Requirements Engineering at a Distance

Adapting the Requirements Engineering process to accomodate different Sourcing Strategies in a structured project environment

D.Y. Kusters
Information and Computing Sciences
Master of Business Informatics
18 June 2018

Universiteit Utrecht

Supervisors
dr. Fabiano Dalpiaz f.dalpiaz@uu.nl
dr. Joske M. Houtkamp j.m.houtkamp@uu.nl

Daily Supervisors
Bram van Dongen
Koen Vermeulen

Abstract

This thesis reports on requirements engineering in the context of sourcing strategies in a structured project environment. We concentrated on four common types of sourcing strategy: offshore outsourcing, offshore insourcing, onshore outsourcing and onshore insourcing, where the latter is only used as a reference. Both requirements engineering and sourcing strategies are a continual source for project failure, yet no concrete set of practices of requirements engineering exists to adapt to a specific sourcing strategy. The aim of this thesis is to evaluating the possibility of such a list and creating best practices for requirements engineering adaptation. The research methods are based on the Technology Transfer Model to create a solution based on both academia and industry best practices. To that end, the steps of this model are applied with initial industry workshops and discussions to ground the current issue followed by alternating two rounds of literature research and two rounds of interviews, with three experts interviewed for each round. The latter was carried out by taking the Dutch cable company Ziggo as a case study, which performs projects in a structured environment by using a method compatible with Prince2. The results from this research provide evidence that a requirements engineering practice can be adapted to a specific sourcing strategy. The relation between adaptation and distance is direct; increases in sourcing strategy distance also increases the number of requirements engineering practices suggested. Accounting for incompatible techniques, a framework was created for offshore outsourcing with 48 best practices, offshore insourcing with 32 best practices and onshore outsourcing with 11 best practices. These sets of techniques can be used in a practical setting, picking and choosing which has most value for a specific case or project to overcome the difficulty aspects of using a sourcing strategy.

Keywords: Requirements Engineering, Sourcing Strategy, Outsourcing, Insourcing

Acknowledgements

First of all I would like to thank my supervisors dr. Fabiano Dalpiaz and dr. Joske Houtkamp. Writing a thesis did not come naturally to me but with their help it became possible. Having the structure provided by weekly reviews from Fabiano was invaluable for my continued progress. My sincere thanks to Bram van Dongen and the information analyst team at Ziggo for their enduring interest and availability for every iterative round that my thesis went through. Enthusiasm was not hard to find and each conversation was engaged with imagination. Heartfelt thanks for my parents who kept thinking out of the box when I myself was not able to find a solution. Giving bigger picture advice is not always easy to give or receive but persistence did wonders. Furthermore I would like to thank Femke, for her everlasting patience, support and a place to come home to.

Table of Contents

Glossary

Some of these definitions are debated in the literature; the reader should consider the formulation presented here as an attempt to use a shared vocabulary in the thesis, rather than an attempt to resolving an open debate in the research community. First there is a list of definitions with their abbreviations in alphabetical order. Then there is a diagram that visually relates these words, putting them into context of each other.

Captive Unit	An offshore branch of the client organisation.
Client Organisation	Entity that is the target source for the to be executed project. In the context of the Ziggo case, this would be Ziggo itself. The client organisation is the business of which “business goals” would refer to.
Collocated	Collocated refers to project members being situated on the same location.
Distance	Unless stated otherwise, distance in this thesis refers to multiple relevant factors of separation between a client organisation and the executing party. The different aspects of distance are geographical, temporal and cultural. An increase in distance may entail a country that is further away, in a different time zone and with a more disparate culture. In the context of sourcing strategies these three metrics have often been lumped together (Agerfalk et al., 2005)
Domestic	An action of part of development is executed domestically; in the same country as the client organisation.
Executing Party	Entity or branch that executes further steps in the software development life cycle.
Farshore	In relation to the client organisation’s geographical location, farshore refers to a relation with an executing organisation that is situated in a country that is ‘far away’. In most literature this refers to India, when viewed from the perspective of western countries where most academic resource originates.
Global Software Engineering(GSE)	“When organizations shift all or part of their software development to another country (referred to as offshoring)” (Verner, Brereton, Kitchenham, Turner, & Niazi, 2014, p. 55). There are many more nuances surrounding this term, but generally speaking a reference is made to software development happening across multiple sites in multiple countries. Terms found in literature which refer to essentially the same phenomenon are: Global Software Development (GSD), Distributed Development and Distributed Software Development (DSD). In this thesis Global Software Engineering is the same as “offshore”, as most case studies on GSE in academia look at straightforward offshore development, whereby some of the initial stages of a project and stakeholders preside in the client organisation country.
Inhouse	A term meaning (part of) development is done by the client organisation (Insource) as well as on the same location as the client organisation (Onshore). Often this refers to a collocated situation, but this is not always the case.
Insourcing	“Leveraging company-internal human resources”(Šmite, Wohlin, Galviņa, & Prikladnicki, 2014) In other words, the client organisation and the executing party are the same entity.

Nearshore	"Leveraging resources from a neighboring country" (Šmite et al., 2014) In other words, nearshore is a type of offshore strategy. In relation to the client organisation's geographical location, nearshore refers to a relation with an executing organisation that is situated in a country that is 'near' the client organisation. In most literature this refers to Eastern Europe, when viewed from the perspective of western countries where most academic resource originates.
Offshore	"Leveraging resources from a different country" (Šmite et al., 2014) In other words, the client organisation and the executing party are not in the same country. Often offshore in literature refers to far off countries, or typical countries to export (software development) labor to, such as India.
Onshore	"Leveraging resources from the same country" (Šmite et al., 2014) In other words, the client organisation and the executing party are situated in the same country.
Outsourcing	"Leveraging external third-party resources" (Šmite et al., 2014) In other words, when the client organisation and the executing party are separate legal entities, and the client organisation contracts the executing party to perform software development activities.
Sourcing Strategy	Sourcing strategy is the chosen relationship between the client organisation and the executing party, with the intent of optimizing in some aspect the development process. In this thesis four separate sourcing strategies will be considered: ● Offshore Outsourcing, where the client organisation and the executing party are two separate entities and are located in different countries. ● Offshore Insourcing, where the client organisation and the executing party are the same entity, but are located in different countries. ● Onshore Outsourcing, where the client organisation and the executing party are separate entities and are located in the same country. ● Onshore Insourcing, where the client organisation and the executing party are the same entity and are located in the same country.

Figure 1 shows how these terms relate to each other, in relation to sourcing strategy. All flavors under sourcing strategy can be divided into the four broader categories as explained above in the glossary.

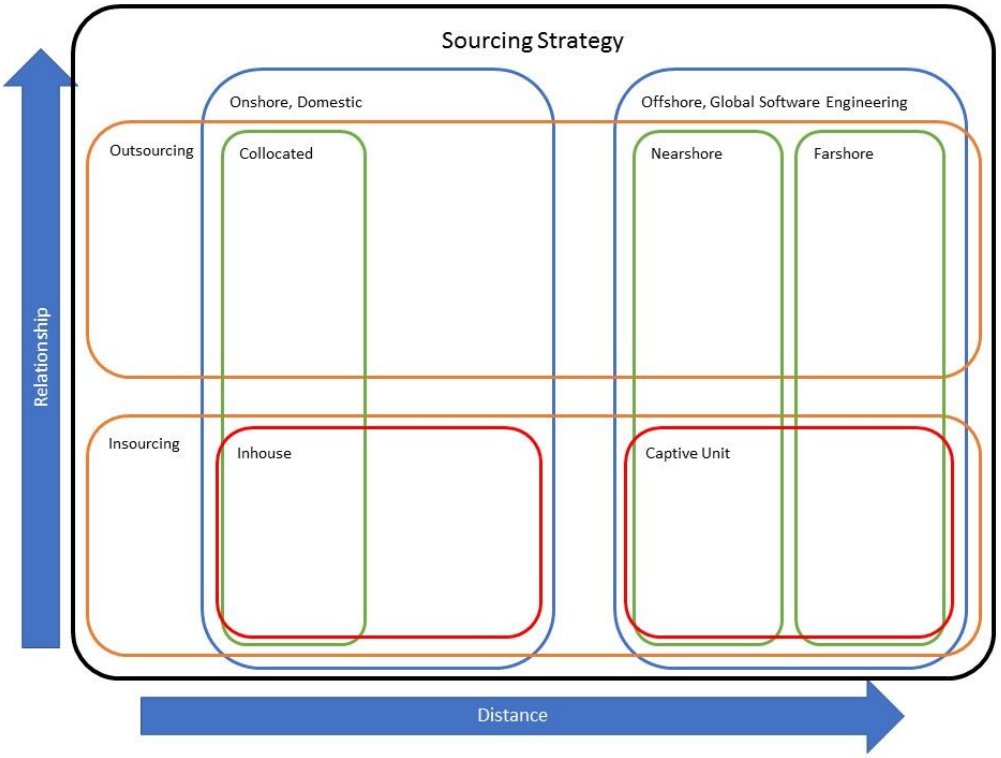

Figure 1. Taxonomy of sourcing strategies

1 Introduction

Requirements form the ground layer of every project, but the process of gathering, evaluating and communicating these requirements is historically (Somerville & Sawyer, 1997) (Hofmann & Lehner, 2001) (Martin, Aurum, Jeffery, & Paech, 2002) and still a main source for project failure (Han & Huang, 2007). Requirements Engineering is even harder when considering special circumstances, such as different development teams at different locations (Bhat, Gupta, & Murthy, 2006). Many causes are blamed for this difficulty, but most can be reasoned back to geographical, temporal and cultural distance that make distributed work challenging (Ågerfalk & Fitzgerald, 2008). Requirements engineering is not the only problem plaguing global software engineering (GSE) and sourcing strategies, but it is consistently one of the most predominant ones. In a study into the early warning signs of failures on offshore software development projects, Philip et al. (2010) found that 5 of 10 of the top warning signs can be related to requirements engineering. This makes requirements engineering a topic of interest when considering sourcing strategy.

Many studies have examined specific sourcing strategies, such as offshore outsourcing, but these studies often consider the entire software development picture and do not go into the requirements engineering details. The same applies to studies that compare sourcing strategies. Examples of this are Nakatsu and Iacovou (2009) who compare onshore and offshore outsourcing and Prikladnicki and Audy (2012) who compare offshore outsourcing with offshore insourcing. These studies however do not have a focus on requirements engineering. Other works study requirements engineering in one sourcing strategy direction, such as Ali and Lai (2015a) who presented a framework for GSE requirements specification and validation. This leaves an interesting gap for an overview when comparing sourcing strategies specifically on the topic of requirements engineering.

Tangential to these issues this study will look into the case of Ziggo, a Dutch cable company, which merged in 2014 with UPC. Ziggo operates a near nationwide cable network, the IT landscape of Ziggo servicing that network and other operations span over 1200 applications that are maintained and developed by a broad range of organisations with varying connections to Ziggo. Within this technology landscape, the new merged company “Ziggo” has to cope with an ongoing merging process, and while both previous companies had similar business goals, their processes and practices were vastly different. Former Ziggo was more adept with waterfall-like project structures with clear and defined project phases and deliverables, while UPC took to the development practices of its parent company Liberty Global and used agile-like methodologies. For requirements engineering this meant that former Ziggo had a very defined and extensive project start-up phase and detailed documents at each stage describing requirements and their many aspects which could slow down a project. On the other hand former UPC had less documentation and knowledge regarding the details of requirements was often tacit and held by those responsible for their implementation, which resulted in a faster progression rate but could have more fault sensitivity. Both approaches to requirements engineering were found to have their own unique issues and now these parties have been brought together, the search is renewed for a requirements engineering approach that is appropriate for the many different stakeholders involved and the various sourcing parties that Ziggo is partnered with. Specifically, the different approaches to requirements engineering and established practices are not always compatible, which limits the interchangeability of project resources. There was a need for a common approach, when dealing with different partners that is grounded in theory but that is also effective and efficient with limited project resources.

The current way of working is a Prince2 project structure, as can be seen in Appendix C and D. This makes the case study of Ziggo relatable to many other companies with a formal project structure

following the common Prince2 guidelines, and the requirements engineering insights when looking at sourcing strategies applicable to formal project structures in general.

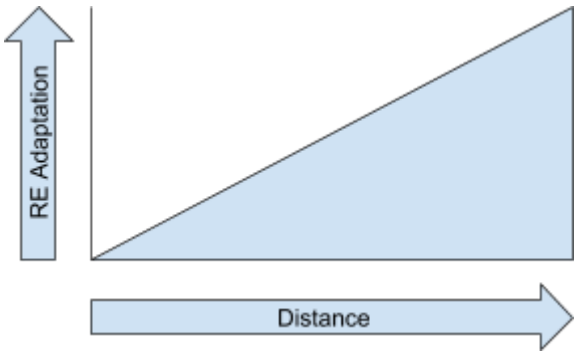

Figure 2. Assumed relationship between requirements engineering adaptation and distance

A quick survey and discussions among experts about requirements engineering at a distance show an expected relationship between requirements engineering adaptation (adapting your requirements engineering practice or adding additional techniques) versus the distance between the client and the executing party. These discussions are summarized in create the diagram shown in *Figure 2*. Noteworthy is that the requirements engineering experts think distance consists of multiple aspects, as the client can be removed also in cultural or temporal distance as well as geographical distance. These aspects of different types of distance can also be seen in the work of (Šmite, Wohlin, Gorschek, & Feldt, 2010) ((Šmite et al., 2010). As seen in *Figure 2*, the amount of prudent requirements engineering adaptation increases linearly with increases in distance. How in practice this affects requirements engineering however is still subject to debate.

As seen in both literature and practice there is a need for a practical solution and differentiation for the sourcing strategy problem when looking at the requirements engineering process. Therefore this research will try to answer if and how different sourcing types (offshoring, onshoring, outsourcing and insourcing) will affect requirements engineering and create a framework on how to approach this problem for each sourcing strategy.

Formally, this makes the hypothesis of this thesis:
“The requirements engineering practice can be adapted to accommodate different sourcing strategies”

And accompanying research question to benefit practice:
“How can the requirements engineering practice be adapted to accommodate different sourcing strategies?”

To that end some basic parameters have to be established. The requirements engineering process needs to be defined, the different sourcing strategies have to be described and generally how these impact software development projects.

This leads to the following sub questions to answer the overall research questions:

1. How to compare requirements engineering in different structured project environments?
2. What are sourcing strategies and how can they be categorised?
3. How do sourcing strategies affect software development in a structured project environment?
4. How can the requirements engineering practice be adapted to accommodate a specific sourcing strategy?

Sub questions one, two and three are answered through a literature review in Chapter 2, which lays the groundwork for subquestion four to be answered. In Chapter 3 the research approach is explained to this end. In Chapter 4 the chosen requirements engineering framework is detailed to map all requirements engineering adaptation on to. In Chapter 5 the results of extensive literature research into sourcing strategy and requirements engineering are covered. These results are analyzed in Chapter 6 on both quantitative as well as qualitative aspects. Chapter 7 presents our conclusions regarding the research questions and hypothesis, followed by the appendices.

2 Literature Review

This chapter serves as a critical evaluation of the literature on the topic of the relevant requirements engineering and sourcing strategies' state of the art. To explore the research question a couple of issues will be discussed such as the requirements engineering process, different typologies of requirements engineering elements, the different existing sourcing strategies and what the sourcing strategies entail.

2.1 The requirements engineering process

There are two orientations of constructing a requirements engineering process. The most common is the activity based requirements engineering approach. In this approach a set of activities dominate the process, with the philosophy that the right set of activities with their own purpose and techniques executed in predefined steps will fulfill the requirements engineering needs (Nuseibeh & Easterbrook, 2000). The other paradigm is the artifact based requirements engineering approach, which advocates the importance of specific requirements engineering artefacts to be produced and has no emphasis on how exactly these artefacts are to be created. An example of artefact based requirements engineering is the AMDiRE approach (Daniel Méndez Fernández & Penzenstadler, 2014).

Activity based requirements engineering approaches have a long history, have earned their place in most software development methods such as the Rational Unified Process (Kroll & Kruchten, 2003) and the waterfall and agile methodologies modeled after the Software & Systems Process Engineering Meta-Model (SPEM) (Ruiz-Rube, Dodero, Palomo-Duarte, Ruiz, & Gawn, 2013). These activities and what process they prescribe can be viewed as modular, and to adapt or create an activity based method to suit a specific situation is the field of Situational Method Engineering (Brinkkemper, 1996).

Situational method engineering can be used to create an activity based process for requirements engineering. The method would allow the evaluation and comparison of different activities and techniques as well as the process as a whole with counterparts to create an optimal requirements engineering process. Documentation for a situational method can be done with the use of a process deliverable diagram (Weerd, Versendaal, & Brinkkemper, 2006). This documentation technique matches up sequentially the activities with their desired deliverables and dependencies.

Some empirical evidence exist to compare these two paradigms of activity and artefact orientation, such as a case study that tests the effectiveness of both (D. M. Fernández, Lochmann, Penzenstadler, & Wagner, 2011). This case study compared the effectiveness of a customizable requirements engineering method called BISA (Méndez Fernández & Kuhrmann, 2009) to a native activity based approach in the context of business information systems. The results show that an artefact based approach is viable in practice and can be flexible and generate artefacts of high quality, but is not necessarily more productive. The creation of an artefact based approach requires significant effort but this is justified for their benefits according to their advocates (D. M. Fernández et al., 2011). The artefact based approach is young however and relatively untested in practice. With most literature on requirements engineering revolving around activity based processes, sources build upon that orientation will be used during this study.

To examine how sourcing strategies differ with regards to requirements engineering, it is useful to break down what requirements engineering elements exist within an activity based requirements

engineering process, as these elements exist in academia and provide a basis for understanding when the sourcing strategy effects are compared.

Traditionally Requirements Engineering is attributed five core activities (Cheng & Atlee, 2007a). These five activities are:

- *Elicitation*, which is the comprehension of underlying goals, motives and objectives.
- *Analysis*, which is the refinement and negotiation of requirements and resolution of conflicts among stakeholders.
- *Specification*, which is the precise documentation of the users' needs and constraints.
- *Validation and verification*, which is about ensuring the well formedness and consistency of the requirements.
- *Management*, which is the coordination of requirements engineering activities and changes to requirements as the project develops.

Between different sources, the list may vary on some specific names of the core activities, or the activities were grouped together differently. For example older iterations of state of the art requirements engineering reviews, such as (Nuseibeh & Easterbrook, 2000) include the activity of agreeing upon requirements, which is about the negotiation between stakeholders on requirements and the activity of communicating requirements. Other lists may contain activities labeled documenting or modeling of requirements.

This leaves a discussion on which 'list' of requirements engineering activities to use. For that purpose the latest version of the most prominent requirements engineering practice frameworks are summarized and discussed here, with the aim of choosing the most useful framework for this thesis.

A framework based on best practices learned from industry is the SWEBOK (Software Engineering Body Of Knowledge). In the Version 3.0 book (IEEE Computer Society, 2014), chapter one is devoted to software requirements, including their process and out of which elements it consists. The activities noted closely align with those given by Cheng and Atlee (2007a), including: Requirements Elicitation, Requirements Analysis, Requirements Specification and Requirements Validation. Requirements Management is not included (in this version of SWEBOK).

Some of the early frameworks of requirements engineering elements or activities aim at evaluating (and improving) the current maturity of the process. Models like the Capability Maturity Model Integration (CMMI Product Team, 2006) included requirements engineering as a subcomponent, but arguably have a wanting detail level. Later frameworks, such as the Uni-REPM by Svahnberg et al. (2015), have been tailored specifically to requirements engineering. This framework contains the following seven main elements:

- *Organisational Support*, which are activities from the surrounding organisation that support the requirements engineering process, such as training of personnel.
- *Requirements Process Management*, which focuses on the structure and formality of the process of requirements engineering. Examples of this are change control and process validation.
- *Requirements Elicitation*, which are the methods of acquiring (from discovery to understanding) requirements from stakeholders.
- *Requirements Analysis*, which are activities around creating a complete frame of requirements without gaps, such as contradictory requirements.
- *Release Planning*, which focuses on packaging requirements in releases in the context of priority and development effort.
- *Documentation and Requirements Specification*, which are activities looking at formatting of requirements into practical and usable artefacts.

- *Quality Assurance*, which are activities that explore the integrity of the developed requirements according to set standards.

The SWEBOK requirements framework notes that requirements engineering is an iterative process and activities often happen at once and are interlinked, but no further details on these links are given, making their relations when set in various sourcing strategies less predictable. The Uni-REPM framework has a strong selling point in maturity measurement and associates activities to maturity level - with additional or more advanced activities being better. This however goes against the notion of optimizing a set of activities to cope with a specific sourcing strategy. Academically a higher level of maturity would be preferred to perform requirements engineering with any sourcing strategy, but practically an organisation might not be able to commit the resources for more advanced methods. Picking and choosing technologies, activities and processes that have proven value for that specific sourcing strategy would be more suitable.

For that reason, this research makes use of the book Requirements Engineering: Fundamentals, Principles, and Techniques by Pohl (2010), which combines all aspects of requirements engineering activities into a holistic framework and demonstrates how activities within the framework affect others. An exhaustive summary on the framework and all containing elements is given in Chapter 4.

2.2 Sourcing Strategy

In the business literature, the term *sourcing* refers to the leveraging of resources (Ritzer & Lair, 2007). When making a choice on how those resources are sourced a sourcing strategy is formed. The term is sparsely used throughout literature, as normally in literature a direct reference to the type of sourcing strategy is made, such as “outsourcing” (Šmite et al., 2014).

Sourcing strategy should not be confused with the term strategic sourcing, although related, Strategic Sourcing refers to the process of continually making choices on optimal sourcing, with the goal of minimizing the costs of production and/or maximizing the value to customers (Anderson & Katz, 1998). As such, strategic Sourcing may result in a sourcing strategy but the two terms are not similar. Strategic Sourcing is part of the supply chain management domain.

Generally speaking, classifications of sourcing strategy can be made against the dimension *location* and the dimension *relationship* between the involved organisations (Ågerfalk & Fitzgerald, 2008; Hofner, Mani, Nambiar, & Apte, 2011). On the location dimension Sourcing can be done either close to the client organisation or far away, with the associated terms “onshore” meaning the executing organisation is in the same country as the client organisation, and “offshore” means that the executing organisation is in a country other than the client organisation, typically in a country far away. On the relationship dimension you find how the client organisation and the executing organisation are related. Two opposites are determined, being “insourcing” which means that the client organisation uses its own resources to execute the task or “outsourcing” in which a third party is contracted to perform a part of the tasks.

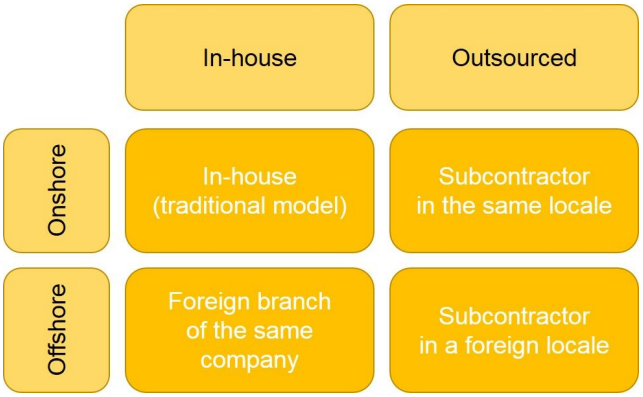

Figure 3. Sourcing Strategy classification (Ågerfalk & Fitzgerald, 2008)

An example of this categorisation is sourcing strategy classification seen in *Figure 3*. Other examples of classification may add a dimension of time or more detail in the location dimension, such as the classification done by Šmite et al. seen in *Figure 4* or even more elaborate taxonomies meant for additional purposes such as effort estimation, for example the specialized GSE Taxonomy created by Britto, Mendes and Wohlin (2016) which includes dimensions of type of architectural model and estimator role.

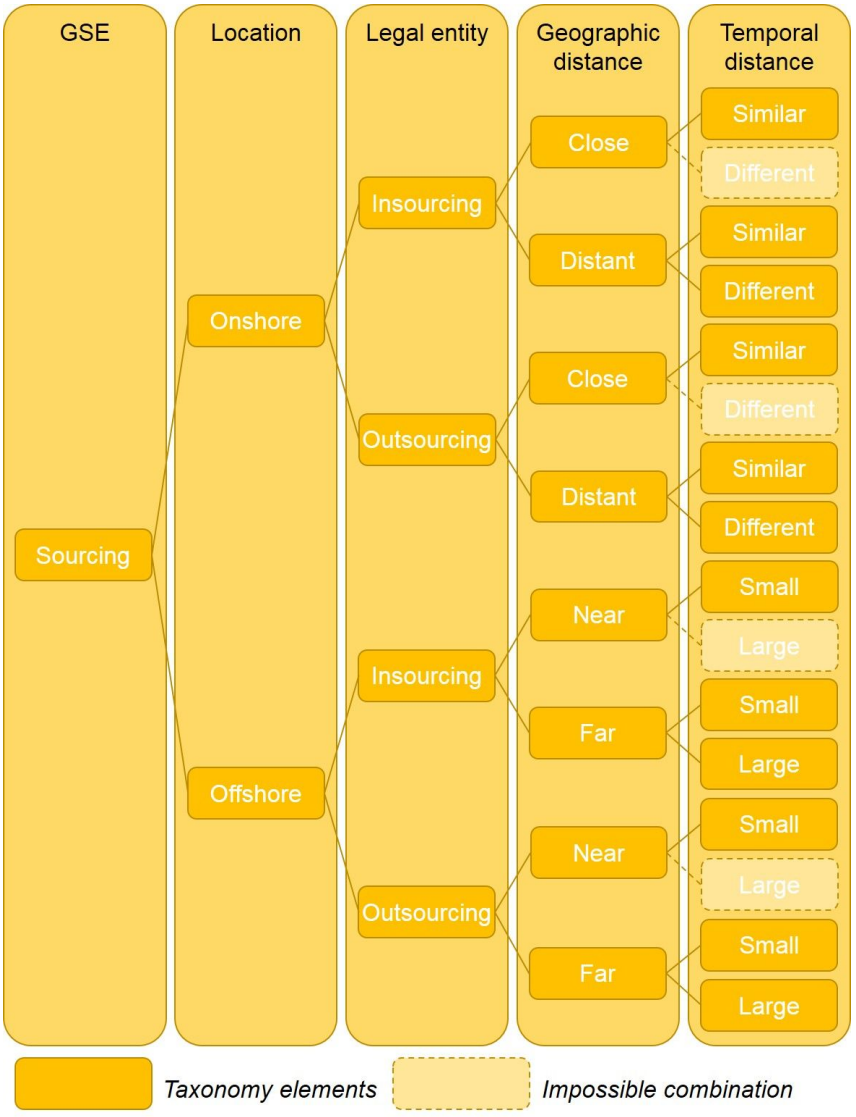

Figure 4. GSE Taxonomy (Šmite et al., 2014)

There are many additional technical terms indicating specific sourcing strategies, such as “nearshore” which means the executing organisation is based in a country close to the client organisation and the opposite definition “farshore” which has the executing organisation in a far away country (Carmel & Abbott, 2007). The full spectrum of sourcing strategies is described below in *Figure 5* and definitions of all terms can be found in the glossary.

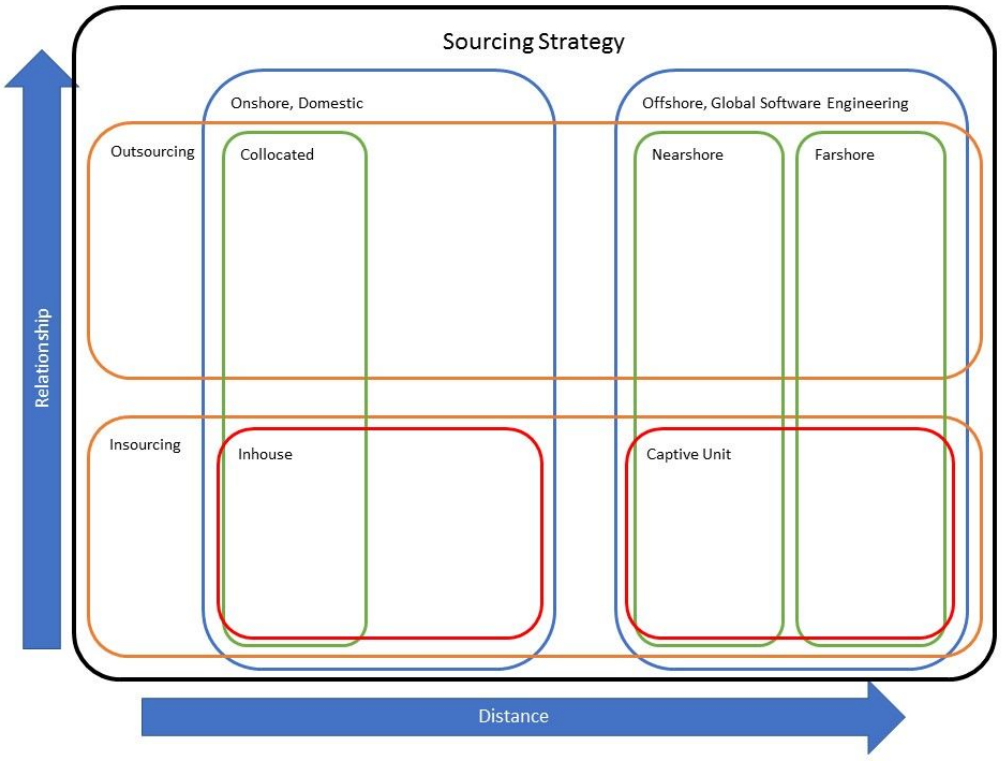

Figure 5. Sourcing Strategy taxonomy on the axis of relationship and distance between organisations

In this thesis, only the four main sourcing strategies as described by Ågerfalk and Fitzgerald (2008) are part of the research, as these are the most common and well known sourcing strategies, with a foothold in academia that makes literature research viable. These four sourcing strategies encapsulate all other variants. These strategies are:

- Offshore Outsourcing
- Offshore Insourcing
- Onshore Outsourcing
- Onshore Insourcing

Some of the literature encompasses multiple sourcing strategies, based on their relation or distance, and thus can be referenced to component terms of the four strategies, such as “offshore” meaning all sourcing which is done not in the country where the client organisation resides, or “outsourcing” meaning the executing party is a different organisation than the client organisation. Another term often crossed in literature is “Global Software Engineering” (GSE) which refers to all software engineering done in multiple geographical locations and is also a catch-all term for “offshore” sourcing (Verner et al., 2014, p. 55).

2.3 How Sourcing Strategy Impacts Software Development

With the rise of outsourcing software development to distant countries came the eventual realization that new challenges emerge when development teams are not located close to each other. Recurring in most articles on this topic are the influencing factors of geographical distance, temporal distance and cultural distance, as is also seen in literature reviews (Šmite et al., 2010). It can be said that with an increase in any of these influencing factors the issues with developing at any distance are proportionally increased. There are however still some differences on how effects of sourcing strategy on software development are classified and what is seen as a core “issue”. Attempts at categorising the key areas have been made and prominent examples of those are examined and discussed below.

Olson and Olson (2000) compared collocated work and work at a distance in a review of ten years of laboratory and field studies of group collaborations in a collocated or remote setting. They described the differences, concluding that distance significantly impacts the process of development and predicted that distance will continue to matter despite technological advances. Olson and Olson found four key areas that differed between the two compared situations. These key areas are:

- *Common ground:* Common ground describes a shared knowledge space in which one participant knows the knowledge of the other and can construct and send information on those premises. Common ground is normally constructed on the fly through conversation and comes more easily through the availability of subtle cues from sources such as body language. When a common ground is not (completely) constructed, misconceptions and miscommunications become more prevalent.
- *Coupling in work:* This refers to the level of communication that is required for the task at hand. Low coupled work can be handed over between two people without much communication, but as soon as the task is complex, non-routine, interdependent or ambiguous, which it often is with software development, we talk about tightly coupled work. This level of work leans heavily on intensive communication and the amount of established common ground. Tightly coupled work is deemed as extremely difficult with increasing distances.
- *Collaboration readiness:* The culture and incentive of the organisation to share and communicate knowledge. Sharing knowledge comes with an extra amount of effort for everyone that wants to do so. With collaboration at a distance this is especially true as often it is facilitated with specific tools or technology that require a certain amount of training to use effectively. When the organisation does not incentivise this effort, the knowledge sharing practices and tools will not be fully adopted.
- *Technology Readiness:* An organisation requires both the technological hardware as well as the culture to adopt new technologies in their way of working. As such, video conferencing in groups requires a video conference room and sufficient networking capabilities on both ends of the conversation, but the organisation also requires habits that suit the technology. Some collaborative tools require frequent sign-ins to be effective, or they might require some change in data collection so the knowledge gained may fit into a knowledge management system without a surplus of effort.

Herbsleb and Moitra (2001) reviewed case studies and experimental designs to find what the difficulties of multiple site (global) development. They found six types of issues, which are:

- *Strategic issues*, which refer to the difficulty of dividing the work amongst sites, including managing which workload can be handled where and what expertise is available.
- *Cultural issues*, which highlights how groups of people can differ in their view on hierarchy, their background and need for structure.

- *Inadequate Communication on two fronts*, the first of which is formal communication, which needs clear and predefined structure to pass on information on the project and dictate ways of working with for example problem escalations. The second is informal communication and the lack of small talk between project members, which builds awareness on project issues and creates a network for project members to find the right skill or knowledge for a task.
- *Knowledge management*, especially referring to active response to changes of the wishes or surroundings of the client organisation and the correct prioritisation of features. Furthermore knowledge on certain issues might be available but if this is not accessible, it might be missed during development.
- *Project and process management issues*, which refer to the seamless handoff of work packages between sites, for example side notes on delivered material that get lost or misread.
- *Technical issues*, which cover the difficulty of maintaining networks between sites and differences between tooling and data formats.

Ågerfalk and Fitzgerald (2008) found three main issues with GSE in an analysis of published literature.

- *Communication*, which refers to the exchange of knowledge and information between the sending and receiving party in such a way that both parties have a similar understanding of that information or knowledge. This is known to be an important factor in all software development projects, but more so when the development teams are separated by a distance as technology has to facilitate all communication and subtle cues that carry information may be lost.
- *Coordination* is the distribution of tasks between teams in such a way that each team contributes fully to the project. This involves how different tasks are dependent on other tasks and teams and how they can be integrated.
- *Control*, which is a set of project management practices that create structures to progress the project within bounds of project restrictions, such as quality, time and budget. With increased distance it can be difficult to convey the desired project management structures.

These lists have many similar elements and many of the issues can be construed to be items of the other lists. It is a common issue in the topic of GSE or outsourcing that there exist a great many overlapping concepts (Šmite et al., 2014). For example, overlap can be found with the “Common Ground” key area of Olson and Olson versus the Cultural issue and the “informal” element of the Inadequate Communication issue as found by Herbsleb and Moitra, as well as the Communication issue as stated by Ågerfalk and Fitzgerald. These elements all refer to the difficulties of creating a communication channel that fully transfers the intended meaning from one party to another. All of these elements view face to face communication as the pinnacle of knowledge and information transfer. There are many more concepts that are similar. More interesting is how they differ.

For example differences can be found in subtle inflections of the lists. Olsen and Olsen stress the importance of work ethic or culture, referring to incentives and established requirements and habits of its employees which together make the effort investment of communication and knowledge practices worthwhile, and see the culture of a company as more as a holistic thing that has to work as a whole. Culture as such is not defined as being different between countries, which is the case for the lists of Herbsleb and Moitra, and Ågerfalk and Fitzgerald. The latter two contain project management practices, but the list of Olson and Olson does not. Herbsleb and Moitra and Olson and Olson include the type of work that is done, but Ågerfalk and Fitzgerald do not.

No attempt will be made to combine these lists, as this is not included in the research scope and the slightly different inflections and perspectives these lists provide can be useful tools for reflecting on the change requirements engineering elements. They can also be utilized in hindsight, to see whether all issues are covered in the altered requirements engineering frameworks.

3 Research Methods

In this chapter the research strategy and methods are explained. First, the overall steps of the research design are formulated. Next, we describe the details of the activities that are part of the high-level research design.

To answer the main research question of how sourcing strategy affects the process of requirements engineering, the research in this thesis was conducted iteratively, with every step supported by a research strategy to create the framework this research was eventually aiming for. This type of iterative research was modeled after the Technology Transfer Model, which takes a problem in industry, studies it academically and creates a solution which was tested against industry needs and academic validation (Gorschek, Wohlin, Carre, & Larsson, 2006). This model, which is pictured in *Figure 6*, was designed and tested with industry relevant requirements engineering research and has a purpose of not only delivering technical or academic results but to validate them in a real life setting and improve the business process.

The Technology Transfer Model consists of seven steps, which are:

1. Explore the issue and identify industry needs
2. Formulate the problem statement while researching the field and domain
3. Formulate candidate solution in cooperation with industry
4. Conduct lab validation (experiments) or validation through literature
5. Perform a static validation, for example by performing interviews
6. Perform a dynamic validation, such as small tests or pilots
7. Controlled release of the solution while remaining open to improvements

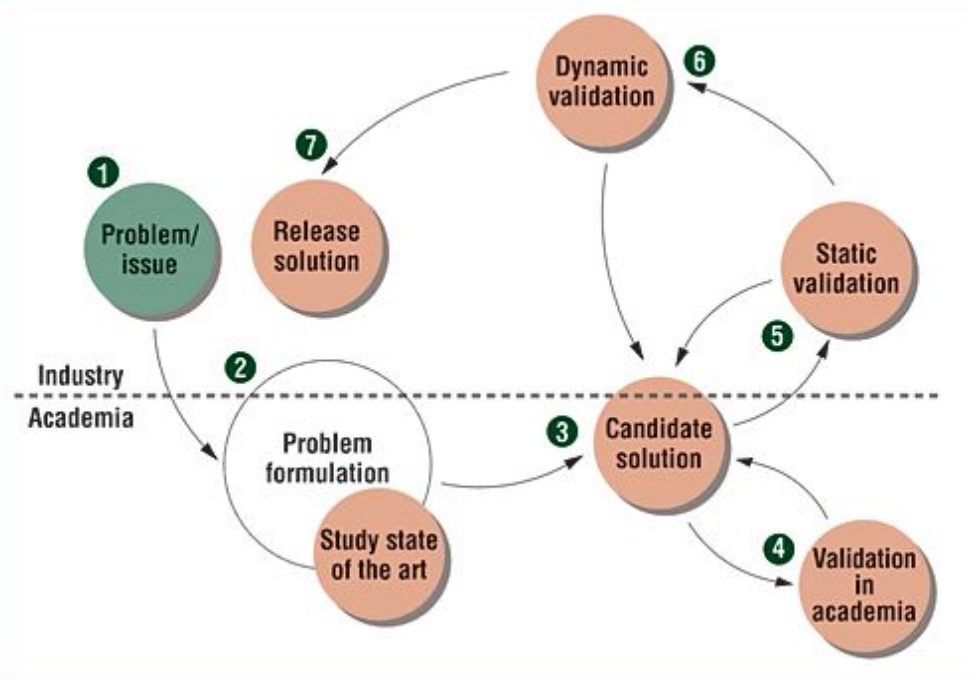

Figure 6. The Technology Transfer Model (Gorschek et al., 2006)

These steps were also followed in this research. In *Figure 7* you can find the adapted model, with each step as occurring in the study design used for this research. Below the image each step can be found in more detail, explaining what is done and what goals are to be achieved.

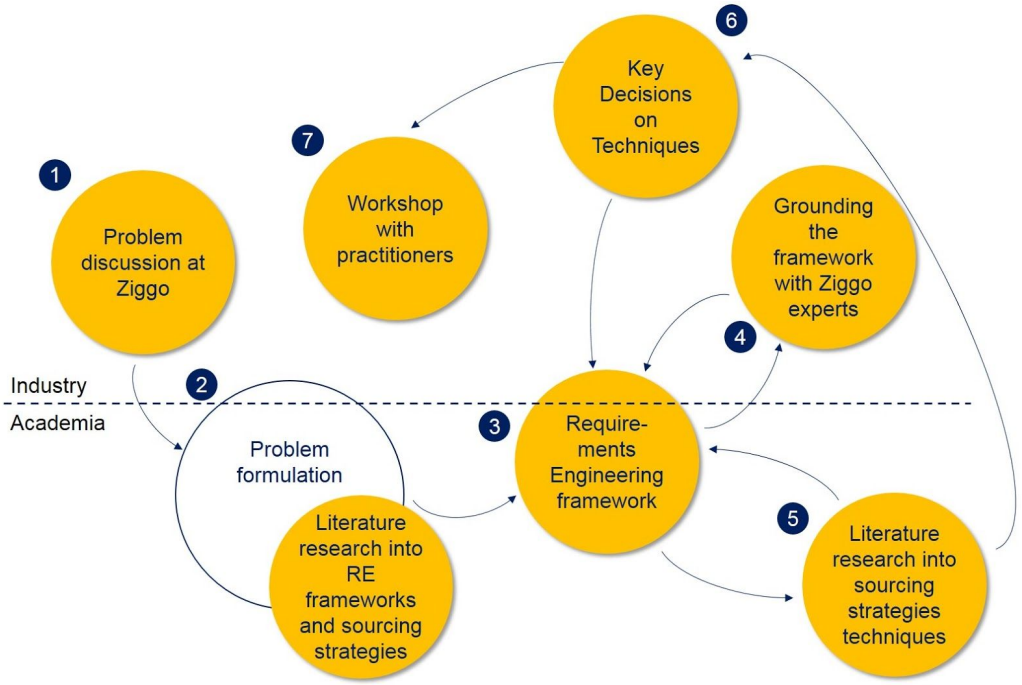

Figure 7. Technology Transfer Model applied to the Ziggo case

1) *Problem discussion at Ziggo*. The original problem of adapting requirements engineering to a sourcing strategy was presented by requirements engineering practitioners and management as an interesting and pressing issue that could benefit from a solution based in academia. From a round of initial informal interviews and group discussions, the graph in *Figure 2* was

coined, but what this graph practically meant or how to best deal with sourcing strategies was unknown. This led to the creation of the hypothesis and research questions as stated in the introduction.

2) *Literature research into requirements engineering frameworks and sourcing strategies.* On this step the ground layer for the rest of the thesis is formed and aspects of the problem that Ziggo presented are explored in literature. To that end, three of four sub questions as presented in the introduction are to be answered:
    - How to compare requirements engineering in different structured project environments?
    - What are sourcing strategies and how can sourcing strategies be categorised?
    - How do sourcing strategies affect software development in a structured project environment?

    The literature research in this section has been performed by a combination of literature search techniques. First, a structured literature research has been done of the past three years on notable requirements engineering journals to capture a state of the art. Journals searched in this capacity are Requirements Engineering Journal (Springer), IEEE Software Magazine and RefSQ. Furthermore, the unstructured literature research technique 'snowballing' has been used, to capture important building blocks for this research. Previously found articles have been analyzed for pertinent sources, which in turn have been scanned for such sources.

3) *Requirements Engineering framework.* To approach how a requirements engineering practice can be adapted to a sourcing strategy, a requirements engineering framework has been used to plot sourcing strategy issues upon and their accompanying techniques to solve that issue. The solution would need properties to concretely propose techniques to implement to overcome problems associated with a specific sourcing strategy, giving a practitioner switching to a new sourcing strategy a guideline to adapt his or her requirements engineering practice.

4) *Grounding the framework with Ziggo experts*. The requirements engineering framework chosen was that Pohl (2010). To place this framework in a real world context, experts at Ziggo were asked in a semi-structured interview how and if each of the requirements engineering elements were currently performed as well as how they envisioned the elements to be used potentially in a practical context. For each element a discussion was held on both of those aspects to solidify the academic requirements engineering framework by Pohl into a practitioners language. Summarized transcripts of the interviews can be found in Appendix E.

5) *Literature research into sourcing strategies techniques*. This step intends to explore how sourcing strategy impacts the different requirements engineering elements as described by Pohl (2010). Each element has been searched upon in combination with the specific sourcing strategy and the overall sourcing strategies are examined on processes and their best practices. This literature research session has featured a less structured approach, as it featured a great many individual searches. Relevant sources found in the previous literature research session have been reused when necessary. The Snowballing technique has been used upon new hits found. To cover more search areas google scholar has been added as a search engine. In total 73 unique papers were extensively reviewed, of which 27 yielded actual issues and techniques to cope with sourcing strategies.

6) *Key Decisions*. Due to incompatible techniques uncovered in step 5, multiple key decisions have been made between sets of techniques. These choices and associated techniques are detailed in Chapter 6. To make this choice, for each sourcing strategy an expert in that specific area was asked to discuss and weigh the pro's and con's of each choice. For these discussions the interviewees were only presented with the overarching key decisions and not the individual techniques or their implications to prevent biased outcomes. The summary of the transcripts can be seen in Appendix 9.2.

7) *Workshop with practitioners*. Although further research would make the findings from this thesis more practically applicable, having a set of techniques for each sourcing strategy is of great benefit to practitioners when facing a sourcing strategy. The final set of techniques was presented and discussed, and practitioners were challenged to see which and how techniques can be implemented in daily requirements engineering activities.

The order of the steps differ slightly on one point when compared to the Gorschek Technology Transfer Model. In the original model step 4, 'Validation in academia' and step 5, 'Static validation' are set in that specific order. The model features two way arrows for both of these steps with step 3, the 'Candidate solution', meaning that these steps are done iteratively. This is also the case for the implementation of the model in my research however predominantly step 4 and 5 were reversed: the solution was first tested against industry and then validated and filled in literature, to more closely follow the spirit of the model: grounding the research in practice. In the case of Ziggo, reversing these steps was more beneficial. Furthermore, a complete dynamic validation is missing in this thesis. Suggestions for dynamic validation are done in the Recommendations for future study section in the conclusion chapter, featuring the building of a situational maturity model or situational method, which can in turn be more practically implemented in case studies.

4 Requirements Engineering Framework

The framework created by Pohl (2010) contains the elements that make up the requirements engineering process and has been tried and tested in numerous practical situations. The applications of the framework lie in reference material for creating a customized requirements engineering process, training and communication about requirements engineering and analysis of existing requirements engineering practices. The following chapter will explain the building blocks of the framework to create a complete picture of requirements engineering elements that can be found in a contemporary requirements engineering process.

4.1 Framework Elements

The framework consists of five elements, with each their own sub-elements. These elements are interlinked and do not necessarily have a sequential order in their appearance on a project, instead, each element persists in its own way during a project lifetime. In the next section each of these elements will be further explained.

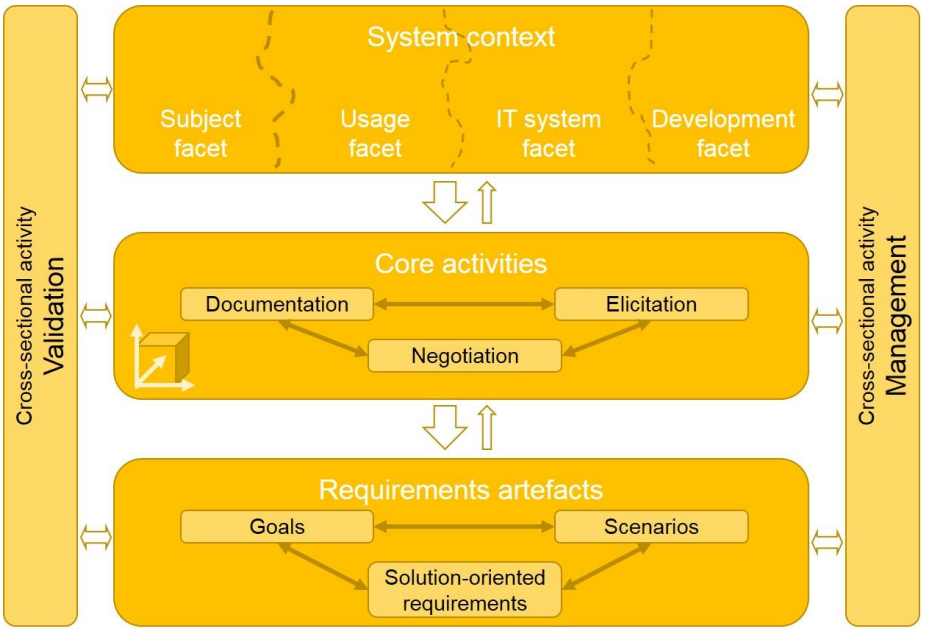

Figure 8. The Requirements Engineering Framework (Pohl, 2010).

4.1.1 System context

The system context considers aspects of the system environment that are important to its development and eventual use. It is vital that the context of a system is sufficiently gathered (Pohl, 2010, p. 44) Based on earlier work (Jarke & Pohl, 1993; Mylopoulos, Borgida, Jarke, & Koubarakis, 1990) four facets are discerned that cover the system context.

Subject facet

The subject facet is about the domain the system will operate in, including the information that will be processed and events and objects it will have to cope with. In the context of a cable company this could mean that a status information dashboard would show a map with locations that have broken down internet or cable services, and the system would have to be able to process the object of location and connection status within that object.

Usage facet
People and possibly other systems will make use of the system in order to achieve a goal or a task. The usage facet concerns how the system will be used and by who. This includes user groups, interactions with the system goals users may want to reach, workflows and possible laws and standards that the system has to comply to.

IT system facet
The system will most likely be deployed in an existing IT infrastructure including other systems, existing hardware, platforms, devices and networks. The IT system facet covers the operational and technical environment that the system will be developed for. This includes policies, restrictions and rules the system and its interaction may need to comply to.

Development facet
The development facet comprises all aspects of the development of the system, such as the method, development tools, techniques, quality assurance and the developing party.

4.1.2 Core activities

The requirements engineering framework consists of five activities, three of them core, contributing directly to the requirements engineering sub-goals of Content, Documentation and Agreement and two cross sectional activities, spanning all aspects of requirements engineering. These activities have various techniques and sub activities and interactions amongst each other.

Documentation
The documentation activity comprises the document building and specification of the requirements found during elicitation. Rules may be set for documentation and specification and supplementary information such as decisions and rationale may be required.

Rules considering documentation prescribe how a document must be built and to assure a level of quality, for example with the use of document templates. Specification rules describe how a requirement must be described, for example within a specific syntax or language or with scenarios.

Elicitation
Requirements elicitation is often also called requirements gathering, and the goal of this activity is to gain an understanding of the goal the system tries to achieve and translate these into requirements. Requirements can be elicited from various sources which depending on the project may or may not be relevant. A prime task of requirements elicitation is to determine which sources are relevant and can be used. Requirements elicited and insights may be gained from stakeholders and existing documentation, or new ‘innovative’ requirements may be elicited through a creative and collaborative process.

Negotiation
The views, wishes and needs of stakeholders may differ and conflict. The goal of the negotiation activity is to detect conflicts and to resolve them. To do this the conflicts have to be made explicit so that they can be reasoned about and steps can be taken to find a suitable solution.

4.1.3 Validation

The first of the cross-sectional activities is validation. The validation activity is about evaluating all inputs and outputs of the other requirements engineering elements. As such there are three sub-activities to validation:

1. Validation of requirements artefacts is the activity of detecting defects and evaluating the quality of requirements within the documentation alone.
2. Validation of activities has the objective of evaluating the performed requirements engineering process. The activities serve as the input for the artefacts on which contracts and development is based, and defects in procedure could lead to defects later on during development.
3. Validation of the consideration of system context evaluates if the proposed set of requirements complies to the specific system context. As the system context serves as the input to the core activities of requirements engineering. As such all four aspects must be taken into account appropriately to ensure the elicitation of all requirements.

4.1.4 Management

The second cross-sectional activity is requirements management and governs the other building blocks of the requirements engineering framework. Requirements management has three distinct goals.

1. Management of the requirements artefacts is the activity of keeping track of all requirements documents and their availability throughout the development cycle. At all times throughout the development process the correct requirements documentation must be at hand for relevant parties to look into. As such storage, change management, configuration management and requirements prioritisation must be taken into account.
2. The observation of system context activity considers changes that may occur in the system context that may impact the requirements already elicited. To that effect changes may need responses of new elicitation and documentation activities or project change management.
3. Management of activities considers the planning and control of the core requirements engineering activities, and may consider a context appropriate workflow of the requirements engineering activities.

4.1.5 Requirements artefacts

Requirement artefacts comprise all documented artefacts. An artefact usually has a prescribed type of documentation, but three types of artefact can be discerned.

Goals
Goals encompass the stakeholders intentions on what the system has to accomplish. As such they include high level objectives of the business, organisation or system.

Scenarios
A scenario describes a concrete example of sequence of steps that satisfy one or more goals the system tries to achieve. As such it documents an example of system usage. Both fulfilment and unfulfillment contingencies of a goal must be considered.

Solution oriented requirements
Solution oriented requirements define the proposed solution in aspects of data models, function and behaviour and include the quality (non-functional) and constraint aspects.

4.2 Static validation of framework

Validation of this framework in the Ziggo project structure was performed by interviewing requirements engineering experts. These experts all had a requirements engineering role and experience working with the past and current way of working within Ziggo. In total three requirements engineering experts were interviewed for this section. Each expert interviewed has at least seven years of experience with requirements engineering and ten years of experience in IT projects. The transcriptions for these interviews can be found in Appendix E.

The interviews had three goals:

- How requirements engineering related to the Ziggo project structure.
- How elements of Pohl's framework were represented in the Ziggo project structure.
- How, when applicable, the current way of working at Ziggo could be improved with regards to requirements engineering.

The interviews show that each element of the requirements engineering framework by Pohl has representation in the way of working by the Ziggo requirements engineering experts. The Ziggo project structure and a summary of its deliverables can be found in Appendix C. In the following sections, for each requirements engineering element of Pohl's framework, there is a short description of how the element is practiced currently within Ziggo. Where applicable there is also the experts inflection how the current practice could be improved.

System context
All facets of system context are uncovered on a high level in the early stages of the Ziggo project structure. These are used as the boundary conditions for a project, and are used to determine a solution direction and impact assessment. Hereafter a budget indication should help figuring out what is possible and what sourcing partner can be attracted. The system context is uncovered by finding and interviewing stakeholders. The system context is expanded upon in later project stages.

Documentation
Because of the UPC, Ziggo and now Vodafone fusions, many types of documentation with different requirement types can be found throughout Ziggo. This creates confusion in the organisation and makes the way of working for requirements engineering experts at Ziggo more difficult. A core principle, for documentation types found in Ziggo, is that early documentation types only have business level requirements and a solution direction. Based on this solution direction later functional and quality requirements are developed in an iterative fashion.

Often missing in requirements documentation is meta information about requirements, such as the requirement owner, its rationale for being included and early project decisions encompassing multiple requirements. implementing this would make adding project members in later stages and bringing them up to speed without repeat of discussions more efficient.

To be unambiguous many forms of requirements documentation contain a lot of similar texts across many artefacts. For example user stories can be very similar with just a few exceptions. This creates enormous artefacts with a lot of creation effort which most stakeholders will not use or read. Having targeted documentation would be more beneficial.

Elicitation
Various types of requirements elicitation are performed at Ziggo. Most dominant are one on one interviews with stakeholders. Earlier rounds of requirements elicitation consider high level system

context to measure the impact and budget of a project. Later stages and more in depth requirements can be gained through interviews, workshops and low fidelity prototyping.

Important for elicitation are the stakeholders involved. For both interviews and workshops it is necessary to include all relevant parties, even if those parties may not contribute much to original project goals. Skipped stakeholders may feel left out and could possibly bog down a project when added too late.

Negotiation
Although prioritization of requirements is not a mandatory aspect for Ziggo requirements engineers, this is often a useful tool to implement, as business stakeholders will often add 'nice to have' requirements to a project even though this is not in line with its main goals. Negotiation about to be added requirements can be resolved on two levels. A final say often comes from the budget holding client stakeholder, but to onboard requirements beneficial from other stakeholders a face to face negotiation guided by the requirements expert is a tool often utilized. For larger projects a " changeboard" or "programme board" is created with senior stakeholders deciding on key requirements.

The downside experienced for requirements prioritization is that they are often interlinked into a process. Meaning that low ranked requirements still might be essential for project success which demands the requirements engineer to reflect requirements prioritization on the system context.

Requirements validation
Validation of documents produced by the requirements engineer is performed by a peer expert. This validation is done on document level, not on artefact level which is a faster process, but might fail to catch flaws in the documentation. For some projects the experts would advise validation on single requirements artefact level, as this would require the requirements engineer to stop and think whether or not a requirement fits in the system context.

The requirements engineer also validates business goals of a project produced by business stakeholders against organisational goals. The requirements engineer should have a better picture of the goals of multiple stakeholders and investigate where these conflict.

Requirements management
There used to be a requirements management system at Ziggo. This tool was abandoned when it did not align with the Ziggo project structure and organisation. Currently requirements documents are often stored in shared cloud solutions, and are communicated mostly via mail. The current Ziggo way of working lacks a system to manage version control of documents which can cause communication issues and work redo. Important for a version control system would be to show stakeholders what aspects changed to make reviewing more efficient.

Change management is performed through a process of requesting changes to a project. Bigger projects have a changeboard which approves or disapproves a change request, which is mostly dependent on available budget weighed against the need of the change.

Currently there is a lack of requirements traceability. This would be beneficial to the Ziggo project structure to evaluate not only if a requirement is filled in according to business goals and rationale, but also traces where in the process faults in the project communication occurred to prevent future faulty project deliverables.

Requirements artefacts

Many different requirements artefacts can be found in the Ziggo organisation. These artefacts are usually produced when a sourcing partner needs them to be created for their internal process. For example the offshore insourcing branche needs a variation of user stories. Our onshore insourcing partner needs excel lists of requirements with more metadata on these requirements. Artefacts are chosen based on the sourcing partner and the project need, as judged by the requirements engineer. Adding visual requirements representations (such as flowcharts and wireframes) are found to be useful when communicating with non-expert stakeholders.

5 Literature Findings about Sourcing Strategies in RE

This chapter contains the results of the literature research. For each sourcing strategy, each requirements engineering element as presented by (Pohl, 2010) will have its associated literature discussed. The results are presented by common issues for that requirements engineering element, showing how multiple authors view the issue and what techniques may be implemented to solve it. For each requirements engineering element an analysis is given on how the different viewpoints and techniques relate to the overall sourcing strategy and how they interact with each other. The sourcing strategies are discussed in reversed order for distance: Offshore outsourcing, offshore insourcing, onshore outsourcing and onshore insourcing.

5.1 Offshore Outsourcing

Requirements engineering takes a forefront with the offshore outsourcing strategy because of the focus on contractual agreements and liability between client and executing party (Prikladnicki & Audy, 2012), making it essential to know what is to be built and who does what. As outsourcing practices are generally temporary by nature, there is far less benefit or time allocated to building relations between onshore and offshore teams. A focus with offshore outsourcing is therefore on quick and cheap completion of the project. This makes it harder to build a case for techniques that take away resources from direct project completion but provides long term or risk reducing benefits (Prikladnicki & Audy, 2012). The core problem of offshore sourcing strategies is the reduction of communication when distance (geographical as well as temporal and cultural distance) increases (Cheng & Atlee, 2007b). The need to cope with this communication reduction is at the forefront of many requirements engineering techniques. These techniques are intended to bridge the gap of the lack of natural communication because of that distance. Lastly offshore sourcing strategies have longer development activities than their collocated counterparts (Herbsleb & Mockus, 2003), requiring the need for timesaving activities or techniques.

5.1.1 System context

Domain knowledge prevalent with offshore team

Establishing the domain knowledge is cited as a prevalent and more pronounced problem for offshore outsourcing compared to onshore variants (Nakatsu & Iacovou, 2009)(Herbsleb, 2007). Differences in culture, organisation and language can lead to loss of data, rework and generally confusion on the desired end product (Verner et al., 2014). Nakatsu and Iacovou note that requirements simply have to be made more explicit and management on their reception has to be on a short leash to make up for the lack of business know how and understanding the underlying business goals. Part of this problem is the lack of shared culture and tacit knowledge based on which information can be transferred explicitly. To allow for a shared culture, Bhat et al. (2006) found some best practices, which include training on culture and communication tools used by both onshore and offshore teams, consensus on

operating norms such as how a meeting works and how to comply to commitments, and the sharing of requirements specification templates.

Lack of informal communication

Akin to the above problem, another issue that is less prevalent in onshore sourcing strategies is the lack of informal communication between stakeholders, which is worse in offshore settings as distance both geographically and culturally decreases communication. (Cheng & Atlee, 2007b) (Verner et al., 2014) (Herbsleb, 2007). Verner et al. state that lack of informal communication is detrimental to relationships between team members and suggests direct communication channels between all members of the team. Herbsleb summarized various studies and cases which made use of instant chat solutions. Although these solutions were not consistent in their results or how they were utilized, generally speaking they were perceived as beneficial for creating discussions and supplementing information acquired from official documentation.

Shared responsibility

Because of a great many stakeholders involved in requirements engineering, especially when establishing system context, responsibility for certain aspects of knowledge is ambiguous and is rated by Bhat et al. (2006) as one of the root causes for offshore outsourcing problems. Their own solutions to this problem include establishing fixed methods for project reporting and performance metrics, which is concurred by Prikladnicki and Audy (2012). The reason for this is that a well defined process shared by multiple companies makes it more intuitive what responsibilities exist and what can be expected of other parties. The simplest form of a shared but well understood process is to adhere to a tried and tested development methodology. Another method to reduce issues of responsibility is to have frequent and small deliverables from all team members to allow greater control and earlier signaling of conflicts or defunkt stakeholders and to better track project progress (Bhat et al., 2006) (Herbsleb, 2007).

Finding expertise

Inherently creating a complete picture of the system contexts requires a diverse set of experts, usually from the client side. Bhat et al. promotes the use of a requirements awareness system, with for each expert an outline of their domain, role and responsibilities. The effectivity of this method is contested however (Herbsleb & Mockus, 2003), as it laborious to maintain the accuracy of domains and even more difficult to create domains of expertise that have conventional vocabularies that are understood perfectly by all parties involved. Instead, Herbsleb and Mockus developed a tool based on work history. When an expert has worked in the past on a specific piece of documentation, the author is logged and can easily be found when more information on that piece is required. Although this method was originally used for developer code, it can apply to requirement documentation as well.

Analysis

For the requirements engineering element of system context most problems are at least roughly related to each other. To sum up, they all focus on one of two things, the first of which is finding the correct knowledge, and the second is transferring this knowledge without ambiguity. As noted for issues regarding accessing the right stakeholders, many solutions revolve around utilizing technology, but the participants or stakeholders have to be willing and have available time when required. Use of technology has to be paired with some management oversight as suggested by Verner et al. (2014).

Noteworthy for the second problem for system context are the varying success of chat tools by Herbsleb (2007). In his analysis of the use of chat tools some cases had positive results and some were more negative. The difference between these two could be the variable of shared culture as described by Bhat et al. (2006). Aligning culture between the offshore and onshore team may allow more consistent results from other technological solutions such as chat tooling.

5.1.2 Documentation

Documentation Standards
Characteristic for outsourced projects is a problem of double standards. Inevitably there are differences in work approaches between companies and this is a dominant issue for requirements documentation (Niazi, El-Attar, & Usman, 2012) (Ahmad, Nasir, Iqbal, & Zahid, 2015) (Bhat et al., 2006). According to Bhat et al. this impacts requirements engineering in two ways. Dissimilar methods and documentation lead to information ambiguity because of differing requirements engineering maturity levels or a loss of requirements engineering work to comply in a translation of standards, both can increase faults in the end documentation. The simple solution suggested in all three articles that specifically note this issue is to have either party take definitive leadership and set a documentation standard, which includes both a template for document structure as well as how requirements are to be described. All parties involved have to comply to this documentation standard. According to Niazi et al. this also promotes a shared understanding of requirements and bring to light cultural differences. Bhat et al. also note that it is crucial to distribute the chosen standard and template to all relevant stakeholders, or the advantage may be lost.

GSE tailored documentation templates
Another way to view documentation standards and templates comes from Ali and Lai (2015b). They advocate the advantages of tailored templates for requirements documentation to suit GSE specifically. Ali and Lai's own template to this end consists of multiple stages; each of these stages has requirements validated on all participating development sites. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE conundrum and safeguards against common GSE traps.

Documentation supplements
Another venue of attack to increase the transparency of requirements is to supplement the documentation with some additional aspects. Niazi et al. (2012) found two high value practices for GSE when considering documentation. The first is to include a summary of the requirements, and the second is to use diagrams to accompany the requirement text. Both will increase the speed of understanding across stakeholders and facilitate early discussions. Many processes or concepts are also easier explained with diagrams. Furthermore, both Niazi et al.(2012) and Herbsleb (2007) found that the building and use of a common vocabulary was immensely valuable. Having a standard definition that stakeholders can look back on reduces the chance that meanings are distorted between sites.

Analysis
The overarching principle to documentation in the offshore outsourcing setting is to equalize between all stakeholders and sites. All authors argue that discrepancies between parties in their documentation approach increase risk to concise requirements engineering. This opens up the question how to choose a preferred documentation standard, which none of the authors address. When development of a project consists of a large number of sites, the choice of documentation becomes increasingly important, with at the end of the spectrum specific templates such as described by Ali and Lai (2015b). When the situation is less severe, often the options are limited to the documentation technique of the executing party or the client party. The advantage of the latter is the ingrained business context and vocabulary that may be present in the documentation. The advantage of using the executing party documentation structure is the often higher maturity and definition of requirements prescribed by developing parties - leaving less open for discussion.

5.1.3 Elicitation

Participation of stakeholders

An often recurring problem happens when stakeholders do not realise the importance of their participation and availability or are averse to helping the project get along as it might change their work environment to their detriment, such as automating away their jobs. Without their cooperation establishing the full system context is challenging at best, which means that obtaining access to these key stakeholders is critical for project success (Niazi et al., 2012) (Verner et al., 2014) (Bhat et al., 2006) (Gotel & Finkelstein, 1994). Having access and identifying stakeholders is increasingly difficult when the stakeholders and developers are far apart, which is the case in offshore sourcing strategies. Gotel and Finkelstein add that not having access or limited access to these stakeholders increases the likelihood of requirements traceability problems. To compensate for this issue, Verner et al. suggest that constant pressure from high level management on the client side is required to ensure the full participation of all stakeholders. Niazi et al. sees the use of technology, such as video conferencing as a solution to this problem. Alternatively occasional visits by representatives of the offshore team may improve direct communication.

Natural Language Processing

Developments in natural language processing allow for an entirely new methods of requirements elicitation. Traditionally, a viable option for requirements elicitation is the painstaking research of the client documentation. This process is time-consuming and can be error prone (Rago, Marcos, & Diaz-Pace, 2013). With natural language processing of these documents, this type of elicitation can be greatly sped up or need little human interference at all. The advantages these techniques give with the offshore outsourcing strategy is the standardization of document processing, as the tools are neutral and not hampered by cultural preconceptions or language barriers that a real life analyst may have. Example techniques like these are presented by Rago et al. (2013), Sharma and Kushwaha (2011) and Lucassen et al. (2015). Rago et al. (2013) developed the QAMiner, a tool that automatically collects non-functional or quality requirements. It can also be used to create the set of concepts that are specifically relevant to the client, which is a jump off point to create a system context specific vocabulary. Sharma and Kushwaha (2011) developed a tool that semi automatically categorizes requirements on their priority and complexity, as well as discerning functional and non-functional requirements. Lucassen et al. (2015) proposed a tool that evaluates the user story requirements artefact and points out low quality areas in the user story to prevent defects later on in the development process. Although these tools may not be mature enough in all settings, they can be used to bridge the distance and cultural gap with unbiased views of the system context.

Analysis

Requirements elicitation requires sources to elicit from. To some extent this requires the client to take responsibility and ensure the availability of the correct business stakeholders. Practice however states this is a recurring issue and requires active inquiry to find, connect with and keep the attention of core business stakeholders. Making sure the project kicks off with management support from the client is an essential tool to keep stakeholders participating. When access to stakeholder resources is limited and frequent, informal and synchronous contact between business stakeholders and the executing party is impractical, a semi automated process might offer help in finding requirements unbarred by a context colored disposition. This does require the client to already possess or prepare documentation about the current system context and generally the vision the to be created solution should fulfil.

5.1.4 Negotiation

Slow negotiation due to cultural or geographical distance
Generally accepted issue when considering negotiation in an offshore outsourcing setting is that it takes longer than with other sourcing strategies (Nakatsu & Iacovou, 2009) (Bhat et al., 2006). A common reason for this is the need for absolute certainty that requirements are understood fully. Bhat et al. (2006) adds to this that during negotiation in an offshore outsourcing setting, multiple facets of the other party have to be dealt with. For the client contact can be with both an onshore delegation of the executing organisation and the actual developers offshore. Likewise the executing party faces the business section of the client as well as the IT department from the client. This means that there are often more stakeholders taking part in any one negotiation and more possible discrepancies between stakeholder goals that have to be settled before the project can continue. This issue increases when these stakeholders have asynchronous discussions instead of synchronous decision making sessions.

Communication
Communication during negotiation of requirements is generally cited as an issue regardless of sourcing strategy, but is pronounced in the offshore outsourcing context (Nakatsu & Iacovou, 2009)(Verner et al., 2014)(Niazi et al., 2012). A major contributing factor lies in cultural assumptions and reference points. Even when a negotiation is performed in the same language, misunderstandings can arise because of colloquial speech and writing based on assumptions only partial to a certain culture or country (Nakatsu & Iacovou, 2009). The most commonly cited solution to this problem is the prioritization of requirements (Verner et al., 2014)(Ahmad et al., 2015)(Niazi et al., 2012)(Bhat et al., 2006) as this places requirements in context of each other and gives away information on what is critical to fulfil business goals. An annotation to this solution is the importance of making it someone's responsibility to make sure requirements prioritization takes place (Verner et al., 2014).

Interaction between stakeholders
Another core problem with offshore outsourcing negotiation found is a lack of common goals among stakeholders. (Bhat et al., 2006) (Verner et al., 2014) Bhat et al. (2006) have a couple of suggestions to remedy this problem, such as explicitly developing stakeholder viewpoints, including stakeholder satisfaction as a successfactor and involving stakeholders in the creation of a common vision. Verner et al. (2014) add that it is crucial to share requirements information with all stakeholders, before negotiation commences. This affects their interaction during negotiation settings creating more confusion. Another aspect of stakeholder interaction is the lack thereof between those of the same level in hierarchies between companies in an offshore outsourcing setting. Negotiation most often happens between higher level stakeholders, although details for the solution may often be relegation to lower levels. It is essential that these lower levels also have an opportunity for open communication to prevent incomplete viewpoints or a mismatch on detail level of the solution (Bhat et al., 2006).

Synchronous Negotiation
A literature review by Smite and Wohlin (2011) recommends negotiation is best facilitated by face-to-face meetings, requiring one or more stakeholders to be flown over to the negotiation site. If this is not possible, synchronous meetings at a distance are a prime alternative. Synchronous meetings at a distance are best aided by rich media such as video conferencing (Bhat et al., 2006), noting that a lack of video (but with quality audio) some cues are lost during negotiation, which include group dynamics, interpersonal behavior, attitudes and level of agreement (D. E. Damian, Eberlein, Shaw, & Gaines, 2003). Synchronous communication at a distance through this method may even

yield some benefits, for example a slightly slower conversation speed which makes it easier to follow a set negotiation process (D. E. Damian et al., 2003). Another general consensus is the utility of a human facilitator during negotiation meetings. (Bhat et al., 2006) (D. E. Damian et al., 2003) (Niazi et al., 2012). Even sparse facilitation can have a big impact on meeting effectivity. Human facilitators can enable stakeholders to make more objective decisions and keep track of discussion points that may otherwise be lost, preventing issues from being unaddressed (D. E. Damian et al., 2003).

Asynchronous Negotiation

When time differences between sites make frequent synchronous communication difficult, some methods of asynchronous communication may be useful supplements.

Damian et al. (2006) explored a method of asynchronous communication whereby stakeholders used an argumentation scheme tool to collect and tackle issues. In stages comments were then given on each of the issues by all stakeholders, exploring the issue fully. After that, the final decision making was done through asynchronous computer conferencing. The study concluded that the actual synchronous negotiation meetings were more effective when these asynchronous practices happened before the meetings, and reduced the time required for these meetings as many of the issues had been eliminated beforehand. Damian et al.(2006) did note that the effectiveness of this method is reduced when complex issues arise as lengthy messages are then used to argument an issue, which can take up a lot of stakeholder time.

Analysis

The general viewpoint of offshore outsourcing requirements negotiation points out that it is generally harder and more time consuming, although the risks may be controllable with the right techniques or methods, such as use of a human facilitator, quality video conferencing or pre-negotiation meeting discussions. When extra time and effort is expected and allotted for negotiation when compared with onshore/insourcing strategies, a rush job with accompanying oversights can be prevented. When resources are available to do so, the pinnacle of negotiation remains face to face meetings with all stakeholders. If a negotiation is deemed critical or time is sparse, flying over stakeholders may be considered an acceptable investment.

5.1.5 Validation

Validation in offshore setting

Inaccurate or low quality requirements artefacts are cause of major problems when keeping to project timelines. When during the project inaccuracies or poor quality items are discovered, a process of intense communication has to occur back down the chain to obtain the correct information. This is less of a problem when all parties are located near each other, as face-to-face communication can happen spontaneously and with little effort to resolve the issue. For GSE, this is a major hurdle to overcome, and a reason why GSE fails to meet project milestones (Ali & Lai, 2015b). Special validation techniques can be used to tackle this problem in two ways: speeding up the validation process and enriching the validation outputs to cope with the special needs of GSE. An example technique like this is the template created by Ali and Lai (2015b) as described in the documentation section. This technique has a validation process that is tailored to accommodate different time zones that are present in offshore settings, reducing chance of overshooting project deadlines. Also considered essential for offshore requirements validation is a common vocabulary (Niazi et al., 2012) (Herbsleb, 2007)(Ali & Lai, 2015b). Cultural and language differences, even different business jargon will make transferring project knowledge far more difficult.

Natural Language Processing

Another technique to consider for requirements artefacts validation are the earlier discussed natural language processing techniques. An example is the method by Sharma and Kushwaha (2011), who created a method of analysing a requirements specification through decomposing the document into single, simple sentences. These sentences are then analysed and categorised as type of requirement and their priority. Next, further analysis can be done on the resulting categorisation, using multiple aspects, such as system inputs and outputs. From this analysis, a complexity analysis can be computed, showing how difficult the system will be to create and maintain, where possible problems may occur and how long development might take. Techniques calculating complexity analysis are now often done when the code has finished. Having this analysis done earlier, exceptions on project bounds may be prevented. Besides enrichment of data on the requirements, this technique also has the potential to reduce the validation time and reduce the difficulty of hand-verifying the requirement artefact, as the method's reduction of sentences into simple statements leaves little room for interpretation.

Analysis

A recurring theme seems to be the need for common vocabulary. Making sure all stakeholders have a clear definition of business concepts specific to the client seems essential. It seems useful to keep in mind that validation takes time, but reduces risk of deviation of project timelines. In an offshore setting, this need is amplified with validation, as managing mistakes with dispersed teams can have more impact than when issues can be settled locally. In a similar fashion, using specific techniques that accommodate an offshore sourcing strategy may require more time up front, but can force all stakeholders to provide a certain level of quality, reducing overall project risk and especially one on project timeline deviation.

5.1.6 Management of the requirements artefacts

Traceability problem

One of the problems for requirements engineering that is more aggravated in outsourcing settings is the traceability problem (Lormans, van Dijk, van Deursen, Nocker, & de Zeeuw, 2004). As among changing requirements, it is often necessary to trace original stakeholders to discuss the impact previously established requirements to see if business goals are still attained with the adapted solution. This is more difficult in an outsourced setting, as reaching the correct stakeholder may require going through many layers of separation and hierarchy. The main solution presented to this problem is the use of a centralized requirements depository or requirements management system (Lormans et al., 2004) (Bhat et al., 2006) (Smite & Wohlin, 2011). Although other research by Ahmad et al. (2015) found that a centralized requirements depository was the least beneficial among all practices as laid out by Sommerville and Sawyer's framework (1997), it was still found to be useful. Use of such a system in an offshore setting does require strong infrastructure: not being able to access a document in any reasonable length of time because of poor downloading/uploading rates, will severely hamper the utility of these tools and may cause version issues when simultaneous work is performed (Smite & Wohlin, 2011)(Herbsleb, 2007).

Another more simple solution to the traceability problem is the use of diligent hypertexting in documentation to create a structure and visible relations between artefacts and their precursor documents (Gotel & Finkelstein, 1994)(Sinha, Sengupta, & Chandra, 2006). This solution requires less expensive tooling and systems (or goes well alongside them) but is more difficult to maintain or to perform change management upon. Sinha et al. (2006) found that it would also be useful for this solution if links could be maintained across document depositories.

Awareness of project members
GSE struggles with less effective and frequent communication between project members. One of the resulting issues with this problem is the difficulty of awareness of other project members, what they are doing and what their expertise is (Herbsleb, 2007). Beyond the previously mentioned centralized depository, this problem requires some additional collaborative tools or features that project members can make use of. A more pronounced solution is a version control system, which can be used to track which project member contributed and thus has knowledge on which part of the solution (Mockus & Weiss, 2001). Configuration management systems which are shared between all teams working on aspects of software development are also useful to this end, as well as rich communication media (Smite & Wohlin, 2011). Another useful tool is the remote sharing of screens to actively show other stakeholders what you are working on (Herbsleb, 2007).

Analysis
Most suggestions named in this section promote the use of centralised global tooling, at least to some extent. An indispensable requirement for this to be possible is a solid infrastructure with reliable and ample network bandwidth. Any connection made also has to be secure. For the offshore outsourcing setting, this requires each new client-execution party to be set up with a centralised system with secure connections, spanning the globe. It seems however, that this is deemed worth the effort, as these systems allow much easier collaboration at a distance and make the software development process more capable of handling changes to the initial requirements. It would be advised that setting up these systems requires time needs to be considered when planning the project.

5.1.7 The observation of system context

Formal mechanisms
Basic change management requires a policy for processing changes to requirements in place, at the minimum, this is especially true for offshore outsourcing (Verner et al., 2014)(Bhat et al., 2006). Poor change controls will result in scope creep and it is essential that proper requirements engineering is performed when a change to the original request is pushed through (Nakatsu & Iacovou, 2009). One quote best presenting this issue is ‘‘the high quality of code and CMMi Level 5 processes that exist for most organizations overseas further increases the likelihood of being delivered exactly what you’ve asked for—so you better make sure you are asking for exactly the right thing.’’(Nakatsu & Iacovou, 2009, p. 63). A formal mechanism generally used to combat poor change controls is accountability on requirements and keeping track of the original goals throughout the project. This may be hampered by the length projects take: The longer the project, the bigger the turnover on project staff, sapping project and business knowledge, as well as accountability (Bhat et al., 2006).

Informal mechanisms
Herbsleb and Mockus (2003) argue that a formal policy alone is not enough to react to changes in requirements in a GSE setting. The problem with formal communication channels is their lack of speed: they will travel through layers of hierarchy up and down the chain from the client to the executing party, each step taking up time. The solution is to mimic a co-located setting, whereby stakeholders on the same level of hierarchy communicate freely between each other as if they were working in the same office. Various tools and technologies exist to stimulate informal communication Herbsleb and Mockus (2003). Tools and technologies that were found to be useful include video conferencing, audio conferencing, instant text messaging, shared calendars and presence awareness features (showing team members coming online). Herbsleb and Mockus (2003) did note that just email and telephone were not enough to stimulate informal communication, furthermore they found some of the more informal tools to be awkward for newcomers. Specialist tools that deal with

requirements change management are also available. Sinha et al. (2006) developed a tool that allowed discussions between random team members and logged the results, allowing all stakeholders to look back on what had been discussed. The tool also allows stakeholders to subscribe to requirements, which alerted them whenever a change to the specific requirement was made. This feature not only helps with the requirements traceability problem, it also increases accountability and helps find key stakeholders.

Analysis
There is some level of conflict between formal mechanisms and informal mechanisms. Informal mechanism strategies focus on faster reaction times, circumventing formal processes, which also guard the quality of the project. This may speed up development time or prevent the need for rework (as work done during a change process may have to be revisited). Informal mechanism methods rely on the skill of individual project members to guard the project quality and require active participation of all stakeholders. Tools as developed by Sinha et al. (2006) may prove to combine the best of both worlds, allowing stakeholders through all phases during development to view the origins and arguments of decisions, combatting project member turnover.

5.1.8 Management of activities

Way of working
Regardless of type of software development process, it is found to be essential to have one. Both Prikladnicki and Audy (2012) and Bhat et al. (2006) found having a well defined and relatable process vital for successful offshore outsourcing, as long as this software development process is shared and understood among stakeholders. Having an established way of working has the advantage of requiring less effort for sharing the used software development process, as these are commonly known or material for learning about these processes is readily available. Regardless of the process, it is advised to create a project structure to relate the dependencies of artefacts and activities (Bhat et al., 2006). An example of modeling and tailoring project structure to specific needs is described by Weerd, Versendaal, & Brinkkemper (2006). This leaves the discussion whether or not agile methodologies are a good match for offshore outsourcing. On one hand, there is a lot of evidence for the use of short and incremental development cycles or deliverables to keep a tight leash on planning and quality and to catch mistakes early (Bhat et al., 2006)(Smite & Wohlin, 2011) (Herbsleb, 2007), which would signal great use for agile development techniques. On the other hand, the fundamentals of agile methodologies advocate having teams collocated, daily meetings and short lines of communication and coordination, which are all hampered with offshore outsourcing (Paasivaara & Lassenius, 2006). Despite this, Paasivaara and Lassenius suggest that a possibility of advantage exists for using agile, if the traps of agile in combination with this sourcing strategy are acknowledged and mitigated. The advantage comes forth from the idea that, when despite the distance the communication practices essential to agile methods are upheld, they can overcome the communication blockades that plague offshore outsourcing.

Delays
One of the more common issues found with offshore strategies is the management of timeframes for activities. The origins of this problem are found in the resolution of issues, as an issue arises in the case of an offshore strategy, generally more people are involved and need to be consulted to resolve the issue (Herbsleb & Mockus, 2003). Communication, coordination and negotiation between this larger group of project members has to occur which causes delays. Herbsleb and Mockus(2003) have estimated offshore sourcing strategies can take 2.5 times longer than with onshore strategies. When switching to an offshore strategies, it is therefore advised to adjust the expected timelines accordingly. A more active approach to reduce the longer development times with offshore strategies is found in

the decoupling of work (Smite & Wohlin, 2011) (Herbsleb & Mockus, 2003) (Herbsleb, 2007). When tasks only require project members that are collocated to collaborate, the delays may be prevented. Despite their own advice, Herbsleb and Mockus(2003) have found in their case studies that decoupling of work may not deliver as promised. After three and a half year of decoupling efforts during such a study found no consistent time savings. They suggest that instead, a solution has to be found in combination of decoupling and novel tools or techniques. A different approach was found by Gotel and Finkelstein (1994), who found the surprisingly easy solution by reducing the amount of project members in a project. To combat the difficulty of finding the right stakeholders and the build healthy communication, it can be valuable to have but a few stakeholders to communicate with. All project members will be able to remember and personally contact the persons responsible for pieces of knowledge. Failing a reduction of team members, Gotel and Finkelstein suggest having champions of interaction, who can act like communication relays for the project.

Analysis
Using an agile methodology, such as Scrum, may come with its own set of risks and solutions possibly making matters more complicated than they normally are with offshore outsourcing. Before opting into this popular software development method, a consideration has to be made whether you have the expertise onboard to handle agile in a more challenging environment. The same counts for decoupling of work or task dependencies between sites. This technique may or may not pay off and the added complication to the overall project has to be weighed against the possible benefits. More solid is the advice of using a run-of-the-mill development methodology, which makes aligning the client organisation and executing party organisation less complicated, as experts in these methods are bountiful or novices can be trained with accessible material. Regardless of methods used, it would be prudent to acknowledge that offshore outsourcing will take longer than onshore variants, and plan activities accordingly.

5.1.9 Requirements artefacts

Task dependencies
A known issue for GSE is the coordination problem, as coined by Agerfalk et al. (2005), which states that a dilemma exists when parties have common interests or goals, but are dependent on each other to complete them. Solving this task dependency problem naturally requires extensive communication and time, especially at a distance. A common conviction about task dependencies is that they are influenced by dependencies in software architecture (Herbsleb, 2007). This allows early design decisions to dictate the amount interaction that teams at a distance from each other require to complete their tasks, given that interfaces between modules in the software are well defined. While more refined solution oriented requirements artefacts are developed, paying attention to function interdependency can steer the software architecture in such a direction that minimal interaction between teams is enabled, reducing development time and possibility for communication errors. Mockus and Weiss (2001) developed a technique to enable this independent structuring in hindsight, looking at change histories and assignments given to developers to discern what parts of the software could be maintained separately with minimal interaction. Such a technique can also be used in reverse, setting up requirements for interaction between sites or responsible parties as it is known which dependencies exist or will arise in the future.

Analysis
Techniques in this section are angled at an overall reduction of communication required between different locations, arguing that the less teams have to coordinate, the less effort and risk is introduced during development. Although these are very beneficial, reducing communication between sites can also impact other aspects of collaboration that may be detrimental. Teams will have less

knowledge of each other, and will less likely play into practical aspects of other teams. If mistakes were made during solution design, the necessary communication required to fix the problem may be more strained, as the lines of communication are less often used or none existent at all. This makes the decision to reduce task dependency and therefore communication a strategic one, weighing risks of failure and changes to the product against faster development times and smoother coordination.

5.2 Offshore Insourcing

Compared to offshore outsourcing, there is more emphasis on people than project deadlines and boundaries when considering the offshore insourcing strategy. As project members are most likely there to be had for multiple projects, investment in their integration and their effectiveness becomes a viable route to improve project success (Prikladnicki & Audy, 2012). Similar to the offshore outsourcing strategy, the offshore insourcing strategy is also plagued by the reduction of communication that comes forth from increased distance (Cheng & Atlee, 2007b). Coping with this reduced amount of communication is central to many of the offered solutions. Also akin to the offshore outsourcing strategy is the elongated development cycle that comes forth from non-collocation (Herbsleb & Mockus, 2003).

5.2.1 System context

Domain knowledge prevalent with offshore team
Establishing the domain knowledge is cited as a prevalent and more pronounced problem for offshore sourcing compared to onshore variants (Hofner et al., 2011)(Herbsleb, 2007). Differences in culture, organisation and language can lead to loss of data, rework and generally confusion on the desired end product (Verner et al., 2014). Hofner et al. (2011) note that one of the three most important things to train offshore employees in, is domain knowledge. This is most relevant when the software the offshore team produces is supposed to function in embedded systems (Prikladnicki & Audy, 2012), concur that training is one of the more essential parts of preparing an offshore insourced team. They recommend having a formal training programme for GSE and company contextual knowledge. Effort on training employees on domain knowledge would take precedence over strict project management according to Prikladnicki and Audy.

Lack of informal communication
As written in the offshore outsourcing section 5.1.1, the problem with the lack of informal communication is as major for the offshore insourcing strategy. The main source for this issue is put forward by Cheng and Atlee (2007b), Verner et al. (2014) and Herbsleb (2007) stems from the fact that distance (both geographically and culturally) decreases communication. Both Verner et al. and Herbsleb promote the use of instant and direct communication channels between team members to facilitate discussions and supplementing information transferred through more official channels.

Finding expertise
Similar to what is put forward for the offshore outsourcing strategy section 5.1.1, finding experts on certain topics across locations all over the world can be difficult, especially when information on expertise is often shared via different social networks that are bound by their location (Herbsleb & Mockus, 2003). According to Herbsleb and Mockus having a simple social network with expert descriptions is insufficient for practical use. Instead, a more competent approach is to have a tool track individual's work history. When an expert needs to be found, work history can be searched to find original authors or related works from different creators.

Analysis
Literature on the three different problem areas surrounding system context for the offshore insourcing strategy put forward different approaches towards the same goal. The underlying goal that can be discovered is the integrating or leveling of the different sites on knowledge, skill and communication, which can be done by training (Hofner et al., 2011)(Prikladnicki & Audy, 2012) or by tools and technology (Verner et al., 2014)(Herbsleb, 2007)(Herbsleb & Mockus, 2003).

5.2.2 Documentation

Documentation standards and templates
Less boldly put forward then for offshore outsourcing, having documentation standards and prepared templates can still be seen as a useful item for offshore insourcing. Making sure all sites use the same approach to documentation is essential to prevent loss of work (Niazi et al., 2012). Niazi et al. also value having documentation standards as they improve the understanding of requirements across sites and bring cultural differences to the foreground where they can be dealt with. A more extensive approach would be to craft a documentation template and standards specifically to GSE settings. Ali and Lai (2015b) created a template and process to this regard. Their template consists requirements validation stages across sites, incrementally improving requirement quality and preventing knowledge gaps and miscommunication about the requirements. The template considers GSE specific challenges, such as time zones, development site needs and geographical locations and includes these as variables to deal with actively to prevent common GSE pitfalls.

Documentation supplements
For subject of documentation supplements in the offshore insourcing strategy the same literature applies as in its offshore outsourcing counterpart in 5.1.2. The core principle is to supplement the requirement documentation with additional information or information structuring to increase the transparency of the requirements. Three useful practices in total were discovered. Niazi et al. (2012) found the inclusion of a summary of the requirements and diagrams to be useful. Both Niazi et al.(2012) and Herbsleb (2007) promoted the use of a common vocabulary (across different global locations) that was included in documents.

Analysis
Literature on requirements documentation in the offshore insourcing strategy is encompassed by the literature of the same topic in the offshore outsourcing strategy, with just fewer authors to its name than the latter. As such, similar analysis will come forth of this topic, albeit slightly weaker. A principle that is put forward by most literature on this front is to equalize across different development sites. Having similar or perhaps even specialized documentation between different global teams will improve the transparency of the included information and prevent miscommunication about requirements.

5.2.3 Elicitation

Participation of stakeholders
Participation of stakeholders is as much an issue for the offshore insourcing strategy as it is with offshore outsourcing 5.1.3. Stakeholders may have conflicting interests when helping a project run smoothly, and they are difficult to rouse for full participation, especially when approached by an offshore branch (Niazi et al., 2012) (Verner et al., 2014) (Gotel & Finkelstein, 1994). Having key users on hand is however critical for project success. To combat this problem Verner et al. suggest having

high level management pressure stakeholders in cooperation (Verner et al., 2014). Having technology reduce communication barriers, such as video conferencing also alleviates this issue. A more pricey option is having representatives fly over to talk with onshore stakeholders, as a direct method of communication.

Natural language processing
Natural language processing as a solution to overcome some of the barriers with offshore strategies works as well for offshore insourcing as described in the offshore outsourcing counterpart section 5.1.3. The main item natural language processing overcomes to benefit requirements elicitation is the cultural and language barrier (Rago et al., 2013). As a natural language processing technique has no preconceptions a real life analyst may have, the technique may be a concise and no-nonsense communication channel across sites. Example techniques are presented by Rago et al. (2013), Sharma and Kushwaha (2011), and Lucassen et al. (2015). These techniques may require some maturing or fiddling to work in practical context within a modern company, but can eventually offer a bridge between cultures and languages, allowing experts around the world to work together without biased views of the system context.

Analysis
The two points of interest described above are directly opposing in their strategy. In the Participation of stakeholders, authors suggest broadening the ties between key stakeholders and offshore branches. Natural language processing bridges the gap by letting technology do the talking between the onshore and offshore locations. A choice may have to be made between these two, as implementing both could possibly be detrimental to either, because having close ties between sites could have experts forgo the formal requirements documents produced by natural language processing as knowledge is thought to be had, and likewise having documents containing all or most relevant information could make stakeholders less likely to reach out to distant colleagues anyhow. In the context of offshore insourcing, a point may be made in favor of close ties between stakeholders, as the intention of an offshore insourced branch is a long term commitment, making personal relations between stakeholders more valuable in the long run.

5.2.4 Negotiation

Requirements communication
Deficient communication of and about requirements is a common problem for GSE (Verner et al., 2014)(Niazi et al., 2012). Most issues that surround this problem are due to (cultural) assumptions present with negotiating parties, even when negotiations are done in the same language. One aspect of communication about requirements that is often overlooked is the prioritization of requirements (Verner et al., 2014)(Niazi et al., 2012). This communication step both gives direct value to requirements, but also cross cultures notes their importance and reveals some context and intention behind the requirements. To ensure the prioritization step occurs, making someone in charge of prioritization is a valid solution according to Verner et al. (2014). Another simple but effective technique to improve communication about requirements, especially during the negotiation phase, is sharing the entire set of requirements with all stakeholders before the actual negotiation takes place (Verner et al., 2014).

Synchronous negotiation
As described in section 5.1.4, negotiation is best done in a face to face setting, which ensures the least possible noise during communication (Smite & Wohlin, 2011)(D. E. Damian et al., 2003). As this is not always possible in an offshore insourcing strategy scenario, there are a few things to take into account when negotiating synchronously. A lower quality connection (video or audio) when

conferencing at a distance will fail to transfer all communication queues, so making sure the supporting technology is working as intended is critical. Communication through technology is also slightly slower than during a face to face meeting, but this may be beneficial as a more structured discussion plan can be followed with the reduced speed of turn-taking communication that is common when conferencing (D. E. Damian et al., 2003). Having an impartial human facilitator present is seen as especially beneficial for multi-location requirement negotiations, which makes such negotiations more effective (D. E. Damian et al., 2003)(Niazi et al., 2012).

Asynchronous negotiation
The alternative to synchronous negotiation comes with the same annotations as the offshore outsourcing strategy, described in section 5.1.4, Timewise often more practical, asynchronous communication between onshore and offshore branches is traditionally seen as lossy communication, but may be improved by communication supplements (D. Damian et al., 2006). Damian et al. created an asynchronous communication method that used an argumentation scheme tool to find issues and structure discussions around them. For this negotiation technique, participants have stages during which to comment and argue certain points. Then a follow-up computer conference is held to facilitate final decision making. This technique was found to be more effective than just the synchronous meetings alone, as they prepared the participants and reduced the in meeting time required to cover all issues. This technique was less effective for increasingly complex discussions, as participants then had to read over lengthy argumentations that arise from complexity.

Analysis
Requirements negotiation is a more time consuming business as stakeholders are hampered by distances and flawed communication to do their communication through. Having appropriate techniques and checklists in place that are exercised religiously reduces the harm this distance introduces to negotiation. The techniques provided in this section do not hamper each other and benefits of using multiple may be cumulative. When resources to do so are available, forgoing many techniques and flying over stakeholders to have face to face meetings remains the most effective solution.

5.2.5 Validation

Validation in offshore setting
As is true for any offshore sourcing strategy, described further in section 5.1.5, deficient requirements artefacts are a source for delays in projects, as faults are discovered further down the road and some backpedaling is required. This is worse for offshore strategies, as the original sources for requirements artefacts are more difficult to track down (Ali & Lai, 2015b). To strengthen the validation step to prevent these delays various techniques can be used. For example GSE templates for documentation can speed up the validation step and prevent common GSE issues, such as timezones. Another technique is including vocabularies in requirement artefacts, ensuring all parties are talking about the same concepts (Niazi et al., 2012) (Herbsleb, 2007)(Ali & Lai, 2015b).

Natural language processing
Also as valid for the offshore insourcing strategy, is the described validation technique in section 5.1.5. Natural language processing can decompose requirements artefacts into formal and logical statements. The tool as described by Sharma and Kushwaha (2011) furthermore analyses the forthcoming requirements and categorizes them as well as giving them a priority. Additionally system inputs and outputs can be determined, which will allow the computation of a complexity analysis.

Having these described formally by a system, a following manual validation is easier and can be sped up as the short statements have little room for interpretation.

Analysis
Ensuring all stakeholders have a concise definition of requirements and included business concepts is a recurring topic. Cultural assumptions and language barriers increase the difficulty of communication of requirements, the validation step suffers along with other communication. To that end both discussion points in this section point at formalizing concepts to help the validation step along. Having a formal documentations standard and having distinct concepts within your requirements and even formalizing (logically) the requirements leave less room for interpretation errors, making validation more straightforward.

5.2.6 Management of the requirements artefacts

Centralized depository system
Making sure requirements are available and visible for all stakeholders at all times is more difficult for offshore sourcing strategies. Artefacts are often saved only locally, and may be shared via mail, creating multiple instances of the same artefacts which can create miscommunication around versions and adaptations to documents. A common solution to this problem is a centralized requirement depository (Smite & Wohlin, 2011). Having such a solution does however present its own issues when facing vast distances. Poor or incompatible infrastructure across different sites may still hamper various centralized tools to work effectively and sluggish systems with poor transfer rates will reduce the utilization or may even have stakeholders go around the depository entirely, making it fruitless (Smite & Wohlin, 2011)(Herbsleb, 2007). An alternative to these systems when good infrastructure cannot be assured is the use of hypertexting between artefacts, creating reference chains to create requirements traceability. The drawback of this more crude system is the increase in effort in maintaining such a setup, especially when considering change management (Gotel & Finkelstein, 1994)(Sinha et al., 2006). Such a system will also need to be able to link between locations and cannot only be locally or version issues will still occur between onshore and offshore teams.

Awareness of project members
Infrequent and ineffective communication between project members are also a major issue for offshore insourcing, similar to the offshore outsourcing strategy as further explained in section 5.1.6. The result of which is difficulty of knowing exactly which project member to contact when expertise is needed (Herbsleb, 2007). Having a centralized depository or version control system organizing requirements artefacts that also tracks authors for specific changes, makes finding and possibly contacting the right person (Mockus & Weiss, 2001)(Smite & Wohlin, 2011). Remote sharing is a tool that goes well with these capabilities, allowing project members to highlight exact problems (Herbsleb, 2007).

Analysis
Having centralized global tooling to manage requirement artefacts is overall considered to be a step in the good direction when considering requirements artefact management. Making sure the tooling works as intended is however key when considering these kinds of solutions. Besides working technically and providing availability of requirements according to needs, the usability of such a tool also needs to be near or beyond the usability of standard text editor and email service, or the tool may find opposition in its use by practitioners, finding the hassle of using the tool not worth the effort. An off the shelf product has the advantage in this section for having to also provide a comfortable user experience to remain competitive. For an offshore insourcing branch, it may well pay off to arrange for a lasting solution. The organisation may benefit this well beyond a few projects, and even provide a

source for reusable artefacts that can track back to original authors, when similar solutions are pursued on a later date.

5.2.7 The observation of system context

Informal mechanisms
The absence of defined organisational policies for change management are a major risk to project success (Verner et al., 2014). But having a formal policy is only one side of the coin. For offshore insourcing having informal mechanisms to cope with change rapidly and effectively is just as important as explained in the offshore outsourcing section 5.1.7. Formal communication is the slow progression at which different parties are notified, having to travel up and down the hierarchical chain (Herbsleb & Mockus, 2003). Herbsleb and Mockus suggest creating a co-located setting across distances, simulating coworkers in the same office environment and cutting across hierarchy. This can be done with the use of tools and technologies such as video conferencing, audio conferencing, instant text messaging, shared calendars and presence awareness tools. A strong note to this, is that just telephony and email are not enough to create informal communication. More specialized tools for requirement communication are also available, as for example created by Sinha et al. (2006). Their tool allowed coworkers to log and discuss requirements and to review these discussions, to reflect back on them. The tool also allowed subscription of requirements, engaging stakeholders in further development decisions to fulfill requirements. Such a tool would reduce the requirements traceability problem and also improve accountability and involvement of stakeholders.

Analysis
The benefit of offshore insourcing opposed to offshore outsourcing would be the possibility to create more intimate relations between team members across distances. If such efforts are successful, allowing and facilitating informal mechanisms of change management in a project would speed up development times and prevent possible rework when a team continues effort in a direction when a proposed change still travels up and down the hierarchies. The downside of informal mechanisms is their lack of processed quality control, and the risk of the mechanism failing when the onshore and offshore team members are not as connected as is needed for such mechanisms to work. Conscious effort has to be supplied to not only provide the capability (such as tools and technology) to communicate freely across distances, but also the willingness to communicate with distant colleagues.

5.2.8 Management of activities

Delays
Managing planning and timeframes steadfastly plague both offshore insourcing and outsourcing, as explained in section 5.1.8. This problem originates from the involvement of more parties to every contention and each party increases the time delays before a decision is made (Herbsleb & Mockus, 2003). Herbsleb and Mockus (2003) estimated from empirical data that GSE can take up to two and a half times longer than collocated work. When a switch is made from onshore to offshore sourcing, it is prudent to incorporate this extra time into activity planning. There are various suggestions to solve this problem less passively, with varying results. One promoted option is to decouple work. Decoupled work would have tasks which require cooperation only require team members that are co-located, which may prevent delays (Smite & Wohlin, 2011)(Herbsleb & Mockus, 2003)(Herbsleb, 2007). This is however a contested solution, as Herbsleb and Mockus (2003) also found that decoupling of tasks may not be as effective, with a study of three and half years having found no time saving benefits. They suggest that the use of tools that support decoupling, sharing supplementary information across

sites may yield more promising results. Another solution is suggested by Gotel and Finkelstein (1994), who suggested reducing the number of involved project members would simplify communication within a project, allowing project members to have knowledge of expertise of each project member. If a small project team is not feasible, having champions of interaction which know the team intimately could be an adequate surrogate.

Roles and responsibilities

Where offshore outsourcing has a focus on project or development methodology and process, this switches to roles and responsibility for offshore insourcing. The reason for this difference comes from the possibility of building a long term competent and effective team with tight relations with the onshore branch with the possibility to exercise a great amount of control, where this is largely impossible for an outsourced team (Hofner et al., 2011). Having the possibility to build such an effective team does come with the responsibility to put in the sustained effort to do so. This lays the effort mostly in the hands of managing the offshore team and the relations between onshore and offshore stakeholders (Prikladnicki & Audy, 2012). Having established such relations offers a step in the right direction for the most important piece of knowledge that an offshore insourced team requires: domain knowledge (Hofner et al., 2011). This furthermore increases the continued involvement of business stakeholders into project development. As critical decisions have to be made by business stakeholders, having more direct lines to these stakeholders and respond effectively to change requests minimizes difficulties normally associated with GSE (Niazi et al., 2012)(Prikladnicki & Audy, 2012).

Working agile

Having frequent deliverables or a short and incremental development cycles is a suggested strategy for offshore sourcing (Smite & Wohlin, 2011) (Herbsleb, 2007). Having a short feedback loop could prevent late notice of issues and keeping planning and quality under control. This aspect alone would suggest agile methodology to be a great fit for offshore insourcing, but this is contradicted by other agile preferred features, such as daily face to face meetings, short lines of communication and intimate coordination between team members (Paasivaara & Lassenius, 2006). Paasivaara and Lassenius do however argue that when these shortcomings are consciously worked with, an advantage can be created out of the disadvantaged agile. The potential advantage stems from precisely the disadvantages that agile has in this context: GSE has problematic communication at the core of its difficulty. If a true agile methodology is incorporated and upheld, especially enforced communication such as daily stand ups and increased cooperation, it could overcome the communication barrier naturally.

Analysis

The more prominent solution overarching the three above discussions is investing in excelling and empowered employees with practical communication channels. Coping with delays may be improved by having small effective teams, an offshore insourced team can be more effective when managed well and agile needs to be strictly enforced to be an asset in this sourcing strategy. Taking these aspects into account before a switch is made to offshore insourcing would greatly benefit the offshore team and the chance of project success.

5.2.9 Requirements artefacts

Task dependencies
As described in more detail in section 5.1.9, coordination is a challenging task for GSE (Agerfalk et al., 2005). The problem arises when multiple parties want to reach a shared goal but are dependent on each other to complete tasks to reach this goal. This dilemma creates a lot of communication overhead, increasing the time required to reach a goal. A natural solution to this problem is to reduce the dependency between teams, by reducing the amount of tasks that requires distant colleagues to complete. This task dependency can be affected by software architecture (Herbsleb, 2007), which allows early design decisions, such as seen in later requirements artefacts with higher fidelity, to influence the amount of task dependency found in the project.

Analysis
Reducing task dependency to decrease the amount of communication required between team members at different locations may speed up the development process during a 'sunny day' scenario, but when issues arise that do require cooperation between teams the lack of familiarity with distant team members and the inhabitual communication may prove to be heftier and less predictable obstacles to overcome. A mistake made in initial design documentation may be far more taxing for teams that are used to low task dependency than those with high task dependency. Using this strategy would need to be well considered for both pros and cons before implementation.

5.3 Onshore Outsourcing

Onshore outsourcing does not suffer the same consequences from distance as the offshore sourcing strategies do. For all types of distance; geographical, temporal and cultural the client and executing party are often close together. What remains are barriers concerning the relationship between client and executing party. As the organisations may have different ways of working, tooling and maturity levels.

5.3.1 System context

Collocation
The number one risk found for domestic outsourcing by Nakatsu and Iacovou (2009) is miscommunication of original requirements, with as reason a lack of understanding of system context and experience regarding the business of the client. A lack of (business) stakeholder involvement is also cited as a major risk. Both risks can be solved by putting outsourced teams together on the same location as the involved business stakeholders As people then naturally communicate on expertise and contextual information (Herbsleb & Mockus, 2003) (Herbsleb, 2007). Knowing who to contact on what subject matter and doing so quickly greatly speeds up the project process. Essential for this solution is the actual distance between offices. Kraut and Streeter (1995) found that beyond 30 meters, communication would perform similarly to offices that are many kilometers apart.

Analysis
When at all possible, forcing stakeholders and project members together in one open space seems the best way to get contextual information across to the executing party and this can be seen as a valid strategy for requirements engineering system context sharing. Alternatives to collocation can be derived from technology solutions as presented in the offshore strategies, but these solutions will have the same drawbacks as they have at a distance. Creating a friendly 'watercooler' environment

and inciting social interaction between business stakeholders and outsourced team members will solve many problems associated with outsourcing.

5.3.2 Documentation

Standard Documentation Structure
Ahmad et al. (2015) researched critical requirements engineering practices among a large group of software development practitioners in the context of outsourcing. The practitioners were asked to rate the 66 practices as laid out by Sommerville and Sawyer (1997), when specifically looking at outsourced projects. Of the 66 practices, only two were predominantly rated as 'high value' practices. One of those was having defined a standard documentation structure for requirements and making sure all involved parties can work and follow that structure. This does require, before a project requirements phase commences, that a document structure is chosen upon or defined, perhaps tailored to the project.

Analysis
Having a requirements documentation standard prepared for a project requires the expertise to build such a document. As often an outsourcing company has a higher maturity level in their field of expertise, it would be advisable to request the outsourcing company to provide or suggest a template and arrange training for client company stakeholders to work with that template, preventing miscommunication about the target content as well as making sure no two documentation standards are in play.

5.3.3 Elicitation

No academic sources were found on requirements elicitation in the context of onshore outsourcing.

5.3.4 Negotiation

Face to face negotiation
Excellent communication is essential for projects, and lack thereof is listed as the number two risk for domestic outsourcing projects (Nakatsu & Iacovou, 2009). Face to face time is one of the most valuable assets to this extent, especially for negotiation activities. Even with perfect video and audio tooling to facilitate communication, some information to the negotiators may be lost (D. E. Damian et al., 2003) and this is greatly aggravated when the quality of the tools or connection is less than perfect. As the cost of face to face time for onshore outsourcing is reduced compared to offshore sourcing strategies, maximizing face to face negotiation moments is advised, collocated work also takes significantly less time than multi site operations. (Herbsleb & Mockus, 2003), as it takes longer for communication on decisions takes place and more people are generally involved in decisions for multi site projects.

Managing expectations
Another major risk found for domestic outsourcing by Nakatsu en Iacovou (2009) was the failure to manage end-user expectations. A solution to this problem is to make stakeholders aware of relative effort required for specific requirements. Stakeholders are then less likely to cite infeasible or difficult requirements as system essentials. Techniques that provide such a solution are the Stakeholder Win-Win approach or the MBASE Approach (Boehm, Abi-Antoun, Port, Kwan, & Lynch, 1999). Such approaches reconcile Win conditions for the client (resulting system capabilities) with Win conditions for the developer (minimal risk to going beyond schedule or budget). When a win condition for the

client poses a significant risk to a win condition of the developer, this is stated as an issue that has to be resolved or agreed upon.

Prioritizing requirements
In an effort to find high value requirements engineering practices for outsourced projects, Ahmad et al. (2015) asked a large group to evaluate the usefulness of Sommerville and Sawyer's (1997) best practices for requirements engineering, specifically in the case of outsourcing projects. Only two practices were rated 'high value', the first was having standard documentation (as described in the documentation section of the onshore outsourcing strategy). The second practice rated predominantly 'high value' was the prioritization of requirements. This practice forces stakeholders to relativize requirements amongst each other, agreeing which takes precedence and preventing future discussions. It also allows scheduling and prioritizing of development work later on in the project (Firesmith, 2004).

Analysis
Both the managing expectations and prioritizing requirements discussion benefit from the first discussion for the negotiation section, namely having one or multiple sit downs with business stakeholders and going over requirements. Essential for these sit downs is not only the one way communication from stakeholders towards the executing party, but also between stakeholders themselves, having them agree on requirements, their relative importance and their contribution to project cost.

5.3.5 Validation

No academic sources were found on requirements validation in the context of onshore outsourcing.

5.3.6 Management of the requirements artefacts

No academic sources were found on the management of the requirements artefacts in the context of onshore outsourcing.

5.3.7 The observation of system context

Informal communication
When requirements change, it can be difficult for formal processes of communication to propagate along hierarchies fast enough to prevent the need for work redo (Herbsleb & Mockus, 2003). Informal communication is much better at reacting quickly to change, especially in a collocated setting as explained in the system context (section 5.3.1) of the onshore outsourcing strategy. Team members can be apprised of upcoming changes before they are formally submitted, preventing work redo and preparing the way for a change in requirements to land softly in the development process. Alternatives to collocation come from the appropriate adaptation of various tools and making sure parties are connected within those tools. Examples of technology supporting informal communication are shared calendars, chat services and presence awareness tools (Herbsleb & Mockus, 2003).

Analysis
As promoted earlier, collocation is the pinnacle of communication, and whenever possible in-house stakeholders and outsourced team members should work in the same environment. When this option is not available, making sure that common day tools, such as chatting, calendars, autocomplete email addresses in mail clients and such, are linked between client and executing companies, so that links

of communication are easily established. The more difficult it is to contact other parties, the less likely or often this will occur.

5.3.8 Management of activities

No academic sources were found on the management of activities in the context of onshore outsourcing.

5.3.9 Requirements artefacts

No academic sources were found on requirements artefacts in the context of onshore outsourcing.

5.4 Onshore Insourcing

Although more traditionally sourced software development projects (onshore insourcing) are still plagued by issues, these issues are more systemic to software development projects in general and less attributed to their sourcing strategy. As such, sparing to no literature is to be found on an "onshore insourcing" strategy. Despite this, having academic sources report on a greater need for one approach or technique with a specific sourcing strategy which is not onshore insourcing, may reflect on what is less necessary when applying the most basic sourcing strategy. To explore this alternative research route some basic parameters can be discussed on what the perceived strengths of onshore insourcing are compared to other sourcing strategies. Herbsleb (2007) described this as followed:

> In a (highly idealized) traditional, co-located project, teams with a history of working together have naturally built up a number of ways of coordinating their work. They have a shared view of how the work will proceed, either because of a shared, defined process or just by acquiring a common set of habits and vocabulary over time. Through frequent interactions, both formal and informal, team members have a clear idea of who has what sort of expertise and how responsibilities are allocated. Information flows freely through the network during the many informal interactions that happen in the hallway, over meals, before and after formal meetings. There is relatively little miscommunication as teams share a common native language as well as national and corporate culture. People are generally aware of what others are working on, know if and how their work affects other people, and know day to day the level of urgency and stress experienced across the project. Prior collaborations have produced long-standing professional and social relationships that provide a context and history within which problems and misunderstandings can be resolved. (Herbsleb, 2007, Chapter 1.2)

When this is considered a best case scenario, having an onshore insourcing strategy that deviates from this view without argumentation may be less effective than it could be. Furthermore, as best practices suggested for other sourcing strategies try to emulate this scenario at a distance, their advised tools and techniques for doing so may be less interesting when this scenario is implemented as is. To that extend each requirements engineering element will be discussed in light of this scenario, and how techniques included to overcome barriers for other sourcing strategies to reach this scenario may have less potential.

5.4.1 System context

Talking about all aspects of system context, Herbsleb and Mockus said “As long as people are collocated, it seems that people’s natural gregarious tendencies can be relied upon to disseminate information**”** (Herbsleb & Mockus, 2003, p. 492). This is also what is reflected in this sourcing strategies’ scenario description, where it is desirable of having project members collocated, with the intention of having them collaborate and share information. This collocation has a strong demarcation of effectiveness, the closer the better, but when two workspaces are further apart than 30 meters, it could as well have been several kilometres (Kraut & Streeter, 1995). For the onshore insourcing strategy it is often financially feasible to have project members and stakeholders do collocated work, most of the time, and may only require some organisational stretching and pressure to accomplish.

As an organisation in the onshore insourcing context often has a single culture and language, training or techniques for shared vocabulary as suggested by Bhat et al. (2006) might be foregone. Another suggestion is having smaller and more frequent deliverables, as for offshore outsourcing, signaling conflicts between deliverables and defunkt project members often happens late (Bhat et al., 2006) (Herbsleb, 2007)). This is less of an issue for collocated work as informal and natural discussion about the project will signal conflicts.

5.4.2 Documentation

Having a vocabulary of terms specific to a business domain as a standard addition to requirements documentation is often cited to be useful in offshore sourcing strategies (Niazi et al., 2012)(Herbsleb, 2007). A big reason for this is the divergence of meaning given to terms across cultures and locations, resulting in interpretation differences of the requirements. In the onshore insourcing setting, there is often but one company language and culture present, with terms converging when stakeholders talk often with each other. Provided that onshore locations and branches of the company are interlinked enough to have similar cultures and practices, having a vocabulary would prove to be less beneficial. The same counts for other documentation supplements and templates, such as charts and diagrams, which are specifically used to explain certain business terminology. Having diagrams explaining requirements can of course be good requirements engineering practice however.

5.4.3 Elicitation

Elicitation can be less difficult with analysts on board that are already actively aware of the system context and are often well known to business stakeholders. This makes a recurring problem of contacting business stakeholders to prevent requirements traceability problems (Gotel & Finkelstein, 1994) as well as difficulty with stakeholder cooperation (Niazi et al., 2012) (Verner et al., 2014) (Bhat et al., 2006) less of a risk to a project. Having specialized tools to determine client concepts and business goals such as the QAMiner developed by Rago et al. (2013)3) are also less relevant, as the assumption for onshore insourced stakeholders can be made that they are more aware of the system context.

5.4.4 Negotiation

Negotiation is most effective at reaching results in a face-to-face meeting. Even other
Synchronous meeting types, such as a video assisted conference call, are less effective (Smite & Wohlin, 2011). For the onshore insourcing it should be feasible to arrange face-to-face meetings for

all negotiations with all stakeholders, considering the importance of agreed requirements. As it is a second best option regardless, video conferencing or other negotiation technology is less essential for this sourcing strategy. For the offshore outsourcing strategy having a human facilitator present at negotiations was a highly acclaimed solution to make sure decisions are made objectively and all meeting points are addressed. This might still be useful, but with more readily available common knowledge on system context with all stakeholders during a negotiation at a collocated setting decision making, might already allow stakeholders to come to decisions beneficial to the client organisation.

5.4.5 Validation

Validation in offshore settings is mostly hampered by the duration for validation to be processed, as discussions at a distance are often asynchronous which causes projects to be late (Ali & Lai, 2015b). This is in the offshore setting solved by increasing the quality of initial validation steps, or laying down a process for validation discussion that increases the tempo of handling validation issues at a distance. When face-to-face meetings can happen spontaneously in a collocated setting, this is far less of an issue. That does however require stakeholders to be able to free up time slots to be available when issues arise, or similar issues to offshoring contexts may occur.

5.4.6 Management of the requirements artefacts

There are two problems that plague management of the requirements artefacts in offshore sourcing strategy settings, which are traceability of requirements (Lormans et al., 2004) and awareness of project members (Herbsleb, 2007). Both issues are prescribed centralized systems that store, categorise and administer requirements and who contributed to them. These solutions can still provide benefits in an onshore insourcing setting, when faced for example with knowledge management issues with a turnover churn of project team members, but are generally less interesting in daily use as the difficulty of talking to and finding original contributors is a lot easier when a walk around the office will suffice. Centralized systems require the discipline and effort to maintain, or they lose their use or might even be detrimental because of out of date information resulting in suboptimal decisions. A strategic decision needs to be made on how fragmented the client company is compared to the effort required to make use of a centralized system. If a company has many diverse locations with little informal communication it can be interesting to invest in such a technology.

5.4.7 Observation of System Context

Both formal as well as informal mechanisms for communicating changes in the system context have been created for GSE. The problem with offshore strategies is the length of time it takes before a change to the original requirements is put through, having to travel through the hierarchy of different stakeholders whilst still implementing good requirements engineering practices (Nakatsu & Iacovou, 2009). This is less of an issue when stakeholders of all levels of hierarchy share an office space, as rumours about a change is more easily spread and a meeting with relevant project members is quickly arranged to decisively cope with the change. The danger of informal change management is the reliance on the proactivity and awareness of individual project members. Relevant project members have to become aware of a potential change in time to perform adequate requirements engineering before the project has moved on and work redo/scope changes are difficult to implement. This coping strategy goes against modern flexible workplaces and working from home paradigms: if there are fewer occasions for informal conversation an informal change management strategy is less effective

and may result in a flawed change management approach. In that case a more formal approach to change management will have to be relied on.

5.4.8 Management of Activities

A great boon to onshore insourcing is shorter project timelines. Herbsleb and Mockus (2003) have estimated offshore sourcing strategies can take 2.5 times longer than with onshore strategies. This phenomenon is mostly attributed to more involved stakeholders and management layers through which communication has to be propagated. This can be capitalized upon further by keeping teams small together, making them more effective as a unit. A collocated setting also allows for the incorporation of various agile methodologies that require people to work together on the same location.

5.4.9 Requirements artefacts

While constructing Requirements artefacts there is the possibility to create decoupling of work. This is done by introducing separation of functionality whilst constructing higher fidelity artefacts in the solution direction, so that the functionalities can be developed on a site, without much coordination with teams located elsewhere (Herbsleb, 2007). This is of potential use when considering projects with teams separated both geographically as well as temporally making communication difficult. For an onshore insourcing setting introducing decoupling of work could be potentially detrimental, as forcing team members to coordinate and communicate frequently allows them to find expertise on specific pieces of knowledge or functionality when the need arises as they are more aware of activities performed by others through their regular contact.

6 Analysis of the Literature Findings

In this chapter, we conduct an analysis of the literature on both qualitative as well as quantitative aspects of the papers, issues and techniques uncovered in Chapter 5. The qualitative section discusses how the choice of a sourcing strategy influences software development in general, expanding on the data in Chapter 4.3, then some inconsistencies between techniques are figured out on a section about key decisions for each sourcing strategy. Note that in none of the coming analysis sections onshore insourcing is discussed, as this considers 'regular' requirements engineering not related to sourcing strategies. No direct techniques were found for this flavor of sourcing strategy. This is further discussed in Chapter 5.4.The following subsection looks for quantitative trends in the findings from the literature.

6.1 Quantitative Analysis of the Literature

Quantitative analysis of the literature will show the relationship between requirements engineering and sourcing strategies and if quantitatively the literature holds up to the hypothesis; "The requirements engineering practice can be adapted to accommodate different sourcing strategies". To do this three metrics by which to quantitatively evaluate the literature are used, which are the number of papers, the number of issues and the number of techniques. As seen in the findings section, the literature is divided into issues that requirements engineering in that sourcing strategy faces. Multiple papers may discuss that issue and suggest different techniques to solve that issue. These three metrics of issues, papers and techniques can be categorised by sourcing strategy, requirements engineering element or both. Categorising them as such can tell something about if requirements engineering can be adapted to accommodate a sourcing strategy, what interesting areas for adaptation are when considering a sourcing strategies and how a requirements engineering practice can be adapted to accommodate a sourcing strategy. Tables for the lists of issues, techniques and numbers of each per element and sourcing strategy can be found in Appendix A.

6.1.1 Sourcing Strategies

In this section, differences will be shown between the sourcing strategies. Showing how they are different in the number of issues that research find, the amount of papers and the number of techniques that are suggested to solve the issues. This can show the need for an adaptation of requirements engineering practice to accommodate the different sourcing strategies.

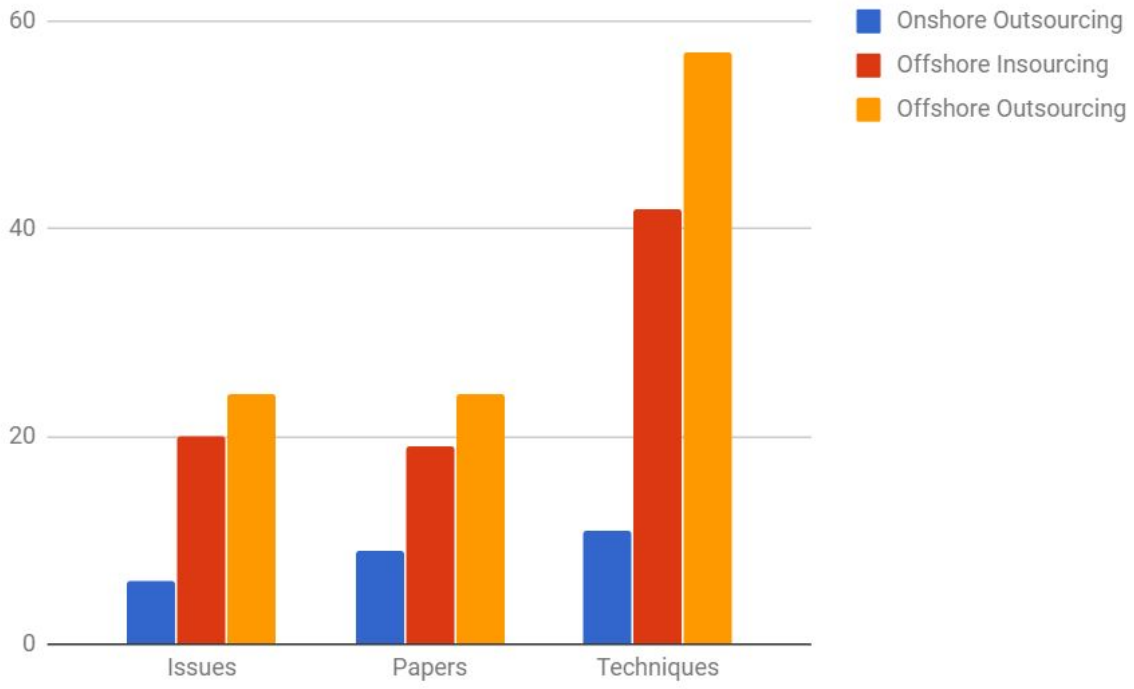

Figure 9. Comparison of sourcing strategies

When looking at the bar graph, it can be seen that it closely resembles *Figure 2* in the introduction of this thesis, whereby experts predicted that the necessity for requirements engineering adaptation would increase as distance increases.

The differences on all metrics between offshore insourcing and offshore outsourcing is less than the differences in metrics between onshore outsourcing and offshore insourcing, which would suggest that the "shoring" aspect of a sourcing strategy is more important than the "sourcing" part of a sourcing strategy. In other words, the distance between client and executing party in terms of geography, culture and time are more relevant than the relationship between client and executing party. There is no exact measure of distance defined in this thesis, thus no solid conclusion can be drawn about the importance of relative distances.

6.1.2 Requirements Engineering Elements

In this section differences will be shown between the requirements engineering elements in light of sourcing strategies. Again the metrics number of issues, papers and techniques are used. This can show the relative interest of the different requirements engineering elements when considering sourcing strategies, and which element deserves more attention when adapting your requirements engineering practice for a sourcing strategy. Note that the axis of this bar chart are switched compared to the sourcing strategy comparison bar chart above.

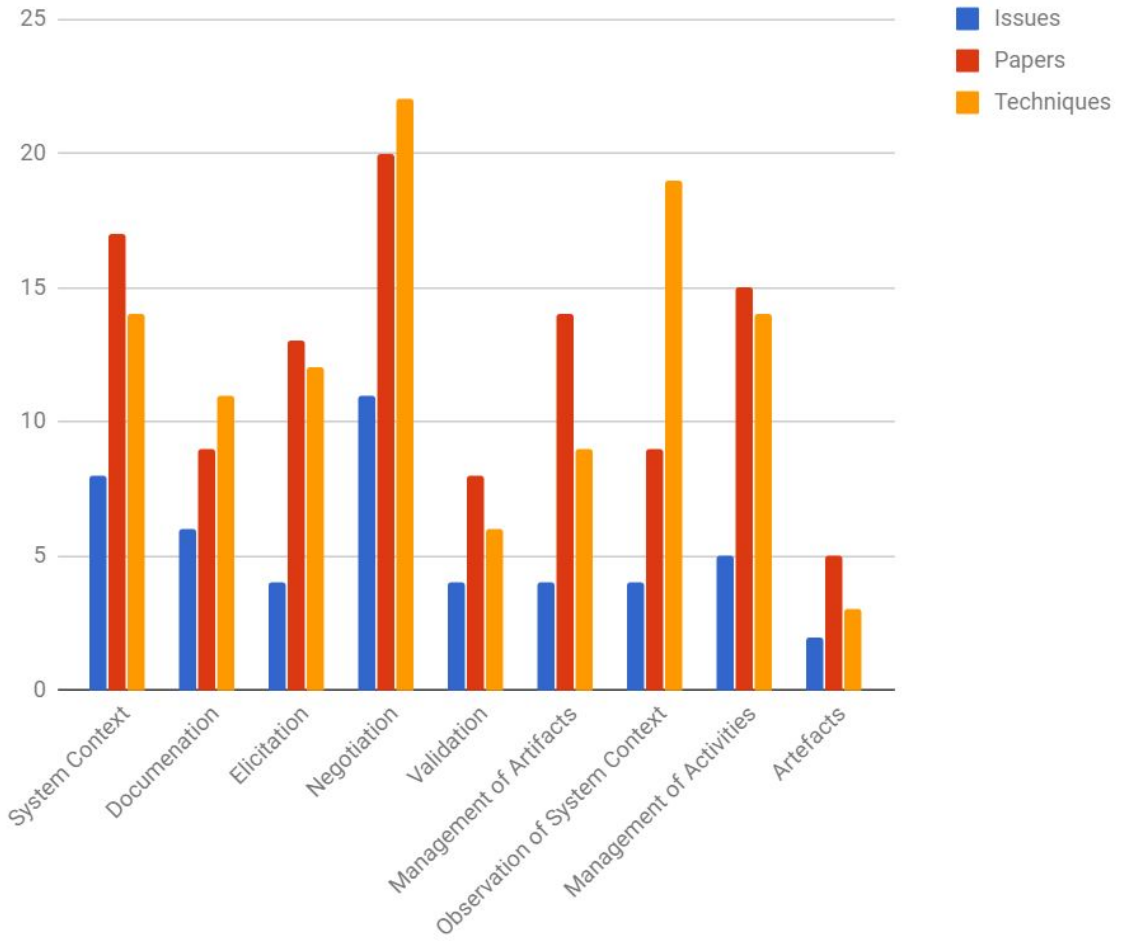

Figure 10. Comparison of requirements engineering elements

When comparing requirements engineering elements it is noteworthy to see the different ratios between issues, papers and techniques. The number of papers suggests an interest in the academic field for that topic. The number of issues, although not weighted in importance, would suggest an element that needs a lot of adaptation. The number of techniques, also not weighted for their effectiveness, can give a rough indication on the effort needed to adapt a requirements engineering element. Furthermore when there is a big difference between the number of issues and techniques for an element, it may suggest a contested area: As so many different techniques are proposed per issue, it could indicate difference of opinion on how to solve an issue or the issue is not resolved to satisfaction as of yet.

Examples of these contested elements are elicitation, management of artefacts, observation of system context and management of activities with more than double the techniques than issues. Observation of system context is the most contested with 4.75 times as many techniques proposed than issues for the element. When considering techniques, only negotiation and observation of system context have the most techniques proposed by authors for. This suggests more effort is required to adapt these elements. For both number of papers and number of issues the negotiation and the system context elements take first and second most important spots. When combining metrics the negotiation element is by far the most important, beating each other element in every metric. This is followed by system context, management of activities and observation of system

context which are all close together. The least interesting element on all metrics is the “artefacts” element.

6.1.3 Both Sourcing Strategy and Requirements Engineering Elements

This section will go into details on how different sourcing strategies compare when looking at the different requirements engineering elements. For this purpose radar charts are used as to easily see both what are areas of interest and how the sourcing strategies compare on that element. When looking at the sourcing strategy comparison in the previous section, a relationship was shown whereby both distance and relationship between client and executing party influences the interest for a topic on all metrics positively. When this relationship does not hold up when comparing requirements engineering elements within different sourcing strategies, it may indicate an area of specific interest for that sourcing strategy.

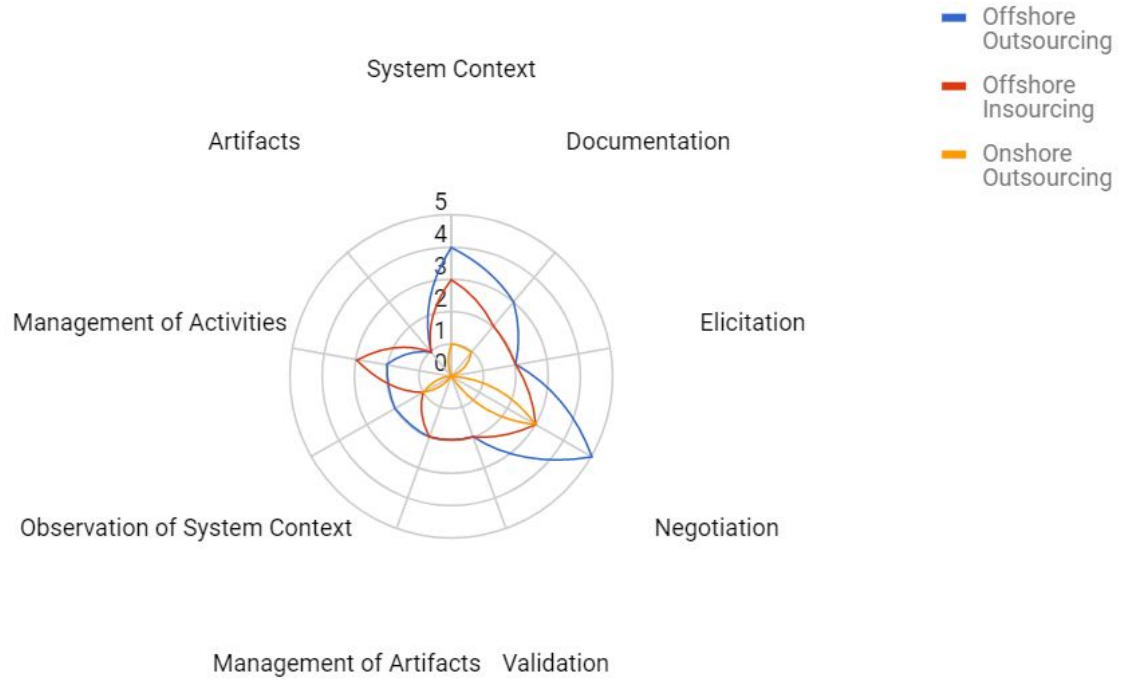

Figure 11. Comparison of Issues

The only place where requirements engineering adaptation relationship with distance breaks when considering the number of issues is for the Management of Activities element for the offshore outsourcing and offshore insourcing strategies. This can be traced back to the extra interest for offshore insourcing when considering the possibility to build a long term effective team and relationships at a distance and the difference in strategy this brings.

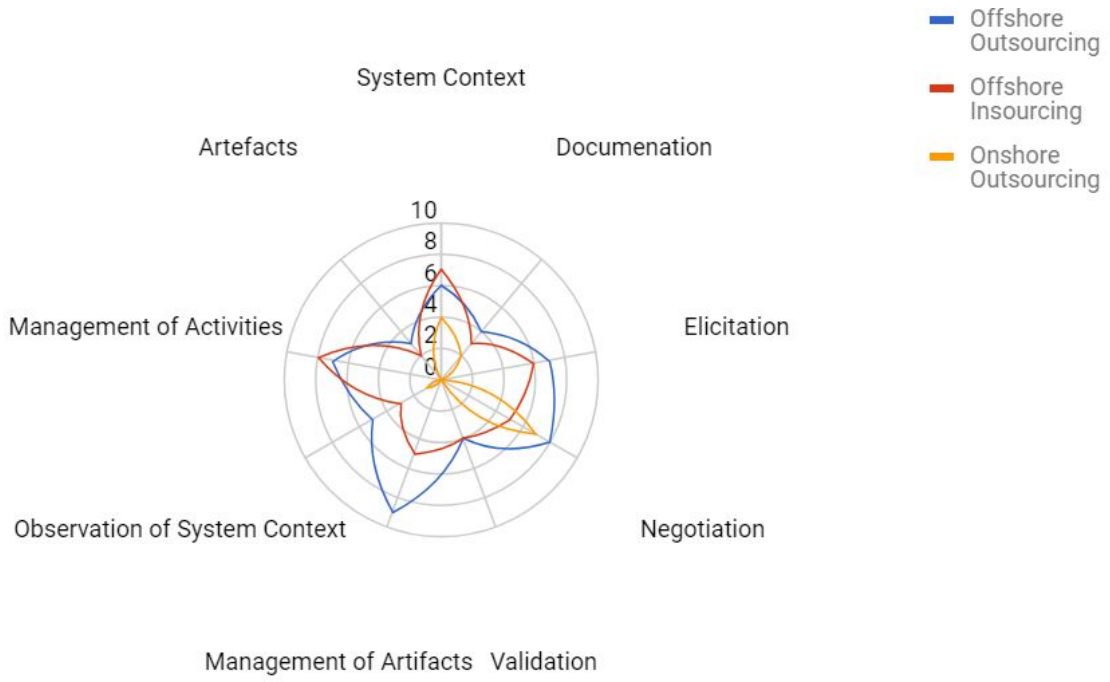

Figure 12. Comparison of Papers

For the amount of papers there are two places where the relationship between distance and requirements engineering adaptation breaks. The first of which is again the management of activities element for the offshore outsourcing and offshore insourcing strategies. The second of which is the negotiation element when comparing offshore insourcing and onshore outsourcing. It can be reasoned that this is natural for this specific element, as negotiation would have to span multiple organisations with outsourcing strategies, adding complexity. This is concurred when looking at the offshore outsourcing strategy, which also has negotiation as an element of interest.

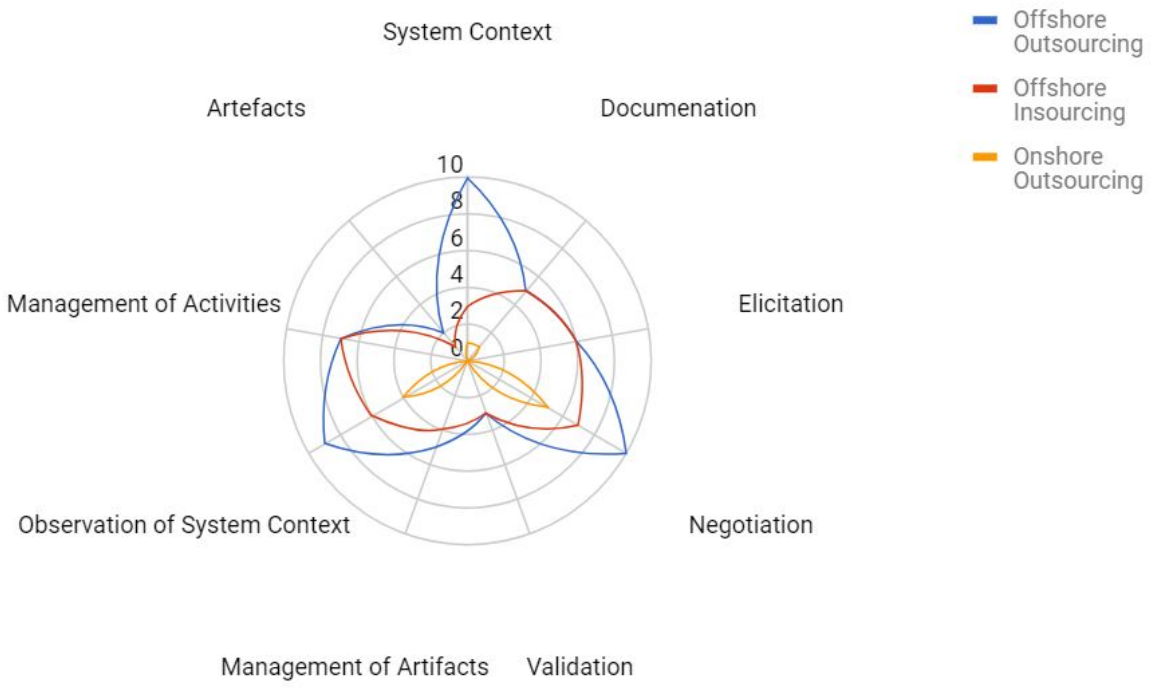

Figure 13. Comparison of techniques

When comparing sourcing strategies on the number of techniques that are suggested for them it can be seen that the relationship between distance and adaptation holds on all points. Only for some areas of offshore insourcing similar numbers are reached as with the offshore outsourcing strategy. Notable are the System Context, Negotiation and Observation of System Context requirements engineering elements where offshore outsourcing has many more techniques than offshore insourcing, which are also the areas on which onshore outsourcing scores on. This would suggest these are areas that outsourcing in general has difficulties with or that these areas need adaptation when it comes to requirements engineering.

6.2 Findings comparison to sourcing strategy literature

Chapter 2.3 draws a relationship between sourcing strategy and software development, and explains how the former affects the latter. As requirements engineering is part of the software development cycle, a logical prediction would be to assume the relationship between requirements engineering and sourcing strategy behaves the same way. In this chapter, a reflection is made on that relationship and how it holds up when looking at requirements engineering specifically. Chapter 2.3 describes lists of effects that share a significant overlap. For that reason, not every item on all three lists will be visited, but every broader category will be touched upon in this chapter. It has to be noted that the items discussed come from literature that covers ‘work at a distance’, which includes the offshore strategies but has less relevance (although not irrelevant) for the onshore strategies.

As a general observation, the suggested techniques from Chapter 5 for adapting requirements engineering to suit a specific sourcing strategy are often not ‘novel’ for requirements engineering. Mostly these are existing techniques also possibly incorporated in standard, non sourcing strategy related requirements engineering practices. A question answered for most literature is which of the techniques work in the context of sourcing strategies and which do not, which allows practitioners in the field to choose a set of techniques depending on their unique circumstances.

Common Ground / Cultural issues
Approaches on this front differ slightly per strategy. For offshore outsourcing a suggestion is made for training the offshore branch on the culture, tools and practices of the onshore team (Bhat et al., 2006). Suggestions for offshore insourcing go a bit further on this level, Prikladnicki and Audy (2012) propose a formal training program for both the onshore and offshore stakeholders, covering working in a GSE setting as well as culture of involved nationalities and business goals and practices. Other than these suggestions, approaches to cultural differences are not suggested.

Coupling work / Strategic issues / Coordination
These sections cover generally how work is divided and coordinated between the onshore and offshore team and have similar solutions for both offshore sourcing strategies. In the literature a case is also made for tightly coupled work as an advantage. The advantage comes from the knowledge of who to contact when an issue arises to quickly solve it. With loosely coupled work it can be cumbersome to find the experts related to an issue. Besides this option, also the suggested decoupling of work (Olson & Olson, 2000) is discussed in literature. This offers the strategic choice between decoupled work or tightly coupled work between the onshore and offshore teams.

Technical issues
Herbsleb and Moitra (2001) discuss possible technical issues that spread across actual hardware and network performance, as well as the difference of tooling and data formats. This former is not mentioned for requirements engineering except for a more general remark that advanced tooling that uses syncing requires network capability to do so on the fly, or the tools become increasingly difficult to use. The latter is a major issue. All authors that mention the issue(Niazi et al., 2012) (Ahmad et al., 2015) (Bhat et al., 2006) agree that a common documentation and tooling scheme among client and executing party is key for project success. This includes the actual sharing of the tooling and documentation standards and their use. Which ‘standard’ to chose is yet open for debate. Suggestions are made to use specific GSE tailored documentation templates (Ali & Lai, 2015b) as well as the use of more common documentation templates and tools with the argument that they are easy to pick up and cope with project team member turnover. Besides this, the outsourcing strategies also have the dilemma of which set of documentation standards and tools to use, those of the client or those of the executing party?

Technology Readiness
The technology readiness section as defined by Olson and Olson (2000) covers both the use of novel technologies and the capability to implement them for the organisation. A number of tools are recommended to supplement requirements engineering practices. Most of these manage knowledge and change management, such as the requirements awareness system (Bhat et al., 2006) and the tool based on work history (Herbsleb & Mockus, 2003). Other tools work to automate part of the requirements engineering practice, such as natural language processing tooling, and evaluation tools such as suggested by Rago et al. (2013). All of these may offer a boon to the requirements engineering practice, but do require a significant change in way of working because the fundamentally change when and how information is stored, and shared. Natural language processing even requires knowledge of training a system on the organisations terms and vocabulary, and demands some skill to interpret the results. The transition to this new way of working across locations and maintaining it is often not explored in depth with the suggestions of the tools, and no comprehensive suggestions are made in literature to implement new requirements engineering technologies.

Collaboration Readiness / Inadequate Communication
Communication channels are discussed often in literature, on how to use instant messaging services, communication tooling and video conferencing. When to use which is less discussed, as is the will to share information. The problem is often phrased as being able to find the right people at the right time, and not so much as to how the right culture is instilled to share knowledge. Suggested solutions for this can be technological and organisational. The former type of solutions include presence awareness features (shared calendars, messaging services that show team members 'coming online'). Organisational solutions for finding the right persons at the right time can be by introducing 'champions of interaction' or just by having smaller project teams which allow every team member to know others personally. Specifically for offshore insourcing, a case is made for direct lines of communication between team members of the same level of hierarchy.

Communication
Ågerfalk and Fitzgerald (2008) discuss communication as the successful exchange of knowledge between sending and receiving party. Suggestions for all sourcing strategies for this section are surprisingly similar, with little difference between offshore outsourcing, offshore insourcing or even onshore outsourcing. Discussions in this category are on synchronous and asynchronous communication, which can be complementary assets for effective communication. Also suggested is the use of a dedicated human facilitator during negotiations. Although some tools are offered as useful, just having a synchronous discussion (flown over or via conferencing) is a very effective tool for communication and decision making compared to just emailing.

Project and process management issues / control
Diversity in advise exists concerning this aspect. There are suggestions for incorporating strict project management policies and adhering to tried and tested development methodologies. On the other hand, authors also point at the slow speed of this approach, preferring more informal tactics and empowering individuals with the capability to make decisions and collaborate personally between onshore and offshore team members to solve issues. Also interesting is the case for agile, which traditionally would not be optimal for work at a distance.

Knowledge management
The knowledge management aspect as defined by Herbsleb and Moitra (2001) is for a large part aimed at controlling changes in the system environment and relegating this to the project in a correct fashion. This is traditionally a large part of requirements engineering, and the same arguments are had in sourcing strategy literature for this topic as for project and process management for both formal change management process as well as informal routes of communication coping with changes. These two routes may be incompatible to some degree: having informal communication between individual project members may be fast, but if they have to wait for formal approval to react to an upcoming change, much of the speed advantage is lost. On the other hand, having a formal process in place to deal with changes ensures a higher standard of quality and guards against scope creep.

Given the many and broad categories described in Chapter 2.3, there is no single technique suggested in the literature to improve requirements engineering with specific sourcing strategies that deviates significantly from the sourcing strategies and software development in general. As seen in the quantitative analysis of the literature, there are but few actual differences in approach between sourcing strategies, there are often just less options to pick and choose from for sourcing strategies that are less distance away. As seen above some of the techniques suggested in literature are to some extent incompatible, which means the organisation has to pick an choose their set of techniques. This will be detailed in the following subsections.

6.3 Incompatible techniques and key decisions

Most techniques as found in literature can be described as part of a situational maturity model, as described by Mettler and Rohner (2009) where the use of all techniques suggested for a sourcing strategy makes the organisation the most mature. However, some of the suggested techniques have some degree of incompatibility, meaning that implementing such incompatible techniques will be detrimental or carry additional risk to the overall requirements engineering practice. This effect can occur for one identified issue, but may also occur on a much broader level, where a technique suggested for one requirements engineering element can influence a technique suggested for another issue entirely. For example, in the negotiation element, in the issue of interaction between stakeholders, a solution is coined to make stakeholders of the same level in the hierarchy communicate with each other to tackle the details of a problem. On the other hand, in the observation of system context element, in the formal mechanisms issue, a focus is put on having a formal and strict policy to handle every change in requirements and scope. The former technique would rely on the expertise of individual stakeholders to speed up the process, the later makes sure the process is done properly to avoid errors down the road. The goals and practice of these techniques are to some degree incompatible. Some of these opposing key decisions are intertwined and can be seen as broader strategies: when a choice is made for one key decision you are also locked in for others. Below the set of key decisions are listed for each sourcing strategy that has a forthcoming key decision to choice for. Briefly the pro's and con's of each choice are listed that can be deduced from the literature.

6.3.1 Key decisions in sourcing strategies

For each sourcing strategy, we present list of the key decisions, their pros and cons and finally the associated techniques for each decision. The full list of techniques can be found in Appendix A. Note that not all techniques discovered in the Findings from the literature (Chapter 5) are listed here, as not all techniques are part of any one key decision, because they are not incompatible with other techniques.

Offshore outsourcing

- Formal Project Change Mechanisms <-> Informal Change Mechanisms

This strategic choice concerns the reliance on formal project mechanisms to guide a change into the project versus the use of informal communication between peers incorporate a change into the project. Formal change mechanisms are better suited to ensure every facet of change management is considered and guarding against unnecessary scope creep. This consideration is however a slow process, while ongoing work continues with old plans, with necessary work redo as a result. Informal mechanisms are faster, allowing changes to spread quickly through the organisation. This does necessitate that there are direct and effective communication lines between peers between onshore and offshore teams.

The techniques for formal project change mechanisms are:

- a policy for processing changes to requirements
- a project structure to relate the dependencies of artefacts and activities

The techniques for informal change mechanisms are:

- ❖ instant chat solution for direct communication between project team members
- ❖ stakeholder interaction between those of the same level in hierarchies between companies in an offshore outsourcing setting
- ❖ a tool that allowed discussions between random team members and logged the results, allowing all stakeholders to look back on what had been discussed
- ❖ reducing the amount of project members in a project
- ❖ having champions of interaction, who know all stakeholders and can act like communication relays for the project

- Decoupled Work <-> Integrated Onshore and Offshore team

Decoupling work is to set down a divide in requirements and early design decisions that allow work to be split between locations with a minimum amount of required communication. This also means that there is less possible miscommunication that can be attributed to cultural differences. On the opposite end, to more quickly resolve issues another strategy would be to increase communication, which allows stakeholders to have more intimate knowledge of expertise of project team members.

The techniques for decoupled work are:

- ❖ decoupling of work across locations
- ❖ paying attention to function interdependency can steer the software architecture in such a direction that minimal interaction between teams is enabled, reducing development time and possibility for communication errors
- ❖ setting up requirements for interaction between sites or responsible parties as it is known which dependencies exist or will arise in the future

The techniques for integrated onshore and offshore team are:

- ❖ training on culture and communication tools used by both onshore and offshore teams
- ❖ instant chat solution for direct communication between project team members
- ❖ a requirements awareness system, with for each expert an outline of their domain, role and responsibilities
- ❖ a tool based on work history. When an expert has worked in the past on a specific piece of documentation, the author is logged and can easily be found when more information on that piece is required
- ❖ stakeholder interaction between those of the same level in hierarchies between companies in an offshore outsourcing setting
- ❖ configuration management systems which are shared between all teams working on aspects of software development
- ❖ remote sharing of screens
- ❖ shared calendars between project members and stakeholders
- ❖ instant chat solution for all project members and stakeholders
- ❖ presence awareness features (showing team members coming online)

- Use of Client Tools and Processes <-> Use of Executing Party Tools and Processes

Often the tools and processes between organisations are not the same. A choice has to be made whether to use those of the client or executing party. The advantage of client tools and processes is the greater understanding of the client's line of thought and business goals, as these tools and processes are often aimed at reaching their specific business goals. The advantage of the executing party's tools and processes are their often higher maturity level. Furthermore the developers are often

far from the business and may need interpret documentation. Their own tools and processes are more familiar to them and having requirements phrased and communicated into their tools and processes may result in fewer interpretation errors. Either way, the client organisation or executing party will have to master to acceptable degree the tools and processes of the other.

The techniques for the use of client tools and processes are:

- ❖ have either party take definitive leadership and set a documentation standard, which includes both a template for document structure as well as how requirements are to be described

The techniques for the use of executing party tools and processes are:

- ❖ have either party take definitive leadership and set a documentation standard, which includes both a template for document structure as well as how requirements are to be described

- Face to face negotiations <-> Tool Assisted Negotiations <-> Video/Audio Conferencing

Most literature agrees that face to face negotiation is superior to other methods, but this would be a choice of practical value. Are face to face negotiations worth the cost for each negotiation or meeting to fly a stakeholder over from location to location? If not, would tool assisted negotiation (tool guided negotiation, often with asynchronous communication and reviewing components guided by the tool's process logic) or perhaps classic video or audio conferencing suffice for most negotiation settings?

The techniques for face to face negotiations are:

- ❖ face-to-face meetings, requiring one or more stakeholders to be flown over to the negotiation site

The techniques for tool assisted negotiations are:

- ❖ a method of asynchronous communication whereby stakeholders used an argumentation scheme tool to collect and tackle issues. In stages comments were then given on each of the issues by all stakeholders

The techniques for video/audio conferencing are:

- ❖ having synchronous discussions instead of asynchronous discussions (by telephone instead of email)
- ❖ synchronous meetings at a distance aided by rich media such as video conferencing
- ❖ audio conferencing

- Standard Documentation & Process <-> GSE tailored Documentation & Process

When practicing an offshore sourcing strategy a wide range of existing project or software development methodologies is available to choose from. Using a well known standard format, such as Scrum, would allow easier introduction of new stakeholders to the project. This is less so for GSE tailored methods, but they have the advantage to take common GSE risks into account and accomodate for them.

The techniques for standard documentation & process are:

- ❖ adhere to a tried and tested development methodology
- ❖ a well defined and relatable software development process shared by all project members

The techniques for GSE tailored documentation & process are:

- ❖ tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE

- Automated Discovery Elicitation <-> Expertise Elicitation

Tooling to perform rudimentary requirements elicitation, such as provided by natural language processing tools, can be utilized instead of traditional expert elicitation. Tooling such as that still requires experts to train the tool and to analyze the output, but is less demanding on key business stakeholder interaction. This can be an interesting option when the client produces a lot of documentation that would otherwise have to be diligently read by an analyst which can be error prone and time consuming.

The techniques for automated discovery elicitation are:

- ❖ a tool that automatically collects non-functional or quality requirements from existing documentation
- ❖ a tool that semi automatically categorizes requirements on their priority and complexity, as well as discerning functional and non-functional requirements
- ❖ a tool that evaluates the user story requirements artefact, and points out low quality areas in the user story to prevent defects later on in the development process
- ❖ a Natural Language Processing method of analysing a requirements specification through decomposing the document into single, simple sentences. These sentences are then analysed and categorised as type of requirements and their priority

The techniques for expertise elicitation are:

- ❖ video conferencing to contact stakeholders
- ❖ occasional visits by representatives of the offshore team

Offshore insourcing

Key decisions for offshore insourcing are a subset of those for offshore outsourcing, and the same descriptions apply. The choices might not be the same, as the arguments for a choice may differ to the offshore outsourcing strategy, and the techniques presented to solve issues can also be different.

- Formal Project Change Mechanisms <-> Informal Change Mechanisms

The techniques for formal project change mechanisms are:

- ❖ no specific techniques for this key decision. It is however present in how requirements engineering is conducted

The techniques for informal change mechanism are:

- ❖ instant chat solution for direct communication between project team members
- ❖ a version control system which can be used to track which project member contributed
- ❖ instant chat solution for all project members and stakeholders
- ❖ reducing the amount of project members in a project
- ❖ having champions of interaction, who know all stakeholders and can act like communication relays for the project
- ❖ building relationships between onshore and offshore team members

- Decoupled Work <-> Integrated onshore and offshore team

The techniques for decoupled work are:

- ❖ decoupling of work across locations
- ❖ paying attention to function interdependency can steer the software architecture in such a direction that minimal interaction between teams is enabled, reducing development time and possibility for communication errors

The techniques for Integrated onshore and offshore are:

- ❖ a formal training programme for GSE and company contextual knowledge
- ❖ instant chat solution for direct communication between project team members
- ❖ a tool based on work history. When an expert has worked in the past on a specific piece of documentation, the author is logged and can easily be found when more information on that piece is required
- ❖ remote sharing of screens
- ❖ shared calendars between project members and stakeholders
- ❖ instant chat solution for all project members and stakeholders
- ❖ presence awareness features (showing team members coming online)
- ❖ building relationships between onshore and offshore team members
- ❖ having direct lines of communication from business stakeholders and offshore project members

- Standard Documentation & Process <-> GSE tailored Documentation & Process

The techniques for standard documentation & process are:

- ❖ no specific techniques for this key decision. It is however present in how requirements engineering is conducted

The techniques for GSE tailored documentation & process are:

- ❖ tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE

- Face to face negotiations <-> Tool Assisted Negotiations <-> Video/Audio Conferencing

The techniques for face to face negotiations are:

- ❖ face-to-face meetings, requiring one or more stakeholders to be flown over to the negotiation site

The techniques for tool assisted negotiations are:

- ❖ a method of asynchronous communication whereby stakeholders used an argumentation scheme tool to collect and tackle issues. In stages comments were then given on each of the issues by all stakeholders
- ❖ a tool that allows discussions between random team members and logged the results, allowing all stakeholders to look back on what had been discussed

The techniques for video/audio conferencing are:

- ❖ having synchronous discussions instead of asynchronous discussions (by telephone instead of email)
- ❖ synchronous meetings at a distance aided by rich media such as video conferencing
- ❖ audio conferencing

- Automated Discovery Elicitation <-> Expertise Elicitation

The techniques for automated discovery elicitation are:

- ❖ a tool that automatically collects non-functional or quality requirements from existing documentation
- ❖ a tool that semi automatically categorizes requirements on their priority and complexity, as well as discerning functional and non-functional requirements
- ❖ a tool that evaluates the user story requirements artefact, and points out low quality areas in the user story to prevent defects later on in the development process
- ❖ a Natural Language Processing method of analysing a requirements specification through decomposing the document into single, simple sentences. These sentences are then analysed and categorised as type of requirements and their priority

The techniques for Expertise Elicitation are:

- ❖ video conferencing to contact stakeholders
- ❖ occasional visits by representatives of the offshore team

Onshore outsourcing

Only one strategic decision remains for onshore outsourcing:

- Face to face negotiations <-> Video / audio Conferencing

The techniques for face to face negotiations are:

- ❖ face to face negotiation meetings

The techniques for video / audio conferencing are:

- ❖ synchronous meetings at a distance aided by rich media such as video conferencing

No mention was given for tool assisted negotiation, but the question remains whether to have all negotiations face to face or if it is acceptable to have most negotiations via video or audio conferencing.

6.3.2 Key decision resolution

Expert interviews were used to resolve the key decisions discussed in the previous sections. During these interviews the experts were explained each key decision and why a conflict occurred. Then they were asked what the pro's and con's of each option is and what, from the perspective of an organisation like Ziggo, would be a preferred direction if such a decision can be made. The experts were chosen for extensive experience with that specific sourcing strategy. For each sourcing strategy one expert was approached for a total of three. A different set of experts was consulted than during previous research steps. Each expert had at least 10 years of experience in IT projects, most of which in relation to that sourcing strategy. The transcriptions of these interviews can be found in Appendix E.

In the following section the results of the interviews are detailed. Each sourcing strategy has its own subsection with discussions for each key decision. The discussions are the condensed versions of the knowledge the expert shared on its domain during the interviews. Then all of the results for all sourcing strategies are compared and analyzed.

Offshore outsourcing

- Formal Project Change Mechanisms <-> Informal Change Mechanisms

No matter how much you specify what you want from a vendor, gaps will remain in your request. To better perform in an offshore outsourcing situation, a tight working relationship with the executing party resolving issues along the way would be prefered. This does require the contracts with the executing party are set up properly to allow this. It is also greatly beneficial to have a business analyst of the client available for contacting from the executing party perspective. Having a direct line to business know how could quickly resolve issues. That said, when a concrete major project decision has to be made this should be finalized through official channels

- Decoupled Work <-> Integrated onshore and offshore team

If different teams work in different locations with their own decoupled deliverables - without communication between these teams - the product will most likely not work as intended. As client you should always facilitate collaboration between people. Preferably any offshore team has a spokesperson or communication liaison onshore to work with the onshore team.

- Use of Client Tools and Processes <-> Use of Executing Party Tools and Processes

The problem with executing party-driven processes is that rarely there is proof of correctly implementing the method that is said is being performed, or that a requirement is correctly covered. However, the executing party should not be forced to operate according to client processes, this will deteriorate the quality of deliverables. To compensate, if a process requirement from the client should be included to the project which demands documentation which shows the execution of the project was sound.

- Face to face negotiations <-> Tool Assisted Negotiations <-> Video/Audio Conferencing

Tool assisted negotiation is a great boon to any project. It helps to format and document requirements in such a way to allow unambiguous requirements engineering. More important however is some extensive use of face to face negotiation. For example, the first two weeks of a project would be wise to consider all team members to be colocated. After this initial period video or audio conferencing would suffice for most situations, because all members are known to each other by this point.

- Standard Documentation & Process <-> GSE tailored Documentation & Process

When the executing party collaborates with the client early in the project, standard documentation suffices. The executing party is by this point known with the client’s documentation and could if necessary reform it to their own standards, without the client needing to reformulate their documentation to cope with GSE challenges.

- Automated Discovery Elicitation <-> Expertise Elicitation

When bigger documentation sets are in play a first go at elicitation may be tool driven. Any steps beyond that need to be done by an expert. There is still too much nuance that can’t be gained from the document by a tool and need further action or elaboration only a person can initiate. A tool would have its place to allow an expert to have easier insight in what documentation may contain.

Offshore insourcing

- Formal Project Change Mechanisms <-> Informal Change Mechanisms

It is all too easy to abuse trust when you rely only on informal relationships to facilitate changes. Changes can get started, hog up time and resources but effort may be wasted when changes are recalled later when approval was lacking. Having a formal forum to discuss both formal and informal change requests or ideas would allow boundaries to be established to any request. This will prevent and resolve issues in later stages of projects.

- Decoupled Work <-> Integrated onshore and offshore team

After the above mentioned boundaries are established, it is useful to have tight cooperation and get information sharing going. Having teams apart from each other will only bring hardships. The organisation should encourage tightly coupled teams with many contact moments and shared documentation.

- Standard Documentation & Process <-> GSE tailored Documentation & Process

This does not really matter. What matters is open and clear communication on what is expected from all parties and what is useful to use in the context at hand. The documentation and process should reach the best results with the least amount of effort.

- Face to face negotiations <-> Tool Assisted Negotiations <-> Video/Audio Conferencing

Projects or initiatives that are critical may find value in face to face negotiations. When time is not constrained simply mailing documents phoning and instant messaging does just fine. When a project is in danger of failing or on a critical timeline it can be very useful to have project members meet face to face.

- Automated Discovery Elicitation <-> Expertise Elicitation

Big documents are a hassle in any situation because no one wants to actually deal with one and go through it all. A strategy to prevent this is to work more incrementally, with smaller steps and documentation. Documents may grow over time, but this is by then material the project members are familiar with which makes it easier to cope with. A requirements engineer should not encounter huge documentation sets with requirements to begin with. With later stages of a project when testers need detailed test scripts larger documents may be inevitable, but it is far better to ensure stakeholders know where to find which relevant information and not have general requests like “review this 300 page document”. A suggestion for this is to partition documentation, with each stakeholder receiving relevant information but no more. These partitions should then have a short umbrella document that covers a high over design so all stakeholders know how their part impacts the whole.

Onshore outsourcing

- Face to face negotiations <-> Video/Audio Conferencing

When discussing high level goals with a broad solution space and possibility for differences in interpretation it is best to be face to face for any negotiation. This is especially true when there is substantial diversity in the participants group part of the negotiations. The problem with video or audio conferencing is the uncertainty of reaching your audience. Without seeing their entire body language it is difficult to see whether or not the audience had difficulty understanding or comprehending your words. Key items for a good negotiation practice. In practical terms having fewer but longer dedicated face to face sessions is better than multiple calls which may discourage or cause confusions. Video or audio conferencing can be used in situations where all parties understand the issue and definitions on a high level, and the negotiation on this point has a goal of filling out details.

Analysis

In *Table 1* the results from the expert interviews are summarized. For most of these a clear favourite was expressed by the sourcing strategy expert, but sometimes a clear decision could not be made or additional notes were given with the decision.

Key Decision	**Offshore Outsourcing**	**Offshore Insourcing**	**Onshore Outsourcing**
Formal project change mechanisms <-> Informal change mechanisms	In favor of informal change mechanisms.	In favor of formal project change mechanisms.	[redacted]
Decoupled work <-> Integrated onshore and offshore team	In favor of Integrated onshore and offshore team.	in favor of Integrated onshore and offshore team.	[redacted]
Use of client tools and processes <-> Use of executing party tools and processes	In favor of use of executing party tools and processes.	[redacted]	[redacted]
Face to face negotiations <-> Tool assisted Negotiations <-> Video/audio conferencing	In favor of project kick-off face to face period, followed by tool assisted and video/audio conferencing.	In favor of face to face communication with critical project timelines or failing projects. Otherwise video/audio conferencing suffices.	In favor of face to face communication with high level negotiation or possibility of miscomprehension from either party. Detail negotiations can be performed with audio/video conferencing.
Standard documentation & Process <-> GSE tailored documentation & process	In favor of standard documentation & process.	Undecided. An optimized process and documentation has to be chosen based on the project at hand.	[redacted]
Automated discovery elicitation <-> Expertise elicitation	Undecided; in some situations automated discovery elicitation may be beneficial, but	Slight favor for expertise elicitation; source material for requirements should	[redacted]

	there are many elicitation steps that a tool cannot complete.	not be large enough to necessitate automated discovery elicitation.	[redacted]

Table 1. Sourcing strategy key decision summary.

Interestingly, there are cases of clear opposition between offshore outsourcing and offshore insourcing. For the key decision of formal or informal change mechanisms, intuitively one would expect offshore outsourcing to be aligned with formal change mechanisms and offshore insourcing with informal change mechanisms because of the different contractual natures between these two. The results from the interviews show the opposite. This may be attributed to differences in personal experience of the experts, or the intuition that is associated with these sourcing strategies is wrong. The consideration for offshore outsourcing towards the informal direction is because of the argument that defining the entire solution space in fine detail to ensure you get what you asked for is an impossible task - practice says gaps will remain. This situation requires a level of informal interaction during the entire the project to fill in the details for best solution. The opposite rationale applies to offshore insourcing into formal change management. Because of the lack of formal boundaries to initiatives and projects, project resources offshore can easily be inefficiently or unjustly allocated. It can be reasoned that the latter can be attributed to exactly the same reason intuitively these two results would be swapped: There is a solid contract protecting change mechanisms for outsourcing where this is not the case for insourcing situations. This would imply that insourcing strategies could benefit from more strictly monitoring agreements made between teams, which would allow further interaction to move more freely as is favoured for offshore outsourcing.

Both for offshore outsourcing and offshore insourcing it is preferred to integrate the dispersed teams as opposed to decoupling their work. This also means it is necessary to ensure they have the ability and empowerment to directly communicate on the same hierarchical level.

All experts for every sourcing strategy agree that some level of face-to-face communication is paramount for project success. Some differences can be spotted however on how much and when. For offshore outsourcing the project kick-off was mentioned as the moment to have intensive face-to-face communication to get to know the other party and stakeholders to make future communication more efficient and effective. For offshore insourcing, face-to-face communication is seen as a trump card to increase the odds of project success when the project is lagging or in danger of failing. For onshore outsourcing all interaction where there can be doubts on miscomprehension face-to-face communication is preferred. That said, the latter has the benefit of lower costs associated with face-to-face communication. Regardless of sourcing strategy or distance, expenditure towards face-to-face communication in every project is worth it.

The different reasoning on automated discovery elicitation is equally interesting. Neither expert was very enthusiastic for this course, perhaps due to their lack of experience with this technology, but the rationale is very different. The expert for offshore insourcing mentioned that the project approach is wrong when a requirements engineer or business analyst is faced with a document large enough for automated discovery elicitation to be effective. Both agree that there are some cases where a tool could be useful, but it would be better to not be in such a situation in the first place.

With these results a preferred framework for requirements engineering in a structured project environment can be created. The key decisions resulting from the experts do not decrease the use of the techniques not chosen, but can encourage a strategy in a certain direction. The final framework with all techniques to adapt to a specific sourcing strategy can be found in Appendix B, including an indication which techniques are prefered.

7 Conclusion

In the introduction of this thesis, the following hypothesis was presented:

“The requirements engineering practice can be adapted to accommodate different sourcing strategies**”**

With the following research questions:

1. How to compare requirements engineering in different structured project environments?
2. What are sourcing strategies and how can sourcing strategies be categorised?
3. How do sourcing strategies affect software development in a structured project environment?
4. How can the requirements engineering practice be adapted to accommodate a specific sourcing strategy?

In the following section, the research questions and the hypothesis will be revisited. For each of these, the relevant findings are summarized and conclusions are drawn. Following that the limitations to the study performed in this thesis will be discussed and what conclusions cannot be drawn from this thesis. Then, a set of recommendations will be given on what future research would advance requirements engineering practice in the context of sourcing strategies. Finally a reflection on the research process and study design will be performed.

7.1 Research Questions

1. How to compare requirements engineering in different structured project environments?

There are two high-level ways to structure a requirements engineering approach. One is based on requirements artefacts, and one based on activities. For this study an activity-based approach was chosen as it was the most common and the most suitable to analyze the existing literature. Three activity-based approaches were discussed: that of SWEBOK (IEEE Computer Society, 2014), the Uni-REPM framework (Kroll & Kruchten, 2003) and the Requirements Engineering framework by Pohl (2010). The first two were discarded for a lack of interconnectivity between distinguished requirements engineering elements and versatility; thus, the framework by Pohl was chosen which had both of those aspects.

2. What are sourcing strategies and how can sourcing strategies be categorised?

From the wide range of discovered sourcing strategies, four were chosen to base this research on as they are the most common in academic literature: Offshore outsourcing, offshore insourcing, onshore outsourcing and onshore insourcing as can be read in Chapter 2.2. The last strategy named is the ‘baseline’ requirements engineering setting, and literature for it is not related to sourcing strategies. Sourcing strategies can be defined by their relationship between the client and executing party and the distance between the two, whereby distance includes geographical, cultural and temporal distance as different attributes that all form a barrier to effective sourcing strategies.

3. How do sourcing strategies affect software development in a structured project environment?

As seen in the literature section Chapter 2.3, the works of Olson and Olson (2000), Herbsleb and Moitra (2001) and Ågerfalk and Fitzgerald (2008) were used to view the impact of sourcing strategies on software development in general. The lists by these authors had a lot of overlap and suggest approaches are needed in the areas of cultural differences, project management, change management, technology and its acceptance and communication. When reflecting back upon that section in the analysis Chapter 6.2, all techniques to adapt requirements engineering practice based on sourcing strategy can be categorised according to the initial sourcing strategy issues as described by the above named authors.

4. How can the requirements engineering practice be adapted to accommodate a specific sourcing strategy?

For each sourcing strategy chosen to use in this thesis literature was analysed and categorised into the requirements engineering elements found in Pohl's (2010) framework, which gave an overview of issues and techniques to solve those issues specific to a requirements engineering aspect. As seen in the analysis, this resulted in sets of techniques for each sourcing strategy with which to adapt a requirements engineering practice. Respectively for offshore outsourcing, offshore insourcing and onshore outsourcing a total number of 57, 42 and 11 techniques were found. For the onshore insourcing strategy no direct literature was found, as this is regular requirements engineering. The number of techniques for the sourcing strategies suggest that the initial assumption on which this thesis was based as seen in *Figure 2* is correct: both the relationship between the client and executing party as well as the distance between the client and the executing party matter, whereby an increase in distance as well as relationship increases the number of techniques found in literature to adapt the requirements engineering practice.

Among the techniques some incompatibility was discovered. This requires a set of choices to be made which techniques to apply for the requirements engineering practice for that specific sourcing strategy. For this inconsistency issue also counts that increases in distance and relationship have a higher number of key decisions inherent to them, respectively 6, 5 and 1. To validate which of these choices is best suited for that sourcing strategy, experts were asked to assess the options and make a choice based on their knowledge and experience which set would be most beneficial for the overall sourcing strategy and requirements engineering practice.

The expert interviews resulted in most key decisions to be made distinctly, which means that these techniques would be preferred in the context of that specific sourcing strategy with a structured project environment. The most interesting and unexpected result was the preference for informal change management in offshore outsourcing, and the preference for formal change management in offshore insourcing. This would suggest the contract which is in place for offshore outsourcing facilitates small changes to be made on the fly, because the boundary for these changes is safeguarded by the contract. Offshore outsourcing might then benefit from a more formal boundary setting to allow the same level of informal change management further down in the project. When the preference for the key decisions is taken into account, the number of techniques is distilled down to 48 techniques for offshore outsourcing, 32 techniques for offshore insourcing and 11 techniques for onshore outsourcing. This does not make the unchosen techniques less valid, but the incompatibility of the techniques requires a choice to be made and in the context of a structured project environment the chosen techniques may be prefered. The framework is as such limited in the preference choice as it is based on a selected set of experts, and this preference may be different in other situations. When using the framework the unchosen techniques can be considered when these make more sense strategically. The complete framework of techniques for each sourcing strategy, including which techniques are preferred or not, can be found in the appendix.

Considering that a set of techniques improving requirements engineering for each sourcing strategy can be assembled, a conclusion can be given to the hypothesis of this thesis:
"The requirements engineering practice can be adapted to accommodate different sourcing strategies"

In Appendix B the completed frameworks of the different sourcing strategies can be found. The use of these frameworks assumes foreknowledge of requirements engineering: These techniques add value to existing requirements engineering practice when faced with the context of sourcing strategies. As such, they can be used by practitioners in the field to improve their requirements engineering process and reduce the risk of project failure related to requirements engineering. Furthermore, they can be used by practitioners used to one sourcing strategy when switching to another, showing what practices are common and what new techniques can be used to tackle the specific trials of that sourcing strategy. As these techniques have not been tested as a combined set such a recommendation cannot be made. Techniques can be chosen whichever best suits the needs of the project and context at hand, and can be considered to have equal value.

7.2 Limitations

A limitation to this study design is an additional validation step on the value of the different techniques. The techniques found to adapt requirements engineering practice are separately studied by authors elsewhere and when, but these techniques are not compared to each other in this study. When searching for a 'best value' set of techniques when not enough resources are available to implement all of them (which is a safe assumption), there is no advice given on which to pick and choose or what technique would yield the best results. Besides that, the experts that made the set of key decisions may have done so from their personal environment and experience which do not reflect all organisations. Different experts or different organisational surroundings may have made these choices differently, and the techniques that are left out may be more suitable in other organisations.

Another limitation involves distance, which as defined from the start of this thesis envelops geographical, cultural and temporal distance. These three aspects are known to cause disparity in sourcing strategies, but the the amount of disparity coming from these aspects compared to the others is however not known, and adapting your requirements engineering practice to just one of these aspects, such as geographical distance, may not offer an accurate image of reality. Furthermore, with the results gained from this research no comparative value can be given to the 'distance' element. This in total makes it difficult at best to compare different 'offshore' locations as defined in this study. Most academic input used refers to an offshore branch 'far' away, with high disparity in geographical, cultural and temporal distance as this is the most common offshoring sourcing strategy set up, but no filtering has been done to only include papers with exactly such a set up. It is my expectation that increases in distance in all its aspects gradually increases the prudency of including more adapted requirements engineering techniques, but this result cannot be concluded from this thesis.

7.3 Recommendations for future study

In this section three recommendations for future study are given that would advance the topic of requirements engineering practice in the context of sourcing strategies further. Each will look into the benefit of further study and offer a suggestion in how the topic can be studied.

Validation of the framework in practice

The validation step planned for this dissertation was changed because of the incompatible requirements engineering techniques - choice was made what to validate considering time constraints. This changed the research set up in hindsight. This makes the study design by (Gorschek et al., 2006) somewhat incomplete, whereby a formal last validation step can still be performed. For the future research suggestions, this validation step would go well together with the creation of a situational maturity model or situational method as described in the following section, but was not included in this thesis.

Situational maturity models and methods

Future research that would greatly improve the practical applicability of the set of techniques found in this thesis is the creation of either a situational maturity model or a situational method, as described by Mettler and Rohner (2009) and Brinkkemper (1996). A situational maturity model would allow a company to measure their own maturity for coping with a sourcing strategy and look for the best value future improvements to aim for. Later studies would be able to gather the maturity of multiple companies and set down a benchmark, which would allow an organisation to view how it measures up against others. For a situational maturity model a study would look at the comparative value of the different techniques for each sourcing strategy found in this thesis, for example by having a pool of experts rank the techniques in order of their value. A situational method for each sourcing strategy allows an organisation to plan their requirements engineering practice and would serve as a handhold for practitioners just entering the field of new sourcing strategy. A study design would look at the flow of different techniques and how and when these would need to be applied for an optimal result.

Comparative distances

As stated in the limitations, the study design in this thesis does not allow for the comparison of different distances. One ‘offshore’ sourcing strategy is as distant as another. Although shown that distance does matter for requirements engineering, not shown is how much it matters, and how much to adapt your requirements engineering practice based on a certain amount of distance. Future research that compares distances and the prudence of adapted requirements engineering techniques would allow an informed decision to be made about requirements engineering when an executing party is closer or further away. A stepping stone for approaching differences in distance and its effect would be to include more sourcing strategies that are commonly known (and discussed in the literature section), which would have more support in academia. For example, *offshore* could be split up into *nearshore* (the executing party is situated in a country close to the client country) and *farshore* (the executing party is situated far away from the client country). Having these two stepping stones would facilitate more informed decisions about adapting requirements engineering practice as well as allowing the creation of an hypothesis on how exactly distance and adapted requirements engineering are related.

Three aspects of distance

Distance in this thesis included three different aspects; geographical, cultural and temporal distance. The influence of these three comparatively to each other is not known. In this study this stops a practitioner from making an informed decision on how much and with which techniques to adapt a requirements engineering practice.

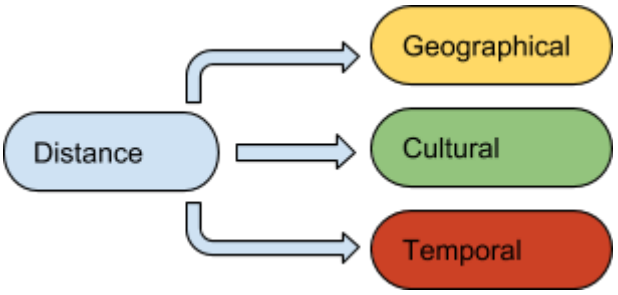

Figure 14. The Three types of distance that influence sourcing strategy.

Future research could compare the different aspects and give them comparative value, which would be a great boon to all sourcing strategy research. A possible study design would find situations where only one of three aspects is significantly different and the others are as equal as possible to see the effect of each. The obvious difficulty for such a study design is objectively quantifying cultural distance, and the probable reason such a study is not done already.

8 References

Ågerfalk, P. J., & Fitzgerald, B. (2008). Outsourcing to an Unknown Workforce: Exploring Opensurcing as a Global Sourcing Strategy. *The Mississippi Quarterly*, *32*(2), 385–409.

Agerfalk, P. J., Fitzgerald, B., Holmstrom Olsson, H., Lings, B., Lundell, B., & Ó Conchúir, E. (2005). A framework for considering opportunities and threats in distributed software development. Retrieved from https://ulir.ul.ie/handle/10344/2347

Ahmad, R. B., Nasir, M. H. N. M., Iqbal, J., & Zahid, S. M. (2015). High Perceived-Value Requirements Engineering Practices for Outsourced Software Projects. *JSW*, *10*(10), 1199–1215.

Ali, N., & Lai, R. (2015a). A method of software requirements specification and validation for global software development. *Requirements Engineering*, 1–24.

Ali, N., & Lai, R. (2015b). A method of software requirements specification and validation for global software development. *Requirements Engineering*, 1–24.

Anderson, M. G., & Katz, P. B. (1998). Strategic Sourcing. *The International Journal of Logistics Management*, *9*(1), 1–13.

Bhat, J. M., Gupta, M., & Murthy, S. N. (2006). Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing. *IEEE Software*, *23*(5), 38–44.

Boehm, B., Abi-Antoun, M., Port, D., Kwan, J., & Lynch, A. (1999). Requirements engineering, expectations management, and the Two Cultures. In *Proceedings IEEE International Symposium on Requirements Engineering (Cat. No.PR00188)* (pp. 14–22). ieeexplore.ieee.org.

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, *38*(4), 275–280.

Britto, R., Mendes, E., & Wohlin, C. (2016). A Specialized Global Software Engineering Taxonomy for Effort Estimation. In *2016 IEEE 11th International Conference on Global*

Software Engineering (ICGSE) (pp. 154–163). ieeexplore.ieee.org.

Carmel, E., & Abbott, P. (2007). Why'nearshore'means that distance matters. *Communications of the ACM*. Retrieved from http://dl.acm.org/citation.cfm?id=1290959

Cheng, B. H. C., & Atlee, J. M. (2007a). Research Directions in Requirements Engineering. In *2007 Future of Software Engineering* (pp. 285–303). Washington, DC, USA: IEEE Computer Society.

Cheng, B. H. C., & Atlee, J. M. (2007b). Research Directions in Requirements Engineering. In *2007 Future of Software Engineering* (pp. 285–303). Washington, DC, USA: IEEE Computer Society.

CMMI Product Team. (2006). CMMI for Development, Version 1.2. Retrieved from http://repository.cmu.edu/sei/387/

Damian, D. E., Eberlein, A., Shaw, M. L. G., & Gaines, B. R. (2003). An exploratory study of facilitation in distributed requirements engineering. *Requirements Engineering*, *8*(1), 23–41.

Damian, D., Lanubile, F., & Mallardo, T. (2006). An Empirical Study of the Impact of Asynchronous Discussions on Remote Synchronous Requirements Meetings. In *Fundamental Approaches to Software Engineering* (pp. 155–169). Springer Berlin Heidelberg.

Fernández, D. M., Lochmann, K., Penzenstadler, B., & Wagner, S. (2011). A case study on the application of an artefact-based requirements engineering approach. In *Evaluation Assessment in Software Engineering (EASE 2011), 15th Annual Conference on* (pp. 104–113). ieeexplore.ieee.org.

Fernández, D. M., & Penzenstadler, B. (2014). Artefact-based requirements engineering: the AMDiRE approach. *Requirements Engineering*, *20*(4), 405–434.

Firesmith, D. (2004). Prioritizing requirements. *Journal of Object Technology*, *3*(8), 35–48.

Gorschek, T., Wohlin, C., Carre, P., & Larsson, S. (2006). A Model for Technology Transfer

in Practice. *Software, IEEE*, *23*(6), 88–95.

Gotel, O. C. Z., & Finkelstein, C. W. (1994). An analysis of the requirements traceability problem. In *Proceedings of IEEE International Conference on Requirements Engineering* (pp. 94–101). ieeexplore.ieee.org.

Han, W.-M., & Huang, S.-J. (2007). An empirical analysis of risk components and performance on software projects. *The Journal of Systems and Software*, *80*(1), 42–50.

Herbsleb. (2007). Global Software Engineering: The Future of Socio-technical Coordination. In *2007 Future of Software Engineering* (pp. 188–198). Washington, DC, USA: IEEE Computer Society.

Herbsleb, J. D., & Mockus, A. (2003). An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, *29*(6), 481–494.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE Software*, *18*(2), 16–20.

Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software*. Retrieved from http://search.proquest.com/openview/481781039249c6bd66960939ddbc810e/1?pq-origsite=gscholar

Hofner, G., Mani, V. S., Nambiar, R., & Apte, M. (2011). Fostering a High-Performance Culture in Offshore Software Engineering Teams Using Balanced Scorecards and Project Scorecards. In *2011 IEEE Sixth International Conference on Global Software Engineering* (pp. 35–39). ieeexplore.ieee.org.

IEEE Computer Society. (2014). *Guide to the Software Engineering Body of Knowledge (Swebok(r)): Version 3.0* (3rd ed.). Los Alamitos, CA, USA: IEEE Computer Society Press.

Jarke, M., & Pohl, K. (1993). Establishing visions in context: towards a model of

requirements processes. *ICIS Chemical Business*. Retrieved from http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1015&context=icis1993

Kraut, R. E., & Streeter, L. A. (1995). Coordination in software development. *Communications of the ACM*, *38*(3), 69–82.

Kroll, P., & Kruchten, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Pearson Education.

Lormans, M., van Dijk, H., van Deursen, A., Nocker, E., & de Zeeuw, A. (2004). Managing evolving requirements in an outsourcing context: an industrial experience report. In *Proceedings. 7th International Workshop on Principles of Software Evolution, 2004.* (pp. 149–158). ieeexplore.ieee.org.

Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2015). Forging high-quality User Stories: Towards a discipline for Agile Requirements. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)* (pp. 126–135). ieeexplore.ieee.org.

Martin, S., Aurum, A., Jeffery, R., & Paech, B. (2002). Requirements engineering process models in practice. *7th Australian Workshop on*. Retrieved from http://www.nyu.edu/classes/jcf/CSCI-GA.2440-001_sp12/handouts/10.1.1.3.6502.pdf

Méndez Fernández, D., & Kuhrmann, M. (2009). Artefact-based requirements engineering and its integration into a process framework. *Institut Für Informatik, Technische Universität München, München*.

Mettler, T., & Rohner, P. (2009). Situational Maturity Models As Instrumental Artifacts for Organizational Design. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology* (pp. 22:1–22:9). New York, NY, USA: ACM.

Mockus, A., & Weiss, D. M. (2001). Globalization by chunking: a quantitative approach. *IEEE Software*, *18*(2), 30–37.

Mylopoulos, J., Borgida, A., Jarke, M., & Koubarakis, M. (1990). Telos: Representing Knowledge About Information Systems. *ACM Transactions on Information and System Security*, *8*(4), 325–362.

Nakatsu, R. T., & Iacovou, C. L. (2009). A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study. *Information & Management*, *46*(1), 57–68.

Niazi, M., El-Attar, M., & Usman, M. (2012). An empirical study identifying high perceived value requirements engineering practices in global software development projects. *Proc. of 7th Intl. Conf. on*.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements Engineering: A Roadmap. In *Proceedings of the Conference on The Future of Software Engineering* (pp. 35–46). New York, NY, USA: ACM.

Olson, G. M., & Olson, J. S. (2000). Distance Matters. *Hum. -Comput. Interact.*, *15*(2), 139–178.

Paasivaara, M., & Lassenius, C. (2006). Could Global Software Development Benefit from Agile Methods? In *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)* (pp. 109–113). ieeexplore.ieee.org.

Philip, T., Schwabe, G., & Wende, E. (2010). Identifying Early Warning Signs of Failures in Offshore Software Development Projects-A Delphi survey. *Aisel.aisnet.org*. Retrieved from http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1465&context=amcis2010

Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques* (1st ed.). Springer Berlin Heidelberg.

Prikladnicki, R., & Audy, J. L. N. (2012). Managing Global Software Engineering: A Comparative Analysis of Offshore Outsourcing and the Internal Offshoring of Software Development. *Information Systems Management*, *29*(3), 216–232.

Rago, A., Marcos, C., & Diaz-Pace, J. A. (2013). Uncovering quality-attribute concerns in

use case specifications via early aspect mining. *Requirements Engineering*, *18*(1), 67–84.

Ritzer, G., & Lair, C. (2007). Outsourcing: Globalization and beyond. *The Blackwell Companion to Globalization*. Retrieved from https://books.google.nl/books?hl=en&lr=&id=XKnmvRATtfAC&oi=fnd&pg=PA307&dq=%22definition+of+sourcing%22+software+development&ots=_ZNKRd2NyX&sig=UVYKXDjUFM-scC1AhHrcQaNlm4Q

Ruiz-Rube, I., Dodero, J. M., Palomo-Duarte, M., Ruiz, M., & Gawn, D. (2013). Uses and applications of Software & Systems Process Engineering Meta-Model process models. A systematic mapping study. *Journal of Software: Evolution and Process*, *25*(9), 999–1025.

Sharma, A., & Kushwaha, D. S. (2011). Natural Language Based Component Extraction from Requirement Engineering Document and Its Complexity Analysis. *SIGSOFT Softw. Eng. Notes*, *36*(1), 1–14.

Sinha, V., Sengupta, B., & Chandra, S. (2006). Enabling Collaboration in Distributed Requirements Management. *IEEE Software*, *23*(5), 52–61.

Smite, D., & Wohlin, C. (2011). A whisper of evidence in global software engineering. *IEEE Software*, *28*(4), 15.

Šmite, D., Wohlin, C., Galviņa, Z., & Prikladnicki, R. (2014). An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineer*, *19*(1), 105–153.

Šmite, D., Wohlin, C., Gorschek, T., & Feldt, R. (2010). Empirical evidence in global software engineering: a systematic review. *Empirical Software Engineer*, *15*(1), 91–118.

Somerville, I., & Sawyer, P. (1997). Requeriments Engineering: A Good Practices.

Sommerville, I., & Sawyer, P. (1997). *Requirements Engineering: A Good Practice Guide* (1st ed.). New York, NY, USA: John Wiley & Sons, Inc.

Svahnberg, M., Gorschek, T., Nguyen, T. T. L., & Nguyen, M. (2015). Uni-REPM: a framework for requirements engineering process assessment. *Requirements Engineering*, *20*(1), 91–118.

Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology*, *56*(1), 54–78.

Weerd, I. v., Versendaal, J. M., & Brinkkemper, S. (2006). A product software knowledge infrastructure for situational capability maturation: Vision and case studies in product management. *Technical Report UU-CS*, *2006*, 1–16.

9 Appendix

9.1 Appendix A: Tables

List of Issues and Techniques for the offshore outsourcing strategy found in literature

	Issues	Techniques
System Context	Domain knowledge prevalent with offshore team	training on culture and communication tools used by both onshore and offshore teams
		consensus on operating norms such as how a meeting works and how to comply to commitments
		sharing of requirements specification templates
		requirements have to be made more explicit
	Lack of informal communication	instant chat solution for direct communication between project team members
	Shared responsibility	establishing fixed methods for project reporting and performance metrics
		adhere to a tried and tested development methodology
		have frequent and small deliverables from all team members to allow greater control and earlier signaling of conflicts
	Finding expertise	a requirements awareness system, with for each expert an outline of their domain, role and responsibilities
		a tool based on work history. When an expert has worked in the past on a specific piece of documentation, the author is logged and can easily be found when more information on that piece is required
Documentation	Documentation Standard	have either party take definitive leadership and set a documentation standard, which includes both a template for document structure as well as how requirements are to be described
	GSE tailored documentation templates	tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE
	Documentation supplements	include a summary of the requirements
		use diagrams to accompany the requirement text
		a written out vocabulary of terms
Elicitation	Participation of stakeholders	constant pressure from high level management on the client side is required to ensure the full participation of all stakeholders

		video conferencing to contact stakeholders
		occasional visits by representatives of the offshore team
	Natural Language Processing	a tool that automatically collects non-functional or quality requirements from existing documentation
		a tool that semi automatically categorizes requirements on their priority and complexity, as well as discerning functional and non-functional requirements
		a tool that evaluates the user story requirements artefact, and points out low quality areas in the user story to prevent defects later on in the development process
Negotiation	Slow negotiation due to cultural or geographical distance	having synchronous discussions instead of asynchronous discussions (by telephone instead of email)
	Communication	prioritization of requirements
		making it someone's responsibility to make sure requirements prioritization takes place
	Interaction between stakeholders	explicitly developing stakeholder viewpoints, including stakeholder satisfaction as a successfactor and involving stakeholders in the creation of a common vision
		share requirements information with all stakeholders, before negotiation commences
		stakeholder interaction between those of the same level in hierarchies between companies in an offshore outsourcing setting
	Synchronous Negotiation	face-to-face meetings, requiring one or more stakeholders to be flown over to the negotiation site
		synchronous meetings at a distance aided by rich media such as video conferencing
		a human facilitator during negotiation meetings
	Asynchronous Negotiation	a method of asynchronous communication whereby stakeholders used an argumentation scheme tool to collect and tackle issues. In stages comments were then given on each of the issues by all stakeholders
Validation	Validation in offshore setting	tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE
		a common vocabulary included in the requirements engineering artefacts
	Natural Language Processing	a Natural Language Processing method of analysing a requirements specification through decomposing the document into single, simple sentences. These sentences are then analysed and categorised as type of requirements and their priority

Managing Artefacts	Traceability problem	use of a centralized requirements depository or requirements management system
		use of hypertexting in documentation to create a structure and visible relations between artefacts and their precursor documents
	Awareness of project members	a version control system which can be used to track which project member contributed
		configuration management systems which are shared between all teams working on aspects of software development
		remote sharing of screens
Observation of System Context	Formal mechanisms	a policy for processing changes to requirements
		accountability on requirements
		keeping track of the original goals throughout the project
	Informal Mechanisms	audio conferencing
		shared calendars between project members and stakeholders
		instant chat solution for all project members and stakeholders
		presence awareness features (showing team members coming online)
		a tool that allowed discussions between random team members and logged the results, allowing all stakeholders to look back on what had been discussed
		a tool that allows stakeholders to subscribe to requirements, which alerts them whenever a change to the specific requirement was made
Managing Activities	Way of working	a well defined and relatable software development process shared by all project members
		a project structure to relate the dependencies of artefacts and activities
		use of agile methodologies (such as Scrum)
	Delays	adjust the expected timelines / planning for outsourced work (it generally takes longer)
		decoupling of work across locations
		reducing the amount of project members in a project
		having champions of interaction, who know all stakeholders and can act like communication relays for the project
Artefacts	Task dependencies	paying attention to function interdependency can steer the software architecture in such a direction that minimal interaction between teams is enabled, reducing development time and possibility for communication errors

		setting up requirements for interaction between sites or responsible parties as it is known which dependencies exist or will arise in the future

List of Issues and Techniques for the offshore insourcing Strategy found in literature

	Issues	Techniques
System Context	Domain knowledge prevalent with offshore team	a formal training programme for GSE and company contextual knowledge
	Lack of informal communication	instant chat solution for direct communication between project team members
	Finding expertise	a tool based on work history. When an expert has worked in the past on a specific piece of documentation, the author is logged and can easily be found when more information on that piece is required
Documentation	Documentation Standards and Templates	all sites use the same documentation standard
		tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE
	Documentation supplements	include a summary of the requirements
		use diagrams to accompany the requirement text
		a written out vocabulary of terms
Elicitation	Participation of stakeholders	constant pressure from high level management on the client side is required to ensure the full participation of all stakeholders
		video conferencing to contact stakeholders
		occasional visits by representatives of the offshore team
	Natural Language Processing	a tool that automatically collects non-functional or quality requirements from existing documentation
		a tool that semi automatically categorizes requirements on their priority and complexity, as well as discerning functional and non-functional requirements
		a tool that evaluates the user story requirements artefact, and points out low quality areas in the user story to prevent defects later on in the development process
Negotiation	Requirements	having synchronous discussions instead of asynchronous

	Communication	discussions (by telephone instead of email)
		prioritization of requirements
		share requirements information with all stakeholders, before negotiation commences
	Synchronous Negotiation	face-to-face meetings, requiring one or more stakeholders to be flown over to the negotiation site
		synchronous meetings at a distance aided by rich media such as video conferencing
		a human facilitator during negotiation meetings
	Asynchronous Negotiation	a method of asynchronous communication whereby stakeholders used an argumentation scheme tool to collect and tackle issues. In stages comments were then given on each of the issues by all stakeholders
Validation	Validation in offshore setting	tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE
		a common vocabulary included in the requirements engineering artefacts
	Natural Language Processing	a Natural Language Processing method of analysing a requirements specification through decomposing the document into single, simple sentences. These sentences are then analysed and categorised as type of requirements and their priority
Managing Artefacts	Centralized Depository System	use of a centralized requirements depository or requirements management system
		use of hypertexting in documentation to create a structure and visible relations between artefacts and their precursor documents
	Awareness of project members	a version control system which can be used to track which project member contributed
		remote sharing of screens
Observation of System Context	Informal Mechanisms	audio conferencing
		shared calendars between project members and stakeholders
		instant chat solution for all project members and stakeholders
		presence awareness features (showing team members coming online)
		a tool that allowed discussions between random team members and logged the results, allowing all stakeholders to look back on what had been discussed

		a tool that allows stakeholders to subscribe to requirements, which alerts them whenever a change to the specific requirement was made
Managing Activities	Working agile	use of agile methodologies (such as Scrum)
	Delays	adjust the expected timelines / planning for outsourced work (it generally takes longer)
		decoupling of work across locations
		reducing the amount of project members in a project
		having champions of interaction, who know all stakeholders and can act like communication relays for the project
	Roles and Responsibilities	building relationships between onshore and offshore team members
		having direct lines of communication from business stakeholders and offshore project members
Artefacts	Task dependencies	paying attention to function interdependency can steer the software architecture in such a direction that minimal interaction between teams is enabled, reducing development time and possibility for communication errors

List of Issues and Techniques for the onshore outsourcing Strategy found in literature

	Issues	Onshore Outsourcing
System Context	Collocation	putting outsourced teams together on the same location as the involved business stakeholders, less than 30 meters apart
Documentation	Standard Documentation Structure	all sites use the same documentation standard
Elicitation		
Negotiation	Prioritizing Requirements	prioritization of requirements
	Face to face negotiation	face to face negotiation meetings
		synchronous meetings at a distance aided by rich media such as video conferencing
	Managing expectations	make stakeholders aware of relative effort required for specific requirements
		stakeholder Win-Win approach or the MBASE Approach: such approaches reconcile Win conditions for the client (resulting system capabilities) with Win conditions for the developer (minimal risk to going beyond schedule or budget). When a win condition for the client poses a significant risk to a win condition of the developer, this is stated as an issue that has to be resolved or agreed upon

Validation		
Managing Artefacts		
Observation of System Context	Informal Communication	shared calendars between project members and stakeholders
		instant chat solution for all project members and stakeholders
		presence awareness features (showing team members coming online)
		collocation of project team members
Managing Activities		
Artefacts		

Comparison of sourcing strategies

	Onshore Outsourcing	Offshore Insourcing	Offshore Outsourcing
Issues	6	20	24
Papers	9	19	24
Techniques	11	42	57

Comparison of requirements engineering elements

	Issues	Papers	Techniques
System Context	8	17	14
Documentation	6	9	11
Elicitation	4	13	12
Negotiation	11	20	22
Validation	4	8	6
Management of Artifacts	4	14	9
Observation of System Context	4	9	19
Management of Activities	5	15	14
Artefacts	2	5	3

Radar chart data for papers

	Offshore Outsourcing	Offshore Insourcing	Onshore Outsourcing
System Context	6	7	4
Documentation	4	3	2
Elicitation	7	6	0
Negotiation	8	5	7
Validation	4	4	0
Management of Artifacts	9	5	0
Observation of System Context	5	3	1
Management of Activities	7	8	0
Artefacts	3	2	0

Radar Chart Data for Issues

	Offshore Outsourcing	Offshore Insourcing	Onshore Outsourcing
System Context	4	3	1
Documentation	3	2	1
Elicitation	2	2	0
Negotiation	5	3	3
Validation	2	2	0
Management of Artifacts	2	2	0
Observation of System Context	2	1	1
Management of Activities	2	3	0
Artifacts	1	1	0

Radar Chart data for Techniques

	Offshore Outsourcing	Offshore Insourcing	Onshore Outsourcing

System Context	10	3	1
Documentation	5	5	1
Elicitation	6	6	0
Negotiation	10	7	5
Validation	3	3	0
Management of Artifacts	5	4	0
Observation of System Context	9	6	4
Management of Activities	7	7	0
Artefacts	2	1	0

9.2 Appendix B: Finished requirements engineering framework

Below are the three requirements engineering frameworks filled in for each sourcing strategy. Each requirements engineering element has a set of associated techniques found in the literature, discussed in Chapter 5. Some of the techniques are incompatible, as discussed in Chapter 6.3. The preferred techniques found through the expert interviews are highlighted bold in the frameworks. The unpreferred techniques are greyed out in the frameworks.

Offshore outsourcing

Requirements engineering Element	Adaptation Techniques
System Context	**training on culture and communication tools used by both onshore and offshore teams**
	consensus on operating norms such as how a meeting works and how to comply to commitments
	sharing of requirements specification templates
	requirements have to be made more explicit
	instant chat solution for direct communication between project team members
	establishing fixed methods for project reporting and performance metrics
	adhere to a tried and tested development methodology
	have frequent and small deliverables from all team members to allow greater control and earlier signaling of conflicts
	a requirements awareness system, with for each expert an outline of their domain, role and responsibilities
	a tool based on work history. When an expert has worked in the past on a specific piece of documentation, the author is logged and can easily be found when more information on that piece is required

Documentation	**have either party take definitive leadership and set a documentation standard, which includes both a template for document structure as well as how requirements are to be described**
	include a summary of the requirements
	use diagrams to accompany the requirement text
	a written out vocabulary of terms
	tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE
Elicitation	**constant pressure from high level management on the client side is required to ensure the full participation of all stakeholders**
	video conferencing to contact stakeholders
	occasional visits by representatives of the offshore team
	a tool that automatically collects non-functional or quality requirements from existing documentation
	a tool that semi automatically categorizes requirements on their priority and complexity, as well as discerning functional and non-functional requirements
	a tool that evaluates the user story requirements artefact, and points out low quality areas in the user story to prevent defects later on in the development process
Negotiation	**having synchronous discussions instead of asynchronous discussions (by telephone instead of email)**
	prioritization of requirements
	making it someone's responsibility to make sure requirements prioritization takes place
	explicitly developing stakeholder viewpoints, including stakeholder satisfaction as a successfactor and involving stakeholders in the creation of a common vision
	share requirements information with all stakeholders, before negotiation commences
	stakeholder interaction between those of the same level in hierarchies between companies in an offshore outsourcing setting
	face-to-face meetings, requiring one or more stakeholders to be flown over to the negotiation site
	synchronous meetings at a distance aided by rich media such as video conferencing
	a human facilitator during negotiation meetings
	a method of asynchronous communication whereby stakeholders used an argumentation scheme tool to collect and tackle issues. In stages comments were then given on each of the issues by all stakeholders

Validation	**a common vocabulary included in the requirements engineering artefacts**
	a Natural Language Processing method of analysing a requirements specification through decomposing the document into single, simple sentences. These sentences are then analysed and categorised as type of requirements and their priority
	tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE
Managing Artefacts	**use of a centralized requirements depository or requirements management system**
	use of hypertexting in documentation to create a structure and visible relations between artefacts and their precursor documents
	a version control system which can be used to track which project member contributed
	configuration management systems which are shared between all teams working on aspects of software development
	remote sharing of screens
Observation of System Context	**accountability on requirements**
	keeping track of the original goals throughout the project
	audio conferencing
	shared calendars between project members and stakeholders
	instant chat solution for all project members and stakeholders
	presence awareness features (showing team members coming online)
	a tool that allowed discussions between random team members and logged the results, allowing all stakeholders to look back on what had been discussed
	a tool that allows stakeholders to subscribe to requirements, which alerts them whenever a change to the specific requirement was made
	a policy for processing changes to requirements
Managing Activities	**a well defined and relatable software development process shared by all project members**
	use of agile methodologies (such as Scrum)
	adjust the expected timelines / planning for outsourced work (it generally takes longer)
	reducing the amount of project members in a project
	having champions of interaction, who know all stakeholders and can act like communication relays for the project

	decoupling of work across locations
	a project structure to relate the dependencies of artefacts and activities
Artefacts	paying attention to function interdependency can steer the software architecture in such a direction that minimal interaction between teams is enabled, reducing development time and possibility for communication errors
	setting up requirements for interaction between sites or responsible parties as it is known which dependencies exist or will arise in the future

Offshore insourcing

Requirements engineering Element	Adaptation Techniques
System Context	**a formal training programme for GSE and company contextual knowledge**
	a tool based on work history. When an expert has worked in the past on a specific piece of documentation, the author is logged and can easily be found when more information on that piece is required
	instant chat solution for direct communication between project team members
Documentation	**all sites use the same documentation standard**
	tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE
	include a summary of the requirements
	use diagrams to accompany the requirement text
	a written out vocabulary of terms
Elicitation	**constant pressure from high level management on the client side is required to ensure the full participation of all stakeholders**
	video conferencing to contact stakeholders
	occasional visits by representatives of the offshore team
	a tool that automatically collects non-functional or quality requirements from existing documentation
	a tool that semi automatically categorizes requirements on their priority and complexity, as well as discerning functional and non-functional requirements
	a tool that evaluates the user story requirements artefact, and points out low quality areas in the user story to prevent defects later on in the development process
Negotiation	**having synchronous discussions instead of asynchronous discussions (by telephone instead of email)**
	prioritization of requirements
	share requirements information with all stakeholders, before negotiation commences

	face-to-face meetings, requiring one or more stakeholders to be flown over to the negotiation site
	synchronous meetings at a distance aided by rich media such as video conferencing
	a human facilitator during negotiation meetings
	a method of asynchronous communication whereby stakeholders used an argumentation scheme tool to collect and tackle issues. In stages comments were then given on each of the issues by all stakeholders
Validation	**tailored template for requirements documentation to suit GSE specifically. The template takes into account elements like location, time zones and specific needs of different places, which creates awareness of the GSE**
	a common vocabulary included in the requirements engineering artefacts
	a Natural Language Processing method of analysing a requirements specification through decomposing the document into single, simple sentences. These sentences are then analysed and categorised as type of requirements and their priority
Managing Artefacts	**use of a centralized requirements depository or requirements management system**
	use of hypertexting in documentation to create a structure and visible relations between artefacts and their precursor documents
	remote sharing of screens
	a version control system which can be used to track which project member contributed
Observation of System Context	**audio conferencing**
	shared calendars between project members and stakeholders
	presence awareness features (showing team members coming online)
	a tool that allowed discussions between random team members and logged the results, allowing all stakeholders to look back on what had been discussed
	a tool that allows stakeholders to subscribe to requirements, which alerts them whenever a change to the specific requirement was made
	instant chat solution for all project members and stakeholders
Managing Activities	**use of agile methodologies (such as Scrum)**
	adjust the expected timelines / planning for outsourced work (it generally takes longer)
	having direct lines of communication from business stakeholders and offshore project members
	decoupling of work across locations
	reducing the amount of project members in a project
	having champions of interaction, who know all stakeholders and can act like

	communication relays for the project
	building relationships between onshore and offshore team members
Artefacts	paying attention to function interdependency can steer the software architecture in such a direction that minimal interaction between teams is enabled, reducing development time and possibility for communication errors

Onshore outsourcing

Requirements engineering Element	Adaptation Techniques
System Context	**putting outsourced teams together on the same location as the involved business stakeholders, less than 30 meters apart**
Documentation	**all sites use the same documentation standard**
Negotiation	**prioritization of requirements**
	face to face negotiation meetings
	synchronous meetings at a distance aided by rich media such as video conferencing
	make stakeholders aware of relative effort required for specific requirements
	stakeholder Win-Win approach or the MBASE Approach: such approaches reconcile Win conditions for the client (resulting system capabilities) with Win conditions for the developer (minimal risk to going beyond schedule or budget). When a win condition for the client poses a significant risk to a win condition of the developer, this is stated as an issue that has to be resolved or agreed upon
Observation of System Context	**shared calendars between project members and stakeholders**
	instant chat solution for all project members and stakeholders
	presence awareness features (showing team members coming online)
	collocation of project team members

9.3 Appendix C: Ziggo project structure

For this research, case studies will be done into the situation of Ziggo. As Ziggo has its own definitions and jargon, it is useful to describe the project structure of Ziggo and compare it to the project structure it is modeled after: Prince2. Doing this will establish how comparable the case study of Ziggo is for other companies that use Prince2, as Prince2 is a standard for project management practices.

First in this chapter is the description of the Ziggo project structure, then the differences between these two project structures are highlighted and their impact analysed.

What is a project, as defined by Ziggo?

A project is aimed at a change that is executed by a temporary project organisation. Projects are performed within set lines of results, time, funding and quality. A project is defined as such when:

- The project knows a pre-defined scope
- The project delivers a unique result
- A project knows a clear start and end date
- A temporary project organisation is required to realise the unique result

Projects within Ziggo use one clear and predefined project structure, to improve company wide comprehension and communication as well as efficiency. Projects have phases to control uncertain elements, improve decision making and to ensure projects are still working to the intended result. Below you can see a chart of the project phases.

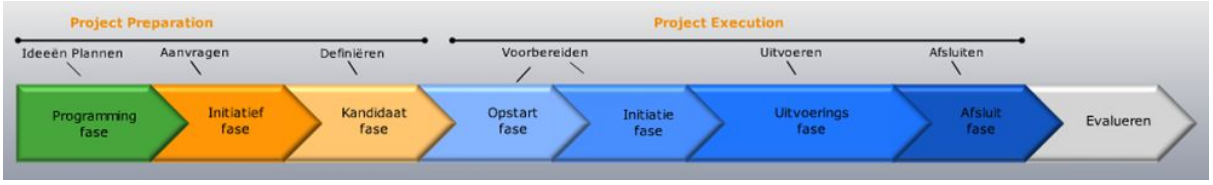

Figure 15. The Ziggo Project Structure.

The phases of the ziggo project structure are explained in the following section. To increase the readability of this section, core concepts are color coded.
Red = Essential concepts
Orange = Organisational bodies / roles
Green = Deliverables that Phase

Programming phase

In this phase the ambitions of the business, knowledge domains, operations, strategic vision and improvements of the business are collected into a integral roadmap. In other words, the business side of Ziggo determines where it wants to go and plans this out into a roadmap. Each goal / project on that roadmap is a roadmap item, which as a document is a description of a proposed project in terms of Delivery Track, project type, organisational needs, client, goal, scope, timescale and budget. For ideas that should be apart of the roadmap, but are not concrete enough to form a project a "Verkenning" (Reconnaissance) can be made. This document includes options on how a goal can be reached but without committing to a specific solution.

Each of these roadmap items are defined further into the Portfolio Plan. A Portfolio Plan is a collection of to-be started and running projects. A portfolio plan serves its purpose from an organisational strategic point of view, or how roadmap items add strategically to the organisation. These plans are also used to prioritize between roadmap items.

These items will be planned as much as possible, but if urgent items come up they need to be decided upon by the Program Board.

Initiative Phase

A roadmap item as described in the programming phase becomes an initiative (which changes the status of an item from "idea" to something that will be done) by submission of a Solution Request (SR). A Solution Request contains a description of the desires and needs the initiative takers (client) and the background and serves as the decision making point to give project a go. A Solution Request will be submitted by a delegate of the client (Person from the business side and not a information analyst or project manager).

In this phase it will be decided who becomes responsible for the budget and project (delivery responsible party). Beside this, a project start and delivery date will be set as well as an indication of project risks, dependencies and budget.

Candidate Phase
An initiative becomes a candidate when a Solution Request has been approved by the Program Board or, depending on whether it is on the Roadmap it will be approved by the programming managers of Program Office Planning (Which manages the integral Roadmap). A Solution request might first prompt a Feasibility Assessment, which further details scope, feasibility, project dependencies and risks and contains a Impact Analysis. Another option is a Pre-Project, which details the execution of the Initiative and the Candidate phases, and any other activities required to prepare for a project.

After the approval of the Solution Request the project is elaborated into an Assignment, which is the agreement between client and contractor which gives the main directions on the project with an estimation on solution, Business Case (justification) and return on investment.

Start-Up Phase
If no Project Manager was assigned to the project thus far this will occur in this phase. The goal of this phase is to get the Assignment as detailed in the previous phase approved. To do this the specifications of the project and the solution direction will be elaborated upon.

Bigger projects end this phase with a Project Brief (PB), which is a document that explores some in the Assignment with more depth, such as the Business Requirements, solution direction, expected result, actualised Business Case, project definition and planning. The goal of the Project Brief is to accurately portray the feasibility and costs of the project to make a final decision on project launch.

Initiation Phase
In this phase all doubts concerning the project execution are removed. Project preparation details are documented in the Committed Project Plan (CPP), which explains the project approach. The Committed Project Plan contains a set of goals and end products, a further elaborated and validated Business Case, project scope and planning as well as steps and (sub)activities. The Plan will be used as a contract between client and contractor and can be used to measure project progress. Furthermore the Committed Project Plan will answer the questions:

- What do we want to achieve?
- Why is this important?
- Who are involved and what are their tasks, responsibilities and their prerogatives?

During this phase there will be periodical project status reports called Highlight Reports. The purpose of Highlight Reports are to give the Program Board insights in the project status, progress and risks, and minimally contain progress regarding time and budget, quality and scope as well as risks and their mitigation approaches.

Execution Phase
In this phase the Project Manager controls the project within set bounds and delivery specifications. Highlight Reports are continued to be produced periodically.

If circumstances arise that make the project go outside of the Committed Project Plan (regarding time, budget, scope or quality) the Project Manager will have to file an Exception Report and reach an agreement upon this with the Program Board.

A Project Manager can delegate responsibility for part of the project to subprojects. When the Execution Phase comes to an end the project result has to be delivered. This delivery happens formally according to the Protocol of Internal Delivery. The Protocol of Internal Delivery contains the information that the client needs to decide acceptance of the project result. If an agreement is reached, the delivered result is handed over to operations.

Closing Phase
In this phase the different parties are held accountable for their part of the project. A Project End Report (PER) is made to inform the Program Board of the project proceedings. The Project End Report also has a "lessons learned" section to benefit future projects. The project is reviewed on differences between the Committed Project Plan and the actual events and resources spend on the project.

9.4 Appendix D: Ziggo project structure compared to Prince2

In this section the Ziggo project structure as described above is compared to the standard project structure known as prince2. The Prince2 project structure is well defined and tried, information on this project structure was derived from the Prince2 training manual. This comparison is visually explained by *Figure 15* and *Figure 16* showing the Ziggo Project structure and the Prince2 method structure respectively.

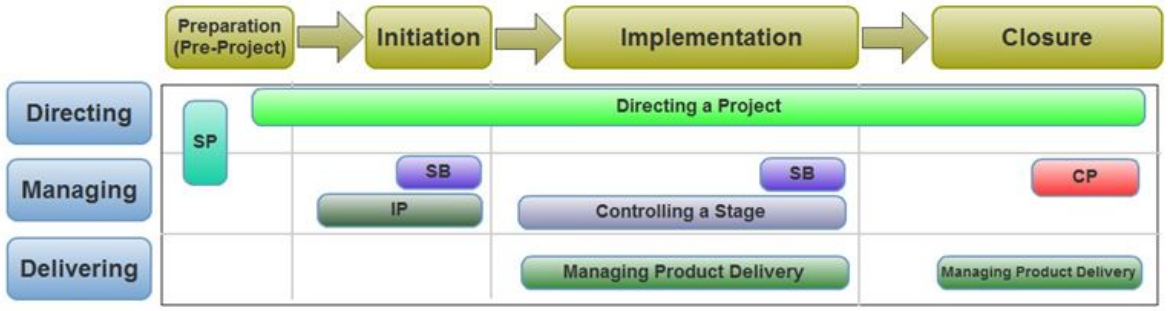

Figure 16. The Prince2 method structure.

To create a mapping between these project structures, all Ziggo project phases are examined and how their counterparts work within Prince2. The phases are examined on roles, documents and main activities performed. Afterwards, the main differences are highlighted and a combined visual graph is made to discern where what phases are situated.

Roughly put, during the programming phase the business side of Ziggo determines where it wants to go and plans this out into a roadmap. The programming phase is formally not part of the prince2 project structure, as it is a more long term strategizing phase. However, most ideas for future projects (should) emerge from this phase and is thus included in the pre-project phase as Prince2 defines it.

During the Initiative Phase the initial start of a regular Prince2 project commences, where this starts up with Prince2 with a Mandate, which is a document containing an outline of the Business Case and will explain the reason why the project is needed. This document serves as the trigger for a project. Within Ziggo the mandate role is filled by the Solution Request document, which is a relatively low fidelity mandate. This document serves as the first decision making point for the project and has to be drafted by a delegate of the business, to get their commitment to the project on board. During this phase a rough planning is made to estimate project start and end dates.

When a Solution Request has been approved by the Program Board (CxO level) or if it has been a planned roadmap item by Program Office Planning the Candidate Phase starts. During this phase more requests or research for information regarding boundary factors of a project may be prompted, such as further planning of the coming (pre project execution) phases, and impact analyses as well as project dependencies and risks. In Prince2 this can be equaled to the Starting Up a Project process, which has as main objective to verify that the project is worthwhile. When these are performed satisfactorily, the Assignment document is drafted. The Assignment is an expanded Solution Request, or a fully developed Project Mandate as seen in Prince2. The Assignment serves as an agreement between (business) client and contractor, and contains a business case, business level requirements and an estimation on solutions.

During the Start-Up phase a project manager will be assigned if not done so before, and will be responsible for the rest of the project. The project manager will draft a Project Brief, which is a document that explores the Assignment with more depth, such as the Business Requirements, solution direction, expected result, actualised Business Case, project definition and planning. The goal of the Project Brief is to accurately portray the feasibility and costs of the project to make a final decision on project launch. With Prince2, the goal of the Project Brief is similar, however, it is slightly differently placed. The Project Brief in Prince2 is the end of the Pre-project stage, but within Ziggo the Start-Up phase, including the Project Brief, are portrayed as part of the Initiation Stage (and part of the actual project execution). Regardless of position, the Project Brief contains a set of requirements on the functional / non-functional (quality) requirements level.

During the Initiation Phase the project plan is finalized and predictions on planning, scope and budget should be close to a hundred percent accurate. All these variables are documented in the Committed Project Plan. The Committed Project Plan explains the project solution. The Committed Project Plan contains a set of goals and end products, a further elaborated and validated Business Case, project scope and planning as well as steps and (sub)activities. The Plan will be used as a contract between client and contractor and can be used to measure project progress. Furthermore the Committed Project Plan will answer the questions:

- What do we want to achieve?
- Why is this important?
- Who are involved and what are their tasks, responsibilities and their prerogatives?

The Committed Project Plan has a Prince2 counterpart which is the Project Initiation Document, and is shortly described in the Prince2 Training Manual as:

> *“A set of documents that contain essential information to start the project; in other words the documents that were created during the Initiation Stage that describes how the project will be done in detail.”*

From the Initiation Phase and onwards reports on the progress of the project are delivered through Highlight Reports, which give insights on status, budget, schedule, quality, risks and their mitigation strategies.

The Execution Phase of Ziggo and the Prince2 Execution Stage are very similar in concepts, and both are performable with any project method (waterfall and agile for example). During this phase, the actual project is executed, possibly in different phases according to the needs of the project. Leading on daily decisions is the project manager, who regularly communicates progress and handles exceptions with the project board. Prince2 officially dictates some processes to occur during this phase, such as Controlling a Stage and Managing Stage Boundaries, which define and close different stages. These processes are less formally executed within Ziggo, but most elements occur in every project, such as reviewing status of a stage, examine and mitigate risks and taking corrective action and correcting planning and business case where necessary.

In the Closing Phase the different parties are held accountable for their part of the project. A Project End Report is made to inform the Program Board of the project proceedings. The Project End Report also has a “lessons learned” section to benefit future projects. The project is reviewed on differences between the Committed Project Plan and the actual events and resources spend on the project. With Prince2 this takes shape as the Closing Stage and the Closing a Project process, which delivers a End Project Report with similar metrics as a Project End Report. This report also creates a plan for measuring the actual reaped benefits of the project.

The Evaluation Phase is only present in the Ziggo project structure, and not in the Prince2 structure. Essentially, during this phase the calculated benefits are measured according to plan as drafted in the Closing Phase in the Project End Report. This process is executed by those operating the products or programs that the project delivered.

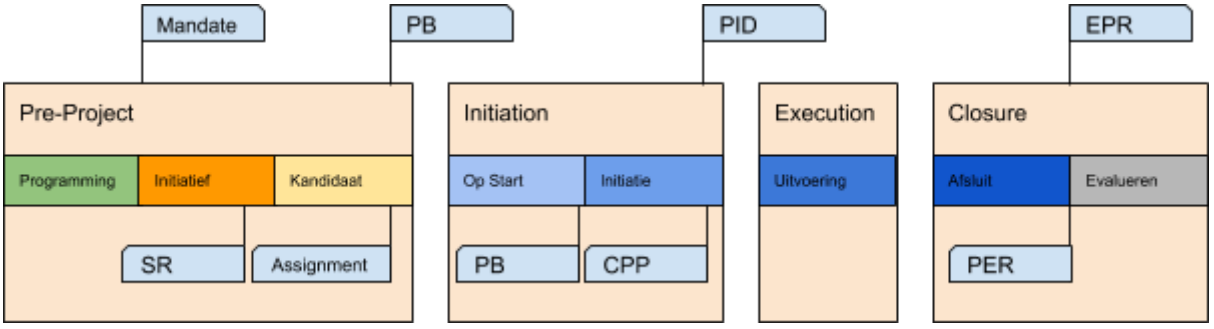

Figure 17. The Ziggo phases mapped to the Prince2.

The overall comparison leaves us with the diagram in *Figure 17*. In the diagram the Ziggo phases can be seen as the colored band in the middle. These phases are encapsulated by the Prince2 phases. Attached below to the Ziggo phases are their associated main documents. Their Prince2 counterparts are labeled above the Prince2 phases.

Main Differences

The first project phase, the Programming Phase, does not officially have a place in the Prince2 project structure, but it concerns activities that are the source for projects, as thus they can be defined as “Pre-Project” as Prince2 describes the Pre-Project stage.

The biggest difference between the two project structures can be found in the documentation between the closure of the Pre-Project Stage and the Initiation Stage. Ziggo has an extra document fidelity layer. Where, during the Initiation Stage, there is only one final stage document, the Project Initiation Document (PID), Ziggo has two, namely the Project Brief (which closes the Pre-Project Phase in Prince2) and the Committed Project Plan. The Ziggo Candidate Phase is closed by the Assignment document, which is a high fidelity Mandate.

The contents of the evaluation phase in the Ziggo project structure are not present in the Prince2 method, but are however hinted to in the plans drafted in the End Project Report (EPR). Therefore this phase is tuned in with the closing stage of the Prince2 project structure.