, 1980)
(Collectivism Culture). The experiments took place under three settings: unsupported, face-to-face CMC, and dispersed CMC. Their research brought outstanding conclusions: “The results showed that the impact of CMC on majority influence was contingent upon national culture”. They highlighted the absence of the Culture factor in previous CMC theories, and they stressed the importance of its inclusion in further CMC and Management theories. It must be clear that CMC effectiveness over organizations is more than just technology and its availability, but it demands organization and culture adjustment (Montoya, Massey, Hung, & Crisp, 2009). Also, the limitations to create a sense of identity and establishing an ethos among participants, typically present in CMC, may bring frictions and misunderstandings in intercultural online environments (St.Amant, 2002).

In response to these problems researchers have tried to overcome some of CMC’s difficulties while including social and context elements while using CMC solutions. Social Software tools (e.g. wikis, videoconference systems, blogs, etc.) are trying to improve the user’s experience, while considering aspects like: trust, culture and knowledge sharing in CMC tools (Abbattista, Calefato, Gendarmi, & Lanubile, 2008).
Chapter 4. Conceptual Model

The goal of this Chapter is to create a generic Conceptual Model around the main research question: **What criteria influence the reduction of time-to-market in software development projects that combine Agile Scrum with Offshoring Outsourcing?** Such goal will be achieved by combining the results of Chapter 2 and Chapter 3. The Conceptual Model will be build in three steps. The first step is to analyze which criteria from Agile Scrum influence the reduction of time-to-market (section 4.1.1). The second step is to analyze which concepts from Offshoring Outsourcing software settings influence the reduction of time-to-market (section 4.1.2). And the third step is to build the Conceptual Model from the combination of both practices, leading to a set of theoretical propositions that could explain the criteria influencing time-to-market (section 4.1.3).

### 4.1.1 Agile Scrum and the reduction of time-to-market

Agile Scrum, being a software methodology and not a project setting as Offshoring Outsourcing, suggests little about the specific project settings. However, we deducted Scrum key concepts, which are comparable and related to the Offshore Outsourcing Challenges being studied:

1) **Team norms and values (AS1).** Agile Scrum promotes Openness, Transparency, Inspection and Adaption, as some of its core values; inviting its practitioners to be proactive and self-driven. This group of norms and values result into a positive effect over the reduction of time-to-market because team members can tackle high-priority tasks depending on project's current status instead of strictly following ‘a plan’.

2) **Team Coordination (AS2).** Agile Scrum has important limitations with regards coordination. The methodology is suitable for small projects (less than nine Development members), but in larger projects teams can experience small productivity returns and coordination overheads (Schwaber & Sutherland, 2011). The reduction of time-to-market is not evident in large projects.

3) **Frequent Time-boxed events (AS3).** Scrum’s events are designed to create intensive interaction between the different actors in compressed periods of time. Each Event conforming the methodology (see section 2.2.3), is strategically located in the development process in order to exchange information and receive feedback from the different actors; delivering the tasks with highest priority in the shortest time possible.

4) **Trustful environment (AS4).** Agile Scrum stimulates group accountability by decentralizing responsibilities from the extinct figure of the Project Manager. In every Sprint, the methodology allows the Scrum team to define its own goal, creating a sense of group commitment towards a shared goal. The Development team, as group, can identify
free riders and the ScrumMaster is a process supervisor. Such balance between trustfulness and supervision benefits the reduction of time-to-market because teams are no suffocated with control mechanisms and at the same time, the group can detect underperforming team members.

5) **Team Communication (AS5).** The methodology promotes Face-to-Face Communication as the most effective and efficient form to share information between the different stakeholders. The reduction of time-to-market is positively influenced, because task clarifications and requirement changes can occur with greater fluency, compared to project using classical software develop methodologies.

Figure 4-1 summarizes the mentioned concepts:

![Figure 4-1. Influence of Agile Scrum over reduction of time-to-market.](image)

### 4.1.2 Offshore Outsourcing projects and the reduction of time-to-market

Deducing whether or not one of the mentioned Offshore Outsource challenges will influence time-to-market can be very complex. Software projects can lay in very specific contexts, limiting the generalization the effect of every Challenge over time-to-market. Therefore, we must review the theory of each Challenge and define which conditions could plausibly improve or hinder the reduction of time-to-market. For example, literatures suggest that development speed in software projects can either increase or decrease,
depending on the time zone configuration on which the project is executed (Espinosa, et al., 2007), therefore we need specific knowledge where the software project will take place in order to deduct a plausible effect from Time Zone Differences over time-to-market. Based on the literature researching each of the top-five challenges affecting Offshore Outsource software development projects, we deducted twelve plausible contexts that could help us to define which criteria hinder or facilitate the reduction of time-to-market. Following, we described each context and its plausible effect over the reduction of time-to-market.

1) **Culture Differences.** National Culture. The first Context (C1) refers to projects with conflicting National Culture (language, norms, and values). In this configuration, team members are likely to be exposed to misunderstandings with regards to requirements, communication styles and power hierarchies; resulting into frictions and plausible project delays. In contrast, projects conformed by Team members with compatible National Culture (Context 2) are expected to possess mutual understanding. Meaning that C1 is expected to hinder the reduction of time-to-market and C2 cause the opposite effect.

Organization Culture: Context 3 (C3) refers to projects involving organizations with conflicting Organizational Culture. Projects in this type of contexts can be affected by frictions between both organizations related to social norms defining ‘appropriate’ and ‘inappropriate’ behaviours in during the project execution. Such frictions are expected to hinder the reduction of time-to-market. In contrast, Context 4 (C4) refers to projects involving organizations with compatible Organizational Culture. Such projects, while not experiencing conflicts with regards Organizational Culture, are expected to have a better performance and ultimately create a positive effect in the reduction of time-to-market.

2) **Coordination.** Literature suggests that software projects using A priori Coordination mechanisms present more difficulties to succeed (reach goals in terms of time, scope, budget and quality) than projects using Ongoing Coordination mechanisms (Fabriek, et al., 2008) . These lead us to the next two contexts. Context 5 (C5): A priori coordination mechanisms plausibly hindering time-to-market. Context 6 (C6): Ongoing coordination mechanisms, improving the reduction of time-to-market.

3) **Time Zone Differences.** Considering four time separation configurations (full, 2/3, 1/3 time overlap and no overlap at all) we deducted the following contexts. Context 7 (C7): deducts that 2/3 and 1/3 time configurations (between 2.5h-5.5h time zone difference) hinder the reduction of time-to-market. Context 8 (C8), deducts that ‘full’ time overlap and ‘no overlap’ time improves time-to-market.

4) **Trust.** Literature investigating the effect of Trust in software development projects, suggest a relation between trustful environments, structural controls and team’s performance (quality and time) (Sabherwal, 1999). Context 9 (C9) deducts that projects with Over/Under structure controls creates stimulates mistrust, low performance and hinders the reduction of time-to-market. Context 10 (10) suggests the opposite effect: appropriate structure controls stimulate trustful environments, high performance and a positive effect in the reduction of time-to-market.
5) **Effective Communication.** Literature has highlighted the importance of physical separation among team members and its effect over communication frequency (Thomas & Günter, 2007). Additionally, reach use of Computer-Mediated Communication tools (CMC), can largely benefit team members to exchange information and facilitate problem resolutions (Kerr & Murthy, 2009). In Context 11 (C11) we deduct that physical separation among the team members next to poor use of CMC tools will negatively influence the reduction of time-to-market. In contrast, in Context 12 (C12) we deduct that physical proximity and reach use of CMC tools will positively influence the reduction of time-to-market.

Figure 4-2 illustrates the mentioned concepts.
4.1.3 Offshore Outsource Agile Scrum projects and the reduction of time-to-market

In Figure 4-3 we combine the previous two models, presenting the plausible criteria influencing the reduction of time-to-market in Agile Scrum Offshore Outsource software development projects.
Figure 4-3. Conceptual Model of criteria influencing Reduction of Time-to-market.
From the previous Conceptual Model, we derived twelve propositions. Following we explain each of them and their plausible effect over the reduction of time-to-market.

<table>
<thead>
<tr>
<th>Offshore Outsource Challenge</th>
<th>Proposition</th>
<th>Offshore Outsource Software Projects using Agile Scrum</th>
<th>Effect over Reduction of time-to-market</th>
</tr>
</thead>
<tbody>
<tr>
<td>Culture</td>
<td>P1</td>
<td>Teams with contrasting National Culture will encounter difficulties to adopt Scrum’s core values causing a neutral effect over the reduction of time-to-market.</td>
<td>Neutral</td>
</tr>
<tr>
<td></td>
<td>P2</td>
<td>Teams with similar National Culture will not encounter difficulties to adopt Scrum’s core values causing a large improvement in its time-to-market.</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>P3</td>
<td>Teams with contrasting National Culture will encounter difficulties to adopt Scrum’s core values causing a neutral effect over the reduction of time-to-market.</td>
<td>Neutral</td>
</tr>
<tr>
<td></td>
<td>P4</td>
<td>Organizations with similar Organizational Culture will encounter difficulties to adopt Scrum’s core values causing a large improvement in its time-to-market</td>
<td>++</td>
</tr>
<tr>
<td>Coordination</td>
<td>P5</td>
<td>Large Offshore Outsource Agile Scrum projects using A priori coordination mechanisms will largely deteriorate its time-to-market</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>P6</td>
<td>Large Offshore Outsource Agile Scrum projects using On going coordination mechanisms will have a neutral effect over time-to-market</td>
<td>Neutral</td>
</tr>
<tr>
<td>Time Zone</td>
<td>P7</td>
<td>Offshore Outsource Agile Scrum projects executed with time differences between 2.5h-5.5h will have a neutral effect over time-to-market</td>
<td>Neutral</td>
</tr>
<tr>
<td></td>
<td>P8</td>
<td>Offshore Outsource Agile Scrum projects executed with no time overlap or in the same time zone will largely improve its time-to-market</td>
<td>++</td>
</tr>
<tr>
<td>Trust</td>
<td>Proposition Description</td>
<td>Impact</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>P9</td>
<td>Agile Scrum’s group commitment and group accountability in projects with over or under control mechanisms will have a neutral effect in time-to-market</td>
<td>Neutral</td>
<td></td>
</tr>
<tr>
<td>P10</td>
<td>Agile Scrum’s group commitment and group accountability in projects with over or under control mechanisms will largely improve time-to-market</td>
<td>+ +</td>
<td></td>
</tr>
<tr>
<td>Effective Communication</td>
<td>Proposition Description</td>
<td>Impact</td>
<td></td>
</tr>
<tr>
<td>P11</td>
<td>Offshore Outsource Agile Scrum projects with physical distance between its collaborators and limited use of CMC tools, will have a neutral effect over time-to-market</td>
<td>Neutral</td>
<td></td>
</tr>
<tr>
<td>P12</td>
<td>Offshore Outsource Agile Scrum projects with physical proximity between its collaborators and reach use of CMC tools, will largely improve its time-to-market</td>
<td>+ +</td>
<td></td>
</tr>
</tbody>
</table>

Table 4-1. Propositions describing the reduction of time-to-market

Having described this, the next step in this research is to conduct a Case Study to help us evaluate our Conceptual Mode.
Chapter 5. Case Study Outline

The purpose of this Chapter is to describe the process configuration and process execution used to collect the empirical evidence to support this research. The Chapter consists of four main sections. The first section (5.1) describes the Unit of Analysis and Unit of Observation that guides this Case Study. The second section (5.2) explains the criteria used to select the companies and projects involved in this investigation. The third section (5.3) contains a description of the Client and the Vendor – as separate organizations – followed by a description of the projects in which they were collaborating. The Data Collection Process is described in the fourth section (5.4).

5.1 Case Study Research
This is an exploratory research aiming to analyze the convergence of two practices (Agile Scrum and Offshoring Outsourcing Software Development). In Social Science, exploratory researches focused on contemporary events are likely to lead to Case Studies (Yin, 2009 p. 9). Therefore we have chosen the Case Study method as our research methodology. The research's context will be multiple projects conducted between one client and one vendor. The Unit of Analysis constitutes projects blending Outsourcing Offshoring and Agile Scrum. The Unit of Observation is conformed by the actors involved in the project execution: Product Owner, ScrumMaster, the Development Team and the Client. The mentioned characteristics led us to conduct an embedded multiple case study.

5.2 Case Study Selection
Selecting the Case Study was a two-phase process. The first step was to delimit the minimum conditions that it should comply with. In this case, the projects should be conducted under the Offshore Outsourcing mode and also they should include Agile Scrum practices. The second step consisted on finding companies willing to collaborate and with projects fulfilling the mentioned requirements. An external advisor of this research led the selection process.

A Telecommunication company from Belgium and its service provider from India accepted to be part of the research. From now on we will call them respectively the Client and the Vendor. Their relationship dates from at least 11 years and our Case Study will focus on two specific projects, which will be described in detail in the following sections. Due to confidentially agreements, we will use fictional names to substitute sensitive project information (e.g. company names, interviewee’s personal information, etc.)
5.3 Case Study Protocol

The case study protocol is considered a good tool to guide the researcher through the phases of the data collection and data analysis; as well as to increase the reliability of the case study (Yin, 2009 p. 79). In this investigation, it will help us to remain focused on the core concepts of the research, especially on the Initial Conceptual Model described on Chapter 4. Additionally, it will help the audience of this research to visualize the steps followed while collecting the data and justifications behind every decision made, thus case study replication can be performed if needed. A Case Study Protocol is usually conformed by the following sections (Yin, 2009 p.81): 1) A Case Study Overview (describing the case study issues and project objectives), 2) Field Procedure (naming the sources of data, procedural remainders and access to the case study ’sites’) and 3) Case Study Questions (describing the core questions the investigator wants to answer while conducting the data collection process). Following, each of these sections are described in more detail.

5.3.1 Case Study overview

This section will provide a description of the Client and Vendor, the conditions under which they started collaborating and the project settings. Due to confidential agreements we are not allowed to reveal the exact information sources, therefore we will give a high level overview of where the project descriptions are originated. The information described in sections 5.3.1.1 and 5.3.1.2 was extracted from three main sources: 1) Client’s official website, 2) Vendors’ Official website, 3) Renowned online newspapers and magazines (The Hindu, ComputerWorld, The Economic Times and CIO-in). The last section (5.3.1.3) mainly emerged from our early-stage conversations with Client and Vendor employees as a matter of preparation for the data collection phase and a series of emails with the Product Owner (Red Project) and the Client’s Release Manager (Blue Project).

5.3.1.1 The Client

The Client is a telecommunication company based in Belgium and founded in early 1990’s. Since its instauration, the Client has been characterized to be a customer-oriented company and pioneer in the implementation of new telecommunication services. Its history shows several examples to empower its customers over the services they receive, resulting in a perpetual quest to improve customer satisfaction. Some of its trademarks on the market have been: the implementation of TV shows on demand, ‘do-it-yourself’ installations and self-configuration services via PCs or smartphones to record TV shows. All these have been materialized through a deep knowledge of the customers’ preferences and constant investment in new technologies.

Currently, the Client has a strong market presence in Belgium and Luxemburg; providing services related to cable television (analog and digital), high speed Internet and telephony (fixed and mobile). Its Corporate Culture and mission are clearly aligned to enrich the customer’s daily life through innovative and reliable services. The Client has policies to stimulate employees to remain focused on the customer’s needs. Some of those policies are: management rewards based on satisfaction levels, efficiency indicators and a customer certification program. The latter is designed to help employees who do not have contact
with the customer to understand the customer’s needs through customer-company activities.

Currently, the Client is confronted with important challenges that could compromise its market position in the near future. Besides the strong market competitors and highly demanding customers; the Client faces new online communication services, which are shaping the customers’ habits to communicate and entertain. Therefore, the Client needs to implement strategic and innovative services to maintain its revenues and guard its market position.

As a response to such challenging environment, the Client has an ambitious Innovation program to improve the network capacity, mobility and convergence of its mobile and fixed telecommunication services. The program is planned to conclude in 2015 and improve the customer experience –company’s standard for many years. With this program, the Client aims to secure its position in the region while complementing its data telecommunication services with entertainment, smart-living and high mobility services.

5.3.1.2 The Vendor

The Vendor is one of the Top-Five Indian companies for IT Outsourcing. It was founded in early 1980’s in South India, region where its headquarters are currently located. By 2002, after entering the NASDAQ stock market, registering US$500 million in revenues and more than 50,000 employees; the Vendor has earned a place in the IT-world map. Just ten years later, in 2012, the Vendor reached revenues of more than US$6 billion and a labor force of more than 130,000 employees worldwide, showing a tremendous company growth.

Currently, the Vendor has offices and development centers in more than 25 countries supporting its strong world presence especially in Europe and U.S.A. It offers a large set of outsourcing services mainly specialized in: Cloud and Mobility services, Consulting, Business Application and Business Process Outsourcing. In the case of Mobility services, it claims to have an important portfolio of satisfied customers, including Telecom Service Providers. It has received several recognitions from international organizations, describing it as one most influential and outstanding IT Outsourcing provider in the world. This has fed its fearless expansion strategy, which despite the economic turbulence of many of the world economies, has resulted into 10,000 new employees just in 2011 (2000 more new jobs than the initial estimation).

Analysts outside of the organization doubt whether the Vendor’s outstanding growth pace could be sustained. Investors have been very critical on the Vendor’s revenue growth compared with its direct Indian competitors. They claim that the Vendor has been underperforming in emerging business areas like Infrastructure Management and Business Process Outsourcing. Additionally, rating agencies have downgraded the Vendor’s corporate credit rating arguing financial instability, directly affecting the Vendor’s stock prices.
Within the organization, important issues have also emerged. In 2010 the Vendor registered a peak of 20% of employee attrition rate. Recently, in early 2012, Top Management announced the suspension of salary increases until revenue indicators would improve. Such decision seems incoherent for many employees due to the high salary increase received by some top executives in 2011-2012. Employees have claimed that underperforming problems could actually come from mismanagement at top levels of the organization.

5.3.1.3 Project Description

Both organizations started collaborating more than 11 years ago. During this period, the Vendor was the Client’s primary IT partner enabling the creation of external and internal IT services and products. It has been a relationship mainly ruled by: waterfall methodologies, time-boxed projects and upfront requirements; reinforcing classical software practices in both organizations. More recently, due to the increasing demand of new Information Systems, the Client decided to partially delegate the development of two projects to the Vendor and therefore compensate its relatively small development capabilities. Both projects experienced the introduction of Agile Scrum practices, however such process has not been trouble-free, hampering their Client-Vendor relationships. The Client has a special interest on improving their software development operations, tuning-up the synergy with the Vendor and reducing its time-to-market. Following is given a general description of the projects that will be analyzed in the case study:

Red Project.

The Red project is a multi release program to replace the current sales order intake system and make it available to multiple channels (both internal and customer facing). The project demands to create new functionalities as well as providing compatibility with legacy systems that currently support the business operations.

The project started in August 2010, aiming to make a functional delivery between Quarter 3 and Quarter 4 (2011). Due to its large complexity, the project was divided in five components: (C1) a new data base and backend services to manage sales orders, core element for the other components, (C2) a module to synchronize old and new services, (C3) frontend module built over C1, (C4) an extension of the master data system and (C5) migration of the old data to the new system. Based on the component’s dependencies, five milestones or releases were planned. All the components, except C3, were developed in collaboration between both organizations. Figure 5-1 depicts the evolution and team confirmation in the development of each component.
Previous projects between both parties were developed under the waterfall methodology and this project was designed to follow the same approach. During releases 1-3, the project progress was typically tracked through Key Performance Indicators (KPIs), provided by the Vendor, and weekly follow-up meetings with all team leaders. However the complexity of the project was very high and there were roughly 60 resources located in three locations: one in Belgium and two in India (Pune and Trivandrum). Task dependencies constantly delayed the teams and they had important difficulties foreseeing problems; hampering the original schedule and compromising the project’s budget.

In July 2011, just three months before the final release, a project evaluation revealed a critical problem: Features from Release 1, 2 and 3 were incomplete and the entire project was considerably delayed from the original schedule. They estimated that the project was not going to be ready for the October 2011 and they would have to postpone the final release (Release 5) at least until March 2012. Informing the different stakeholders that not only that the project was delayed, but also that they would have to wait until March 2012, was not an option. As a response, the Vendor (under the Client’s supervision) decided to prioritize tasks, speed up the development and deliver at least the core functionality.

At this point of the project, team members had two assumptions: 1) They only had to finish some missing features from Releases 1-3. 2) New functionality was not required therefore no Analysis or Design tasks were needed. Based on these assumptions, the Vendor decided...
to implement a *Scrum-But* approach (follow some concepts from the methodology, but adapt it to the current status of the project).

After Release 3, they introduced one extra team for C1 and C5 and several Scrum concepts were taught in order to implement the new plan. Every sub-team had a Daily Scrum meeting and followed there was a Scrum-of-Scrum meeting where representatives from each sub-team were giving a general overview. During the first iteration, personnel from the Client side travelled to the Vendor’s offices in India. During one week they attended to the Scrum-of-Scrum meetings and they met team members. No major warnings were received suggesting a project failure; therefore the Client and Vendor continued with the plan.

The first sprints in preparing for release 4, the teams seemed to perform better. However problems came back to the project: teams were still presenting coordination problems and estimated task duration was very different compared to actual task duration. The two core assumptions were not completely met: Releases 1-3 were missing functionality and new requirements appeared. After delivering some of the core functionality, the project is not considered complete. The Client decided to stop the project because an initial calculation of the remaining tasks revealed high costs to finish it. Currently, the Client is reevaluating the project’s role on the TI Master Plan and is yet to decide the next steps to follow.

**Blue Project**

It started in April 2012 and it is the continuation of a previous version originally from 2011. In terms of budget, complexity, time and team conformation, the Blue Project is a smaller project compared to the Red Project. The project consists of developing an application that would allow end users to share and synchronize personal information (e.g. agenda) with other end users, independently if they are affiliated with a different mobile provider. The project is considered a Business-to-Business project because it demands collaboration from other Companies (mainly telecommunication companies) to enable the service. Although, it can also be considered a Business-to-Customer project because it eventually will be service for the final customer.

Regarding the software methodology used, it is relevant to mention that the Blue Project originally followed a waterfall approach. The large majority of the requirements and design elements were defined upfront. Afterwards, just before starting the development phase, the Vendor requested to introduce Agile Scrum practices into the project. They suggested this change in order to reduce time-to-market and the duration of the development cycles. They used the list of requirements and added priorities. They used this list as a Product Backlog, but it turned out to be a TO-DO list. The project did not have a Product Owner, under the premise that the scope was defined and no change in the requirements were expected.

**5.3.2 Field Procedure**

The case study is expected to have main two data sources: Interviews and Official project documentation. Following we describe each of them:
1) **Interviews.** Interviews are suitable tools to disclose information in case studies that involve human affairs (Yin, 2009 p. 108). As it was concluded on Chapter 2.3, the main actors conforming Agile Scrum are: the ScrumMaster, Product Owner, Development Team; hence our main interest is to interview these actors. Additionally, due to the fact that this research is focused on two-organization relationships, we also are interested to interview representatives from the Client side. This does not mean that additional project members would be excluded to participate in the research.

The first step to follow will be to contact the Client’s IT Supply Manager (the contact point for our research) and request his approval to conduct the interviews. His strong position in the Client’s hierarchy could benefit us to have access to different project members and gain credibility as researchers in both organizations. After having access to an initial list of interviewees, we will contact them via email introducing ourselves, describing the research and requesting a 30 minutes meeting. Appendix 9.2 describes the Interview Letter planned to be used to fulfill the mentioned elements.

After receiving a positive reply, we will proceed and schedule individual interviews. We expect to have face-to-face and virtual interviews. As it has already mentioned, the Client’s headquarters are located in Belgium, but the majority of the Vendor’s employees are in India. Due to budget limitations and time constrains proper of this research we will be able to perform the interviews under the same conditions. (e.g only face-to-face meetings). Depending on the interviewee’s convenience, virtual interviews will occur via videoconference tool (e.g. Skype, GoToMeeting, etc.) or telephone. Additionally, we aim to audio record the interviews to facilitate transcription and interpretation.

2) **Official project documentation.** Documents like requirements, design documents, schedules are common tracking and agreement mechanisms in outsourced projects. Having access to project documentation would allow us to have deeper knowledge of the decisions and issues present in the projects.

The procedure to access this type of data source will be very similar to the previous data source: 1) Request the approval of the Client’s IT Supply Manager to have access to the documentation. 2) Contact project members to request documentation. 3) Analyze the data, linking it with the rest of the research.

**5.3.3 Interview Format**

The main purpose of the protocol’s questions is to keep the investigator on track as data collection proceeds (Yin, 2009 p. 86). The interview was designed as *semi-structured interview* to help the respondent to introduce and clarify concepts along the discussion and also allow the interviewer to guide the discussion towards the research topic. The purpose of this section is to highlight which topics we plan to discuss and the goal behind it.

---

6 Recording will only take place if interviewees accept it.
Table 1-1 summarizes the three main topics aimed to cover and its justification with regards the Initial Conceptual Model (Chapter 4).

<table>
<thead>
<tr>
<th>Topic</th>
<th>Relation with Conceptual Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roles and Responsibilities</td>
<td>We aim to validate the interviewee’s participation in the project. Also we aspire to confront the theoretical findings disclosed on Chapter 0 related to Agile Scrum’s Roles, Responsibilities and Artifacts with a practical project. Additionally, we expect to confirm or reject our assumptions related to the core Agile Scrum concepts affecting Client-Vendor relationships. Example of introducing phrase: “Could you explain me your roles and responsibilities within the project?”</td>
</tr>
<tr>
<td>Client-Vendor Relationships</td>
<td>Through this topic we aim to disclose how both organizations interact, as well as means to communicate. Covering this topic we will disclose connections with CMC’s mediation role in the project and also we expect to reveal the Agile Scrum Events where Clients and Vendors converge. Example of introducing phrase: “Could you explain me your interaction with the Client/Vendor? How often do you meet? How do you interact?”</td>
</tr>
<tr>
<td>Challenges</td>
<td>The last topic refers to the Outsourcing Challenges and their possible presence along the Agile Scrum framework. We aim to cover each of the five mentioned challenges (Culture Differences, Coordination, Time Zone Differences, Trust and Effective Communication) and invite the interviewee to bring anecdotes whether or not they consider they are a challenge in the project. This will help us to refine the initial Conceptual Model and confirm or reject our initial deductions. Example of introducing phrase: “Literature mention that [challenge] can be a challenge in this type of projects. Have you faced any problems with this aspect? How has it been the effect working with Agile Scrum?”</td>
</tr>
</tbody>
</table>

Table 5-1. Interview topic list.

It may be the case that due to the interview’s flexibility and openness planned for the interview, the interviewee could have already mentioned a specific problem; therefore it would not be needed to discuss every single topic. The reason behind this is let the interview flow more naturally and give the opportunity to disclose more elements to the research.

Concluding the interview, we will ask whether there is any remaining topic or aspect that interviewee would like to add to what has been already discussed. By doing this, we enable the interviewee to disclose relevant topics that might have been excluded from the initial literature review.

5.4 Data Collection Process
The initial decision to contact the Client’s IT Supply Manager requesting his collaboration to collect the data resulted to be very effective. He became a facilitator in the process of
accessing the project members and project documentation. We conducted interviews with the following team members:

- **Red Project**
  - The Release Manager and Project Manager (Client’s employees)
  - The Release Manager, ScrumMaster and one collaborator from the Development Team
- **Blue Project**
  - The Release Manager and Project Manager (Client’s employees)
  - The Release Manager, ScrumMaster and one collaborator from the Development Team.

All the interviews were recorded and later literally transcribed resulting into 45 pages of Case Study documentation that later will be used to facilitate the analysis process of this research and increase its validity. Appendix 9.3 describes in further detail the interview settings (e.g. duration, location, interviewee’s role, etc.).

In addition to this information, the team members from both projects provided us documentation revealing the project’s hierarchy in order to improve our understanding of the project configuration. Moreover, they provided us a project template ‘Product Backlog’. Among other elements, this ‘Product Backlog’ contains: KPI (Key Performance Indicators), a Requirement List with delivery dates assigned to each task, and burndown charts showing the Teams’ Scrum velocity. Serving as a historical documentation record of the project evolution, constantly updated by all the project members, makes it a very reliable document for our research and helpful for confronting the information from the interviews.
Chapter 6. Case Study Results

“... Frankly speaking, you are just mapping all the things that I want to talk to you about and all the things we have faced ...”

-Development Team Member, Vendor, Blue Project,

The goal of this Chapter is to describe the Case Study results emerged from the Red and Blue Projects. The main assets used to reach such goal were the data collected through semi-structured interviews and project documentation described in Chapter 5. In order to process and analyze the Case Study evidence, we used a four-phase approach suggested by Corbin & Strauss (2008). Following we explain each theses phases:

- **Phase 1- Exploration.** It consists on analyzing the data using an Open Coding technique, where the researcher aims to reduce amount of data but remain open to concepts that can be important for the research.

- **Phase 2- Specification.** This phase is focused on grouping the concepts found on Phase 1 into dimensions and characteristics. The researcher strives to find differences and similarities between the categories created from lower-level concepts.

- **Phase 3- Reduction:** At this stage, it is recommended to select the main code(s) or concept(s), which are mainly aligned to the research’s core goal.

- **Phase 4- Integration.** This is an integration phase where the researcher intents to create links and disclose causalities between the core concepts. Literature related to the topic plays again an important role because it will help to support or reject the possible connections discovered on the previous phases

Having these phases in mind, we structured this Chapter as follows: The first section (6.1) depicts Phase 1 and 2, discussing the open coding and axial coding process. In the next section (6.2) we analyze the process to fulfill with the Reduction Phase. The last phase (Integration Phase) is described in section 6.3. In section 6.4 we evaluate our Conceptual Model based on the Case Study findings and in section 6.5 we describe the general conclusions of the case study.

6.1 Exploration and Specification

The interviews conducted with Red and Blue Project team members resulted into 18 and 35 pages of literal transcripts respectively. In order to increase the research’s internal validity and illustrate how the interviews took place, we enclosed one of the interview transcripts (see appendix 9.4). Due to the large amount of data to process, we used Atlas.it (a qualitative research tool) to facilitate the coding and analysis process.
Initially we revised each transcript and broke the data into concepts helping us to summarize the information provided by the interviewee. Such process is also known as Open Coding (Corbin & Strauss, 2008 p. 195). During the Open Coding process, we encountered concepts that were known due to our initial literature review (e.g. Sprint, burndown chart, language miscommunication and mistrust), but also we detected concepts that were out of our initial knowledge base. Some of these new concepts were: Jira (issue resolution tool), Change Request, Scrum-But (term defining partial implementations of Agile Scrum) and Single Point of Contact (SPOC). We concluded this stage with roughly 210 codes for both projects.

For the Specification phase, we used the Case Study Protocol (Section 5.3) to contextualize the codes and relate to the topics we initially discussed during the interviews. We created a matrix consisting of Topics (columns), Roles (rows) and codes (cells). Following, we rehashed the matrix into codes emerging from the Client’s employees and codes from the Vendor’s employees. The purpose of both matrices was to give an overview of ‘who said what’ and later facilitate the identification of shared and contrasting codes. Both matrices can be found on Appendix 9.5.

6.2 Reduction
This probably was the most difficult phase to conduct during the data analysis process. At this stage we had to discard less important concepts and try to focus on elements that would let us understand the real events behind both projects. The criteria used to determine what was ‘relevant’ during the interviews was conformed by: 1) Code Occurrence (number of times that a code appeared within the interview and within the project) 2) Interviewee’s concept relevance. A concept could be counted once in an interview, but if the interviewee dedicated significant amount of time to explain why it was a problem, it should be more relevant than just a concept that was merely ‘listed’ once. 3) Sensitivity. The Researcher’s sensitivity can be used to identify important concepts within the data. Sensitivity is a skill acquired by the researcher through the time, but it can be enhanced with the in-situ experience as well as with professional experience (Corbin & Strauss, 2008 p.32,33). Followed, we present a series of images illustrating the most relevant concepts grouped per Outsource Offshore Challenge.
Figure 6-1. Concepts related to Culture Differences.

Figure 6-1 depicts the main concepts related to Culture Differences. Client’s employees recalled three main topics: 1) The abrupt introduction of Agile Scrum in both projects in both projects. 2) How Vendor’s employees avoid disclosing problems. 3) Discussions and frictions between both organizations mainly started by differences on ‘ways of working’. On the other hand, Vendor’s employees (right side on Figure 6-1) agreed that their long-term relationships with the Client lighten any ‘culture misunderstanding’. Furthermore, they emphasized that there is a difference of how the Client and they (the Vendor) perceive project progress.

For the second Outsource Offshore Challenge (Coordination), we encountered an interesting trend. With the exception of the Red Project in which the Project’s Configuration was too complex, Client’s employees mostly agreed that they did not face any problems with Coordination, as the Single Point of Contact (SPOC) was very good at coordinating teams. Vendor’s employees also complimented the SPOC and its bridging role, but they disclosed more problems. For the Red Project, introducing Agile Scrum was a learning process when coordination was a challenge and they perceived communication overheads resulting in extra effort (time) from many of the team members. Blue Project members named Jira and reports (burndown charts) as key tactics to reach coordination. Figure 6-2 illustrates the previous concepts and relations.
Time Zone Differences (Offshore Outsource Challenge 3), was the challenge with the least concepts associated. There was a general agreement from all the interviewees that time difference between India and Belgium (3.5h-4.5h) was not a problem, but more a constant that they have managed to overcome. There were two outlay concepts. 1) ‘Short window’. This concept emerged from a Product Owner (Red Project) commenting that during stressing moments a 5hour window to coordinate could be too small. 2) ‘Time Difference is an advantage’. This concept emerged from a Development Team member (Blue Project), explaining that after overcoming coordination problems during the first Sprint, the team was able to leverage the Time Difference. Figure 6-3 illustrates the mentioned concepts.

Figure 6-2. Concepts related to Coordination.

Figure 6-3. Concepts related to Time Zone Difference
Trust (Challenge 4) was a dense challenge to analyze. On one hand Client’s employees pointed that Trust is built at personal level, but Transparency from the Vendor was also needed. Late deliverables, the abrupt change to Agile Scrum and change of estimations were events that largely led to conflicts and mistrust towards the Vendor. On the other hand Vendor’s employees aim to be Transparent and build Trust through reports (performance reports, burndown charts, etc). Client’s inexperience with Agile Scrum caused misunderstandings related to team’s Scrum Velocity. The Vendor, justified problems (e.g. delays) through the premise that Blue Project is a Pilot project for the introduction of Agile Scrum and the Client must realize that errors could appear. Figure 6-4 summarizes the previous ideas.

In the last challenge (Effective Communication) the code related to SPOC and its importance was highlighted again in both organizations. However, Client’s employees stressed that even though project’s status was given on daily a basis, the effectiveness of Sprint Demo meetings were limited by the quality of the teleconference. Another relevant code was the Vendor’s employee turnover. The Release Project Manager (Blue Project) emphasized that Vendor’s employees need time to be adjusted to the ‘Client’s way of communicating’ (direct and punctual) and effective communication between team members was damaged every time a team member leaves the project. Figure 6-5 illustrates our findings.
It can be expected that there were codes that did not fit into the challenges and had to be discarded. Especially when we asked to the interviewees to openly discuss any missing topic, we identified codes related to project technical details, third party companies and future projects. These codes are important to understand the projects’ specificities, but also they limited the generalization of our analysis, therefore we decided to leave those codes as background information rather than include them in our main analysis.

6.3 Integration
In this phase we aim to compare the initial findings from the previous phases (Exploration, Specification and Reduction) with the Literature Review conducted in Chapters 2 and 3.

6.3.1 Culture Differences
Literature suggests that Outsource Offshore projects can suffer Culture incompatibilities in two sub-dimensions: 1) National Culture - related to language barriers, values and norms associated to each country or region. 2) Organizational Culture - referring to mismatches on ‘ways of working’. The post-interview analysis revealed that both sub-dimensions of Culture Differences are present on the Client-Vendor relationships. All the interviewees agreed that there were differences between both organizations and to some extend, between both countries. Followed the main findings are presented.

Misunderstandings related to language and expressions were common topics during the interviews. Even though English was the official language for both projects, Interviewees mentioned that they encountered difficulties to understand the other’s party accent or ways to expressing themselves.
“We (the Client and the Vendor) communicate in English but still the language is different. It is not our native language, so sometimes we try to say something but it is understood in a different manner. This is a bit of a problem I would say.”

- Functional Expert, Vendor’s employee, Red Project

In addition to language misunderstandings, we found more profound problems related to ‘Communication Style’. Client’s employees, majority are Belgians, behaved and admitted to be, very abrupt in comparison with their counterpart in India. Client’s employees did not hesitate to give direct answers regardless these are ‘good’ or ‘bad’ news. On the other hand, Vendor’s employees were very cautious to disclose issues and problems. Many Client’s employees recalled situations when Vendor’s employees were reluctant to give ‘bad’ news, limiting the problem resolution process.

“Flemish, we are very straightforward, very abrupt. Most of us are not afraid to say ‘No’, or we are not afraid that something is ‘wrong’ … here we don’t immediately know that something is going wrong, they (the Vendor) are very afraid to lose face and just say: ‘oh this is not working’”.

- Project Manager, Client’s employee, Blue Project

“Couple of times I sat with one of the developers and said: ‘Ok you clearly see that here there are some issues here, can you put it on emails, so at least everyone is aware of them and we can tackle the problem’. Simply they didn’t want to do it, because that would be escalated. Escalation in their culture is not a good thing. It will reflect directly on their performance but that will not solve the problem.”

- Product Owner, Client, Red Project

Statements from the Client’s employees suggest that Vendor’s employees present a ‘Problem Avoidance’ behavior. Such comportment is consistent with our Literature Review discussed on section 3.1 and with Carmel and Tjia’s research (2005 p.181), where many Indians prefer to avoid confrontation.

During the interviews we also experienced traces of the ‘Problem Avoidance’ behavior by the Vendor’s employees. Even though we circulated an Invitation Letter explaining the purpose and confidential settings where the research was going to take place (Appendix 9.2), we encountered Vendor’s employees who were not very open to talk about problems on their projects. One interviewee reconfirmed that he was not going to be quoted on the research. Another interviewee, every time we were asking him about a specific challenge, he was replaying first with ‘tactics’ or ‘how they were dealing with the specific problem’ and then continuing with his explanation. This called our attention because even though we were not involved on the project daily routines, we could perceive the communication difference between Client and Vendor’s employees.

National Culture Differences explained on section 3.1, can help explaining why employees from each organization differ on how they communicate, in this case: Why Client’s employees
are more direct than Vendor’s employees? Nevertheless, we should not overlook that company mechanisms can incentivize or discourage employees to reveal certain information. In section 5.3.1.2 we discussed that in early 2012, Vendor’s Top-Management stopped all the salary increases for the Vendor’s employees until the company’s indicators (e.g. quality) are improved. Such corporate direction can amplify the Problem Avoidance behavior in the Vendor’s employees.

In addition to differences on Communication Styles, our analysis revealed important variances on how Client’s employees and Vendor’s employees perceive project progress and overall status. On one hand, Client’s employees rely more on their perception and less on indicators and reports. On the other hand, Vendor’s employees are highly-data driven, paying much attention to processes and standards.

“We measure typically our standard or processes, and we define standards to see how we are progressing. But the Client as such is more [remains quite] driven by …[remains quite]… there is an overall feel in terms of how is the progress. They are more into perception, their approach is quite different.”

–Project Manager, Vendor, Red Project.

Client’s employees were very open to talk about how differences on communication styles and preferences to ‘measure’ project progress, led to constant frictions between both organizations.

“A lot of the Client’s employees think that sometimes it’s excuses (late deliverables)... Also towards Business (final system users), there is some reluctance towards their deliverables, the way they work, because it takes so long, and time-to-market is too long. They always have to pay extra, there is always a CR (Change Requests). So if we want to go to full Agile, definitely discussions regarding CR and SLA (Service Level Agreements) have to be opened up again.”

–Release Project Manager, Client, Blue Project

6.3.2 Coordination

As we described in section 3.2, Sabherwal (2003) suggested that clients and vendors in Outsource software projects can have 4 types of coordination mechanisms to align their tasks. The first two take place before initiating the project 1) Coordination by standards and rules, 2) Coordination by plans and goals. The remaining two take place during the project execution: 3) Coordination by formal mutual adjustment, 4) Coordination by informal mutual adjustment. During the interviews, we found evidence of a mismatch on how the Client and the Vendor were expecting to coordinate their tasks. The strong Organizational Culture from the Vendor towards performance indicators and status reports was clearly reflected on how Vendor’s employees were acting in the projects. Often, their counterparts at the Client’s side characterize them as very rigid. On the other hand, the Client is a much more loose organization. Its Culture towards perception over indicators, makes the Client more flexible compared to the Vendor. We plotted both organizations on Sabherwal’s coordination typology, based on our analysis from case study information. Figure 6-6
illustrates the different coordination mechanisms used by each organization. Additionally on the right-lower part of the graph, we highlight the Scrum's preferable coordination mechanism while developing software.

Figure 6-6. Client and Vendor’s ruling coordination mechanisms, based on coordination typology from Sabherwal (2003) and Scrum’s concepts

Another relevant finding was the Vendor’s tactic to locate a Single Point of Contact (SPOC) to facilitate the coordination of onshore and offshore teams. Interviewees from Red and Blue project highlighted the importance and effectiveness of having a liaison resource. Ranging from tasks as setting conference meetings, up to conducting Sprint Demos and following-up clarifications; the SPOC was considered a key employee on both projects.

“In general we have an on-site SPOC, so everything is coordinated through him, so he is responsible to set up the calls, to set up the conferences, and he is always there, the one thing I have to say is, he has been pretty good about it, a lot better than my other SPOCs for other project”

- Project Manager, Client, Blue Project

The SPOC is a very good tactic to lighten coordination problems between both organizations, however such tactic has two main drawbacks: 1) As the Project Manager suggested, its effectives differs from person to person. 2) SPOC’s responsibilities are mainly limited to daily and implementation activities. When the projects were confronted with problems such as: change of requirements, project delays or unexpected errors; differences on coordination mechanisms between the Client and the Vendor largely prevail, creating frictions at the project and organizational level.
“I think they [the Vendor] hold too much to the rules and processes. I find it a little bit... sad actually. Because at the end we are a customer, and sometimes we don’t feel that way. For instance: We have a new member, and he is very attached to the rules, and if something critical appears, and we need to make an extra patch. He says no because we only have 2 patches per week.”

- Release Project Manager, Client, Blue Project

Recurrent discussions to related ‘follow the plan’, project delays and scope definition; hampered mutual Trust and eroded Client-Vendor relationships.

“We had a lot of discussions whether something was in scope or not or a new request. And that discussion we typically lose it. At the end we don’t spend more time or afford. That means that you don’t Trust anymore the comments or reactions coming from your counterpart, you simply say: ‘ok we ignore because this is not taking us anywhere’.”

- Product Owner, Client, Blue Project

Another relevant finding was the close connection between Effective Communication and Coordination largely affected by the project configuration. Comparing the both attempts of introducing Agile Scrum practices into Red and Blue Project, Red Project had more difficulties coordinating tasks than Blue Project. Red Project interviewees agreed that the fact that the project consisted on roughly 60 people, multiple sub-teams in multiple locations in India (Prune and Trivandrum) and highly complex tasks; made team coordination a relevant challenge to overcome.

“Every day there used to be a Scrum of Scrums between three ScrumMasters offshore, two from here and the Project Manager, it used to be over telephone, maybe Live Meetings. But that communication overhead was there and that communication had to go one level down”

- Project Manager, Vendor, Red Project

Different tactics suggesting how to scale up Agile Scrum have emerged (Cohn, 2010 p.327), however Jali & Wohlin (2010) in their systematic literature review concluded that there is no sufficient evidence to support that Agile Methodologies can efficiently be implemented in large distributed projects.

### 6.3.3 Time Zone Differences

Time zone Differences was a very interesting challenge to analyze in Red and Blue Project. Espinosa et al (2007) studied four time zone configurations (fully overlapped, 2/3, 1/3 and no overlap) and how non-collocated teams can experience tradeoffs between speed and accuracy. We asked all the interviewees if they have faced problems with the Time Zone Difference between India and Belgium (3.5h-4.5h). Initially all the interviewees agreed that

---

7 The Product Owner expressed that he was not sure how many people were involved. He had a list but he was very sure that the Vendor included more employees, but he never met them.
they did not find problems with such difference. They expressed that it has been a constant since they started collaborating almost 11 years ago and that they have been able to get use to it.

“For us it was not a problem (Time Zone Difference) because in a normal project, even in a not Agile project. We manage that and since we have been working in this kind of model for quite a long time, this is not one of the impediments.”

–Functional Expert, Vendor, Red Project.

Nevertheless, when we continued asking if there was any moment during the Sprint that Time Zone Difference could make a difference and interviewees more related to the software implementation commented that Time Zone Differences can have an effect while coordinating tasks and clarifying requirements.

“...For instance the Time Zone Difference is sometimes a benefit and sometimes not. You can finish something here and they will pick it up there in the morning. That’s a good thing. The problem is that the window to be all together is something like 4 to 5 hours and that’s short. That means that all the meetings and all the communication need to happen within that window.”

Product Owner, Client, Red Project.

Vendor’s employees admitted that in early stages of the Agile Scrum introduction, they were suffering delays because they needed clarifications and the Client was still not at the office. They had to overcome this problem via the SPOC.

“Initially it (Coordination) was an issue. If someone had to discuss something with the onsite people, they had to wait 3.5h until the other team would arrive. After first Sprint we used it as an advantage, we passed the issue to the SPOC and next morning we would have 3.5h to continue with the resolution. After Sprint 1 or 2 we use it more as advantage than as a disadvantage. You have to plan things much better. That was done in the second Sprint. We said: we need to plan this and this and we considered the 3.5h difference.”

-Development Team Member, Vendor, Blue Project.

Our conclusion from Time Zone Differences found that the possible tradeoffs between speed and accuracy under a 3.5h-4.5h setting, can be overwritten by the outsourcing settings. The vendor must guarantee the accuracy and delivery time expected by the client regardless meeting such expectancies would imply more effort (e.g. man-power) from the vendor’s side. Looking in more detail level within the projects. For high-level employees (e.g. Project Managers, Release Managers) Time Zone Differences are less relevant, but for low-medium employees, it becomes more important especially for coordinating tasks or requirement’s clarification.

6.3.4 Trust

Trust was one of the most complex challenges to analyze. In order to fully understand the role of Trust on this Case Study, first we must refer again to the context where Red and Blue
project were executed. Starting from the Red Project, we must understand that there is a perception difference on the final output of this project. Especially among Client's employees, the Red Project is considered a failure. On the other hand, Vendor’s employees partially agree with this view and they prefer frame it as a 'learning project' in which the Client and the Vendor made their first steps towards Agile Scrum practices. Regarding Blue Project, it is relevant to mention that it is a smaller and less complex project, which started months later after Red Project’s closure. In both projects, the introduction Agile Scrum was an unilateral decision from the Vendor.

Recalling this series of events is important because, even though there was no employee overlap between both projects, many events that occurred on the Red Project are recalled during the Blue Project’s interviews. Events such as: the abrupt introduction of Agile Scrum practices, exceeding the original time and budget and missing functionality; continue being evoked even during the Blue Project’s interviews, meaning that there is a scar between both organizations that had led to mistrust.

“Why they (the Vendor) wanted to Blue Project become Agile after Red Project went out under the train! Red Project doing even more Agile than Blue Project. Why they want to do that? I wonder!! Do they want to prove that they can do it?? Well now I sound very pessimistic, but well they only have proved that they can do Iterative development but not Agile. I think the Vendor might look it differently.”

- Release Project Manager, Client, Blue Project

Having said that, we can continue disclosing the results associated to this challenge. On section 3.4 we discussed four types of Trust typically present in non-collocated Software Development projects (Sabherwal, 1999): 1) Calculus-based. Trust built around control mechanisms (e.g. contracts). 2) Knowledge-based. Mutual credibility built by previous client-vendor experiences. 3) Identification-based: Trust reached when both parties identify a common goal that bonds them. 4) Performance-based: Trust based on meeting tangible goals while executing the project (e.g. meet milestones, reach quality standards, etc.). Our analysis revealed that the Vendor’s employees mainly rely Knowledge-based and Performance-based types of Trust. During the interviews Vendor's employees were constantly recalling their long-term relationship with the Client as proof of their mutual trust and understanding. Additionally, Vendor’s strong Culture towards indicators, metrics and data (see Chapter 6.3.1) was also reflected on their tactics to gain Trust:

“Trust is something that goes beyond, has to be there besides waterfall or Agile ... We created a template for each phase, and each phase had a simple technique to show the burndown of each phase, and an easy mechanism for the guys to track it. ... That's how we try to increase our Trust levels. I'd rather to call it Transparency.”

- Release Manager, Vendor, Blue Project
Contrasting to the Vendor, the Client had a different view on Trust and how to maintain it. The Client fits into Performance-based type of Trust with the particularity that they are little interested on indicators and their Trust is based on personal perception.

“The Trust do not come from the Vendor, but from individuals. Recently, I worked with a young guy, and he was superb! ... But that it is not always the case. So it is really a person-to-person situation. I think that for Blue Project they have very qualified people. Definitely there is good Trust in that Team, but in Red Project there was no Trust. No Trust between the Project Manager, No Trust between the Vendor, it was mayhem…”

- Release Project Manager, Client, Blue Project

Furthermore, limited knowledge on the Agile Scrum methodology from both parties and the ‘Problem Avoidance’ behavior present on the Vendor’s employees, have strengthen the differences on ‘how to gain trust’. Especially Client’s employees recalled constant discussions they had with Vendor’s employees about budget, team’s Velocity and deliverables, among others.

“…when we were in the planning or scoping part of the iteration we went offshore (to India) and we met part of the team there. To see if actually it was feasible to have all the functionality for the 12.1 or 12.2 release and then we got the impression that it will work. We didn’t get any big warning sign that it will not work. The first three iterations we got a good show: everything on time, no problems. But after the 4th iteration it was when suddenly all the estimations went up by 50%. I am not sure if this is specifically because of offshoring, I think it can be a typical issue with outsourcing in general. The partner is not always confident at the beginning to report the actual situation. …”

–Product Owner, Client, Red Project

## 6.3.5 Effective Communication

The interview analysis revealed two main sources to from which the Client and Vendor were communicating. Followed is explained each source, its limitations and interaction with the rest of the challenges.

### 1) Person-to-Person.

Interviewees agreed that even though the official language of the project was English, both parties had difficulties to communicate. Blue and Red Project registered problems while understanding highly-detailed requirements and also understanding each other’s English accent. Their main tactic to overcome this situation was again the Single Point of Contact (SPOC). He was the key actor to leverage miscommunication problems and speed up unsolved requirements. Nevertheless, Client’s employees expressed that every new SPOC entering to the project had to be molded to be less strict with rules associated with the Vendor’s Culture.

“For every offshore team, there is a person on site. So that’s how the communication flows, and how I said before, if that person is very good, then not no problems. But in general I think it is good. Only if this new person is coming you have to build again that relationship. But I would
expect the Vendor to put more emphasis on it, because we go through the same thing over and over. For instance, from the RPM (Release Project Manager) position we had 3 new RPM in the last year. So you see, then they come, they know all the procedures, all the calculations for every Change or Requirement (CR) and Service Level Agreement (SLA) and what they should and they should not accept. And you have to break that down a bit cuz the practicality is out of it. Every time you have discussions about production issues found. That it is very painful.”

-Release Project Manager, Client, Blue Project

Even though interviewees from both projects described the SPOCs as very effective bridging both organizations, the SPOCs could not take care of every communication and coordination aspect surrounding the projects. The Lack of a Product Owner (Blue Project) and a ScrumMaster (Red Project) hindered task clarification and the project’s progress, resulting into a Sprint cancelation and constant changes of Sprint tasks.

“There is a communication gap between Business (final system customer) communicating to the Client-Analysts and from the Analysts to us. There are cases when the requirements are not described properly and the interpretation is not correct. I would use that word: interpretation. I think here is very important the role of the Product Owner. The alignment between the solution description and the Product Backlog is not there. I would suggest a Product Backlog design and then the solution description explaining what is in the Product Backlog. It would give us a better understanding.”

-Development Team Member, Vendor, Blue Project

2) Computer-mediated applications.
Red Project and Blue Project mainly used three electronic means to communicate: emails, Jira (issue resolution tool) and teleconference. In general, the first two means (emails and Jira) were very effective helping team members to communicate with each other. However teleconference lacked of quality and this hampered general trust. Specially during the Sprint Demo meetings – key moment to stimulate the progress perception – team members were hardly able to communicate and this discouraged Client’s team members but specially Business (final system users).

“Sometimes, because of the telephony the voice breaks. What happens is that during the first demonstrations, during Sprint 1. We tried to give them over videoconferences, but the voice communication was not very clear. So what we got was that Business was not engaged in the entire discussion. We decided to make the presentations onsite.”

-ScrumMaster, Vendor, Blue Project

Our analysis shows that the Red and Blue Project were not maximizing the usage of Computer-Mediated applications. Both organizations were ‘just communicating’ and not really trying to reduce the lack of physical presence. Improving real-time communication through video and teleconference, as well as Instant Messaging, can largely improve the communication effectives. Currently both organizations rely too much on their SPOCs and
this have created communication bottlenecks and limited the involvement of key players like Business.

6.4 Conceptual Model Evaluation
This multiple-case study was triggered by the need to collect empirical evidence that could help us to validate our initial assumptions of what criteria influences time-to-market in Offshore Outsource projects using Agile Scrum. In Figure 4-3, we presented a Conceptual Model describing different criteria influencing time-to-market, which resulted in twelve propositions. Unfortunately, the multiple case study did not provide us all the settings to evaluate all the propositions. Following we present the propositions that we could find empirical evidence

**Proposition 1:** Teams with contrasting National Culture will encounter difficulties to adopt Scrum’s core values causing a neutral effect over the reduction of time-to-market.

**Case Study result:** This proposition assumes that the positive effect from Scrum core values (Openness, Transparency, Inspection and Adaptation) will be cancelled out by negative effect of National Culture incompatibilities between the team members, creating no effect over time-to-market. The multiple-case study reported a different effect. Team members from the Red and Blue Project had several difficulties to adopt Scrum’s core values, causing personal misunderstandings, mistrust and resulting into project delays. There was a clear mismatch between Belgium and Indian employees on how they behaved within the project. All these caused the erosion of their Client-Vendor relationships and an extra effort (time, money) to conclude the projects.

**Proposition’s revised statement:** Teams with contrasting National Culture will encounter difficulties to adopt Scrum’s core values causing a negative effect over the reduction of time-to-market.

**Proposition 3:** Organizations with contrasting Organizational Culture will encounter difficulties to adopt Scrum’s core values causing a neutral effect over time-to-market.

**Case Study result:** Empirical evidence suggested us that differences with regards Organizational Culture are very influential in Offshore Outsource settings. The Vendor was mainly data driven, guided by performance indicators. The Client was mainly driven by perception and overall perspective of the project. Change on requirements (before and after the Scrum introduction) led to contractual discussions and apparent low performance. Our deduction is that contrasting Organizational Cultures in Agile Scrum projects hinders the reduction of time-to-market.

**Proposition’s revised statement:** Organizations with contrasting Organizational Culture will encounter difficulties to adopt Scrum’s core values causing a negative effect over time-to-market.

**Proposition 5:** Large Offshore Outsource Agile Scrum projects using A priori coordination mechanisms will largely deteriorate its time-to-market.

**Case Study result:** Empirical evidence supported this proposition. Due to contractual agreements and Organizational Culture, the Vendor conducted the projects with A priori coordination mechanisms, expecting to have all the requirements upfront and little space
for changes without leading to renegotiations. Specially in Red Project (more than 60 employees) Client's employees recalled constant discussions with the Vendor that limited the project's acceleration.

**Proposition 7:** Offshore Outsource Agile Scrum projects executed with time differences between 2.5h-5.5h will have a neutral effect over time-to-market

*Case Study result:* This proposition was confirmed in both projects. Employees from both organizations agreed that the time difference between India and Belgium (3.5h-4.5h) did not play a role in the project’s final result. The effective decision to assign a Single Point of Contact (SPOC) between both organizations helped them to leverage the time difference.

**Proposition 9:** Agile Scrum’s group commitment and group accountability in projects with over or under control mechanisms will have a neutral effect in time-to-market

*Case Study result:* Mistrust was present in the Red project and the Blue project. Clients and vendors had to exchange Key Performance Indicators to build trust because Scrum’s velocity was not enough to report the project status. There was no sense of group, but instead a clear difference between the Client’s employees and Vendors’ employees, meaning that each group of employees had their own responsibilities. When the projects demanded a group commitment resulting into an extra effort (e.g. extra deployment in the same week), contractual discussions entered to scene, causing discussions and fast deployments.

*Proposition’s revised statement:* Agile Scrum’s group commitment and group accountability in projects with over or under control mechanisms will have a negative effect in time-to-market

**Proposition 11:** Offshore Outsource Agile Scrum projects with physical distance between its collaborators and limited use of CMC tools will have a neutral effect over time-to-market

*Case Study result:* This proposition was supported by evidence from both Projects. The explanation why actually time-to-market was not negatively affected, is the location of a Single Point of Contact. Interviewees accepted that even though they had problems communicating each other, the SPOC was always lightening coordination and communication problems.

In Figure 6-7 we illustrate the modifications made to the Conceptual Model. Bolded propositions are supported by empirical evidence from the Case Study (P1, P3, P5, P7, P9, P11). Non-bolded propositions require further research to accept or reject them (P2, P4, P6, P8, P10, P12).
Figure 6-7. Refined Conceptual Model of criteria hindering Reduction of Time-to-market.

(Bolded propositions are supported by empirical evidence from the Case Study)
6.5 Conclusion

This Chapter disclosed the results of the multiple Case Study based on the Red and Blue Project. The introduction of Agile Scrum practices was an unilateral decision from the Vendor as a response to lighten team and organizational problems causing late deployments. Eventually, the Red Project –the first project to implement Scrum practices – failed meeting all its requirements, but still the Vendor suggested to include Scrum practices into the Blue Project. Our analysis disclosed that both projects did not experience major improvements in the reduction of time-to-market, however it is also important to mention that both projects did not fully implement Scrum as the literature suggests.

Based on the context in which the Red and Blue project took place, we deduct that the criteria with largest negative influence in the reduction of time-to-market were: Culture Differences (National and Organizational), Coordination and Trust. Large differences at personal level and organizational level with respect how the project should be executed hindered Scrum’s potential to reduce time-to-market. In addition Effective Communication and Time Zone Differences, seem to have a neutral effect in the reduction of time-to-market due to the effective role of the Single Point of Contact.
Chapter 7. Conclusions

This research was initiated with the intention to disclose what criteria influence the reduction of time-to-market in Offshore Outsource Scrum projects. The main question driving this master thesis was:

Q.1. What criteria influence the reduction of time-to-market in software development projects that combine Agile Scrum with Offshoring Outsourcing?

To answer it, we first researched what is Agile Scrum framed in the following sub question:

Q1.1. What are the core principles, actors, responsibilities and processes behind Agile Scrum?

Our research concluded that Agile Scrum is composed by seven main principles: (P1) Individuals over processes, (P2) Working software over comprehensive documentation, (P3) Customer collaboration over contract negotiation, (P4) Responding to change over following a plan (P5) Transparency, (P6) Inspection and (P7) Adaptation. Next to this, Scrum requires three actors: A1) The Product Owner, who is the bridging element between the project outsiders and the rest of the Scrum Team. A2) The ScrumMaster. He is a process manager not a project manager. A3) The Development Team. It is a multidisciplinary team in charge of assessing and developing the Product Backlog items.

Scrum actors interact through a series of processes or events, which are critical for the success of the team following the methodology. Those events are: (E1) The Sprint Planning, when the ScrumTeam agrees on the goal to achieve for the current Sprint based on the Product Backlog items. (E2) The Daily Meeting is a moment during the Development event when all the ScrumTeam plan and synchronize their daily activities, having as the Sprint Backlog as the main reference. (E3) Development, after agreeing on the goal, the Development Team internally decides how to tackle and execute every Sprint Backlog item. (E4) The Sprint Review is a meeting where the original goal is revised the Product Owner decides whether or not a system release is executed. (E5) Retrospective is when the Scrum Team analyzes the positive and negative aspects of the recent Sprint cycle.

Agile Scrum’s ability to reduce time-to-market largely depends on transparent, open, and trustful project environments. Pigs (team members) must be able to fluently communicate with each other, and Chickens (Project outsiders) must show high levels of commitment and collaboration towards the project’s goal.

The second sub question of this research was designed to disclose what are their most common problems: Q.1.2. What are the main challenges faced in Offshoring Outsourcing
Software Development projects? To answer it we used as a starting point a systematic literature review conducted by da Silva et al. (2010) that studied challenges present in Offshoring settings, in from which they concluded the top-five challenges tracked between 1999-2009 were: Cultural Differences, Coordination, Time Zones differences, Trust and Effective Communication. After gaining deep knowledge on why each area could represent a problem for Offshore Outsource projects, we deducted why projects fail reducing time-to-market.

Confronting the core concepts behind Agile Scrum and main problems of Offshore Outsource settings, resulted into an Initial Conceptual Model, depicted on Figure 4-3, which explains the plausible criteria influencing time-to-market. Aiming to refine our initial model, we conducted a multiple case study conformed by two projects between telecommunication company, based in Belgium (the Client) and an Indian outsourse service provider (the Vendor). Followed we present the main findings of our research.

7.1 Findings
We can synthetize our findings into two main topics

1) Identification of criteria influencing the reduction of time-to-market.
From theoretical grounds we deducted a set of twelve prepositions that could explain what influence the reduction of time-to-market. The Case Study provided us empirical evidence to evaluate 6 out the twelve prepositions. Following we list the criteria influencing time-to-market supported by empirical evidence:

**Proposition 1:** Teams with contrasting National Culture will encounter difficulties to adopt Scrum’s core values causing a negative effect over the reduction of time-to-market.

**Proposition 3:** Organizations with contrasting Organizational Culture will encounter difficulties to adopt Scrum’s core values causing a negative effect over time-to-market

**Proposition 5:** Large Offshore Outsource Agile Scrum projects using A priori coordination mechanisms will largely deteriorate its time-to-market

**Proposition 7:** Offshore Outsource Agile Scrum projects executed with time differences between 2.5h-5.5h will have a neutral effect over time-to-market

**Proposition 9:** Agile Scrum’s group commitment and group accountability will have a negative effect in time-to-market in projects with over or under control mechanisms

**Proposition 11:** Offshore Outsource Agile Scrum projects with physical distance between its collaborators and limited use of CMC tools will have a neutral effect over time-to-market

Additionally, we also found a very interesting project-specific finding. Agile Scrum was used in the Red and Blue project as a mechanism force a faster software development and lighten the internal problems that were affecting both organizations before the introduction of Agile Scrum. The abrupt and unplanned adoption Agile Scrum practices in ongoing projects was a symptom of hidden agendas responding to major problems. In the case of Red Project, introducing Agile Scrum practices during the last development phase was a 'last-minute
call’ to overcome recurrent communication and coordination problems, which eventually leaded the project fail in its final original agenda.

Both organizations (the Client and Vendor) assumed that compromising team members produce to faster deployments under a semi-Scrum environment would solve the real problems occurring in the project. They lost focus on the source of the problems: poor coordination, mistrust, culture differences and ineffective communication.

2) Implicit evolution of the Agile Scrum methodology.

Further research related to Agile Scrum must take into consideration that Agile Scrum has been under refinement led by its own founders and close collaborators. In Scrum’s early literature (Schwaber, 1995) the methodology was formed by three phases: Pregame-Game-Postgame. Nevertheless, Scrum was later simplified focusing just in one element: the Sprint. We could not find any scientific evidence discussing this evolution. The first suggestion of this simplification dates from a publication of Schwaber et al (2002) and later reconfirmed in eight years later (Schwaber, 2010). This finding is highly important for future researches, because it tracks a gap between the original and the current version of the methodology. Former investigations (Dullemond, et al., 2009; Larman, 2004) have missed this simplification of the methodology and they have wrongly assumed the initial version from 1995. If researchers are not aware of this evolution they will encounter a theoretical and empirical differences between the current Scrum implementation and Scrum’s initial literature, compromising their findings.

7.2 Recommendations

In this section we aim to answer our last question: Q.1.3. What recommendations can be made to clients and vendors aiming to reduce time-to-market while conducting Agile Scrum Offshore Outsource Projects?

The first recommendation we should address to clients and vendors refers to the initial decision to start a project with cross-nation companies. Clients and vendors must deeply assess the implications related Offshore Outsource projects before starting one. Literature suggests that Offshore Outsource projects requires a cautious coupling processes, going beyond than just defining the means of communication or scheduling meetings in a convenient time for all the participants. Coordination, Trust and Cultural elements do play a role in this type of settings. Vendors and clients must realize that when they incorporate Agile Scrum practices, software development projects obtain a socio-technical connotation.

Returning to the core interest to reduce time-to-market, we must refer to the five dimensions present in our Conceptual Model presented in Figure 6-7.

Culture Differences. With regards cultural aspects, incompatibilities related to National Culture and Organizational culture while using Agile Scrum seem to be some of the core criteria hindering time-to-market (Proposition 1 and 3). Organizations and employees must
increase their cultural awareness align their processes towards a common point between both organizations. Reducing Culture Differences and promote a team sense, must be a priority for clients and vendors if they want to implement Agile Scrum. Team building sessions can be good tactic to bring organizations and team members to a common ground.

**Coordination.** This investigation suggests that organizations should avoid *A priori* coordination mechanisms while using Agile Scrum (Preposition 5). Organizations must be aware one of the core concepts behind Agile Scrum is to embrace change, not reject it. Therefore *Ongoing* coordination mechanisms can be the most suitable way to conduct these projects. However, Ongoing coordination mechanisms can bring high variable costs to the vendor and consequently bring discussions about contract agreements and fees. Both organizations should find the most suitable type of contract for the project conditions, avoiding highly fixed contracts (e.g. Lump sum Turn Key) and very loose contracts (e.g. cost plus percentage fee). Removing contract discussions from the

**Time Zone Differences.** Time zones are fixed and organizations must overcome them. The projects analyzed in the Case Study did not present major difficulties to deal with time differences but it also revealed the importance of having a Single Point of Contact (SPOC) in order to avoid project delays. Agile Scrum requires daily interactivity between its collaborators, without such daily interaction, the methodology starts loosing its ability to communicate issues in a fast way. The SPOC seems to be an effective way to deal such time differences, however teams should not substitute team collaboration promoted by Scrum and just rely on the SPOC. Otherwise will find difficulties to find a shared identity and in contrast they will be two separate organizations with a shared pool of tasks.

**Trust.** When high performance is aimed, literature suggests avoiding strict and loose structure controls. Such statement seems very complex to manage in client-vendor relationships. Every party aims to protect their own stake in project. We suggest to tackle this through two flanks: 1) Culture dimension. Trustful environments are closely related to National and Organizational Culture. By bringing organizations and team members together, they will start knowing each other and identifying their core interests. 2) Coordination mechanisms. As we experienced in the case study, changes of requirements usual in Agile Scrum projects, could be perceived as just strategic behavior from one of the parties. The starting point for both organizations should be aim align their coordination mechanisms and contractual agreements, thus trustful environments can bloom.

**Effective Communication.** Outsourcing Offshoring software development projects by default imply the usage of Computer-Mediated Communication tools. However, the Clients and Vendors must be aware that Agile Scrum demands frequent and effective communication between the actors. If Organizations want to reach maximum levels of performance, organizations must use CMC tools, not just to communicate but to minimize physical absence to the lowest levels possible. CMC-tools should be seen as a project investment and not as a project spending.
The main recommendation to Clients and Vendors aiming to conduct this type of projects and reduce time-to-market is that they should not focus in only one of the mentioned dimensions. This research showed that Scrum projects under Offshore Outsource settings demand not one, but a set of conditions to succeed. Organizations must be aware that one specific tactic (e.g. improve the usage of Computer-mediated Communication tools) will guarantee all the benefits promised by Agile Scrum.

7.3 Reflection

This research is the result of several feedback iterations guided by the graduation committee. Agile Scrum and Outsourcing Offshoring software development are very broad topics and I had to constantly refine the research in order to reach the current state. Additionally, I encountered the problem that the projects conforming the Case Study (Blue and Red Projects) were not completely implementing Agile Scrum. On one hand I gained knowledge on how organizations can use Agile Scrum as a mechanism to overcome problems and force team members to accelerate the development process while projects are delayed. On the other hand, it limited the generalization of this investigation and made more difficult to find a common point between the empirical data and literature.

Regarding nowadays literature found in during this investigation, I must say that I found two clear types of literature discussing the topic. The first type is group of ‘best practices’ with little scientific support, originated from self-called ‘experts’ and largely relying on their own experiences. The second group - academic papers - largely relies on a limited group of very promising investigations that were conducted by Scrum’s co-authors (Sutherland, et al., 2008; Sutherland, Viktorov, Blount, & Puntikov, 2007). Jalali et al. (2010), in their Systematic Literature Review in non-collocated Agile Practices concluded that most of the Agile practices were modified based on the context and project circumstances. With current professional and academic experience on Agile Scrum, I also agree with such conclusion.

Practitioners are combining concepts from different Agile methodologies based on the project’s needs (e.g. including Pair Programming into Scrum projects). Future researchers could experience difficulties to find empirical evidence in ‘pure’ Scrum projects because of such combination of practices. Additionally, several organizations have emerged issuing certifications and claiming to be Scrum authorities (e.g. ScrumAlliance, Scrum.org and MontainGoatSoftware). In a not too far future, I would expect more modifications in the methodology emerging from any of the mentioned organizations. Therefore further research will be needed to validate if nowadays’ scientific findings remains to these new editions of Agile Scrum.

7.4 Limitations and Further research

This section highlights the main limitations of the current research and encourages researchers to continue exploring topics related to Agile Scrum and Offshoring Outsourcing projects.
7.4.1 Research Limitations

Clearly this research is not free of limitations and it is our responsibility to acknowledge them. One of the main limitations refers to the topic of Offshoring Outsourcing. In order to answer the second sub question of this research: *What are the main challenges faced in Offshoring Outsourcing Software Development?* we used as a basis a systematic literature review studying the problems in Distributed Software Development Projects covering scientific publications from 1999-2009 (da Silva, et al., 2010). There is an obvious time gap between when this Systematic Literature Review was conducted and the moment when this research takes place. Further research can overcome this limitation use more recent literature discussing Offshore Outsource Challenges.

Additionally, related to the Case Study execution and analysis, we must say that even though during the interviews we identified shared concepts among the participants, we did not reach Saturation. Due to limitations associated to this research (mainly time and budget related) we could not extend the data collection process until reach Saturation. Additionally, the access to team participants was hindered by the interviewee’s availability. Especially in the case of the Red Project, interviews were more limited because of the fact that the project was officially halted and many of the collaborators were either relocated or they left the organizations by the time the started the investigation. Furthermore, it is important to highlight again that both projects (Red and Blue Project) had a partial and late introduction of Scrum practices, limiting our analysis of a ‘pure’ Agile Scrum project, but at the same time disclosing other problems.

7.4.2 Further Research

There is much space for future research and many questions to be answered. We encourage researchers to continue investigating the following areas:

1. **Complement the current research.** We encourage researchers to extend this research with quantitative techniques and also conduct case studies in other industries besides telecommunication.

2. **Contract agreements in Agile Scrum Offshore Outsource Projects.** Literature analyzing on how Agile Scrum is coupled with Outsourcing practices at a contractual level is very limited. Finding the most suitable type of contract for this type of projects and avoid discussions about Service Level Agreements (SLAs) and Change of Requirements (CRs), can be crucial for the projects.

3. **Agile Scrum Offshore Outsource Projects with multiple organizations.** Multi-organization Scrum projects is a topic with little literature available. Having multiple sourcing providers is not uncommon on nowadays business environments. How to coordinate organizations and teams in such environments, can be the next step to follow.

---

Moment when the researcher stops finding new concepts (Corbin & Strauss, 2008)
4. **Plausible irregular adoption of Agile Scrum practices at a country level.** While analyzing the Agile Scrum Certifications issued by the Scrum Alliance (the most renowned Agile Scrum organization), we encountered very interesting information. The top 5 countries holding more Agile Scrum Certifications are: USA, India, UK, Germany, and Finland (See Figure 1-1). It can be interesting and useful for companies and researchers, to investigate why such behavior among practitioners and why mainstream innovation and technology countries such as Japan, South Korea and China are not ranked.
Chapter 8. References


Chapter 9. Appendices

9.1 Agile Principles

(Agile Alliance, 2001b)

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

**To be send by email to the interviewee.

(Company),
(Date)

Dear Participant,

I am a student from the master program: Management of Technology at Delft University of Technology, The Netherlands. I would like to interview you about your participation on Agile Scrum Offshore projects. The purpose of this research is to derive recommendations to Clients and Vendors on how to combine Agile Scrum and Offshore Outsourcing Software Development in a single environment and improve the Client-Vendor Relationships. We will achieve this in two phases. The first phase: look at the core processes of both models and compare them through a structured analysis based on a literature review. The second phase: conduct a Case Study to illustrate our findings. In this last phase is where we request your cooperation.

Our initial analysis shows that this type of projects present many challenges for Clients and Vendors. Some of those challenges are: Time Zone Differences, Coordination, Effective Communication, Culture Differences and Trust. I would like to interview you about any challenge you have experienced while being part of Agile Scrum projects in Offshore Outsource settings.

The individual meeting will last 30 minutes and the results will be completely confidential. In order capture all the ideas discussed, I am aiming to record the interview only if the participant agrees. The results of this research are completely confidential, therefore any sensitive information will not be revealed (e.g. project’s name or employee's name). After concluding the research, I will circulate a copy of my findings among the participants.

Thank you very much for your time and collaboration.

Sincerely,

Adolfo Sanabria
Email: adolfosan84@gmail.com
Mobile: (+31)0681016084
Master Student – TU Delft, NL
### 9.3 Interview Reviews

#### 9.3.1 Red Project

<table>
<thead>
<tr>
<th>Interview#1</th>
<th>Role: Product Owner/Architect</th>
<th>Organization: Client</th>
<th>Mode: Face-to-face</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: June 21st, 2012</td>
<td><strong>Duration:</strong> 35m</td>
<td><strong>Responsibilities:</strong> At the company level he performs the role of IT Architect and Governance. At the project level he was in shared the responsibility to guard the project's requirements and design elements, which along the document were transformed into the Product Backlog.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interview#2</th>
<th>Role: Project Manager</th>
<th>Organization: Vendor</th>
<th>Mode: Face-to-face</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: June 21st, 2012</td>
<td><strong>Duration:</strong> 32m</td>
<td><strong>Responsibilities:</strong> He was in charge of coordinating the different offshore sub teams from Vendor's side. His responsibilities were mainly aligned testing teams to the rest of the development process.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interview#3</th>
<th>Role: Functional Expert</th>
<th>Organization: Vendor</th>
<th>Mode: Phone call</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: July 6th, 2012</td>
<td><strong>Duration:</strong> 30m</td>
<td><strong>Responsibilities:</strong> She played the role of Functional Expert next to other colleague. She was in charge of guiding the development teams while clarifying requirements and avoiding functional mismatches.</td>
<td></td>
</tr>
</tbody>
</table>
### 9.3.2 Blue Project

<table>
<thead>
<tr>
<th>Interview#1</th>
<th>Role: Project Manager</th>
<th>Organization: Client</th>
<th>Mode: Face-to-face</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: June 21st, 2012</td>
<td>Duration: 25m</td>
<td><strong>Responsibilities:</strong> At the Client, the role of Project Manager is divided in two perspectives: Business and IT. The first is related to business cases and customer management. The second involves project scope, schedules and deliverables. She is assigned to the second perspective.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interview#2</th>
<th>Role: Release Manager</th>
<th>Organization: Client</th>
<th>Mode: Face-to-face</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: June 21st, 2012</td>
<td>Duration: 27m</td>
<td><strong>Responsibilities:</strong> She guides the release process since its start until the deployment, helping to solve issues that could block the project. Due to her former experience with Agile Scrum, she was asked to monitor the Agile Scrum process in Blue Project and help other team members in the learning process.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interview#3</th>
<th>Role: Release Manager</th>
<th>Organization: Vendor</th>
<th>Mode: Face-to-face</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: June 21st, 2012</td>
<td>Duration: 35m</td>
<td><strong>Responsibilities:</strong> His role is to govern a release, starting from project preparation up to the release itself. He maintains discussing which project can qualify for the release. Planning and tracking the day-to-day activities is also part of his role.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interview#4</th>
<th>Role: Development Team Member</th>
<th>Organization: Vendor</th>
<th>Mode: Phone Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: July 5th, 2012</td>
<td>Duration: 30m</td>
<td><strong>Responsibilities:</strong> He is part of the Development team and his responsibilities are aligned to develop the Sprint Elements. He is an emerging ScrumMaster when the official ScrumMaster is not available.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interview#5</th>
<th>Role: ScrumMaster</th>
<th>Organization: Vendor</th>
<th>Mode: Phone call</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: June 29th, 2012</td>
<td>Duration: 1h</td>
<td><strong>Responsibilities:</strong> He is ScrumMaster Certified from the ScrumAlliance. He was in charge of building the Product Backlog and guide the Client in the introduction of the Scrum concepts their processes.</td>
<td></td>
</tr>
</tbody>
</table>
9.4 Transcript Sample

Interview General Information
Date: June 21, 2012
Place: Belgium
Interviewer: Adolfo Sanabria Laporte. Male 28 year old
Interviewee: Female, 35-40 years old.
Project: Blue Project
Job Position: Release Project Manager
Organization: The Client

A: Stands for Adolfo
I: Stands for interviewee.

A: First of all thank you for joining us in this research.
I: You are welcome

A: Maybe, can you tell me which is your role within the project? Which are your responsibilities and how you interact with the Vendor.

I: Ok, for the 12.4 release in which Blue Project has been released, I am actually the RPM, the Release Project Manager. So I see the entire release, and how the teams solve out the problems and be sure that open issues or ... or issues are resolved, from the technical and from the practical point of view. With respect Blue Project, I've been asked to see and monitor their Agile Project, although it is not a full Agile project ... probably you already know that... to just monitor a bit because I just have a previous job I have Agile experience. I worked in another company and we had end-to-end set up: from Product Owner to Testing...

A: So it was completely, pure Agile..

I: Well, it is never completely pure Agile, there are some practical issues that you have to keep on mind, but it was further along than it was here basically, because the Product Owner was part of it. The product that we developed was slightly different. The product was very technical product. The developers had a lot opportunity to mold it, to put functionality on it, which is different here. Some of the products we are doing here very marketing minded and driven towards its costumers. That said, we did Agile, I know what it is, and I had a very positive in experience in my previous company.

A: Ok
I: So with Blue Project, they just asked me to monitor and helped out when is it needed, so I attended the Retrospectives and Demos every second Friday and just follow any questions they might have.

A: Ok.

I: Additionally, we want to move towards this way of working and use it in more projects in the future. So we want to see how Blue Project comes out of it, how is it been deployed, what the quality is, what is the time duration, what kind of issues were around of it. And then use it to add some improvements to projects in the future. I think we are looking at 13.1 (the year 2013) to use it as well, if we get positive feedback out of Blue Project.

A: Ok. Just to clarify, Blue Project started this year (2012) or last year?

I: Blue Project, this version, started this year, but I think the project was once on the road map, but I don’t know exactly what reason that they pushed it off. I think it was before my time here.

A: But you joined when it started...

I: Yeah, yeah, I started this version, that’s why I say that this is a very waterfall approach, so you go with requirements, design, your built, your ST/IT (System Testing/ Integration Testing) support. So basically for Blue Project, we went to its basis and everything was done, including design, so it was more iterative development than Agile. So they broke it out into 6 or 7 sprints of 2 weeks. They took all the requirements and design that already had analyzed and basically delivery it in chunks of 2 weeks and deliver it as such. So it is not true Agile.

A: Ok.

I: So for the testing itself, it is the first time for the Client that they will not test at the end when they have the whole packet but as when they move along. So again I think we are having a learning curve. So it is not fully Agile.

A: Ok, that is very good background information for me to fully understand the project.

I: I know that you already talked with PersonX (Vendor’s employee), I think we both come from different views on it. Personally I think that [silence for 2sec.] they already think that they are more hot than they are ...

A: [laughs] Different perceptions

I: It is ok, I am ok with that. Yeah different perceptions.
A: Ok, well based on my literature, we have identified different problems that are typical to this kind of environments when you have projects which are outsource offshore and also Agile. In my Introduction letter I named them: Culture Differences, Coordination, Trust, Time Zone Differences, Communication…. Maybe I would like to ask you if you have faced any of these problems.

I: First of all Blue Project became an Agile Project on request of the Vendor. They kind of went off. Which is good from a vendor, they come with new ideas, with new approaches. I was kind of lay off for us, because they said ‘we want to do this’ and it was kind a decision that they made out for us. Which also means also that we had to get business in the loop as well. I think Business is very uncomfortable with it. I think most of the people within the technical area, with the breakdown, they can see pieces of a certain delivery and they can say ‘yeah this is what I want’ without seeing the full picture. But I think Business, especially because they are so new to the methodology, they are a little bit uncomfortable. The first demo they only can see a fake screen, only a Mock-Up, the next time that you can add something, the next time that you can delete something. So they are very uncomfortable on giving their input at every intermediate step. I think for The Client, if you want to move forward with it, we definitely need to bring Business in the loop. I think also at the moment, how we are organized, like Product Owner… I think it would be very hard for Business to step in and take that role, which is normally the case. I think it will be very difficult; there is a long road for us to go ahead. I don’t think everyone is aligned yet. I am thinking about Business. Also we were kind of force to go into it. I don’t think it is a bad idea, but it was kind of a short notice… a short agree: We need to do it now, because else… you know that kind of thing… which is ok .. its fine. Of course you already have a couple of people a bit more reluctant to have a more open mind.

A: Yeah,

I: I think also the PM (Project Manager), was new to the Agile approach, and she needed some learning curve... you know... Also how we did all the design before, the risk was very low.

A: Ok.

I: But yeah I think definitely Business is uncomfortable.

A: Ok,

I: Within the team itself, I think all in the Client, were expecting from us was to go full blond ... you know ... with all the Product Owner and stuff... but how it was in short notice ... how we already have all the design specifications from the Ateam (team within the Client) they only broke it down and they played as the Product Owner. We didn't have a changing project. We did know 100% what we had to do basically. Definitely there we need to learn as well: Who is going to do what. That is one of the objectives for 13.1. Between the Client
and the Vendor, we can resolve this. Who is the Product Owner, What does the testing mean? ... So there is still a lot of unknowns

A: You think the Vendor should have played a more teaching role... Make the Client go through the methodology? Do you think it was their responsibility?

I: I think it is a two-in-a-box job, we are not giving it only to the Vendor. I think we all have to learn. I don’t think there is only one party. I don’t think they are experts on it. I know other companies that I'd rather to guide in the process both of us, also in the Vendor-Customer capacity because I did Agile, and there was no Vendor... we were the only ones and it was so much easier. I think when you have a Vendor, and that the Vendor needs to rely on the Customer to help out with that methodology, you might even need a third party to assist because else there is always this friction. I think in a lot of discussions you have, there is friction towards the Vendor. With respect budget, with respect they way they work, with respects the rigidity. I was amazed they asked to make Blue Project to become Agile because they are quite attached to their rules and their processes and even too much. With Agile... it's agile ..it is the word. If something changes you have to pick up. For instance in the waterfall, if you get a CR (Change Request). You need approvals, you need governance. With waterfall you usually have the pressure at the end, but with Agile you have it all the time, and you don't know what it will be at the end. I don’t think the Client is ready for that. If you see Business, they are so uncomfortable with that.

A: There is idea of Culture Differences; do you think there is any difference between the Client and the Vendor? Especially at personal level that could affect the relationships...

I: Oh yes, yes. Flemish we are very straightforward, very abrupt. Must of us are not afraid to say 'No' or we are not afraid that something is ‘wrong’. I think the Indian culture, and also in the previous company I worked. I was in the testing team, and we really had to build that team to acknowledge that something was going wrong. Also here (in the Client) we don’t immediately know that something is going wrong, they are very afraid to lose face and just say: 'oh this is not working'.

A: mm Ok.

I: But I think the Client has learned to deal with that. But there is always discussion, there is always confrontation: 'Why things are not delivered'. A lot of the Client’s employees thing that sometimes it’s excuses: 'Just do it'. We try to work as good as possible as good as we can. Sometimes it is just easier in certain teams and sometimes it is more difficult. Also towards Business, there is some reluctance towards their deliverables, they way they work, because it takes so long, and time-to-market is too long. They always have to pay extra; there is always a CR (Change Requests). So if we want to go to full Agile, definitely discussions regarding CR and SLA (Service Level Agreements) have to be opened up again.

A: Ok,
I: If we go full Agile, that should be rehashed, reformatted, because the way it is now it will not work. It is very waterfall.

A: How do you usually coordinate your tasks within the team? Do you think it is aligned with the Agile methodology, following the Product Backlog, and so on...?

I: Again for Blue Project, the Agile part, is pure development. They chose this project because it is a stand-alone piece of code. They have sole ownership. They are doing development and IT and we do the rest. Our function is to back sit Product Owner basically. It is very clean-cut I think.

A: Ok. Have you faced problems with Time Zone Differences? Do you think it is a problem?

I: I don’t think it is a problem, I don’t see it as a problem.

A: You think it could actually it is a benefit?

I: I think it is not relevant, it is just development. We have that demo every two weeks, and that’s when everyone get’s together, and everyone has a view of what is happening. But when you move to the testing phases, which again I can only stress that it is not Agile, then of course it does play. A team picks up issues and couple of hours the other analyzes it. But at this stage, it is not relevant.

A: Ok, Have you faced problems with trust? I know that the Client and the Vendor have been working for years, but do you think there are issues related to Trust?

I: It all depends on who do you work with. So yes, there are some issues with Trust. Towards new people, especially new people coming to the team. I have been in the Client for already two and a half years, and you see, as soon as new people are coming especially in leading positions: being a PM (Project Manager), RPM (Release Project Manager) a Technical Lead, you have to rebuild that Trust.

A: Mmm

I: The Trust do not come from the Vendor, but from individuals. Recently, I worked with a young guy, and he was superb! I could just give him a task and ask him ‘do this’. And he would come back, and communicate with us, with the technical analyst of functional analyst here, and just whatever needs to be done, it is done. But that it is not always the case. So it is really a person-to-person situation. I think that for Blue Project they have very qualified people. Definitely there is good Trust in that Team, but in Red Project there was no Trust. No Trust between the PM, No Trust between the Vendor, it was mayhem.

A: ok, ok.
I: Have you faced any problems with Communication. The fact that most of them are abroad.

A: Typically they have onshore-offshore situation. For every offshore team, there is a person on site. So that’s how the communication flows, and how I said before, if that person is very good, then not no problems. But in general I think it is good. Only if this new person is coming you have to build again that relationship. But I would expect the Vendor to put more emphasis on it, because we go through the same thing over and over. For instance, from the RPM position we had 3 new RPM in the last year. So you see, then they come, they know all the procedures, all the calculations for every CR and SLA and what they should and they should not accept. And you have to break that down a bit because the practicality is out of it. Every time you have discussions about production issues found. That it is very painful. They all are very good developers, very good analysts and they all are individuals.

A: Yeah of course they are also persons right! This research tries to help to overcome those frictions, those problems.

I: The frictions definitely come from the boundaries of certain processes, for instance SLAs. There is no gesture, there is no… They also don’t think ahead for instance ‘I would fix this the customer would be so much happier’. I think they hold too much to the rules and processes. I find it a little bit... sad actually. Because at the end we are a customer, and sometimes we don’t feel that way. For instance: We have a new member, and he is very attached to the rules, and if something critical appears, and we need to make an extra patch. He says no because we only have 2 patches per week. But we explain that it is the last release, and we need it. So that back and forth is very very stressful and it drains energy, from where it is supposed to be going.

A: I understand. Well, I think we have covered basically everything I wanted to discuss. We talked about Culture Differences, Coordination, Trust, Communication, Time Zone Differences. Do you think out of this five challenges, is there anything that we haven’t covered?

I: Well the only thing that I would like to add. Well your research is about one vendor right? Also in your letter you mentioned it.

A: Yes,

I: I think in other cases here within the Client, there are other cases with multiple Vendors, and definitely we are not ready to go Agile. If I build my piece, how are we going to coordinate with others? I think moving forward to Agile will not work for all the projects. I think also what it is important, is transparency. Which sometimes also lacks. What it is happening on the other side? Maybe that’s coming back to communication, sometimes you have good communication sometimes not. Sometimes you get the feeling that behind what they want, behind what they decided, there is a bigger thing. Why they wanted to Blue
Project become Agile after Red Project went out under the train! Red Project doing even more Agile than Blue Project. Why they want to do that? I wonder!! Do they want to prove that they can do it?? Well now I sound very pessimistic, but well they only have proved that they can do Iterative development but not Agile. I think the Vendor might look it differently.

A: Ok thank you for all these information ...

[Closing chat informing the next steps to follow for the research]
### 9.5 Specification Coding Phase

#### 9.5.1 Codes from Vendor’s Employees

<table>
<thead>
<tr>
<th>Role</th>
<th>Topic</th>
<th>Culture Difference</th>
<th>Coordination</th>
<th>Time Zone Differences</th>
<th>Trust</th>
<th>Effective Communication</th>
<th>Additional Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager – Red Project</td>
<td></td>
<td>Long-term relationship, Culture Awareness, Verbose communication, Perception communication, Data Driven communication, Key Performance Indicators,</td>
<td>Learning process, Communication overhead, Sprint Backlog</td>
<td>Time Difference Irrelevant, Asynchronous coordination</td>
<td>Velocity misunderstanding, Burndown chart, VPN. Cross-team Velocity</td>
<td>Unclear goals, Learning process,</td>
<td>Not Agile, Change, Sprint Cancelation, Little Business involvement</td>
</tr>
<tr>
<td>Functional Expert - Red Project</td>
<td></td>
<td>Long-term Relationship, Internal Coordination, Task Dependencies, SPOC,</td>
<td>Time Difference Irrelevant,</td>
<td>Unexpected events</td>
<td>Jira, Teleconference, SPOC, Language misunderstandings</td>
<td>No SPOC from Client, Product Owner</td>
<td></td>
</tr>
<tr>
<td>Release Manager – Blue Project</td>
<td></td>
<td>Culture Awareness sessions, Culture understanding, Appreciation, Define Expectations,</td>
<td>SPOC, Involvement of Business, Client’s appreciation?, Certifications, burndown charts, Daily Meeting,</td>
<td>Leverage Time Zone Dif.</td>
<td>Performance reports, Burn down reports, define ‘good feeling’, openness, transparency,</td>
<td>Language misunderstanding, teleconference</td>
<td>Red Project, Expectations</td>
</tr>
<tr>
<td>Scrum Master – Blue Project</td>
<td></td>
<td>(lack) Involvement, Long-term relationship, Discussions about quality, way of working</td>
<td>Shared Space, sprint duration, unforeseen problems, cross-team coordination,</td>
<td>Time, Money discussion, Pilot Project, benchmark, velocity, predictability,</td>
<td>Teleconference, videoconference, Sprint Review, Involvement Business, SPOC,</td>
<td>Reconciliation process</td>
<td></td>
</tr>
<tr>
<td>Development Team – Blue Project</td>
<td>Way of working, Resolution time, Not national difference, Sense of Urgency</td>
<td>Multiple-follow up, additional Effort, SPOC, Jira, TZD Delay, TZD advantage, Red Project, Involvement of Business, Quality Improvement, perception, assumptions, Communication gap, Requirement interpretation, Product Backlog, Technical Limitations, On-site Demo, Distractions, Third-party dependencies,</td>
<td>unforeseen delay, daily report, transparency, pilot project, Internal Confidence, Positive Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## 9.5.2 Codes from Client’s Employees

<table>
<thead>
<tr>
<th>Role</th>
<th>Topic</th>
<th>Culture Difference</th>
<th>Coordination</th>
<th>TimeZone</th>
<th>Trust</th>
<th>Effective Communication</th>
<th>Additional Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Manager – Blue Project</td>
<td>Resistance to change, Frictions (budget, time, time, payment), Rigidity, processes, Directness, Culture awareness, lose face, Frictions (time, money, delivery), Rules attachment, Customer orientation</td>
<td>SPOC, Product Owner-like</td>
<td>Time Difference irrelevant</td>
<td>Person-to-Person, rebuild trust, Red Project, Vendor’s transparency, Vendor’s capacity</td>
<td>SPOC, Employee Turnover</td>
<td></td>
<td>Agile with multiple vendors</td>
</tr>
<tr>
<td>Project Manager – Blue Project</td>
<td>Language misunderstanding, Face-to-Face communication, Vendor’s Imposition to Include Scrum, Red Project, Structured Processes, No training.</td>
<td>SPOC,</td>
<td>Time Difference irrelevant</td>
<td>Money, Scope, Estimation,</td>
<td>SPOC</td>
<td></td>
<td>Agile-But, Change of Perception</td>
</tr>
</tbody>
</table>