Outsourcing decisions in the Dutch software industry.

A multiple-case study

Abstract

Outsourcing decisions regarding the development of software has become a major topic for software developing companies in the past decade. Through ongoing globalization it has now become a major topic for small- and medium enterprises (SME's). Current literature mainly views outsourcing decisions regarding software development from an organizational perspective and has been focussing on large enterprises. Views from SME's and from an operational perspective are neglected. Through a multiple-case study this research provides insights in the outsourcing behaviour of SME's in the Dutch software industry. This research seeks to explore the decision making of SME's on both organizational and operational level to provide practical implications for SME's.

Author:	*Jisse Plaggenborg*
Student number:	*S1623788*
1st Supervisor:	*Dr. G. Blaauw*
2nd Supervisor:	*Dr. R. Loohuis*
Company supervisor:	*Mr. C. Thönissen*

31 July 2016

UNIVERSITY OF TWENTE.

Colophon

Date:	29 / 07 / 2016
Place:	Enschede
Academic year:	2015/2016
Version:	1.0
Project reference:	Master Thesis

Author:	Jisse Lars Plaggenborg
Student number:	S1623788
Email:	J.L.Plaggenborg@student.utwente.nl
Education:	Business Administration
Specialization:	Marketing & Strategy
Institute:	University of Twente
Faculty:	Behavioural, management and social sciences

Examination committee

Dr. G. Blaauw
Dr. R. Loohuis

Supervisor Heutink ICT

Mr. C. Thönissen

Preface

This thesis is written to finalize the Master of Business Administration programme at the University of Twente. Through this preface I would like to reflect on my period as a student and to thank everyone that contributed to this research.

Starting as a business administration student in 2011 at Saxion University of Applied Sciences, I knew that I had a commercial drive. However, I did not know what I wanted to become. After doing an internship at Raffles Media in London I knew that I wanted to pursue a Master's degree to gain a deeper understanding of marketing and strategy before starting my career. It was in London that I found out that I wanted to work in an international environment where I can fully make use of the capabilities that I possess. Through experience in several part time jobs I realized that I find it satisfying to solve peoples' problems through selling them value-adding solutions. I am happy that I was able to develop myself both at university and at the same time at my employer, the Odin-Groep.

I would like to take time to thank everyone that supported me throughout my research. In particular, I would like to thank dr. G. Blaauw for guiding me and providing me new insights throughout the research project. Furthermore, I would like to thank dr. R. Loohuis for his help and critical input which greatly improved the research process. Next, I would like to thank Coen Thönissen, operational director of Heutink ICT, for all time and effort invested in this research project. In addition, I would like to thank everyone at Heutink ICT. I really enjoyed my time in the office and of course all the games of poker during the breaks. Finally, I would like to thank my family and girlfriend for supporting me throughout my studies.

Jisse Plaggenborg
Hengelo, July 2016

Table of contents

1. Introduction

This chapter starts with an introduction to make-or-buy decisions and an academic call for research. Next, an introduction of Heutink ICT and their call for research will be discussed. Finally, the main research problem and sub-questions are presented.

1.1 Make-or-buy decisions

Make-or-buy decisions have been widely discussed by researchers in the last decade. Through globalization outsourcing is becoming so sophisticated that even core functions can be moved outside the company. This changes the way firms think about their organizations, value chains and competitive positions. "It's no longer ownership of capabilities that matters but rather a company's ability to control and make the most of critical capabilities, whether or not they reside on a company's balance sheet." (Gottfredson, Puryear, & Phillips, 2005). Gottfredson, Puryear and Philips (2005) argue that it is no longer a question whether to outsource a capability or activity but rather how to in- or outsource every single activity in the value chain.

There are several reasons mentioned in the literature why firms outsource their activities. Isaksson & Lantz (2015) present the two most discussed reasons: achieving cost benefits and/or focusing on core competencies. Other reasons are achieving best practices by acquiring external capabilities/activities (Kakabadse & Kakabadse, 2002), to transform fixed costs into variable costs (Alexander & Young, 1996) or as a tool to rapidly adapt to changing environments (Leavy, 2004). Many organizations outsource some of their activities. However, a significant proportion of these activities fail to fulfil the expectation that the organizations have of them (Bhattacharya, Singh, & Bhakoo, 2013). Following Bhattacharya, Singh & Bhakoo (2013), most failures are caused by communication gaps, opportunistic behaviour, a lack of control and/or language barriers.

Through globalization, Information Systems (IS) assets can be provided and performed anywhere in the world and at any time. Using IS and IT (Information Technology) firms can manage their distant relationships as it allows them to communicate independently of where they are located (Shao & David, 2007). IS assets become more and more important for daily business activities and firm performance, therefore the Information System Development (ISD) process needs to be effective and efficient (Avison & Fitzgerald 2006). "Information System Development is a process that involves the analysis, design, technical implementation, organisational implementation and subsequent evolution of Information Systems" (Livari & Hirschheim, 1996). Bergkvist & Fredriksson (2008) define ISD as: "A relationship where the client contracts or sells ISD assets, people and/or activities to an IT-supplier. The IT-supplier manages these assets and provides services for monetary returns over an agreed time period" (Bergkvist & Fredriksson, 2008).

Not only companies in need of software are looking to benefit from outsourcing. Even software developing companies want to profit from lower developing costs and faster release cycles (Carmel & Argarwal, 2001). During the years many large software developing companies established offshore developing centres in low-wage countries. In order to achieve economies of scale and scope they started to offer their services to third parties. In the first stages, outsourcing a company's software development could be described as externalizing the whole IT department to external vendors (King & Torkzadeh, 2008). Later on, selective outsourcing was introduced, which could be described as outsourcing only certain activities of software development. Outsourcing software development has now become an option for a large number of software developing companies (Kramer, Klimpke & Heinzl, 2013).

In the current literature, make-or-buy decisions regarding Information Systems are widely discussed. However, hardly any study focuses on outsourcing at the software component level. Kramer, Heinzl and Spohrer (2011) started doing research in this area. They proposed a decision making heuristic for software components based on a single case study. Their research addresses relevant information regarding decision making and combines software engineering and outsourcing decisions in one study. Later on Kramer, Klimpke and Heinzl (2013) explored sourcing decisions of behaviour of SME's (Small- and Medium

Enterprises) in the software industry in Germany. They are calling for more research in this area. More specifically, they call for research in the field of component-based outsourcing decisions, especially involving SMEs from the software industry. Therefore, this research aims to contribute to this field of research through exploring how outsourcing decisions at the software component level are made. The purpose of this study is to find out what software components are considered appropriate to be outsourced and to find out why these are considered appropriate according to software developing companies. Consequently, the following research question is formulated:

Research question

"How and on what basis are in-or outsourcing decisions regarding software components made in SME's?"

In order to be able to answer the main research question, several sub-questions have been formulated. As this research is of exploratory kind, the following sub-questions allow us to further explore the topic of outsourcing decisions of software developing SME's in the Netherlands.

Sub-questions

- Who is responsible for making the outsourcing decision regarding software components?
- When are in-or outsourcing decisions regarding software components made?
- Which characteristics are determinants in making an in- or outsourcing decision regarding software components?
- What factors influence the in-or outsourcing decisions regarding software components?

Kramer, Heinzl & Spohrer (2011) derived component characteristics from three different theories. In our study we combine both the resource-based view and transaction cost economics into an economic perspective while the systems theory forms a technical perspective. This study focuses on the combination of these perspectives as they seem to be complementary to each other regarding the outsourcing decision of software components. Both perspectives will be further elaborated in the literature review.

The call for research by Kramer, Klimpke and Heinzl (2013) specifically asked for research in SME's in the software industry. Therefore, eight small- and medium software vendors in the Netherlands that are already outsourcing (parts of) their software developing activities will be approached in order to participate in this research. The aim of this qualitative multiple case study is to get insights in how decisions are made regarding the in-or outsourcing of software components within the software industry. As there is hardly any research conducted at the component level this research will be of exploratory kind. Semi-structured interviews will give insight into how in-or outsourcing decisions unfold and what influences these decisions whereas a questionnaire lacks depth in order to fully understand what characteristics and factors have influence. Therefore, semi-structured interviews will be conducted. A cross-case analysis will be conducted to complement the findings of this research.

This study seeks to contribute to the existing literature in a number of ways. First, it contributes by adding more empirical knowledge in the field of outsourcing decisions at the software component level. Next, it complements the existing literature by identifying characteristics and factors that influence the decision making of in-or outsourcing decisions. Furthermore, it extends the current literature by researching in another western country. Afterwards a cross-country comparison study can be conducted to find out differences and/or similarities across countries. Finally, this research provides practical information for software vendors in order to improve their decision making regarding the in-or outsourcing decisions of software components.

1.2 Heutink ICT

Heutink ICT is total supplier of ICT products in the primary education sector. ICT offers many possibilities for personal education. With smart applications every student can learn at their own level and at their own pace. Heutink ICT offers solutions and services for primary educational institutions to efficiently use ICT that optimally fits their vision (Heutink-ICT, 2015). Heutink ICT is one of the four companies within the Odin-Group. The Odin-Group exists of Heutink ICT, Lesscher IT, Previder and Web2work and they all have their own

expertise. Appendix 1 shows an organogram of the organization. Through narrow collaboration these companies share their knowledge and expertise within the organization (Odin-Groep, 2015).

Recently Heutink ICT has started developing and selling software products. A couple of software developers of Web2work have been taken over in order to develop software for Heutink ICT. Therefore, Heutink ICT cannot yet be seen as a real software developing company. Heutink ICT has difficulties in assessing and steering its software developing activities. In a functional perspective they are making what was meant to be made. However, are they working with the right techniques? Are they working in a good way (efficient, effective, productive etc.)? Heutink's operational director places severe question marks at which software developing activities they could (or maybe should) best do themselves and which activities they should (or could) better outsource to other organizations? As example the operational director mentioned BMW: "Can BMW successfully sell the same 'good' cars without having the knowledge to design or build such a car? Do they have to know how an engine works? Should they be able to assemble the car themselves or are they more successful when they outsource these activities?"

Heutink ICT has the ambition to sell and deliver products of high quality. However, in order to be successful, the question arises whether they should outsource their current software developing activities or should they not? They want to know which strategy to use in order to successfully develop and sell software in the future in the primary education sector. Heutink ICT calls for research, however, this research aims at general applicability and thus not necessarily at Heutink ICT. This allows us to explore the decision making regarding make-or-buy decisions in several different software developing companies.

2 Literature review

A lot of research has been conducted regarding make-or-buy decisions and even software outsourcing has its own research stream (ISD). The current literature regarding software outsourcing has only discussed at the organizational and operational level and a handful of studies aim at strategy, risk, success factors and capabilities (Lacity, Khan, & Willcocks, 2009). This research aims at both the economic and technical perspectives, therefore this chapter starts with an introduction to outsourcing. Furthermore, light will be shed on the economic perspective of outsourcing in terms of the resource-based view and transaction economics. Moreover, software development, its working methods and the system theory will be discussed.

2.1 Outsourcing

The discussion of outsourcing software development relates to the question what to make and what to buy. Wherein making can be described as the internal development of software objects (software artifacts, components, packages etc.) and buying can be described as externalizing the development of software objects (Kramer, Heinzl & Spohrer, 2011). To be more specific, this paper focuses on the development of software applications and does not differentiate between outsourcing to vendors in the same country or to vendors elsewhere in the world.

According to the literature reviews of Dibbern et al. (2004) and Lacity et al. (2009) there are five major issue areas in the IS outsourcing debate namely: *why* to outsource, *what* to outsource, *which* decision process to take and *how* to implement the sourcing decision. Several theories have been proposed and applied to the IS sourcing phenomena. Nevertheless, when focusing on *what* to outsource only a handful of theories is proposed. In the following paragraphs the resource-based view (RBV) and transaction cost economics (TCE) will be discussed as they are the two main theories to explain the economic perspective in relation to outsourcing. Next to these theories the Systems Theory (ST) will be discussed as it makes use of the modularity of software elements which has been proven to be helpful for the development of complex systems such as software systems (Picot & Baumann, 2007) (cf. Kramer, Heinzl, & Spohrer, 2011).

2.2 Economic perspective

There are two main theories that influence the make-or-buy decision namely: Resource-based view and transaction cost economics. The resource-based view is based on a firms resources, capabilities and the way they apply these capabilities and resources in order to gain competitive advantage (Barney, 1991). The transaction cost economics (henceforth TCE) is based on cost saving initiatives (Williamson O. E., 1981).

According to the resource-based view the way a firm is able to apply its capabilities and resources results into competitive advantage. If a firm is not able to provide the required resources in order to develop a product or service internally, resources may be acquired externally to complement the firm's assets (Tallman & Fladmoe-Lindquist, 2002). However, resources with high asset specificity (high strategic significance + highly heterogeneous distribution) should be provided by the firm itself in order to protect its core competencies (Kramer, Heinzl, & Spohrer, 2011).

The TCE perspective builds on a cost savings motive to decide whether to make-or-buy an activity. It examines whether it is economically beneficial to externalize certain activities or if it is better to perform them in-house. Organizations examine three characteristics of the transaction in order to decide whether to make-or-buy a certain activity namely: Asset specificity, the frequency of transaction and the degree and type of uncertainty to which they are subject to (Williamson O. , 1989). Starting with the first characteristic, asset specificity relates to the degree to which an asset can be re-used in a different setting by other users than it initially is meant for, without the loss of productive value. Furthermore, the frequency of transaction determines the efficiency of the transaction. At some point the critical production volume has been reached, indicating that as of this point it might be economically beneficial to insource the activity. Finally, the uncertainty of the transaction relates to the length of the contract between the buyer and the supplier. Long-term contracts are hard to realize as over-time contractual misalignments could appear and firms could take advantage of this phenomenon (Whyte, 1994).

In software development it is assumed that human asset specificity plays the most important role in outsourcing contexts as the knowledge to develop complex systems is owned by the employees of software companies. Companies intention to outsource certain software components is mostly related to high transaction costs as components with complex development tasks have significantly higher transaction costs than components with simple development tasks (Williamson O. , 1989) (Dibbern et al., 2004). This could lead to higher costs as managing the complexity of tasks and processes exceeds the expectation of cost savings through outsourcing (Dibbern, Winkler, & Heinzl, 2008).

Kramer, Heinzl & Spohrer (2011) divide human asset specificity into three separate specificities for software objects namely: business related specificity, functional specificity and technical specificity. First, business related specificity relates to the required knowledge of operational and management processes that are supported by the software product. Second, functional complexity is related to the knowledge needed by software companies in order to translate business processes into a logical software design or the knowledge needed by external developers to understand the software code. Finally, technical specificity is related to the skills and techniques required to implement the desired software system. These specificities form the contextual component properties of a software component and serve as input for an outsourcing decision.

Based on both TCE and RBV perspectives the existing literature gives advice regarding outsourcing decisions by suggesting firms to outsource non-core functions and keep highly complex functions in-house. According to Stratman (2008) the best candidates for outsourcing are well understood, standardized service processes, that are not core capabilities and do not require direct customer contact. In contrary, core functions are complex tasks that are less defined, have no standardized procedure, and therefore includes a deep understanding of software development. When outsourcing core functions this could endanger the success of the project as the tasks are more complex and challenging. This leads to an increase of additional transaction and production costs (Kramer, Heinzl, & Spohrer, 2011).

2.3 Technical perspective

Next to the economic perspective, the technical perspective also seems to play an important role in the outsourcing decision. The combination of factors from an economic perspective and a technical perspective form the base for the outsourcing decision. Equal to the economic perspective, the technical perspective is also based on a theory discussed in the literature namely the system theory. In order to understand how the system theory affects software engineering, the software engineering discipline will be explained in terms of what it is, what phases it consists of and what working methods are used.

2.3.1 Software engineering

"Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use" (Sommerville, 2011). There are five different phases in software engineering namely: Requirements engineering, design, software development, testing and the deployment & maintenance. In the first phase (requirements engineering phase) it is set out what the software system should do and to define constraints on its operation and implication. This happens independent of how these requirements are going to be accomplished as several methods exist in order to develop a software system. The most popular methods will be discussed in a following paragraph. In the second phase the architecture of the system is designed. The software architecture shows how a software system is organized. The way the software architecture is designed influences properties such as performance, security and availability of the software system. This results in a document that is used as guideline in the developing phase. In the third phase the software is being developed. The components are build using the architecture design from the design phase and the requirements document from the requirements engineering phase. In the fourth phase (testing phase) the software system is being tested in order to find the presence of errors. This is mostly done by a separate team as it is hard to find one's own mistakes. Finally, after testing the software, the software is going to be deployed. After using the software new user demands arise and the software may be adjusted due to these demands (Sommerville, 2011). This study focuses on both the requirements engineering phase and the design phase of software engineering since decisions regarding the working method used and design of components are expected to be made in these phases.

2.3.2 Software engineering process models

In order to understand the latest developments in software engineering this chapter describes software process models and methods that are currently used in software engineering. Software engineering processes can be divided into process models and process methods. A software process model is a simplified version of the software process. It provides developers a framework of the process without showing specific activities (Sommerville, 2011). In the book of Sommerville (2011), three different process models are mentioned: The waterfall process model, incremental development and reuse-oriented software engineering. Each process model has its own advantages and they are not mutually exclusive.

The waterfall model (or traditional model) is a plan-driven process which indicates that before working on processes, the processes must be planned and scheduled. It consists of the five phases mentioned in the previous paragraph. Each new phase should not start before the previous phase has been finished. However, in practice these phases overlap each other as flaws from previous phases are recognized in the next phase. At the end of each phase a document has to be prepared in order to make progress visible to managers. After each iteration these documents have to be adapted, which results in more costs. Commitments must be made early and thus makes the waterfall model less suitable to respond to changing customer demands (Sommerville, 2011).

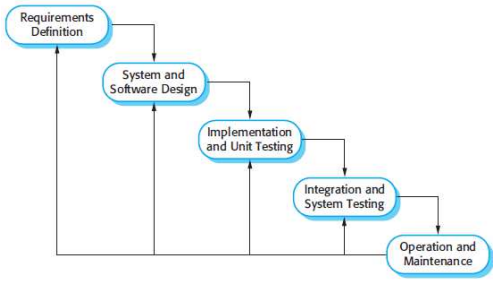


Figure 1: The waterfall model *(Sommerville, 2011)*

Incremental development is a process of developing, asking feedback and then use the feedback to further develop the product through several incremental versions. This model is popular in current business and sets the base for agile developing methods. Within incremental development the specification, development and validation activities are interleaved rather than sequential. It reflects the way we resolve problems. After realizing a mistake has been made, it is possible to track back and resolve the problem. Furthermore, customers can evaluate the intermediate versions at an early stage which enhances the capability of coping with changing customer requirements. Finally, it is possible to deliver the software faster. Even though not all functionalities may be directly available, customers can use the software at earlier stages than is possible with the waterfall model (Sommerville, 2011).

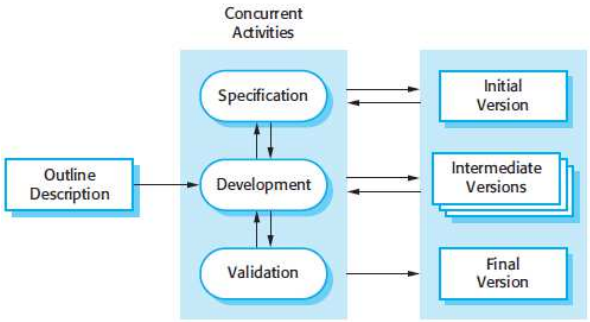


Figure 2: The incremental model *(Sommerville, 2011)*

Reuse-oriented software engineering is the process of finding, and if needed, modifying developed software parts to incorporate them into a new product. It has the advantages that it reduces the amount of software needing to be developed and thus reducing risks and costs and increasing the delivery speed. A limitation is that compromises have to be made and that it might not fully meet the customer requirements. Furthermore, some of the control is lost as new versions of the reusable software are not under control of the organization that uses the reusable software (Sommerville, 2011).

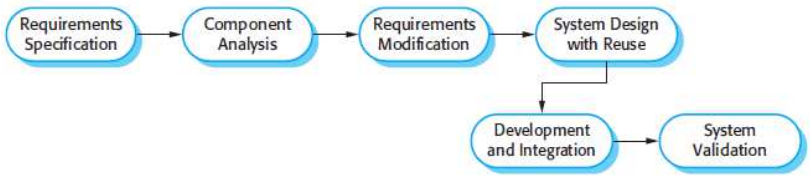


Figure 3: Reuse-oriented software engineering *(Sommerville, 2011)*

In large software development projects these models are often used together. When developing sub-systems of a larger systems, each sub-system can be developed using different approaches. Those parts that are well understood, could be specified and developed using a waterfall-based process. Those parts that are complex and difficult to specify should always be developed using an incremental approach (Sommerville, 2011).

2.3.3. Working methods

The process models set the base for the working methods that software companies use. As each process model has different characteristics, the working methods are also different which in turn might influence the outsourcing decisions of software developing companies. Therefore, it is important to understand which working methods are mostly used within the software engineering domain and how characteristics can be derived from different developing methods used.

Most software development processes follow the Rational Unified Process (RUP). The RUP enhances team productivity and is based on experience of thousands of software engineering projects. It provides a framework to assign and manage tasks and responsibilities in a software engineering project. The Rational Unified Process uses similar stages as the Waterfall model and allows software developers to produce high-quality software that meets the needs of its users within time and budget (Kroll & Kruchten, 2003). It integrates with other tools as it allows to use industry best practices like the spiral model, Unified Modelling Language (UML) and software automation. Therefore, the RUP can be seen as a hybrid model that integrates static models with that dynamic elements of software engineering. It starts with a profound base in the construction phase and it allows incremental development in the other phases (Sommerville, 2011). As a consequence of this hybrid model, UML models and diagrams as well as component based development steps are required, this causes the need for object-oriented programming. Software components are structured into components, packages and classes. This enables the identification of software object characteristics and allows evaluation on component, package or class level (Kroll & Kruchten, 2003). The characteristics of these components could determine the outsourcing potential of a software component.

Another approach is the Agile working method. Business environments change rapidly due to operating in a global context. Often they have to respond to new customer demands or competing products resulting in software that has to be developed quickly. Agile methods concentrate on the goal of the project and the demands of the stakeholders. Documentation and designing models are time consuming and therefore seen as a waste of time. Agile working methods is mostly practiced by small teams where customers are involved in the development process (Sommerville, 2011). There are several Agile working methods, however, Extreme Programming (XP) and Scrum are the best-known approaches. These approaches are flexible and use prioritization of requirements. This allows the most critical developments to be implemented first, mostly the ones that are of great importance to all stakeholders (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003). This could influence the outsourcing decision as the level of prioritization of the software component might determine whether it is eligible for outsourcing.

Finally, in the Open Source Software Development approach components and programs are created on voluntary base by individuals. Each individual is a team member and co-developer. Their task is to develop new features or correct mistakes made by others. Collaboration of the whole community by peer reviewing the development process is highly important for success in Open Source Software Development. Mostly this is controlled by a core group of active developers. As open source software is mostly free to use, most companies focus on selling support on the software rather than the features of the software itself (Sommerville, 2011). Open source software exists of a lot of independent sub-systems, those features have to be integrated within the whole system. The coupling of these interdependent sub-systems could determine whether a component is eligible for outsourcing or not.

Process model	Working methods
Traditional	Waterfall model
	Spiral
	V-model
Incremental	RUP
	Agile methods
Reuse-oriented	Open-source

Figure 4: Overview of working methods per process model used in this paper

2.3.3 Systems Theory

The systems theory perspective makes use of the modularity of software components, which has been proven to be helpful when developing complex systems (Kramer, Heinzl, & Spohrer, 2011). Software systems can be seen as complex engineered systems (CES) as software systems consist of several components and processes with interdependencies during development. The complexity of a system can be defined by the interconnections and/or dependencies in a system architecture (Tripathy & Eppinger, 2011). It is difficult to either study, design or source CES as one system. Therefore, is it suggested to decompose CES into sub-systems or decompose even further into components so that each component becomes a black box that hides the details from other components. Decomposition of CES is necessary in order to identify components that can be designed or outsourced (Tripathy & Eppinger, 2011).

Figure 5 shows the architecture of a system decomposed into subsystems and subsequently into components. The dotted lines show the information flow and the straight line shows the dependencies between components. At the component level coordination needs between the development of components may arise. When one of these components is outsourced it becomes challenging to coordinate the development across different locations. Therefore, it is difficult to completely outsource the development in a single phase. It is suggested to either outsource both components or develop both in-house. Smaller components without significant coordination needs are the easiest to outsource (Tripathy & Eppinger, 2011). This is also confirmed by the research of Mirani (2006)(cf. Kramer, Heinzl, & Spohrer, 2011) who suggests that small components with low complexity, that are well defined and whose development process is highly structured are more likely to be successfully delivered in an outsourcing relationship.

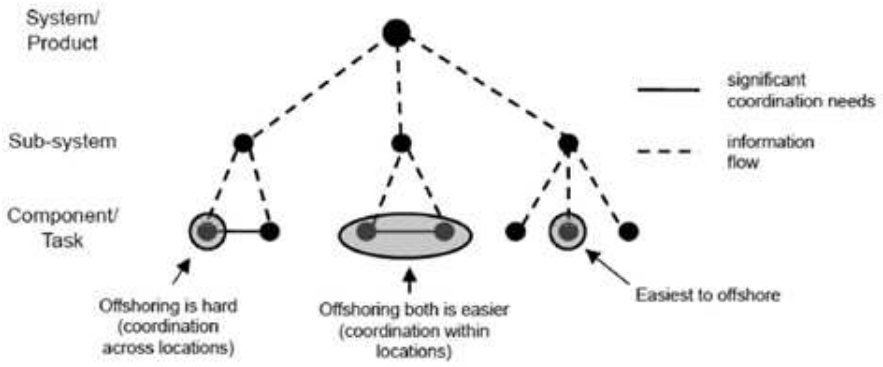


Figure 5: Outsourcing potential (Tripathy & Eppinger, 2011)

According to Kramer, Heinzl, & Spohrer (2011) the systems theory approach requires a systems architecture to be modular in order to be efficient. "Modularity is reached by decomposing a system into numerous subsystems or modules that have a minimal degree of interdependence and that only interact by defined interfaces" (Kramer, Heinzl, & Spohrer, 2011, p.123). The aim of the systems theory is that each sub-system can be developed independently, and therefore can be developed out-house. In the end, all separate sub-systems can integrated in one complete system.

2.4 Summary of the literature review

According to the literature, there are two main input factors for effective resource allocation in software developing companies. The first input factor comes from an economic perspective and is based on both the resource-based view and the transaction cost economics. Based on the economic perspective, factors that determine whether a component is eligible for outsourcing are: business related specificity, functional complexity and technical specificity as explained in paragraph 2.2. Furthermore, the level of strategic significance and the required customer contact are taken into account. Based on the economic perspective it is suggested to outsource the development of those components that are well understood, standardized

service processes, that are not core capabilities and do not require direct customer contact. Core functions are those tasks that are complex and less defined, have no standardized procedure, and therefore includes a deep understanding of software development. In short, the economic factors that determine the outsourcing decision of components are:

- Required knowledge regarding operational and management processes supported by the software product. (Business related specificity)
- Required special knowledge of translating specific business processes into a logic software design or the knowledge needed by external developers to understand the software code (Functional complexity)
- Required programming skills and techniques (Technical specificity)
- Level of strategic significance (Core/non-core)
- Level of required customer contact (High/low)

The technical perspective is based on the systems theory and mostly builds on a modular approach. The technical factors include the coordination needs between the development of components, the size of the component, the level of interdependency of the component, the required customer contact and the priority of development. Based on the technical perspective it is suggested to outsource the development of components that have:

- Low coordination needs
- Small component size
- High level of interdependency
- Low amount of required customer contact
- Low priority of development

In contrary, it is suggested to insource the development of components that have:

- High coordination needs
- Large component size
- Low level of interdependency
- High amount of required customer contact
- High priority of development

2.4.1. Conceptual model

Now that the input factors are determined, a conceptual model can be proposed. Figure 6 displays the abstract model based on the literature. Both the economic and technical perspective seem to influence the effective resource allocation of a company. In turn, effective resource allocation seems to determine whether activities should either be insourced or outsourced.

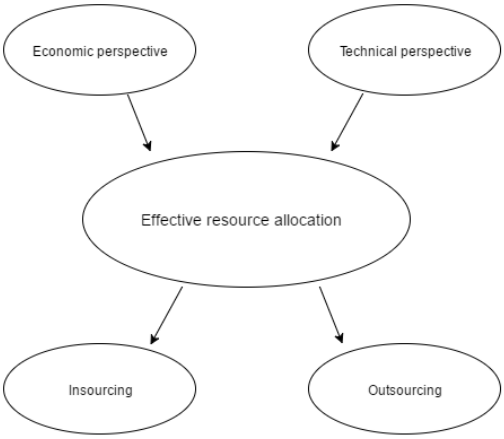


Figure 6: Abstract model based on the literature

As can be concluded from the literature, it is important to keep core activities in-house and outsource those that are well understood, standardized processes that are non-core and require little customer contact. Furthermore, large components with high coordination needs, a low level of interdependency, a high amount of required customer contact and a high priority of development should be kept in-house. Based on this conclusion, the following figure of software development phases and activities is drawn.

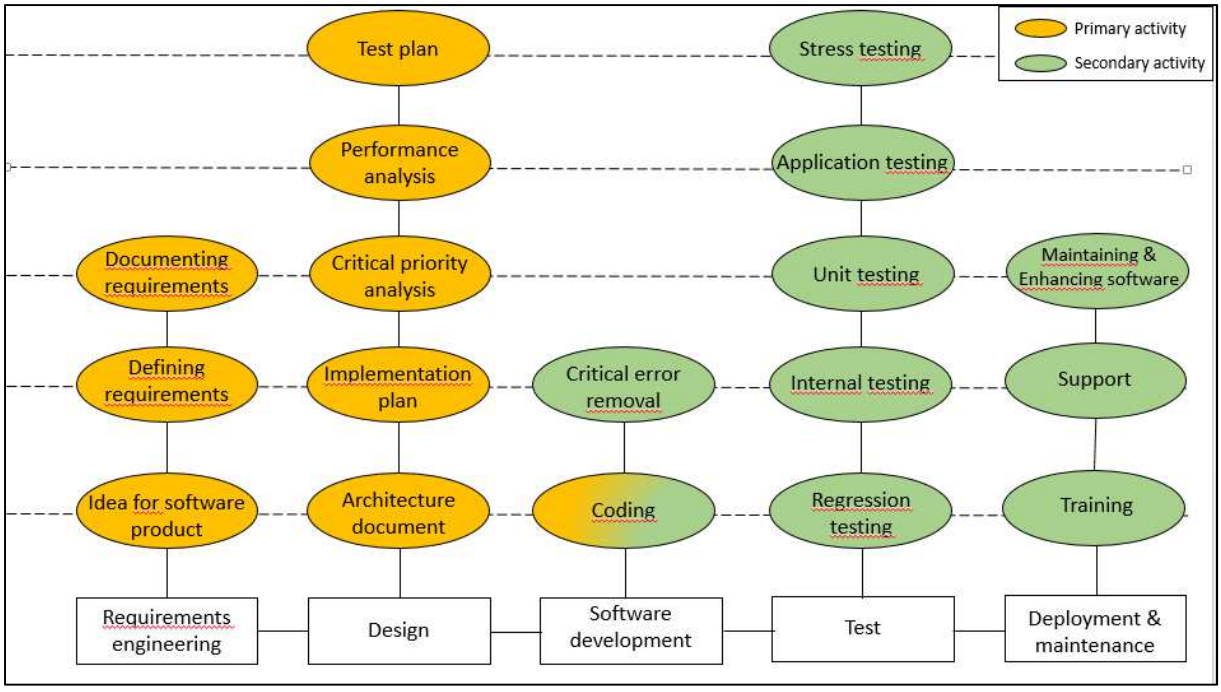


Figure 7: Software development activities divided into primary and secondary activities

Figure 7 contains all the phases and the most important activities of software engineering from the idea until the maintenance of the software. The purpose of this figure is to find out what the primary activities of software developing companies are. No differentiation has been made between the process model of development. Even though the way of displaying is based on the waterfall model and might indicate a sequential process, these activities can also be carried out incremental or cyclical and thus do not affect the working method used.

Nagar (2013) patented a method and computer program that assists in outsourcing software development business processes based on literature and other patents. This method includes characteristics such as the

ability to perform tasks remotely with minimal impact on business processes, the existence of well-defined protocols and the ability to easily monitor the task; having a minimal dependence on other tasks and allowing the organization to retain control over the business process.

Requirements engineering & design phase

Both the requirements engineering and design phase are not suitable for outsourcing. They maintain a pivotal position in defining product requirements and being able to translate them into a software product. Allowing external vendors to (partly) take over this process could lead to losing control over product definition and direction. As requirements are mostly based on customer demands, this could lead to inefficiencies as customer information has to be passed on to the external partner. Furthermore, defining the architecture, design specifications and coding against the specifications is crucial. Developers may work more efficiently when design documents are made available that clearly written document that accurately describe the product requirements. This requires architects/senior developers having either or both technical and product knowledge. Technically sound developers can then write the code based on the specifications. Therefore, it can be concluded that both the requirements engineering and design phases are crucial in the software development process. Outsourcing either the requirements engineering phase or the design phase could have a negative effect on the core competencies of a software developing company in the long run (U.S. Patent No. US 8,566,138 B2, 2013).

Software development phase

Coding the software against the architecture and specifications may be well suited for outsourcing. It presents minimal risk to the company itself while promising substantial cost benefits. Furthermore, it may be useful to reach for an external vendor when more developing capability is needed. In contrary, the geographic distance between locations might lead to communication problems. Therefore, it is of utmost importance that the coders understand the architecture and are able to correctly code against the specifications. It is suggested to implement well-defined processes and communication protocols to protect against this risk (U.S. Patent No. US 8,566,138 B2, 2013). Moreover, companies should take into account both economic and technical component characteristics which seem to enhance outsourcing success. Hence, it is important to assess which components should be developed in-house and what components of the coding activity can be successfully outsourced. Literature has suggested an approach of assessing components, now it is time to find out if and how companies assess component characteristics and how it influences the outsourcing decision.

Testing phase

As the testing phase is closely aligned to the coding phase, it is a good candidate for outsourcing for some software developing companies. Using the documents prepared in the design phase the product will be tested if it works and if all features are present according to the documents prepared. Risks can be minimized through remotely monitoring the product quality. Carefully monitoring testing reports reduce the risks of outsourcing. It might be easier to outsource both the development as the testing phase to a single partner. They may perform the task more efficiently by using the knowledge gained coding the software (U.S. Patent No. US 8,566,138 B2, 2013).

Deployment & maintenance phase

The deployment and maintenance phase is also a good candidate for outsourcing. An external vendor might be able to deploy and configure the software either on-site or in the cloud. As part of this duty, they have the responsibility for making sure that the product operates correctly for the customer. This could include setting up training manuals and classes to meet the customers' needs. This kind of information does not include company sensitive information. This phase might require contact between the development teams and the deployment/maintenance teams to ensure that the deployment and support teams have adequate knowledge to assist the customers (U.S. Patent No. US 8,566,138 B2, 2013).

Furthermore, the supporting activities are also candidates for outsourcing. Support staff handles problems that customers experience while using the released product. Risk of transferring institutional knowledge is minimal. The biggest challenge here is to measure the quality of customer support. In order to accurately measure customer support the software developing company has to be aware of the customer issues and how the outsourcing partner handles these issues. To minimize the risk here, the software developing company should have an escalation process wherein the company itself handles the support of customer problems (U.S. Patent No. US 8,566,138 B2, 2013).

3 Method

In order to find answer to the research question the method of data collection and data analysis has to be determined. This chapter covers the way the data has been gathered and how it was analysed. Qualitative research has been chosen over quantitative research as it provides more in-depth results (Barriball & While, 1994).

3.1 Data collection

As mentioned in the introduction of this proposal semi-structured interviews were selected due to several reasons: First, they are well suited for the exploration of opinions and perceptions regarding complex issues. Second, they enable probing to gain more information and a better understanding of the respondents' answers (Barriball & While, 1994). Finally, the group of respondents vary as they work for different companies. Therefore, structured interviews make sure that any differences in answers are due to differences in respondents rather than differences in the questions asked (Barriball & While, 1994).

The semi-structured interviews gave the interviewee a format which guided him throughout the interview. The interviews were held one-on-one rather than in focus groups as bias may occur when respondents are influenced by strong opinions of other respondents (Barriball & While, 1994). The interviews were held face-to-face as it is impossible to examine non-verbal communication in over the phone interviews. The interviews lasted between 30 and 60 minutes each. The sample size consists of eight software developing firms in the Netherlands with a maximum of 250 employees. There is no exact way to determine the amount of interviews needed, due to time restrictions the amount of eight cases has been chosen. This amount of cases allows us to compare the results between cases (Diefenbach, 2009). The interviewee has to be an expert regarding outsourcing decisions whilst being in a management position in order to avoid problems with the unit of analysis (Kramer, Klimpke & Heinzl, 2013). To ensure confidentiality anything that is discussed during the interview remains between the interviewee and the researcher. The interviews took place in a private room away from other professionals. After permission of the interviewee the interview was recorded on tape.

3.2 Data analysis

The collected data is literally transcribed on the computer using Atlas TI Software. Both the name of the firm and the name of the interviewee have been anonymized. Each firm is randomly assigned to a code word based on the NATO phonetic alphabet. The transcripts of the audio recordings have been used in order to categorize the answers in order to answer the sub-questions.

Data from each interview has been transformed into a story. Next, each story has been analysed by means of answering the sub-questions of this research. A conclusion has been drawn based on the findings of the literature review. Finally, in the cross-case analysis, the objective of the analysis was to discover patterns across all cases. These patterns are further elaborated in the discussion section.

Appendix 3 shows the interview protocol that has been used during the interviews. Both figure 5 (modularity of a system) and 7 (phase model) were used as key objects to find out what characteristics are currently considered in the outsourcing decision and in which phases the decisions are made. Likewise, the questions who is the decision maker and what influences the decision are of equal importance as it allows to explore how outsourcing decisions are being made. Especially figure 5 and the guiding text from the literature made it possible to discuss the characteristics of components in depth and to find out if aspects from both the economic and technical perspective were taken into account.

3.3 Reliability and validity

It is impossible for researchers to control for all possible scenarios. However, the researcher can influence the research project in both a positive and a negative way. The circumstances where the interview takes place, the approach and manners towards the interviewee and the friendliness towards the interviewee can help to secure the validity and reliability of the data (Barriball & While, 1994). Therefore, the interviewer was equal in all cases and guidelines have been set-up prior to the interviews. Furthermore, audio taping has been used in order to provide detailed insight in the performance of both the interviewee and the interviewer. It helps to validate the accuracy and completeness of the data collected. Furthermore, it helps to prevent data collection errors (Barriball & While, 1994).

4. Results

This chapter presents both the within-case and cross-case comparisons. Each of the eight cases have been summarized into a narrative story. The stories represent how the outsourcing decision has been made and what factors have influenced these decisions. Additional, the cross-case analysis reveals converging or diverging processes across these cases. These results are then used to answer the research question.

Case	Size	Respondent's position	Type of software	Development method	Outsourcing experience	Outsourcing outcome
A	< 250	Head of product management	Standard software	Mostly Scrum	Project: Development of an app	Failure
B	< 250	Product manager	Individual software	Traditional/Waterfall	Continous: All development	Success
C	< 250	Chief Executive Officer	Tailor made software	Mostly Scrum	Outsourcing vendor	-
D	< 250	Chief Technology Officer	Tailor made software	Scrum	Continous: Extra capacity	Success
E	< 250	Chief Financial Officer Head of software development	Individual software	Scrum	Continous: Extra developers	Success
F	< 250	General Manager	Standard software	Traditional/Waterfall	Several projects: Extra developers	Failure
G	< 250	Head of software development	Individual software	Scrum	Project: Development of a module	Failure
H	> 250	Head of software development	Individual software	Scrum	Continous: Extra capacity	Success

Table 1 Overview of the cases

Table 1 provides an overview of every single case. As anonymity has been promised each case has been masked using the first letter of the code word that has been assigned. Except for case H, each case fits the size of a small- and medium enterprise in the Netherlands (Kamer van Koophandel, 2016). Furthermore, the table provides information regarding the interviewee's position to ensure that every interviewee is both an expert regarding outsourcing decision and active in a management position. All enterprises are active in the software industry however, a distinction can be made in the type of software that they offer. First, standard software, are software solutions from vendors such as Microsoft or IBM. These can be adapted to the needs of clients through developing and offering specific modules. Second, individual software is software developed by the company itself that can be implemented at their clients. Finally, tailor made software is software specifically made for their clients. All firms that participated in this research are outsourcing themselves, have experience in outsourcing or are outsourcing vendors. Their working method during the outsourcing experience, a short description of their outsourcing experience and the outsourcing outcome of each case is given. In the next section both the within-case and cross-case analysis are presented.

4.1 Within-case analysis

4.1.1 Case 1: ALPHA B.V.

Overview case A	
Company size	< 250
Developing method	Scrum
Technology	Microsoft / IBM
Product	Standard software
What has been outsourced	Development of an app
Outsourcing decision maker	Director
Influencing factors	Capacity / time-to-market
Outsourcing country	Romania
Outsourcing outcome	Failure

Case description

Alpha B.V. is a software developing company that focuses on Microsoft and IBM technology. They provide standard software which is mostly tailored to the needs of its customers. Their software engineers are free to use any developing method that suits them best. However, most developing activities are based on Scrum. They aim to connect people and knowledge and therefore they have decided to develop a SharePoint Online app. Due to a shortage of available developers, the former director decided to outsource the development of this app. They had reached out to a nearshoring partner in Romania. An early version of the app had already

been built by Alpha B.V. on a different platform. Knowing how it should work and how it should look like they thought that this was the product, like no other, that had a chance of being outsourced successfully.

Next to the need of extra developing capacity there were some other relevant factors influencing the outsourcing decision. As all the requirements were clear for Alpha B.V. the app had to be developed fast and the outsourcing partner could offer it at a low price. After they had documented the initial ideas of the requirements engineering phase, the outsourcing partner designed the architecture of the product. Soon after Alpha B.V. agreed on the architectural design the outsourcing partner came back. They were convinced that the initial plan could not be built in this way and thus some of the requirements had to be dropped.

"One of our lead developers mentioned upfront that, what we wanted to make, was not possible. It was naïve to think that the nearshoring partner was capable of building the product even though we thought it was not possible. And thus, naïve to quickly sign the contract and thinking that it was their problem now. It does not work like that."

During the outsourcing project Alpha B.V. faced several problems. Even though they knew that the developers would build exactly what they would ask for, they were still surprised by the extent to which developers are overdoing it. A mistake had been made in the architectural model and the developers had coded the product with the mistake in it. Alpha B.V. mentioned that if they had any questions they should look in the documents and if that works, build it like that. Forgetting that there was a mistake in the documents. Fortunately, the damage was not so big so it could be corrected by the outsourcing partner. Alpha B.V. mentioned that the next time something like this happens, they should ask them first.

Another problem that they faced had to do with the non-functional requirements. Certain tests had to be done with 5000 users. When they started to test they mentioned that they only had 5 licences and thus were not able to simulate 5000 hours. This problem came up when the product was delivered. They said that it was tested, however, Alpha B.V. noticed that this was not the case. The product stopped working with more than 10 users. Thus, the product did not what it was supposed to do. Alpha B.V. had to put their own effort to test the product. After that, the development pace declined. After a couple of months Alpha B.V. asked what was happening with their findings. The outsourcing partner started to work on the project again but now with different developers. This affected the consistency of the code and thus the code quality of the code declined. Finally, a year after the product's initial release, Alpha B.V. decided to stop the outsourcing project without success.

Case analysis

In terms of the outsourcing decision it is possible to answer the who, why, how, what and when decision in this case. Alpha B.V.'s decision to outsource was made by its director as they were in need to extra developing capacity. Furthermore, the speed of development was of importance as they wanted to take the product to market. Thus, the time-to-market was of importance. Finally, the costs of development were also taken into account. Even though the costs were less important, the low cost development offered by the outsourcing partner made it more interesting to make this decision. In this case they chose a partner in Romania. They had been contacted by several outsourcing companies. As a colleague already had experience with the Romanian partner, they decided to work with them. No differentiation has been made between company, country or culture characteristics.

In terms of what to outsource a product has been chosen that had to be rebuild. Documents of the requirements engineering phase were prepared and had to be carried out. The outsourcing partner took care of the design of the product, the development and the testing of the product. Thus, the product could be developed independently, did not require a lot of communication between developers and did not require significant customer contact.

Case Conclusion

From an economic perspective the decision was made by the director and motivated from a resource-based view. Developing capacity and time-to-market are the main factors for the outsourcing decision. The choice for nearshoring has been made based on available offerings and costs. None of the economic factors have been taken into consideration. However, unintentionally they seem to be in accordance with the literature.

It seems that from a technical perspective a software component has been chosen which is, according to the literature, eligible for outsourcing.
The failure of this case cannot be explained by component based outsourcing. There seemed to be a lack of communication from both sides. However, this is outside the scope of this research and therefore an explanation cannot be given.

4.1.2 Case 2: BRAVO B.V.

Overview case B	
Company size	< 250
Developing method	Waterfall method
Technology	-
Product	Individual software
What has been outsourced	Continous development
Outsourcing decision maker	Director
Influencing factors	Finance / capacity
Outsourcing country	Bulgaria
Outsourcing outcome	Success

Case description

BRAVO B.V. is a software developing company focusing on the retail sector. They bring together workforce management, payroll and HRM in a single cloud solution. This company had already outsourced their software developing activities to Bulgaria before the interviewee arrived as project manager at the company in 2002.

BRAVO B.V. was quite a small company back then that had the ambition to become big. In order to become bigger they were in need of more developers. However, they did not have the financial resources at that time. In Bulgaria on the other hand, developers costed only 25 euro's an hour compared to 85 euro's in the Netherlands. The director decided that outsourcing was necessary in order to grow.

The product that BRAVO B.V. offers is an individual product which is applicable to all kinds of retail stores. Their developing activities were based on the traditional waterfall methods as Agile working methods were not known at that time. Based on the waterfall model, only the coding activities and the first tests were done by the outsourcing team in Bulgaria.

In the period before the project manager arrived the process of outsourcing seemed quite chaotic. "Our director visited Bulgaria and started shouting at the developers that the development process was not quick enough and that everyone was not working hard enough. There was no clear line in what they had to do and in what time it had to be finished."

As of 2002, the project manager took over the outsourcing project. Through several meetings he first defined a clear plan together with the project manager in Bulgaria. First, the requirements and the architecture were built in the Netherlands. Next, the coding and some first tests were done in Bulgaria. Finally, the project manager visited Bulgaria every 2-3 weeks to fully test what has been built. Prerequisites were that it should look like how BRAVO B.V. wanted it to look like. Furthermore, they wanted comments attached to every piece to be able to quickly get an overview of what each part is and what BRAVO B.V. could do with it. Finally, the source is owned by BRAVO B.V. and the outsourcing partner is responsible for it.

The project manager had to overcome several problems that arose. The first problem was that too many people still felt involved with the outsourcing project. There were 6 or 7 people that had already worked with the Bulgarian partner. They thought that they still had some kind of responsibility over the outsourcing activities. All these people started communicating with Bulgaria which made the communication really inefficient and unclear.

The second problem is related to the understanding of business processes between two countries. "Retail stores in Bulgaria open in the morning and close at 17:00. Employees are either polishing their nails or cleaning the shop. In terms of retail software this means that, for them, it has to support the opening/closing of the store and the checkout process. Here in the Netherlands, managers are concerned with peak hours and how many employees they need at what point of time."

This indicates that business related processes in the Netherlands are not understood by Bulgarian developers. Therefore, BRAVO B.V. decided to fly-in the Bulgarian developers and explain the processes to them before starting on the code.

"Software serves a purpose. If they do not understand the purpose of the product it is extremely difficult to build software that fulfills the purpose."

Sometimes BRAVO B.V. had more than one Bulgarian company working for them if they were in need of more capacity and speed in the developing process. In these situations communication was key. Bringing those companies together and make them work together towards a simultaneous goal takes a lot of time and effort. Through investing a lot in communication BRAVO B.V. was able to make a success of their outsourcing activities.

Case analysis

In this case the outsourcing decision was made by the director. The main factors influencing this decision stems from the resource-based view. The financial resources were not available to grow according to their ambition. Outsourcing the development of software allowed them to increase their developing capacity.

After restructuring the outsourcing activities, both the requirements engineering phase and the design phase were done in the Netherlands. The development phase was done in Bulgaria and after the initial tests, the testing and deployment & maintenance phases were carried out by BRAVO B.V. itself. As BRAVO B.V. was working according to the traditional waterfall model these phases were carried out sequentially. This makes it possible to divide the software engineering activities between the company itself and their outsourcing partner.

Important in the decision of what phases the Bulgarian team should do was business specificity. As the Bulgarian developers did not understand the business processes of the retail stores in the Netherlands, extra effort had to be put into teaching them about these processes.

Case conclusion

Even though this is an older case the information is still relevant. It seems that the activities carried out by the Bulgarian developers align with what is suggested in the literature and what has been proposed in this research. The first two phases of software engineering are carried out by BRAVO B.V., the development and the initial tests are done by the outsourcing partner and during the test phase they took back the software to deploy and maintain it. This approach allows BRAVO B.V. to protect their intellectual property while being able to hold the outsourcing partner responsible for the code.

Key in this case was communication and a clear approach. After the project manager took over the project he visited Bulgaria every 2-3 weeks to gauge process with both the Bulgarian project manager and developers. Furthermore, they let the Bulgarian developers visit the Netherlands to increase their knowledge regarding business processes.

4.1.3 Case 3: CHARLIE B.V.

Overview case C	
Company size	< 250
Developing method	Scrum
Technology	Microsoft
Product	Outsourcing vendor
What has been outsourced	-
Outsourcing decision maker	-
Influencing factors	Expertise / capacity
Outsourcing country	Romania
Outsourcing outcome	-

Case description

In comparison to case 1 and 2, case 3 is based on a company in Romania that does software developing activities for companies in the Benelux. As they do not outsource themselves, a different interview protocol has been used. CHARLIE B.V. exists almost 20 years and has been active in nearshore outsourcing for the last 15-16 years. With a sales office in the Netherlands and a development unit in Romania, they develop software for their clients mostly based on Microsoft technologies. Their developing unit consists of five departments: DotNet, Java, PHP, mobile solutions and a full testing department.

According to CHARLIE B.V. there are certain rules that apply to the outsourcing decision. The first rule is that software engineers should communicate with software engineers. Their outsourcing activities mostly start with functional input from the client, assuming that the client knows what he wants. Furthermore, the client should have knowledge about the branch/sector it is active in and a certain role deviation(i.e. project manager/scrum master/product owner). This requires a company to either be an ISV(Independent Software Vendor) or to have an own IT department. It allows to work together with the developers in Romania on a professional level. These companies mostly base their outsourcing decision on capacity or expertise. Costs is less relevant. Mostly the smaller companies find it difficult to attract the knowledge and expertise of certain technologies.

In most cases the ISV delivers the functional requirements and how it should look like. CHARLIE B.V. builds the architecture and after approval the system will be build. This is either done on a fixed price/fixed date model or in an Agile/Scrum way wherein extra capacity is made available to the client.

According to CHARLIE B.V. 99% of the industry exists of remote staffing. In this case the outsourcing partner is a recruitment agency that makes sure that the developers with the right expertise are made available to the ISV and that the ISV is responsible for the micro-management of these developers. This indicates that the outsourcing agency is not bothered with the product of the ISV and only rents developers to them. "We think that this concept is completely wrong. Our biggest added value is in the know-how that we possess. We have almost 20 years of experience in how to build an application. Not only on a technical side but also how to facilitate the long range relationship. What tools do you use? How do you share source files? In what frequency and through which channels do we communicate? That is our business and we take joint responsibility in the project."

"The biggest mistake an ISV's can make is to not put enough time and effort into the development process. Giving an assignment to build a certain product and then coming back in 3 to 6 months to check if the product is finished or not is typically a mistake that still happens often. As customer you need to take the project in your own hands. Use TeamViewer and make sure that what happens in Romania is truly in accordance with the requirements of the product. It is recommended to visit Romania every view months, or even less, to discuss the development together with the client and the development team. There is no more efficient way than meeting face-to-face."

Another problem that often arises is software developing companies that work project based. This means that there is an extra link between the customer and the software developers. In most cases this is the project

manager. The end user mostly does not know exactly what he wants yet until he sees the product. The project manager does not possess all of the expertise. Issues arise in the transfer of knowledge and information.

The last problem that CHARLIE B.V. sometimes faces is that the client's IT staff does not always accept the outsourcings relationship. If someone of the management team has decided that a part of the development has to be outsourced it sometimes occurs that the current staff is afraid of outsourcing parts of their job. This could lead to a less co-operative mindset of the client's IT staff as they might fear for a loss of control or fear for losing their job in the future. The support base of the client is one of the success factors.

When CHARLIE B.V. starts assisting companies with the development of their software product there is a huge knowledge gap. Every ISV has its own expertise/specialism/knowledge base. It takes time for an outsourcing partner to understand all the ins- and outs. This process could take several months. Help is needed from the people who know the software product so the first phase consists of knowledge transfer. It is not reasonable to assume that CHARLIE B.V. directly starts to work in the core of the product as this needs a lot of guidance to outsource this effectively. Therefore it is suggested to outsource a more isolated part of the software product with less dependencies. There is a bottom line in the amount of work that should be outsourced. Small projects can better be done by the ISV itself. However, when using an Agile/Scrum method wherein everything is packed into user stories this does not seem to be relevant.

Even more important than transferring the knowledge is the business logic of a company. The way a business works should be written on paper and this should be the essence of what the software is able to do. If a company is able to write on paper what the business logic is for that specific sector or specialism it becomes easier to make a success of the outsourcing relation.

Case analysis
This case includes an outsourcing vendor. Even though the decision making process is not equal to those of other cases, several interesting aspects that can contribute to the outsourcing decision can be derived.

First of all, most ISV's do not outsource for cost efficiency purposes. The availability of certain expertise or extra capacity are factors influencing the outsourcing decision of ISV's. Furthermore, advise has been given regarding a preferred structure.

- Software professionals should be communicating with the outsource developers to avoid miscommunications.
- A certain role deviation should be in place. For example: Product owner, scrum master or project manager.
- The leading role should communicate thoroughly with the outsourcing partner.
- The leading role should monitor progress using the tools available. Preferably visit the outsourcing partner on site every few weeks.

Important factors influencing the eligibility of outsourcing are also mentioned.

- Every projects starts with a knowledge gap. It is suggested to start with isolated components with less dependencies. This makes the outsource developers familiar with the software product.
- Business logic or specific business processes should be clearly explainable. If this can be written down in a clear sense, it is easier to understand for the outsource developers.
- Small projects are less suitable for outsourcing due to the knowledge gap and business logic issues. It takes time to understand the software product.
- Project-based outsourcing is even more complex due to changing needs of the end customer.
- The support base should be high. Developers might fear that they lose their jobs if their outsourcing relationship succeeds.

Case conclusion

Comparing this case to the literature both the business related specificity and the functional complexity seem important factors whether a project or product is eligible for outsourcing or not. Also the required customer contact in terms of project based outsourcing seems an impediment for outsourcing success. From an economic perspective, the magnitude of the product or project seems also relevant. This is not mentioned in the literature regarding the outsourcing of software components.

From a technical perspective, isolated components with less dependencies seem to be easier to develop for external developers. These components seem to be a good start for tackling the knowledge gap. When ISV's use a developing method based on Scrum, the technical perspective seems less relevant as everything is packed into user stories.

4.1.4 Case 4: DELTA B.V.

Overview case D	
Company size	< 250
Developing method	Scrum
Technology	Microsoft
Product	Outsourcing vendor
What has been outsourced	All development activities
Outsourcing decision maker	Director
Influencing factors	Capacity
Outsourcing country	India
Outsourcing outcome	Success

Case description

Offshore development is the core service that DELTA B.V. offers. Through Agile teams they develop software in their own development center in Bangalore, India. Two other services that they provide are Agile consultancy and training and an own standard software product. In short sprints of two weeks DELTA B.V. delivers working and tested software to their clients. In 1993 the director of ZULU B.V. decided that they were going to start with outsourcing. There was a huge demand for DotNet developers while the availability of DotNet developers was minimal. Coming from India, he decided to outsource the development to his home country where a lot of higher education developers were available. Later on, DELTA B.V. was born as a spin-off of ZULU B.V. DELTA B.V. has been outsourcing for 13 years now. They have experienced that working offshore increases the chance of misunderstandings. Therefore, they have decided to work with a short-cyclical process. This method is now known as Scrum.

DELTA B.V. does not have developers based in their home country (the Netherlands). One main reason for this decision is based on what they noticed at a competitor. As work comes into the company, management does not look at what is the best solution. Instead, they look at the availability of the most expensive developers. The Dutch developers are more expensive than the Indian developers. Therefore, they receive work that first comes in. This is in most cases fun and interesting work for developers. The Indian developers on the other hand, receive work that is left over. This is mostly boring and uninteresting work.

In most cases the client provides the product owner and DELTA B.V. arranges a team of Indian developers including a Scrum Master. In some situations the client also includes several developers of their own. This set-up requires a project at least to have a length of 9 months in order for it to be effective. DELTA B.V. wants to attract a team of passionate developers that want to stay for a longer period of time. If a team stays intact for a longer period, they work more efficiently. This requires to offer them perspective.

"An Indian developer is not so different from a Dutch developer. If the work offered is fun, interesting and challenging and the company can offer him perspective, he wants to stay. If there is uncertainty, he will leave. To ensure continuity, quality and engagement there needs to be a healthy flow of interesting work. Otherwise it is impossible to build a good team."

It requires a lot of time before a team can work efficiently and effectively. Not necessarily to learn new technologies but to understand the business processes in a different country. ***"There was this company that wanted to outsource the development of mortgage software to India. In India a mortgage does not exist. These developers cannot ask themselves how the software should behave according to a process since they have no experience in how the mortgage system works. This counts for more things in the Netherlands, not just mortgages."*** Thus, it takes time and effort to find a team, get them to work efficiently and finally retaining them.

A problem that DELTA B.V. faces is that clients are not used to working Agile. They think they know how it works but when a sprint is finished and a working and tested product has been released they lack interest and tell us to come back later when it is fully finished. Another problem is the joint relationship. When something goes wrong every party involved should first look at itself. For the bigger companies this tends to be difficult. When something goes wrong it is always the others fault. ***"It is just like marriage. When***

something goes wrong it is always the other person's fault. This sets the base for a bad marriage. Try to discover how all parties involved can improve."

Case analysis

The initial decision to outsource their software developing activities was made by their director due to a shortage of capacity. His country of origin had an important role in which country has been chosen. As a lot of higher educated developers were available, India has been chosen.

As a Scrum working method is a hard requirement for DELTA B.V. a strict structure is needed for their outsourcing services. Understanding of Scrum is a must and thus a role deviation is necessary. The client delivers the product owner while DELTA B.V. delivers the rest of the team. In some cases two or three developers of the client are included within the team. As Scrum uses a short-cyclical process, the technical perspective seems less relevant. Each Scrum team divides the work within their team through user stories. Through a short-cyclical process every 2 weeks working and tested software is delivered. This enforces the product owner to stay on top of the project, which increases the success of outsourcing.

From an economic perspective, the business related specificity is really important. It takes time to learn the business processes in a different country and thus it takes time for a new team to work efficiently and effectively. Technical specificity seems less important in this case as it inferior to business related specificity. Programming skills and techniques can be learned more easily than specific business processes that do not exist within the country of development. Also in this case the magnitude of the product or project is relevant. Products or projects that take less than 9 months to develop are considered not eligible. This is mostly caused by the knowledge that has to be transferred and to acquire a decent team of developers. If DELTA B.V. can offer job security over a longer period of time, preferably with interesting work, it is easier to find and retain a good team of developers.

Case conclusion

In sum, from an economic perspective both the business related specificity and the technical specificity play an important role. Also in this case, the magnitude of the product or project has to be taken into account. It does not seem to be efficient and effective if a small project is outsourced due to the business related specificity and the recruitment and retaining of developers.

From a technical perspective the component characteristics seem to be less relevant. Using a Scrum method all developers work in the same team and their work is divided through user stories. In this case all developers are working in India and thus the deviation of components based on characteristics is not relevant.

Scrum's short cyclical process enforces the clients to stay on top of the outsourcing project. This seems to be a huge success factor. Both the magnitude and the developing method seem influencing factors for outsourcing decisions.

4.1.5 Case 5: ECHO B.V.

Overview case E	
Company size	< 250
Developing method	Scrum
Technology	-
Product	Individual software
What has been outsourced	Part of the development
Outsourcing decision maker	Director
Influencing factors	Capacity / spreading risks
Outsourcing country	Ukraine & Serbia
Outsourcing outcome	Success

Case description

ECHO B.V. is providing software in the global logistics sector. Through an online Transport Management Solution (TMS) they help their clients to drive value through their supply chain. As they are working with large global clients their amount of work varies often. This requires scalability in the amount of developers. For a smaller company like ECHO B.V. it is hard to attract and retain their software developers in the Netherlands. An investor of the company thought that a scalable workforce was a key to success and thus they started to outsource some of their activities to India. As ECHO B.V. lacked outsourcing experience they had difficulties in managing an outsourcing relationship.

In the begin phase, some employees of ECHO B.V. visited India to guide the Indian developers and to do some kind of hand-over. This was not enough. After some time ECHO B.V. realized that they could not use the software that the Indian developers delivered. According to ECHO B.V. this was mostly due to their own fault. What they did was that they gave the Indian developers a poor specification of their needs and what it should do. And that is exactly what the India developers delivered. However, this was not enough as it was not re-usable for the developers in the Netherlands.

"We have not put enough resources into the outsourcing relationship in order to ensure quality. It was probably not their fault as we should have given them more guidance. We should have been on top of their work. Strictly monitor what they were doing and guide them throughout the process."

ECHO B.V. also stumbled upon some cultural differences. The Indian developers that they were working with, lacked pro-activeness in bringing in their own ideas about the software product. Furthermore, they have underestimated the amount of times they should repeat a question in order to make sure the Indian developers truly understood what was meant. This projected ended about ¾ year later in June 2012 as it did not work out as expected. However, ECHO B.V. has not given up on outsourcing yet.

Since 2015, ECHO's developers are using a new working method. As they were lacking effectiveness in the predictability of delivery, they have proposed Scrum instead of the waterfall model. This helped them to better estimate the required time to develop certain functionalities. Even after changing their working method ECHO B.V. was still not able to meet promises made to their clients as their software was not ready on time. This was partly caused by developers being hunted by other firms. It required a lot of time and effort to find and replace their developers. ECHO's management team therefore decided to start a new outsourcing pilot.

At this moment ECHO B.V. is nearshoring in both Ukraine and Serbia, using two different outsourcing companies. Their developing team consists of 3 teams. Two fully external teams consisting of developers, a tester and a scrum master. The third team is a combination of 3 Serbian developers and 3 Dutch developers. The product owner is always someone from ECHO B.V. itself to keep tight control. Their sprints last 2 weeks and afterwards the functionalities are tested. Integration tests are done in the Netherlands when all developing teams delivered their parts of the software that have to be integrated in the whole application.

Every team divides the work for themselves. The custom User Interface (UI) of ECHO's application is rather complex and difficult to understand for people outside the company. Therefore, most of the UI work is done

in the Netherlands by developers who are specialized in this specific part. All teams are capable of doing all the work. However, work on the UI side seems to take longer when the outsourcing teams work on it. Therefore it has been decided that something that is highly complex or a new functionality starts with the Dutch developers and later on the stories are expanded to the outsource teams.

Another important aspect is that, in both Ukraine and Serbia, people can choose where they want to work. Therefore it is important to offer the outsource teams interesting work. This makes it easier for the outsourcing companies to maintain their employees and that increases the efficiency of the teams. Bringing the developers on-board of the teams is time consuming and costs us a lot of money. When replacing a developer he/she has to get used to our system and thus impacts the productivity.

For ECHO B.V. outsourcing is also some part of spreading risks. A lot can happen within a country beyond a company's control and this makes a company less vulnerable. It is also easier to terminate employee contracts in some countries outside the Netherlands. ECHO B.V. is capable of scaling the amount of employees through those outsourcing partners.

Case analysis

This case contains two different outsourcing experience wherein the first experience was unsuccessful and the second experience is successful and still ongoing. In the first case, an investor decided that outsourcing could solve their scalability problem. However, due to a lack of resources put into the outsourcing relation and an underestimation of cultural differences this experience ended without success.

The second experience was a decision made by the management team as they were unable to quickly replace developers. Moreover, the amount of work kept fluctuating. This led to not being able to meet the agreed deadlines with their clients. Furthermore, the spreading of risks across countries has been taken into account. Instead of offshoring their developing activities to India, a nearshoring approach has been chosen. Both Ukraine and Serbia are closer to the Netherlands and learning from the first experience they know that they have to monitor their progress closely.

As they have two external teams and one hybrid team, work across teams is divided based on complexity. Their custom UI is highly complex and hard to understand for external developers. Thus the highly complex components start within their own teams and is later on expanded to the outsourcing teams. Within teams the work is divided by the teams itself as they are using a Scrum working method.

An important factor here is that the outsource developers also have to be retained to maintain productivity. This can be achieved through offering them interesting work and thus only offering them simple development work is not effective.

Case conclusion

From an economic perspective it seems that technical specificity is an important factor in deciding which team is going to develop a software part. Mostly the development of highly complex components starts in teams that understand the code.

From a technical perspective the component characteristics itself seem less relevant as the developers themselves divide the work within teams. Furthermore, letting the outsource teams develop only the simple work might cause difficulties in retaining developers in the outsource countries. It takes time to replace the developers and that affects the productivity.

4.1.6 Case 6: FOXTROT B.V.

Overview case F	
Company size	< 250
Developing method	Waterfall method
Technology	Drupal
Product	Standard software
What has been outsourced	Part of the development
Outsourcing decision maker	Management team
Influencing factors	Capacity
Outsourcing country	Moldavia
Outsourcing outcome	Failure

Case description

FOXTROT B.V. is specialized in the technical development of websites and web applications. These websites and web applications are built using the Drupal Content Management System which allows them to contribute from the Drupal open-source community. Compared to the other cases FOXTROT B.V. does not have its own individual software. They develop project-based on the demand of its clients using a Scrum developing method.

In 2008 several bigger projects were coming in and FOXTROT B.V. was only able to accept these projects if they increased their resource pool without paying an excessive amount of money. An outsourcing company approached FOXTROT B.V. in being able to offer resources instantly. The management team of FOXTROT B.V. has decided that they were going to start a pilot with this company.

The outsourcing partner connected FOXTROT B.V. to individual developers in low-wage countries in the same time zone. In this case, they started with a developer from Moldavia and later on developers from Ukraine were added. At this point in time, FOXTROT B.V. was still making use of the traditional waterfall developing method.

A project manager of FOXTROT B.V. was guiding the external developers. In the beginning the Moldavian developer had built a website almost completely on his own with some technical guidance from the Dutch developers. They worked with an online tool which allowed them to upload a .psd file with the graphical design. For every aspect of the design they could point out how this part should behave and how it should work. The Moldavian developer started working on this and after completion the project manager tested whether the functionality did what it was supposed to do. Some bugs and issues where going back to the Moldavian developer until the product was finished. The experience with this individual Moldavian developer was successful and therefore FOXTROT B.V. decided to continue and to expand their outsourcing activities.

After the first project a 2nd and 3rd developer were added and both were connected to a project manager of FOXTROT B.V.. Several cultural problems arose. Out of politeness they seemed to give a social correct answer or at least an answer to avoid confrontation. In the Netherlands someone could tell and explain why he is not able to finish his work on time. That might even be understandable and is explainable to clients. The following quote gives an example of what FOXTROT B.V. experienced.

"One of the developers was really good. However, whenever we asked him when he was finished he would tell us: "in an hour.". After two hours we still heard nothing and even the day after it was still not finished. "Ya, ya, ya, I am just fixing a bug." Is what he send back to us. We were thinking what is this for kind of bullshit and what is his story? This doesn't give us any transparency. What does I am almost finished mean?"

"In the begin phase there is a 1000 hours left of a project. But at the end problems start to arise when we are short on time. Then you need to operate as a team and these external developers are not going to make the difference in this game. Maybe they do .. in a negative way. At this point we knew that he was not operating in the same team. "

Even though literature has not been consulted in the decision of what they were going to develop externally. Less complex and more isolated components with a structured task with a clear definition of done have been formulated. In this case less complex does not mean simple or fully isolated. The simple components are bought from the Drupal community which is less time consuming and more cost efficient.

The management of FOXTROT B.V. had to make a new decision. Either they were going to put in all effort to change their behavior and make them feel comfortable to the Dutch way of communication and really make them part of the team or the outsourcing pilot has partly failed. Due to the economic crisis in the Netherlands it became easier to find new developers and they have decided to quit the outsourcing pilot.

Case analysis

FOXTROT B.V.'s outsourcing decision has been made by their management team in order to be able to attract new projects. Their developing capacity was not sufficient in order to be able to accept these new projects. Starting with a single extra developer that did the right job it seemed that their outsourcing experience was going to succeed. Adding extra developers for other projects did not lead to the same result. These external developers were not able to deliver software on time and as FOXTROT B.V. is responsible for delivering the software to their clients they were not able to finish their projects on time. Their project managers were not able to gain transparency from these developers as they were not able to monitor what they were doing.

Case conclusion

Comparing this case to the literature is hard. Even though none of the factors of the economic perspective have been taken into account. It seems that these factors were not influencing the outsourcing outcome.

From a technical perspective, FOXTROT B.V. tried to reduce the complexity of the software components that the outsourcing developers had to develop. However, this did not influence the outsourcing success as the developers were still not able to deliver the software on time.

In conclusion, the failure of this case cannot be explained by component based outsourcing. It seems that mostly the cultural differences and a lack of monitoring progress caused this failure. The question whether this is the case or not falls outside the scope of this research.

4.1.7 Case 7: GOLF B.V.

Overview case G	
Company size	< 250
Developing method	Scrum
Technology	-
Product	Individual software
What has been outsourced	Part of the development
Outsourcing decision maker	Management team
Influencing factors	Developing capacity / speed
Outsourcing country	India
Outsourcing outcome	Failure

Case description

GOLF B.V. provides solutions and services for the education sector. One of these solutions is an individual software product which is a digital learning an working environment for children. The development of this platform started 5 years ago. GOLF B.V. has difficulties in generating pace in the development process. The software is mostly developed based on their own roadmap. GOLF B.V. also works with large tender processes. This can cause difficulties in the development roadmap as priority is given to the development of specific customer requirements as described in the tender inquiry.

In order to increase the development pace their management team decided to start two different pilots. One of these pilots was to outsource certain development activities to an outsource company. This company had approached GOLF B.V. and was based in India. Thus, the decision to outsource to a certain country was irrelevant in this case. them to show that they were willing to support them in what they do best. No literature has been consulted as the outsourcing partner claimed to have a lot of experience in outsourcing and GOLF B.V. decided that they were willing to support them in what they do best.

The other pilot was to temporarily add a Dutch external developer in their own team at their office. Both capacity and money were criteria in the outsourcing decision in order to see what was the most effective and efficient.

After setting up the requirements GOLF B.V. found an isolated part which was well documented that was going to be developed externally. Even though the software part had been carefully selected they still faced several problems. The biggest problem that they faced was that Indian developers did not understand the functional domain. For example, in the Netherlands, kids have to do a CITO test at the end of their primary education. This test does not exist in India. Therefore it was hard for the Indian developers to understand how the end product should work according to the functional domain.

"If we would have asked them to develop a calculator, it would not be a problem. They know what a calculator looks like at how it should work."

Combining this with a cultural difference in the sense that the Indian developers did not came back with questions if they did not fully understand what has been explained to them. This is a rather normal process in the Netherlands.

"I recently found out that if an Indian developer answers the question if he understood what was meant that if he answers with 'yes' that it means that he has understood what you have said rather than what you mean."

GOLF B.V.'s software is rather complex and every module has to be integrated in the system. Half of their Dutch development team was busy in facilitating the outsourcing process. It took them way more time, effort and money to develop the module than they would have if they had developed it themselves.

These issues has led to the conclusion that hiring an outsource company to do some of their development activities was unsuccessful while integrating the Dutch external developer has led to more success.

Case analysis

The outsourcing decision has been made by the management team. Mostly based on creating extra development speed. In order to do so, the development capacity had to be increased. The country of outsourcing has not been chosen. Their decision was based on the company that they wanted to work together with rather than in which country they had established themselves in. As no literature has been consulted it seemed, also according to GOLF B.V., a bit naïve to simply think that it will be alright.

In terms of what to outsource, the development of a profile page for their individual software has been outsourced. This specific part has been chosen because it was well documented and demarcated. After setting up the requirements the outsourcing partner took over and started to develop.

It seems that GOLF B.V. have underestimated the amount of effort that has to be put into an outsourcing relation.

Case conclusion

From an economic perspective it seemed that the education sector in India is completely different. The Indian developers did not know the functional domain of the education sector in the Netherlands. As can be concluded from other cases it requires time and effort to let external developers get used to the Dutch functional domain. Both the functional complexity and the technical specificity seemed less relevant in this case.

From a technical perspective their individual software seemed rather complex. It has a lot of dependencies that have to be integrated into the system. Comparing this to the literature, a lot of knowledge has to be transferred before external developers can work efficiently and effectively. This cannot be done in such a short period.

In sum, the business related specificity and the highly complex software of GOLF B.V. seemed to be factors that have affected the outsourcing relationship. It requires a lot of time and effort in both communication and monitoring what the outsourcing partner is doing. This has been underestimated by GOLF B.V.

4.1.8 Case 8: HOTEL B.V.

Overview case H	
Company size	> 250
Developing method	Scrum
Technology	-
Product	Individual software
What has been outsourced	All development activities
Outsourcing decision maker	Director
Influencing factors	Capacity
Outsourcing country	India
Outsourcing outcome	Success

Case description

HOTEL B.V. develops software for the facility management and real estate industry. Compared to the other cases, HOTEL B.V. is a slightly bigger company with approximately 600 employees. They are active all across the globe and have development teams in the Netherlands, France, Canada and have their own development centre in Haiderabad, India. Their individual software offers CAFM (Computer Aided Facilities Management) and IWMS (Integrated Workplace Management Systems) to optimise their facility management processes and real estate portfolios. Most developing teams use Scrum as a starting point for their working method. They are free to design their own working method that fits them best.

Their first outsourcing experience was between 2008 and 2012. They started to outsource the development of a certain module/product for their individual software. The outsourcing decision was made by both the head of product development and the commercial director. There was not enough developing capacity to build the product and therefore they have decided to let another Dutch software developing company develop it as a partner product. This company was chosen as they had already delivered custom made software for one of HOTEL B.V.'s biggest clients. This custom made software has been used as a start for the product.

Eventually this partnership yielded a product that could be implemented in their own individual software. However, issues came up after the implementation of the product. HOTEL B.V. had their own roadmap of development and releases. The Dutch partner was constantly 6 months behind in developing the external product. This caused a lot of integration problems for the individual software that HOTEL B.V. was offering as HOTEL B.V. still had the responsibility towards their clients. The Dutch partner worked on a project base and also had obligations to other clients. They were not able to continuous develop the product of HOTEL B.V. even though it was a commercial success. Furthermore, they were Dutch and the support that they could offer was only for Dutch clients. HOTEL B.V. already had several international clients which they had to provide support for themselves.

The last problem was that the product was built as a DotNet application while the individual software of HOTEL B.V. was built on a JAVA stack. As an individual web application this was not a problem. However, HOTEL B.V. was not able to take on this product on its own as they did not possess the DotNet expertise. A decision was made to let another company rebuild the product on a JAVA stack and to add some extra functionalities. This was done by a company in India as they were able to provide international support for the product and they were able to grow together with HOTEL B.V. due to the high availability of developers in this country. This company in India was not able to yield a product as the combination of technical migration and being able to build it on a JAVA stack including extra functionality was too much. Thus, this project was considered as a failure.

Since 1998 HOTEL B.V. had the idea to create their own development centre in India. The former director made this decision based on two important factors. The availability of developing capacity was the most important factor to make this decision. HOTEL B.V. was not able to attract enough developers in the Netherlands to support their product development. Due to the technical and functional complexity of their

product the director decided that it should not be outsourced. It takes a lot of time to gain and transfer the knowledge of their product and they did not want this to lay into the hands of an outsourcing company.

In 2008, during the first outsourcing project, HOTEL B.V. started to work together with an outsourcing partner that already had a lot of experience in India. HOTEL B.V. demanded that it retained overall management and control on all processes. The outsourcing partner provided staffing services and were facilitating HOTEL B.V. in assembling their first Indian developing team. This team had to use a Scrum development method and was a hybrid team mixed with Dutch developers. Their strategy was to copy the developing quality of the home-based team and then split up this team into a hybrid and into a full Indian developing team and expand both teams. As of 2010 they had established their own entity in India. All personnel moved from their partner to HOTEL B.V.'s own entity. They have steadily grown ever since.

Their teams in India have two main functions. Most of their teams are concerned with the development of their individual software product. Their product development is based on their roadmap and release planning. The other function is the development of custom tailored solutions on top of their individual software product. These are client and project driven using a traditional project approach.

For HOTEL B.V. it is extremely important to retain the developers in their developing teams. As they have invested a lot of time and money to learn them about the functional domain and about the individual software itself.

"The lesson that we have learned is that if you are going to outsource or develop offshore the continuity of the employees is the most important success factor. We have to retain our developers for at least 3 to 5 years. It takes 1 or 2 years for them to learn the product and the functional domain and after that they start to work on an efficient level. If they leave within two years it only costs money and the quality of development keeps decreasing."

HOTEL B.V.'s retention rate is extremely low, even compared to European standards. For them, their offshore development strategy works and they claim to be successful.

Case analysis

As this case consists of two separate experiences, both experiences will be shortly highlighted. The outsourcing decision has been made by the management team as they did not have enough capacity to build the product that they wanted to make. Two separate outsourcing partners participated in this experience.

The first partner created the product as a partner product. Even though this product was a success due to their work on project basis they were not able to keep up with the roadmap of HOTEL B.V.. Furthermore, they were not able to support their international clients. Thus, a decision has been made to outsource the entire product to India on a different platform. This required the product to be rebuild. The Indian company was not able to migrate their product from DotNet to JAVA and to add certain functionalities. Their technical capability was not sufficient to build the product.

The second outsourcing decision has been carefully considered over a long period of time. Their intention has always been to establish their own development centre due to the technical specificity and functional complexity of their product. India has been chosen due to the high availability of educated developers. As they did not possess any knowledge about India or its labour market a decision was made to use an outsourcing partner. This partner has been chosen due to their extensive knowledge in the Indian market.

Overcoming the knowledge gap and being able to retain the developers in India was their biggest challenge.

Case conclusion

Their outsourcing decision has been motivated mostly from a resource-based view. Due to difficulties in attracting and retaining developers in the Netherlands their management team decided to look for opportunities externally.

From an economic perspective, both business related specificity and functional complexity played an important role in deciding to keep their knowledge in-house. These factors caused that a huge knowledge gap had to be overcome and causes the urge to retain the developers in India for as long as possible.

No differentiation of workload is made between hybrid teams and full Indian teams. Their Scrum based teams decide for themselves who is developing what. Thus, the characteristics defined from a technical perspective do not seem to be of influence here.

4.2. Cross-case analysis

The goal of this research is to find out how outsourcing decisions are made at SME's within the Dutch software industry. In order to do so it is important to find out *who* made the outsourcing decision, *what* has been outsourced, *why* has this decision been made and *which* factors have influenced the sourcing decision.

Who made the outsourcing decision?	A	B	C	D	E	F	G	H
Investor					x			
Director	x	x		x				
Management team						x	x	x
Project manager								

Table 2: Outsourcing decision maker

In most firms the responsible role for outsourcing is either the director or their management team (mostly including the director. For case C the situation is different. As this is an outsourcing vendor itself, they do not outsource their activities. In case E it was the investor that determined that outsourcing was necessary in order to grow. "A new investor came in and he said: we have to be scalable. Thus, one of the measurements became scalability and that meant that we had to work together with an outsourcing partner." (Interviewee case E). This already implies that the decision is made regardless of the project.

When is the decision made?		A	B	C	D	E	F	G	H
Before the project								x	
During the project	Requirements engineering phase	x							
	Design phase								
	Development phase								
Always outsourcing			x		x	x	x		x

Table 3: Point in time when the decision is made

Table 3 provides an overview of when the decision is made at a certain point in time. The implication that has been made in the previous paragraph is motivated by interviewee E in the following quote: "We have a team, that team gets smaller as developers leave. As we are a software developing company we continuous develop software. It is hard for us to replace them so let's look at an offshore/outsourcing model." Equal to case E it seems that most other firms decide to outsource regardless of the project. Interviewee F emphasizes: "In our situation, multiple projects run at the same time. We have a shortage in our pool of resources. Thus, the decision to outsource is made regardless of a project." Whereas firm H has a similar motivation: "For us, capacity and the continuous availability of development capacity was the most important driver to look externally."

In contrary, firm A made the outsourcing decision during the project as they came up with an idea to build an app and they found out that they did not have the resources to develop it. After setting up the requirements they realized that this app had the potential to be successfully outsourced. "Knowing how it should work and how it should look like they thought that this was the product, like no other, that had a chance of being outsourced successfully." (Interviewee A). Firm G also behaves differently. Their outsourcing decision was made before the project. Deciding that they wanted to do a pilot with external developers. The word 'pilot' indicates that it is something temporary. After the decision to outsource was made a software part has been found. Thus, their outsourcing decision was made before the project.

Factors influencing the decision	A	B	C	D	E	F	G	H
Resource-based	x	x		x	x	x	x	x
Product-based	x							x
Cost-based		x						

Table 4: Overview of influencing factors

According to the overview of table 4, most firms' outsourcing decision is resource-based. All cases have a shortage of developers as the main reason to outsource except for case B. Financial benefits were the main reason to outsource. Of course a shortage of developers also influenced their decision. However, cost benefits was their main driver: "Everything comes down to finance. At the time that you need extra man power in the Netherlands to develop it, that means that you need to have the financial resources available to do so. Paying 25 euro's an hour in Bulgaria or 85 euro's in the Netherlands, that is a huge difference."

Both firm A and firm H have also taken product characteristics into account. For case A the main reason was extra developing capacity to outsource. However, they also had a product that had to be rebuild which was, according to them, the perfect product to outsource. For firm H on the other hand, their product characteristics was the reason for them not to outsource. Instead, they have established their own offshore development centre in India. "Our product is technical and functional complex. The knowledge that those developers build up is essential. Thus, the model wherein a developer comes into the team, works 6 months on it and then leaves .. That means that we are only training, training, training and are never able to deliver a product."

This indicates that only two out of 7 (8 - case C) firms take product characteristics into account in their outsourcing decision. Both case C and case D provide outsourcing services and they are able to provide insights from a different perspective. According to firm C a company has to look at its product characteristics to determine their outsourcing strategy. "Especially in certain projects it takes time to understand all the ins- and outs of a product. This process could take several months. In the first phase of the outsourcing relationship it is harder to directly develop complex components that are part of the core of their product. Thus, it is suggested to start the outsourcing relationship with components that are not part of the core of the product and has less dependencies."

Firm D agrees that complex components with a lot of interactions are harder to develop. However, they suggest that, exactly because of that, it is a reason to outsource. "Things that are hard should be done as often as possible, only then you get good at it. Thus the fact that it is hard is in my opinion the worst reason not to do it. If you are going to fitness while you are not fit and you start jogging it is hard and you start to feel dizzy. Is that a reason to say that you shouldn't be jogging? No, if you want to achieve something then you should let it strengthen you. Of course the beginning is hard but after a while you will get good at it."

Only two out of all cases have been looking at their product characteristics before making the outsourcing decision. Some others have looked at the component characteristics, after the initial outsourcing decision has been made. The following table will provide an overview of what component characteristics have been used in order to determine what to outsource.

Component characteristics taken into account	A	B	C	D	E	F	G	H
Functional domain		x		x				x
Business related specificity			x	x				x
Functional complexity			x		x			x
Technical specificity		x	x		x			x
Isolated components						x	x	
Less complex			x		x	x		
Size of the project	x			x				x
New functionality					x			x

Table 5: Overview of component characteristics taken into account

Technical specificity plays an important role in the outsourcing decision. This is mostly related to see whether the outsourcing partner has the required knowledge and expertise to be able to build what desired. Firm B sometimes divides the work between two different outsourcing partners. In that case they assess who has the expertise to be able to build the software part. Additionally, firm C experiences that their smaller clients

often do not have a broad range of expertise. "Our clients are software companies. However, it is difficult for the smaller clients to manage all technologies. If you do not possess the expertise than you should develop it externally."

Furthermore, the functional domain and business related specificity are of importance. According to interviewee C this is crucial: "What really matters is business logic. How does my business work? What information has to be put into the software? How is the information processed? How does the outflow of information look like? This is the essence of what the software should do. If you are not able to write on paper what your business specialism or sector specialism is than you have a huge problem that is not only software related. If you can write your business logic on paper than outsourcing should not be a problem." Firm D adds: "They have to build up domain knowledge, that takes a lot of time. This is a normal process due to the fact that they do not use the same business processes."

Firm G on the other hand, experienced the negative effect of not taken the functional domain into account. The Indian company that they had worked with did not understand the functional domain of the education sector in the Netherlands thus the developers did not understand what was meant to be build.

Functional complexity and less complex components are also of importance in the outsourcing decision. Firm E emphasizes: "Whenever something is of high complexity or a new functionality it automatically starts within our own teams. Later on it will be expanded to our outsource teams." Firm F also started with looking for a less complex part to be outsourced. However, according to them outsourcing is becoming a lot more interesting when highly complex software components are eligible for outsourcing.

Moreover, the size of the project is of importance. Firm A, C and D all mention that it is not interesting to outsource short team projects due to both the overhead costs and the learning curve. C uses overhead costs as motive: "You need at least 3 developers and 3 months of work will it be of some use. The overhead costs are in communication and gets absorbable through both having a low cost and a sufficient size." While firm D only takes on projects for at least 9 months to offer their offshore developers perspective. "To ensure continuity, quality and engagement there has to be a healthy amount of work. Otherwise it is impossible to form a good team. It is a win-win situation."

Moreover, isolated components are considered by both firm F and G. Firm C adds that advises to start an outsourcing relation with a more isolated part of the software that is not part of the core of the product. In this way developers can get used to the software code. As firms tend to look at component characteristics to find out what they should outsource. The next questions of the interview was whether they use standard procedures to determine what to outsource.

Use of standard procedures	A	B	C	D	E	F	G	H
Yes								
No	x	x		x	x	x	x	x

None of the interviews use standard procedures to determine what they are going to outsource. This is explainable by the fact that for most of the firms it was their first outsourcing experience. However, even those firms that are more experienced in outsourcing, do not use standard procedures. Interviewee F emphasizes that they stopped working with their outsourcing partner. However, if they had proceeded the outsourcing relationship they would have had completely optimized processes. Even though firms do not seem to have standardized processes they do think classifying software components might influence outsourcing success.

Classifiying software components	A	B	C	D	E	F	G	H
Has influence on outsourcing succes	x		x	x	x	x		x
Does not have influence on outsourcing succes		x	x	x			x	

Starting with the firms that do think it influences outsourcing success. According to firm H:" If you do not take both the economic and technical aspects into account you will certainly go wrong at some point." Firm E adds that: "If you are in a somewhat more comfortable position where you have enough capacity available

it will be very useful. If you are in such a position you can classify what has to be done first and give the low priority story's to an outsourcing partner."

While firm A, E, F and H think it has influences outsourcing success. Both firm C and firm D think that it can have influence on success in certain situations. Firm C: "I agree with the literature regarding the classification of software object. However, with an Agile/Scrum approach wherein everything is packed into story's this seems less relevant. In certain projects, especially in the begin phase of the outsourcing relationship, there are complex pieces that are part of the core of their product that are harder to outsource." Firm D adds: "Within software development there is hardly any work that is easy or easy to transfer knowledge. If it is simple than it is routine work. Routine work is more like call center work or data entry work. That might be interesting to outsource, however that is not really software development. If there is a high amount of routine work required, it is not complex then it might be possible to outsource. Your motivation should then be based on cost savings."

Finally, firm B and G think classifying software components does not influence the outsourcing success. Firm B mentions: "I think it is the easy way. I do understand it because companies want to invest as less as possible to maximize profit. The trick is to bring together two companies and invest in communication. Communication is the most important aspect."

In short, it seems that Dutch SME's decide about what to outsource based on component characteristics wherein several characteristics can be derived as important for the outsourcing decision. It seems that classifying software objects might influence outsourcing success. However, no procedures have been standardized yet.

5. Conclusion

The aim of this research was to unveil how Dutch SME's in the software industry make decisions regarding the outsourcing of software components. Therefore, this study provides insight into eight cases within the Dutch software industry that have experience in outsourcing. Through a literature review and semi-structured interviews data was collected and analysed.

This explorative study shows that in most cases the director and/or his management team is responsible regarding the outsourcing decisions of the company. Even though three different types of software developing companies participated in this research their reason to outsource is equal. An outsourcing approach is mostly chosen due to difficulties in attracting and retaining software developers. This motivation stems from the resource-based view and transaction cost economics and thus it can be concluded that the economic perspective plays an important role in the motivation of the outsourcing decision.

The results show that the outsourcing decision is made regardless of the magnitude of developing activities required. A project indicates that it is temporary. Most participating firms choose an always outsourcing approach and thus the decision is made regardless of a project.

In terms of *what* to outsource interesting results can be derived from this research. Only a small part of the participants take product information into account in their sourcing decision. Whereas all firms focus on *what* to outsource after the initial sourcing decision has been made. The *what* to outsource decision could be made with the help of component-based decision making. According to our respondents several component characteristics should be taken into account. First, both the functional domain and business related specificity should be taken into account. If developers do not understand the context wherein the software should work, they are not able to ask themselves how it is supposed to work. Furthermore, functional complexity and technical specificity are important characteristics. Outsource developers should be able to understand the software code and need to possess the knowledge and techniques to be able to develop a software product. Moreover, the complexity and dependencies of a component have to be taken into account. If the component is complex and/or has a lot of dependencies it takes more time to be able to understand the code and being able to develop efficiently. This also counts if new functionality has to be developed. Finally, the size of the developing activities is considered. If the amount of work is not of sufficient scale, outsourcing seems an expensive and time consuming yet unavoidable option.

Most component characteristics that are taken into account stem from an economic perspective. The characteristics derived from a technical perspective seem less relevant due to a Scrum working method which is a widely used practice in the Netherlands. Scrum teams divide the workload themselves, management is not involved in this process. Through a short-cyclical process software is developed and tested within 2-4 weeks. Every morning a stand-up meeting is held, this can also be done virtually. Progress is monitored and struggles are taken away. Classifying software components from a technical perspective seems to become less relevant when outsource developers work in the same team. This also increases the amount of interesting work for external developers as they do not only receive the small and easy developing work. In turn, this makes retaining them easier and increases productivity in the long run.

However, when outsource teams do not work in the same team it is suggested to start with less complex components with non or only a few dependencies. This is used as a start point to decrease the initial knowledge gap. It seems of utmost importance to provide a product owner as lead into these outsource teams to be able to guide and to monitor them. However, this falls outside the scope of this research.

In sum, most characteristics derived from the economic perspective seem to be relevant in the outsourcing decision whilst those from the technical perspective seem to become less relevant due to a short-cyclical working method. The next chapter compares the findings of this research to what has been found in the literature. In the final section of this thesis, limitations and suggestions for further research will be given.

6. Discussion

In this chapter the results of the within-case analysis and the cross-case analysis are compared to the findings of the literature review and the answers to our research questions are presented. As there is hardly any research done on component based outsourcing and no research has been done regarding the outsourcing decisions of SME's in the Dutch software industry this research was of exploratory kind. This study sheds light on the outsourcing decisions made by Dutch SME's in the software industry.

In the current literature, research regarding the outsourcing of software development is mostly from an organizational perspective. Both the Resource-Based View and Transaction Economics are widely used theories in this research area. An important shift in the focus of outsourcing decisions has taken place in the last decade. Early literature focuses mostly on cost benefits while current literature focuses on effective resource allocation. This study provides evidence that cost benefits only plays a minor role in the outsourcing decision of SME's. Being able to effectively allocate your resources seems to be the main reason for SME's to outsource.

Only a few studies have researched the operational character of software development. For example, Lacity, Khan & Whillocks (2009) provide an overview of the available work in the areas of strategy, risk, success factors and capabilities. Furthermore, Kramer, Heinzl & Spohrer (2011) proposed a decision making heuristic for software components. Several characteristics determine whether a software component is eligible for outsourcing or not. Later on, Kramer, Klimpke & Heinzl (2013) did research in Germany where they compared the decision making from SME's to those of large enterprises.

Section 4.2 starts with answering the question about *who* is responsible for the outsourcing decision. The results found in this research regarding the responsible roles for the outsourcing decision seem to slightly differ to those found by Kramer, Klimpke and Heinzl (2013). In their research project managers seem to have a lot of influence in the sourcing decision. However, we have found that in all cases the director was involved, either alone or through a management team. This difference is explainable as in their study the director was mostly also the project manager or at least responsible for the project.

Next, the *when* is the decision made regarding software components is answered. This research shows that in most of the cases the decision is made regardless of a project. Applegate & Montealegre (1991) and Dibbern et al. (2004) mention that mostly larger enterprises decide about outsourcing during their strategic decision making. Kramer, Klimpke & Heinzl (2013) found that SME's decide if outsourcing is going to be applied in a specific project. Thus, they decide before the project. This contradicts with the findings of this research. Our cases seem to decide on a strategic level regardless of projects. Thus, the firms participated in this research seem to behave equally to larger enterprises.

Regarding the factors influencing the outsourcing decision, the findings of this study corresponds to existing literature (Dibbern, Goles, Hirschheim, & Jayatilaka, 2004). The outsourcing decision is not related to cost savings anymore. This research confirms that SME's outsource in order to gain extra capacity, and thus motivate their decision from a resource-based view. Additional aspects are product based decisions that are mostly made after the initial outsourcing decision itself. Thus, the theoretical model proposed in section 2.4.1. is confirmed. Both economic and technical factors are taken into account during the outsourcing decision. As both economic factors and technical factors have to be taken into account, the decision making process becomes rather complex.

Kramer, Klimpke & Heinzl (2013) have found that SME's make decisions on operational level after the initial outsourcing decision. Their studies focuses on the specific software development phases using the traditional Waterfall method. They have found that outsourcing decisions on operational level are made within these phases. However, in practice it seems that most software developing firms use Agile working methods. Within Agile, processes are short-cyclical and the phases of software development are interleaved. It seems that the literature in this specific area of research is lagging behind practice. Instead of making outsourcing decisions after the initial outsourcing it seems more logical for software developing companies, that use Agile working methods, to use the component-based outsourcing characteristics in the initial outsourcing decision on product level rather than on component level.

Finally, this study tried to answer *which* characteristics are determinants in the outsourcing decision regarding software components. Comparing the component characteristics found in this research to those found by Kramer, Heinzl & Spohrer (2011). It seems that our interviewees use slightly different characteristics. From an economic perspective both the functional domain, business related specificity, functional complexity and technical specificity play an important role. However, our interviewees did not take the strategic significance nor the amount of customer contact into account. In contrary, they did take the magnitude of the amount of work into account.

From a technical perspective, only complex components and the amount of dependencies were taken into account by our respondents. The level of coordination needs, size of the component and the required customer contact were not taken into account. This is mostly explainable by their working method as our respondents use Agile as a working method. Their teams divide the work themselves, management is not involved. Qualifying components based on these characteristics seems therefore less relevant than literature suggests. As this research was done on management level, it was impossible to find out whether developers themselves use any of the characteristics found in the literature.

This study has clearly confirmed that some of the characteristics are indeed taken into account. Mostly after the outsourcing decision has been made. However, it is arguable whether some of these characteristics should be taken in account before or during the outsourcing decision and if they should be made on component level or on product level. For example, both the functional domain and business related specificity could be taken into account during the outsourcing decision. If a country, sector or niche has specific business processes that are not common in the rest of the world or cannot be clearly written on paper, it suggested not to outsource to a country where these specific business processes do not exist. Unless you are planning to invest a lot of time, money and effort into the outsourcing project.

Furthermore, if a company wants to be able to make use of developers that continuous work on their individual or standard software. Both the complexity and the magnitude of development work should be taken into account. If a software product is very complex and has a lot of dependencies, it seems harder to outsource the development of these components. If the amount of work is less than a couple of months in combination with a complex software product it is suggested not to outsource. Several other opportunities such as hiring freelancers seems more efficient.

Finally, this explorative study tried to shed light on the existence of standard procedures regarding the classification of software objects. Even though firms agree that classification of software objects could potentially influence the outsourcing success, no standardized procedures for classifying components exist.

In sum, this study provides new insights into component-based outsourcing through evidence from eight cases from the Dutch software industry. Both resource-related and product-relevant information provide input in the outsourcing decisions of SME's in the Dutch software industry. As outsourcing decisions do not only seem to be made from an organizational perspective it is now possible to further explore the decision making on operational level.

7. Limitations and suggestions for further research

This study should be seen in the light of its limitations. As only eight cases have been used statistical generalization is hardly possible. According to Yin (2013) statistical generalization is not the goal of the multiple case study method. Analytical generalization on the other hand is possible (Yin, 2013). Each interviewee might have answered each question slightly differently as open questions have been used compared to a questionnaire. An example is that the component characteristics in table 4 have been identified throughout the whole interview rather than asked in a single question. If someone mentioned that they had started with looking for an isolated component while answering the question to tell something about their outsourcing experience, the isolated component box has been checked. Finally, the results are based on single case interviews and this might influence the rigor of the results. However, as all interviewees are highly involved in the outsourcing decision of their firms they can be considered as valuable sources (Yin, 2013).

This study has explored how SME's in the Dutch software industry make their outsourcing decisions with a special focus on component-based outsourcing. As current literature is still exploring this topic it seems that research on component-based outsourcing level is lagging behind practice. While current literature still focuses on the traditional waterfall method and its software development phases, software developing firms within the Netherlands mostly use an Agile working method. Agile working methods seem to influence the outsourcing decision as, from a technical perspective, the classification of components seem to become less relevant. Further research should focus on how the differences between working methods influence outsourcing decisions. Moreover, the classification of components seems to divide software development in complex and interesting work and simple, easy and boring work. The latter could indeed be easier to outsource. However, this research has provided input that this could negatively influence the outsourcing project in the long run. As this is outside the scope of this research, this topic seems interesting for further research. Finally, whereas Kramer, Klimpke & Heinzl (2013) compare the decision making behaviour of SME's in Germany to large enterprises it is now possible to compare the decision making of software developing companies across countries. However, this falls outside the scope of this research and thus is suggested for further research.

References

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New Directions on Agile Methods: A comparative analysis. *Software Engineering, 2003. Proceedings. 25th International Conference on IEEE*, 244-254.

Alexander, M., & Young, D. (1996). Outsourcing: Where's the value? *Long Range Planning, vol. 29(5)*, 728-730.

Applegate, L., & Montealegre, R. (1991). Eastman Kodak Organization: Managing Information Systems Through Strategic. *Harvard Business School* , Case 9-192-030.

Barney, J. (1991). Firm resources and sustained competitive advantage. *Journal of management, 17(1)*, 99-120.

Barriball, L., & While, A. (Journal of advanced nursing, 19(2)). Collecting Data using a semi-structured interview: a discussion paper. *1994*, 328-335.

Bergkvist, L., & Fredriksson, O. (2008). Outsourcing Terms: A Literature Review from an ISD Perspective. *ECIS 2008 Proceedings. Paper 129*.

Bhattacharya, A., Singh, P. J., & Bhakoo, V. (2013). Revisiting the outsourcing debate: two sides of the same story. *Production Planning & Control, 24:4-5,DOI: 10.1080/09537287.2011.648491*, 399-422.

Camel, E., & Agarwal, R. (2002). The maturation of offshore sourcing of information technology. *MIS Quarterly Executive 1*, 65-76.

Carmel, E., & Argarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *IEEE Software, 18(2)*, 22-29.

Dibbern, J., Goles, T., Hirschheim, R., & Jayatilaka, B. (2004). Information systems outsourcing: A survey and analysis of the literature. *Communications of the ACM 35(4)*, 6-102.

Dibbern, J., Winkler, J., & Heinzl, A. (2008). Explaining variations in client extra costs between software projects offshored to India. *MIS Quarterly 32(2), ,* 1-30.

Diefenbach, T. (2009). Are case studies more than sophisticated storytelling?: Methodological problems of qualitative empirical research mainly based on semi-structured interviews. *Quality & Quantity, 43(6)*, 875-894.

Gottfredson, M., Puryear, R., & Phillips, a. S. (2005). Strategic Sourcing: From Periphery to the Core. *Harvard Business Review*.

Heutink-ICT. (2015, December 21). *Homepage*. Retrieved from Home: http://www.heutink-ict.nl/

Isaksson, A., & Lantz, B. (2015). Outsourcing strategies and their impact on financial performance in small manufacturing firms in Sweden. *International Journal of Business and Finance Research, Vol. 9, No. 4, ISSN: 2157-0698.*, 11-20.

Kakabadse, A., & Kakabadse, N. (2002). Trends in Outsourcing: Contrasting USA and Europe. *European management journal, vol. 20(2)*, 189-198.

Kamer van Koophandel. (2016). *Jaaroverzicht ondernemend Nederland.* Kamer van Koophandel.

King, W., & Torkzadeh, G. (2008). Information Systems Offshoring: Research Status and Issues. *MIS Quarterly, 32(2)*, 205-225.

Kramer, T., Heinzl, A., & Spohrer, K. (2011). Should this software component be developed inside or outside our firm? A design science perspective on the sourcing of application systems. *New Studies in Global IT and Business Service Outsourcing*, 115-132.

Kramer, T., Klimpke, L., & Heinzl, A. (2013). Outsourcing Decisions of Small and Medium-Sized Enterprises: A Multiple-case study approach in the German software industry. *System Sciences (HICSS), 2013 46th Hawaii International Conference*, 4236-4245.

Kroll, P., & Kruchten, P. (2003). *The rational unified process made easy: a practitioner's guide to the RUP.* Addison-Wesley Professional.

Lacity, M., Khan, S., & Willcocks, L. (2009). A review of the IT Outsourcing literature: Insights for practice. *The journal of strategic information systems, 18(3)*, 130-146.

Leavy, B. (2004). Outsourcing strategies: opportunities and risks. *Strategy & Leadership, vol. 32(6)*, 20-25.

Mirani, R. (2006). Client-vendor relationships in offshore applications development: An evolutionary framework. *Information Resources Management Journal, vol. 19(4)*, 72-86.

Nagar, A. (2013). *U.S. Patent No. US 8,566,138 B2.*

Odin-Groep. (2015, Februari 11). *Over Odin.* Retrieved from Odin-Groep: http://www.odin-groep.nl/OverOdin

Picot, A., & Baumann, O. (2007). Modularität in der verteilten Entwicklung komplexer Systeme. *Journal für Betriebswirtschaft, 57(3-4)*, 221-246.

Shao, B., & David, J. (2007). The impact of offshore outsourcing on IT workers in developed countries. *Communications of the ACM, 50(2)*, 89-94.

Sommerville, I. (2011). *Software Engineering.* Boston: Pearson Education.

Stratman, J. (2008). Facilitating offshoring with enterprise technologies: Reducing operational friction in the governance and production of services. *Journal of operations management*, 275-287.

Tallman, S., & Fladmoe-Lindquist, K. (2002). Internationalization, globalization, and capability-based strategy. *California Management Review, 45(1)*, 116.

Tripathy, A., & Eppinger, S. (2011). Organizing Global Product Development for complex engineered systems. *IEEE Transactions on engineering management, vol. 58, no. 3*, 510-529.

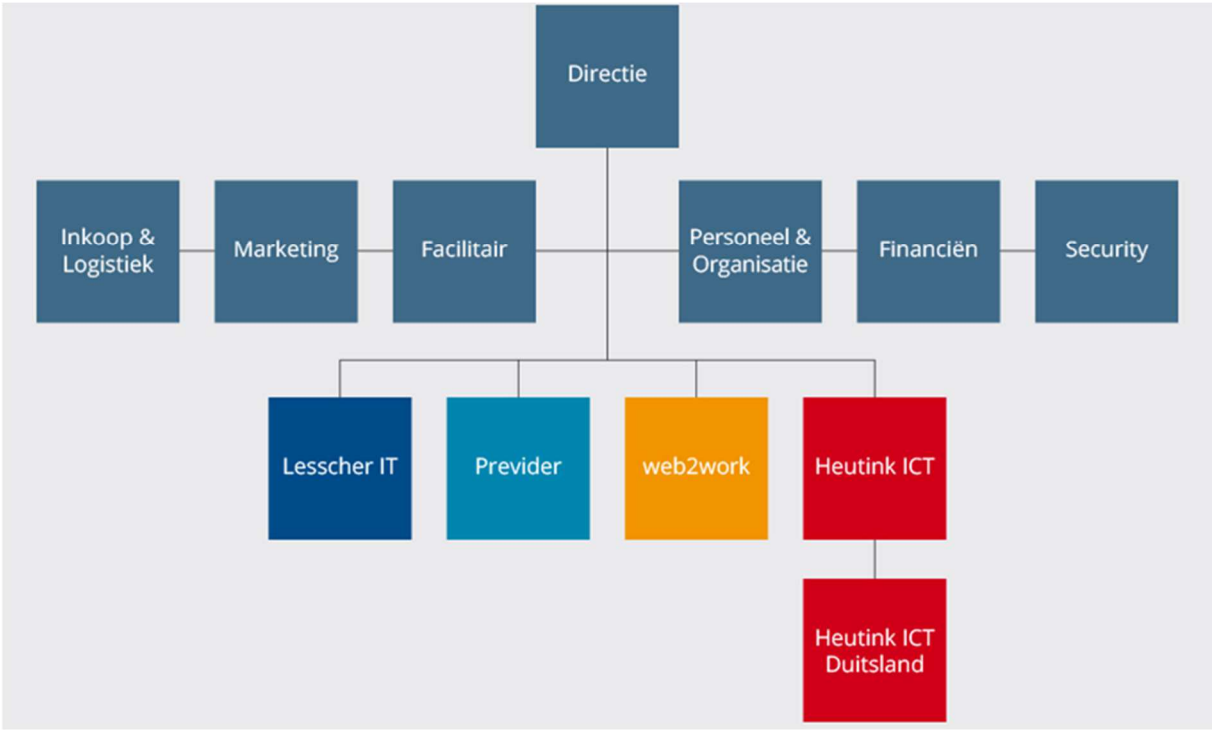
Whyte, G. (1994). The role of asset specificity in the vertical integration decision. *Journal of economic behavior & organization*, 287-302.

Williamson, O. (1989). Transaction cost economics. *Handbook of Industrial Organization, Vol. 1*, 135-182.

Williamson, O. E. (1981). The economics of organization: The transaction cost approach. . *American journal of sociology*, 548-577.

Yin, R. K. (2013). *Case study research: Design and methods.* Thousand Oaks: Sage Publications.

Appendix 1



Appendix 1: Organizational Chart Odin-Group *(Odin-Groep, 2015)*

Appendix 2

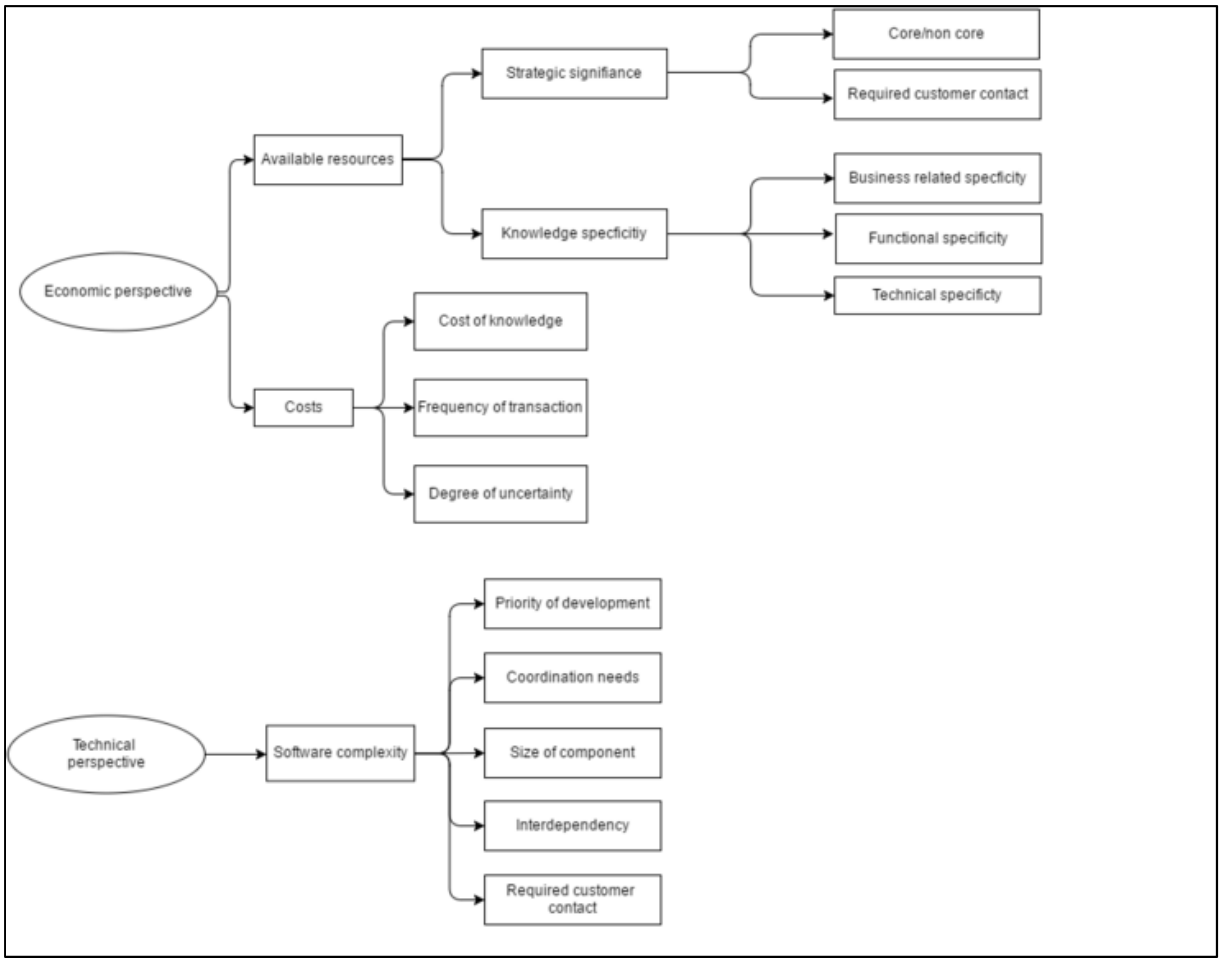


Figure 4: Tree chart of the abstract model to discover the characteristics

Appendix 3

Interview protocol used during the interviews:

Interview protocol

Opening

1) Introductie van de interviewer
2) Het doel van het interview en een kort overzicht van de vragen.
3) Toestemming vragen tot het opnemen en anonimiteit van de respondent garanderen.

***** Voice recorder starten vanaf dit punt

4) Algemene informatie over de respondent
    a. Naam
    b. Leeftijd
    c. Aantal jaren ervaring in de ICT branche/outsourcing
    d. Relatie van de respondent m.b.t. Outsourcing
5) Zijn/haar rol in de organisatie

Het bedrijf

1) Vraag om een introductie van het bedrijf
    a. Wat voor software ontwikkelen jullie?
    b. Hoe lang doen jullie dat al?
2) Wat voor een werkmethode gebruiken jullie?
    a. Vraag om een stap-voor-stap omschrijving van hun software ontwikkelproces.

Outsourcing algemeen

1) Wat verstaat u onder het begrip outsourcing?

2) Welke ervaring(en) heeft uw bedrijf met outsourcing?

3) Wie heeft bepaald dat er voor outsourcing is gekozen?

    ***** Plaatje laten zien van de fases van softwareontwikkeling o.b.v. het waterval model

4) Kunt u mij aangeven op basis van het plaatje in welke fase besloten is dat er geoutsourcet moest worden?

5) Wat is/zijn de reden(en) waarom er voor outsourcing is gekozen?

6) Welke factoren hebben volgens u invloed gehad op deze beslissing?

7) Wat hebben jullie exact geoutsourcet?
    a. Waarom hebben jullie ervoor gekozen om dit uit te besteden?

Als het 'programmeren/coderen/ontwikkelen' is genoemd:

i. Al het codeer werk of deels?
ii. Indien deels: Hoe is bepaald welke deel werd geoutsourcet??
iii. Wie heeft deze beslissing genomen?
iv. Zijn er nog andere factoren van invloed geweest?
v. Op grond van welke criteria zou u bepalen welke deel van het programmeerwerk wordt uitbesteed?

8) Zijn jullie tegen problemen aangelopen?
    a. Waarom wel/niet?

9) Op basis van het plaatje, welke fase(s) acht u geschikt voor outsourcing?
    a. Waarom juist deze?

Nu gaan we wat dieper in op het onderwerp door het te linken aan de literatuur.

Theorie over het outsourcen van het programmeerwerk

In recente literatuur worden verschillende kenmerken genoemd die een software object classificeren als 'in aanmerking komend voor outsourcing'. Dit gebeurt zowel vanuit een economisch als vanuit een technisch aspect. Het economisch aspect is gerelateerd aan de kosten van de kennis waarover het personeel moet beschikken om een object te kunnen coderen in combinatie met de mate van het strategische belang van het bedrijf. De technische factor kijkt naar de eigenschappen van het object dat ontwikkeld dient te worden.

***** Eigenschappen laten zien

Economisch aspect
Vanuit het economische aspect komt een onderdeel van het programmeerwerk in aanmerking voor outsourcing wanneer:

- Door de ontwikkelaars weinig kennis benodigd is van management en operationele processen die ondersteund dienen te worden door het software product. (Business related specficity)
- Externe software ontwikkelaars weinig speciale kennis nodig hebben om de software code te begrijpen. (Functional complexity)
- Er rekening gehouden wordt met de benodigde vaardigheden en technieken om het onderdeel te kunnen programmeren. (Technical specificity)
- Er weinig strategisch belang is. (geen kerncompetentie)
- Er weinig klantcontact vereist is.

***** Plaatje laten zien van modulair architectuur

Technisch aspect
Op technische gronden komen onderdelen van het programmeerwerk in aanmerking voor outsourcing wanneer het:

- Kleine objecten.
- Onafhankelijk van elkaar ontwikkeld kunnen worden.
- Weinig coördinatie tussen ontwikkelaars vereisen.
- Weinig klantcontact vereisen.
- Lage ontwikkelprioriteit hebben.

De volgende vragen hebben betrekking op de theorie.

1) Herkent u het classificeren van programmeerwerk dat in aanmerking komt voor outsourcing?
    a. Zo ja, maken jullie hier gebruik van of hebben jullie hier gebruik van gemaakt tijdens een project?
    b. Waarom wel/niet?
    c. Welke kenmerken van software objecten acht u geschikt voor het bepalen of een product geschikt is voor outsourcing?
    d. Waarom deze?

2) Maken jullie gebruik van standaardprocedures voor het classificeren van het programmeerwerk?
    a. Hoe gaat dat precies in z'n werk? / Waarom niet?

3) Denkt u dat het selecteren van het programmeerwerk op basis van het classificatie invloed kan hebben op het succes van outsourcing?
    a. Waarom wel/niet?

4) Welke fase(s) van softwareontwikkeling zou volgens u het meest geschikt zijn om het programmeerwerk te classificeren voor outsourcing?

Afronden interview

1) Heeft u nog suggesties om toe te voegen aan dit interview?
2) Kent u nog andere mensen die eventueel interessant zijn om te interviewen?
3) Hartelijk bedankt voor uw tijd!
4) U ontvangt een kopie van mijn verslag zodra deze is afgerond en goedgekeurd door de universiteit.