Linköping University

Master's Thesis in Industrial Engineering and Management

Department of Management and Engineering

Division of Economic Information Systems

2014

Value Creation from IT Systems Integration: A Benefits, Openness and Price Model Perspective

A CASE STUDY OF MID-SIZED COMPANIES FROM THE SWEDISH RETAIL SECTOR

HARALD BREGE & PETTER HAMPUSSON

LIU-IEI-TEK-A--14/02086--SE

Examiner: Alf Westelius

Supervisor: Markus Radits

KEYWORDS

APPLICATION INTEGRATION, OPEN SOURCE, PRICE MODELS, SOFTWARE PRICING, IT IN RETAIL, VALUE OF IT, VALUE CREATION, CUSTOMER VALUE

Preface

Writing this report has been both very interesting and challenging. When we first sat down with Entiros in early February we had many ideas of what could be interesting to study. Together we developed the idea of studying value from the perspectives of benefits, openness, and price models. The purpose of the study has been revised many times since, but these core concepts have always been in our focus. Being able to really bite into such an interesting breadth of issues, developing a model for value, turning that model into an interview guide and finally gathering data from real companies has been very exciting. Seldom is one given the time to develop ideas and analysis as fully and uninterrupted as during a thesis work and we have both really appreciated the opportunity to do so. Given the obstacles we encountered, we are able to say that we are proud of this final report we produce as students.

We would like to thank the many people that have helped us and guided us through the process of writing this thesis:

- Our examiner **Alf Westelius**, whose experience and insightful comments have helped us set up and complete this study.
- Our supervisor **Markus Radits**, who has always been available to discuss matters and help us improve both the content and language of the report.
- Our opponents **Anton Nilsson** and **Christian Tennstedt**, who with sharp eyes and clever suggestions have helped improve the report significantly.
- **Stefan Wik** and **Markus Weinhofer** at Entiros, who gave us the opportunity to do this thesis work and have helped us produce relevant results along the way.
- The expert trio **Thomas Rosenfall**, **Daniel Kindström**, and **Johannes Bynke**, who have helped us validate our work and provided us with many interesting approaches for the analysis.
- All the kind people and companies who participated in the study and gave us so much interesting and useful information to work with. You may be anonymised, but you all know who you are!

Linköping, the 18th of September 2014

Harald Brege & Petter Hampusson

Summary

This thesis is aimed at finding out how integration of IT systems creates value for companies and is conducted as a qualitative case study, where six companies are interviewed about their integration solutions. The interviewees were five CIOs and one Managing Director. Four of the companies interviewed have genuine Application Integration solutions, where a middleware platform is the hub of the integration system, while the remaining two used solutions based around a system of point-to-point integrations.

The value of an integration solution will be considered a combination of the benefits an integration solution provides, openness aspects, and the price models used to pay for the system. Value is defined as what the company gains, in monetary terms, in exchange for what it pays for an offering (Anderson, Kumar, & Narus, 2007). This definition further defines the attractiveness of an offering as the value minus the price. When discussing the value of an IT system, it is important to consider the difference between the potential value, which is the maximum the system can deliver with an ideal environment and usage, and the realised value, which is some fraction of the potential value that a company actually gains, of the system (Davern & Kauffman, 2000; Smith & Nagle, 2005). Affecting the realisation of potential value are certain factors, called conversion contingencies, which are things like preparation of implementation projects or efforts at using all aspects of a system.

Benefits are analysed according to a framework that divides IT systems benefits into five categories: operational, managerial, strategic, IT infrastructure, and organisational benefits (Shang & Seddon, 2000). These categories are focused around, respectively, productivity gains, enhanced planning capabilities, new strategic capabilities, better IT administration, and process improvements. We conclude that most companies gain several large benefits in the operational and managerial types, while the other three types have fewer reported benefits. We conclude that there seems to be quite a lot of unrealised potential value in the integration solutions, if the view of the potential of integration from the technological side is used. We also conclude that companies in certain environments and with more complex organisational structures seem to have a larger value potential than others, meaning they have more to potentially gain from an integration solution.

For openness, five aspects of open source software are studied: lock-in, cost, security, flexibility/modifiability, and community. These aspects are mainly derived from literature on open source. The first conclusion we make regarding openness is that most of the CIOs seems not to be fully aware of what the term truly entails. Companies' opinions regarding open source can be seen on a range between two extremes: those who want to modify or develop software and those who only want to use standard systems. The former category has more to gain from the aspects of cost and modifiability than the latter, but both categories can gain from the lock-in, security, and community aspects.

The combination of factors that creates the price of an offering can be described as a price model. To study the value of price models of integration solutions, the SBIFT model (Iveroth, et al., 2013) is used, where the price model is divided into five dimensions, scope, base, influence, formula and temporal rights. None of the interviewed companies were satisfied with the alternatives for price models currently on the market. It was concluded that the dissatisfaction mostly stemmed from the facts that the companies had little opportunity to affect the price model, meaning they could not adapt it to better fit their internal conditions, the complexity of the license agreements, and that it was hard to get vendors to cite a price for a system. Price models that would be more attractive are e.g. models with a larger variable part, like transaction-based ones, or models that affect the time scale of the contract, even though no single model seemed more attractive to all companies.

Definitions

- **Application Integration (AI)**
  A more advanced form of integration that makes use of a middleware platform. (Themistocleous & Irani, 2002)
- **Integration System**
  A system that can handle integration between several discrete applications and IT systems. It can be created with several point-to-point integrations or using some middleware platform as a basis for the integration.
- **Integration Solution**
  An integration system that is placed into some kind of real IT environment at e.g. a company and has to handle integration tasks between systems.
- **Value**
  What a company gains, in monetary terms, in exchange for the price it pays for an offering. (Anderson, Kumar, & Narus, 2007)

Table of Contents

Figures, Lists and Tables

1 Introduction

This chapter introduces the reader to this thesis work. It is intended to give a background to the topic of integration, explain the problem that is going to be studied and what is problematic and interesting about it. Finally, the purpose of the thesis is presented, along with the research questions, delimitations, and a short discussion of how the latter affect the conclusions drawn from the analysis.

1.1 Background to Integration

The need for integration of IT systems emerged out of the transition from the monolithic enterprise mainframes, which used to handle all applications, to a multitude of hardware systems, e.g. workstation PCs (Linthicum, 1999). It has also become more common for companies to buy supplementary software to complement their ERP system in different ways; for all talk of ERP systems being flexible enough to handle all organisations' needs, many researchers (Hasselbring, 2000; Linthicum, 1999; Markus, 2000; Sprott, 2000) have found that such is seldom the case. Instead, companies turn to the so-called Best-of-Breed approach, where several smaller systems that better fit their specific organisational needs are purchased. During the 2000s it has also become more and more popular for companies to buy and use cloud software, also known as SaaS (Software-as-a-Service), especially for complementary systems.

This development leads to an increase of new applications and computer systems that an organisation needs to handle and a corresponding increase in the need for integration of these systems (Linthicum, 1999). The modern discipline for handling such integration tasks is called Application Integration (AI) and has the purpose of making applications adaptable to business and technology changes and to retain legacy applications and technology (Hasselbring, 2000).

1.2 Thesis Assignment

This thesis work is made on behalf of Entiros AB, a Swedish systems integrator founded in 2010 with approximately 15 MSEK revenue and 17 employees. The company offers delivery, implementation, service and operation of AI solutions for customers, along with strategic consulting in AI matters. The offering is based around the open source platform Mule ESB by American developer MuleSoft. In order to tailor solutions and value proposals to better fit customer expectations and needs, Entiros wishes to have more insight into how different aspects of integration solutions combine to create customer value. Since the company works with open source software and since the IT business has seen a diversification in the price models used, they also wish to know more about how openness aspects and price models affect the customer value of an integration solution.

1.3 Problem Definition

So far, few case studies have been made regarding value and benefits from integration and most of those are more focused on system adoption or implementations and their corresponding success factors. Our literature survey only found a small number of articles that are focused on trying to evaluate what kinds of benefits or value companies actually gain from their integration systems, e.g. Themistocleous and Irani (2001a). On the other hand, some authors (Hasselbring, 2000; Linthicum, 1999) rather seem to take it for granted that the integration systems turn out to be beneficial, since the technology shows such great potential.

Yet other authors, such as Davern and Kauffman (2000), show that there is a difference between the potential value of an IT system and the value that a company can actually gain from it. This is hard to reconcile with the view that AI systems are beneficial and valuable in every case, because of their great potential. Thus there is a need for a study of the benefits and value that companies in the real world actually manage to gain from their integration solutions and to examine whether the results integration brings holds up to the expected potential or not. Because of the lack of previous studies in the area, this examination will have to be exploratory, with the goal of finding potential connections between benefits, open source, price models, and value. Value is considered as what a company gains, in monetary terms, in exchange for the price it pays for an offering (Anderson, Kumar, & Narus, 2007).

Entiros uses *openness* as one of the keywords, both internally and when approaching potential customers, mostly influenced by them using an open source integration platform as the base of their offerings. Open source contains a number of factors, which proponents often say are very valuable for organisations using the software. Just to mention a few, there is the large community that searches for bugs in the code, the very low initial cost for the software, and the reduced reliance on a software vendor. The question is, however, whether companies actually feel they can garner any value from these factors.

Finally, there are the matters of price and payment. The software business has always had a few issues with traditional price models, which do not work quite as well as in other cases (Shapiro & Varian, 1999). These days, the number of available options increases, and with the rise of cloud computing, software with non-traditional price models is becoming increasingly common among companies. This raises the question of how the price model used to sell it affects an integration solution, and whether the price model itself might even become something that creates value or if it just affects the attractiveness of an offering.

1.4 Thesis Purpose

The purpose of this thesis is to explore what value medium-sized companies in the Swedish retail sector can gain from their integration solutions, with value viewed in terms of solution benefits, openness aspects, and the price model used.

1.5 Research Questions

- ***What connections are there between the benefits and the value of a solution?***
  We expect to see higher perceived value from companies that reap more benefits from their systems, and vice versa.
- ***What connections are there between openness aspects and value?***
  We expect to see an uneven interest in open source and associated factors and that the perceived value will be affected by the interest in open source in general.
- ***What connections are there between price models and value?***
  We expect that the price model will have some impact on an offer's attractiveness, but that there will be several different views on the value question and which model is preferred.

1.6 Delimitations

To ensure that we have an easier time finding case companies, we will limit ourselves to use Swedish companies. Further, we have decided to focus on one business sector, to have less variation in the setting. According to a 2010 study by the Swedish market researchers Radar Group, the retail sector is one of the largest consumers of integration (Radar Group, 2010), which led us to choose that sector. On the advice of Entiros, we also chose to limit the study to mid-sized companies, approximately 250 to 3,000 MSEK in revenue.

The consequences of these delimitations is that other sectors that are large users of integration, like banking and manufacturing, will not be studied, which reduces the generalizability of the study. Further, since we focus on mid-sized companies, the large retail chains like H&M or ICA will not be studied, which might give a different picture of integration in the retail sector. The choice of only using Swedish companies also reduces the generalizability on an international scale. A study that had a broader size scope would likely have other views on the price model aspects, due to larger and smaller companies having very different positions to negotiate from. Looking at more business sectors would also likely alter the value results, especially if we also studied sectors that do not have the same need for integration.

2 Methodology

This chapter will present the methodology used by the authors of this thesis report. Its purpose is to provide the reader with insight into all stages of the thesis-creation process and to prove that good academic principles have been followed.

First the approach and process to be used in the thesis are explained, followed by an overview of the processes for gathering theoretical and empirical material for the analysis, along with a criticism of the sources. Finally, there is a discussion on the validity and reliability of the report's findings.

2.1 Methodological Process

This section explains the scientific approach of this thesis work and what kind of analytical techniques will be used in drawing the conclusions to the research questions.

2.1.1 Research Direction

Lekvall, Wahlbin, and Frankelius (2001) define four types of research directions: the *explorative*, the *descriptive*, the *explanative*, and the *predictive* direction, who are explained briefly in **Fel! Hittar inte referenskälla.** below. Because of the lack of previous studies in the area, this thesis will take an *explorative* direction. This means it will focus on trying to find potentially interesting connections and examine how value might be created rather than giving any definitive answers to what value is created and which aspects confer what value.

- ***Explorative:*** Used to make initial forays into less known theoretical areas to try to find hypotheses for later studies to examine.
- ***Descriptive:*** Used when it is interesting to paint a picture of how something is, without putting any deeper effort into explaining why it is that way.
- ***Explanative:*** Used for studies with the aim of explaining not just how something is but also why it is that way.
- ***Predictive:*** Used to when it is necessary to try to predict how something would change because of certain actions.

List 2.1 – Research Directions (Lekvall, Wahlbin, & Frankelius, 2001)

2.1.2 Research Approach

There are many aspects that need to be considered when choosing the research approach for a study. Lekvall, Wahlbin, and Frankelius (2001) have identified several dimensions that need to be determined when planning a study.

Depth or Breadth

- ***Case study:*** This type of study is focused on a more in-depth analysis of one or a small number of objects without any intentions of making comparisons with other objects or trying to generalise the findings.
- ***Cross sectional study:*** This type of study is instead focused on a broader study of a larger number of objects with the intention of drawing conclusions about some larger group or making extensive comparisons. The cross sectional study exists in two types: the survey study and the experimental study.
    - *Survey:* A cross sectional study where objects are passively observed.
    - *Experimental:* A cross sectional study where some experimental variable is manipulated and the resulting effects on the objects are studied.

List 2.2 – In-depth vs. Breadth Approaches (Lekvall, Wahlbin, & Frankelius, 2001)

Since this thesis aims to enter a relatively unexplored area of research, we have chosen the case study approach. If the researcher is conducting a case study for theory building, Eisenhardt and Graebner (2007) argue that even a single case study can richly describe the existence of a phenomenon and that multiple cases provide an even stronger base. Multiple cases also enable comparisons that can be used to clarify if a phenomenon is apparent in just one case or if it can be found in several. This can help to create a more robust theory, since it is better grounded in empirical evidence. (Eisenhardt & Graebner, 2007)

It should be noted however that the cases used in this study is smaller and less rich than the cases used by Eisenhardt and Graebner (2007) in discussing the benefits of a case study. As we do not have more than one person in every company each case lacks views from different perspectives. The cases are also just based on one interview at a certain point in time. But even if the cases are less rich than ideal, we believe that the case study approach is still suitable for this explorative study.

Qualitative or Quantitative

- ***Qualitative:*** With a qualitative approach the data to be gathered is expressed in words and analysed with thought models and written arguments. This approach is more common when more complex or poorly understood phenomena are studied, since it does not demand that the researchers already have a clear view of what aspects are going to be interesting before the study begins. The process of gathering data tends to be less structured, since the interaction between interviewee and interviewer are more important for the data richness.
- ***Quantitative:*** With a quantitative approach the data to be gathered is expressed in mathematical terms and analysed using statistical tools and mathematical models. This approach is more common when the phenomena being studied are relatively well understood and can be structured properly, since they can be studied and calculated more precisely with a quantitative approach. The process of gathering data is more structured, since the data is gathered as variables are mapped onto nominal, ordinal or numeric scales.

List 2.3 – Qualitative vs. Quantitative Approach (Lekvall, Wahlbin, & Frankelius, 2001)

Since this thesis studies complex and subjective phenomena we will use a qualitative approach, which is commonly used together with a case study approach. The *reliability* in a qualitative study may be limited in many cases, as circumstances when performing the study are unlikely to remain unchanged and the interactions between researcher and interviewee at a certain time will be unique. However, *validity* can instead be higher compared to a quantitative study. When questions in a quantitative study are structured and quantifiable, they are at the same time often simplified and shallow with the risk of missing the point of the study. In a qualitative study, the researcher has the opportunity for deeper interviews that are better in explaining complex situations than simplified questions in a survey. It should be noted that the difference between a qualitative and a quantitative study does not have to be as definite as it may seem from the definitions. Many studies have the characteristic of both and it is not always clear which approach is the best for a certain study. (Lekvall, Wahlbin, & Frankelius, 2001)

Time Scale of Study

- ***A specific point in time:*** The study aims to study what the situation is like at a specific point in time, usually “now”.
- ***Development over time:*** This type of study often called a time series study and intends to study the development of some phenomenon or phenomena over a certain period of time.

List 2.4 – Time Scale of Study (Lekvall, Wahlbin, & Frankelius, 2001)

This study is interested in exploring what value creation could look like and not how it has evolved over time. Though we obviously would not have had time during a master’s thesis to follow the companies over time, it would still have been possible to ask for the companies’ views on the development. However, since we are chiefly interested in exploring the lay of the land, we feel we would do well to focus on the situation right now during 2014. For an in-depth study over time it could be harder to find interviewees who are knowledgeable about what things were like some years ago, and it would require more interviews, which we didn’t have time for.

Data Types Used

- ***Secondary data:*** With this approach only secondary data is used. A study of this kind is often called a desk study.
- ***Primary data:*** With this approach primary as well as secondary data is used. A study of this kind is called either a field study or a laboratory study, depending on where data is gathered.

List 2.5 – Data Types Used (Lekvall, Wahlbin, & Frankelius, 2001)

This thesis shall use both primary and secondary data. This primary data will be collected in a field study of companies that use integration and will be collected through personal and telephone interviews. The secondary data is mostly in the form of existing theory in the fields of integration, open source and price model and not data from previous studies on the value companies can gain from integration. We will also collect some information on the integration market from Entiros and different integration vendors' websites.

2.1.3 Scientific Approach

The scientific methods of handling a problem and reaching certain conclusions about it, can be narrowed down to two approaches: the *deductive* and the *inductive* approach. With the deductive approach, the base of the study is a theoretic framework or model from which one or a few hypotheses are formulated. These hypotheses are then tested against reality by the use of observations and can be approved or disapproved depending on the result of these observations. The inductive approach, on the other hand, can be described as quite the opposite, where the study starts by making several observations of some phenomenon or phenomena, which can then be generalised into some theoretical model. (Le Duc, 2007)

It should be noted that it is not always easy to specify that a study is using one approach or another. Sometimes it can be more of a sliding scale where some elements of the study are of the deductive type where others are inductive. This is common in a third approach that is called *abduction,* which can be described as a mix between the other two approaches. This can be the case if the researchers first conduct a pre-study of more explorative nature using an inductive approach and then use the results from that study to perform a deductive primary study. (Le Duc, 2007)

During the work with this thesis, both the deductive and the inductive approaches has been used to some degree. Because of the fact that no existing theoretical model that could be applied to all the research questions was found, we decided to build a framework of our own where different aspects of integration benefits, open source, and price models are connected to value. These aspects have been picked from the sources that constitute the thesis' theoretical framework. This model-building phase can be considered inductive. In the next phase of the study the model is used in the case study to determine what information is needed from the interviewees and to analyse its results, which makes this phase more deductive. It is however not deductive in the way that we try to validate or denounce hypotheses created from the model, but rather in the sense that empirical data is gathered from interviews and then the model is used for making sense of those observations.

2.1.4 Workflow Overview

This study has followed a method framework called the Wahlbinian U (Lekvall, Wahlbin, & Frankelius, 2001). The framework is not intended to be an exact model of how a study progresses, but rather to describe the stages that are present in every study and how they are connected. The framework goes from the initial problem that should be handled, through problem clarification, planning, data gathering, analysis, and conclusions, to end with making some recommendations regarding the problem.

In Figure 2.1 below the workflow framework is described. The framework is iterative rather than a waterfall, so previous steps can be returned to after some later steps have been initiated. In our case, we went back to refine the theoretical framework, handled in the "Defining the study" step, both after having started the data gathering and during the analysis. We also stepped back and refined the study's purpose both when the study had been defined and during the analysis. When we saw that certain areas had a potentially low validity during the early analysis we reiterated and conducted the expert interviews, both to gather more information and to ensure the validity of the study.

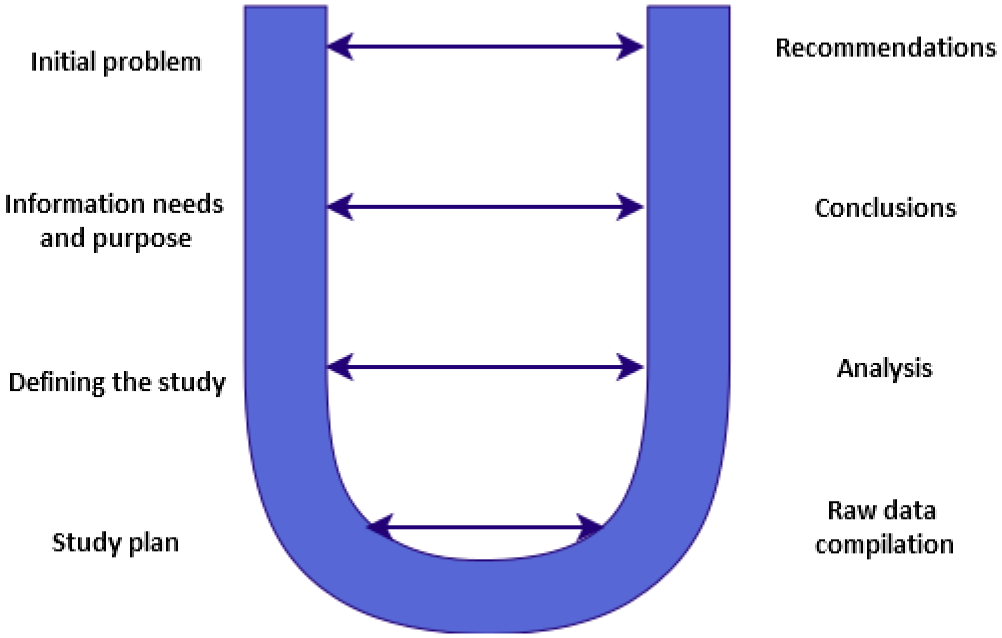

Figure 2.1 – The Wahlbinian U method framework, from Lekvall, Wahlbin & Frankelius (2001)

2.2 Theoretical Material

This section describes the methods used to gather literature for the theoretical framework. Both the process of gathering and the perceived quality of the results are highlighted.

2.2.1 Literature Study

The purpose of the literature study was both to gather material that could compose a theoretical framework to be used in the thesis and to give the authors a broad overview of the state of academic research in the area. The theoretical framework needed to cover a number of areas in order to be usable in analysing all aspects of the thesis' purpose and research questions. From initial discussions with Entiros, our supervisor, and our examiner, we decided that the areas that would be most relevant to the thesis were Application Integration, value theory and the value of IT, openness, open source, price models, and software pricing.

To gather initial material for the study, searches were made on the university library's *UniSearch* search engine and on *Google Scholar*. First the search was focused on finding articles on integration. Keywords used were "*integration*", "*systems integration*", "*application integration*", "*EAI*", and "*enterprise application integration*". A problem that arose was that searches including the term "*integration*" often gave results that were focused on other types of integration than IT, e.g. integration of immigrants into Swedish society. To avoid these kinds of results the more specific search terms, like "*application integration*", were used later on.

In selecting the literature for the framework, we first picked the articles with the most citations for each search term. Then we read the titles and abstracts and skimmed the articles to determine whether they were relevant for our study. Those that made it through the filtering process were read in their entirety and, if they were still considered relevant, were included in the thesis framework. After a basic understanding of AI was reached, further gathering continued with similar methods. Other keywords used were e.g. "*value of IT*", "*value of Application Integration*", "*value of open source*", "*benefits of open source*", "*pricing of IT*", "*software pricing*", "*price models and value*". When we got an understanding of important aspects of open source we used keywords in combination with "*open source*" to find articles with specific aspects like "*lock-in*", "*flexibility*", "*security*" and "*community*".

After a sizeable amount of material had been gathered in this fashion, a deeper search was made into the references of those articles and books that were most relevant to the purpose of the thesis. This was done in several iterations, to ensure that important articles and books in each field were not left out. Material was also picked from earlier experience in the areas and based on recommendations from examiner, supervisor, and opponents. By applying these methods a mix of older, well-cited sources and newer research was gathered. Since both authors have worked with Entiros previously, we see ourselves in a confident position to judge the coverage of relevant topics.

2.2.2 Source Criticism

An initial problem with the literature study was the difficulty in finding articles that had researched the value aspects or benefits gathered from integration solutions. There were difficulties in eliminating the search results that were entirely irrelevant to the purpose, which could mean that keywords were not properly formulated. Further, very little relevant research in the area of AI was found that had been written within the last 10 years, indicating that some other important keywords might have been missed out. To try to get around this problem we tuned the search engines to prioritise newer articles in later iterations of the search process, which led to some newer articles being found.

The theoretical framework is based on articles and books from many authors and is thus more likely to give a nuanced and up-to-date view. The framework itself does have a few problems though, mainly that certain passages lean quite heavily on just one or two sources. On the one hand this is due to a lack of sources in the area and on the other, it is due to time pressure. Another problem is that certain sections rely mainly on older sources, meaning there is a risk there have been new developments in the field that have been left out. This problem is linked to the fact that it was hard finding relevant newer sources for certain areas. There is also the problem that a large section of the theoretical framework is based on studies and articles regarding software in general, which may bring up factors of less relevance for a study focused on integration solutions. To mitigate these problems we conducted three expert interviews, with the aim of validating our framework. The interviews indicated that we had managed to get a good grasp of the topics and that there were no major deficiencies in any areas.

2.3 Empirical Material

This section will explain the choice of having an interview study to gather empirical material and how the study was thought out, prepared, and executed.

2.3.1 Interview Study

In the early phases of the thesis work the empirical material was intended to be gathered by sending out a survey. The goal of this would have been to gather quantitative information that could have been analysed statistically to provide answers to the purpose and research questions of the thesis. After discussing the matter with the thesis' examiner, the empirical approach was changed to semi-structured interviews instead. Interviews give a better chance of explaining complex or ambiguous aspects of the empirical findings, since the interviewees can be asked to provide explanations for answers and any misunderstandings of questions can be solved. The interviewee can also give immediate feedback on the questions asked and comment on any relevant areas that have been missed out.

By making the interviews structured, these positive aspects of interviews can be captured while an interview guide provides structure to the interview, to ensure nothing important is left out. In the semi-structured case, the interview guide is not necessarily followed to the letter; questions that for some reason are not interesting in a certain interview can be left out and questions and answers can be allowed to go outside the boundaries of the interview guide. Interviews of this kind, however, will provide qualitative rather than quantitative data, meaning the analysis will take on more subjective aspects as well. (Lekvall, Wahlbin, & Frankelius, 2001)

2.3.2 Interview Guide

Since the interviews where to be made in a semi-structured manner, an interview guide was created. The guide has five parts: introduction, benefits, openness, price model and wrap-up. The introduction contains general questions on the company and their integration solutions, while the benefits, openness and price model sections contains questions meant to provide empirical material to be analysed. Finally, the wrap-up is where the interviewee can comment on the interview.

The questions in the benefits section of the guide are based on Shang's and Seddon's framework for categorising benefits in ERP systems (Shang & Seddon, 2000; 2002), explained in section 0, and the benefits found by Themistocleous and Irani in their previous study of benefits of AI solutions (Themistocleous & Irani, 2001a), explained in part in section 0 and in full in Appendix II. The questions in the openness section ask if the interviewee's company considers open source software important and which, if any, of the openness aspects considered by many authors are truly value creating (Stallman, 2009; Computer Economics, 2005; AlMarzouq, Zheng, Rong, & Grover, 2005). In the price model section the goal has been to identify what kind of price model the interviewee's company pays by, as per the SBIFT framework (Iveroth, et al., 2013), and whether the price model itself could be considered to add value for them.

The guide was validated by testing it with Entiros. Their sales manager Johannes Bynke was interviewed on what integration needs companies in the retail sector usually had, to help build an idea of what answers we should get. He also participated in a test interview using the guide, to help ensure that the questions were easy to understand for someone from the business side of IT and that they asked for the things they were supposed to. The interview guide was also sent to Entiros Product and Marketing Manager Markus Weinhofer to further see that we asked questions that were interesting to the integration business. Finally, the guide was improved using input from the first few interviews as well, e.g. adding questions on satisfaction with current price model used.

2.3.3 Target Companies

In order to be able to draw some conclusions from the study, we decided to target a rather narrow segment of the integration market. After initial discussions with Entiros and studies of reports by the Swedish research company Radar Group (Radar Group, 2011), we decided to focus on medium-sized companies in the retail sector. Medium-sized means, in our case, between 250 and 3,000 MSEK, corresponding to 100 or more employees. This range was chosen on recommendation from Entiros, since smaller companies do not handle integration on a large enough scale, while larger companies each have their own ways of handling integration, making them less interesting to study. The retail sector was chosen since it was one of the sectors that were in the greatest need of integration solutions (Radar Group, 2011), has rather many companies in the right size category, and was also considered to be easier to approach for a study than e.g. the banking business, which is also transaction heavy and thus have a large need for integration. The banks are, however, much larger companies and their information more sensitive, which makes them less suitable for the study.

The companies for the study were found by searching the online database *Retriever Business*, which holds information on all Swedish companies. The search was narrowed to SNI (*"Svensk Näringsgrensindelning",* Swedish business sector division) code 47 ("*Detaljhandel förutom med motorfordon*", retail except sales of motor vehicles) and looked for all limited companies (Swedish *aktiebolag*) with more than 250 and less than 3,000 MSEK in revenue for 2012. The result was a list of 217 companies, which had to be further filtered to remove companies that were obviously not in full control of their IT administration or software purchases. For example there were many stores from the *ICA Maxi* chain on the list, which are franchisees to the main ICA group and are therefore not in full control of their IT. Besides such franchisees, we also removed car dealerships and companies that were more manufacturers than retailers, as estimated from their websites and annual reports. Car dealerships were excluded because they are not classified as retail in the SNI system and were only present because they had some store selling car-related articles, which did not by itself turn over more than 250 MSEK per year.

Remaining was a list of 107 companies, which were all contacted by e-mail and asked to participate. Initially companies were also contacted by phone, but because of great difficulties in reaching the CIOs at the companies this approach was dropped. One follow up e-mail was sent out two weeks later to those who did not answer the first one. In the end 60 of the companies never answered any of the mails. From those who did answer, only 6 agreed to partake in the interview study. Those who answered but did not wish to participate mostly cited lack of time, although some did not talk about these issues as a matter of policy. One problem with the study was that it came just before summer, when companies were most likely busy to get everything done before the vacations. The companies that participated were however quite diverse and so we got several different views of the topics of the study. Thus we feel that the explorative purpose of the thesis can still be fulfilled, even though there was a big loss from the target population.

2.3.4 Company Interviewees

Since the study aims to gather information regarding their integration solutions and the companies' views on open source aspects and price models, the interviewees need to be in some management position at the company. The management position is necessary find someone who both has insight into the company's IT infrastructure, priorities for their software procurement and can answer questions on how these are used on both the strategic and operational levels. Most of the theoretical material the framework is based around also takes a senior management perspective to the topics, e.g. the benefits framework (Shang & Seddon, 2000). We would obviously get more nuanced answers if we could talk to people at different positions, so ideally we would have been able to interview many people at the same company. However, no company gave us the opportunity to interview more than one person. Instead we interviewed the companies' CIOs, who according to Entiros and Thomas Rosenfall would most likely have a good mix of the technical and managerial perspectives. It is also reasonable to assume that the view of the CIO on these issues is rather dominant in the company.

2.3.5 Interview Techniques

The interviews were conducted as a mix of personal and telephone interviews, but the same techniques and questions were used for both formats. Most of the interviews were approximately one hour long, except for the one with Furniture Company, which was about two hours, and the one with Leisure Product Company, where we one got half an hour. Both of the thesis authors acted as interviewers together at all but one interview, where a scheduling misunderstanding made it impossible for both authors to be present.

One of the interviewers, the same for all interviews, had the main task of asking the questions and leading the interviewee through the three topics of the interview guide. The other interviewer handled taking notes and asking some follow-up questions that had been missed. All of the interviews were conducted using in a semi-structured format, based on the interview guide in Appendix I.

In general, a broad question for e.g. one of the benefit types was presented to the interviewee, whose answer was then written down. Depending on how well the answer covered the question we either asked one or more follow-up questions or moved on to the next area. What we were looking for was a broad covering of the question topic with some examples to provide nuance. Follow-up questions were often asked just to get examples and were often based around previous knowledge of how other companies handled things or things that had come up during discussions with Entiros.

The main difference between the types of interview was that during the personal interviews the interviewees took the initiative to a greater degree, while they were more passively answering the questions during the telephone interviews. They were also somewhat more accommodating with examples in the personal interviews, while we more often had to ask for examples in the telephone interviews. Because the interviewees had a more active role in the personal interviews, the examples they provided could sometimes be outside the scope of this study and more effort had to be put from the interviewers' side to get back on track.

2.3.6 Expert Interviews

We also conducted three expert interviews during the later stages of the study. These were intended to ensure the theoretical framework was valid with regards to the questions and purpose and to provide interesting angles and information for the analysis of the gathered material. The interviewees are described below:

- *Daniel Kindström*: Associate Professor in Industrial Marketing at Linköping University. Specialises in e-business, ICT and services and has previous experience from the consulting business. Interviewed mainly about value and price models.
- *Thomas Rosenfall*: Junior Lecturer in Industrial Marketing at Linköping University. Specialises in open source theory and also has several years of practical experience working in the area. Interviewed mainly about open source.
- *Johannes Bynke*: Sales Manager at Entiros. Has almost 20 years of experience from the IT business and has previously worked for, amongst others, Infor, Lawson and Aditro. Interviewed mainly about the benefits and potential value of integration.

These interviews were mostly unstructured, with a few broad topics that were to be discussed, which gave the interviewees free hands to provide whatever information they thought would be most useful and interesting. The interviewees were also asked to comment on strengths and weaknesses of the approach and theoretical material used in the thesis

2.4 Method Criticism

This section will go into greater depths in evaluating the validity and reliability of this report and its results.

2.4.1 Planned Generalizability

Since this study is intended as an exploratory case study of the value integration creates in the Swedish retail sector, the generalizability is not a primary focus. In the ideal case we would have gotten more companies to participate in the study, until we reached some kind of saturation, where adding more cases no longer brought up new issues or opinions. We would also ideally have had several interviewees at each company, with different positions, so we could make each case richer and with more perspectives. In such a case, the generalizability of the study would have been better than what was actually achieved, but still not that great compared to a much larger survey study of the topic, since there is no guarantee that the cases studied are a good reflection of the sector in general.

2.4.2 Validity

As has been mentioned previously, using a qualitative case study approach is likely to improve the validity of a study (Eisenhardt & Graebner, 2007; Lekvall, Wahlbin, & Frankelius, 2001). Since both of the authors have worked with Entiros before and are rather familiar with the basics surrounding integration, the base validity can be considered good. In order to ensure that the validity of the study remained high we also turned to experts, both at the university and at Entiros. For example, the expert interviews with Thomas Rosenfall and Johannes Bynke confirmed that CIOs were a good choice for the interviews. Entiros was consulted prior to the interview study, to validate that the interview guide asked the right questions.

The validity of the conclusions is decreased because of the fact that only one person per company was interviewed, which makes their subjective image of the situation more important. Making several interviews at each company could have mitigated this problem, but no company was interested in splitting the interview over more people. The validity was however kept reasonably high thanks to the position of the interviewees, meaning they likely have a relatively good insight in the study topics. Most interviewees also accepted follow-up questions after the interview via e-mail, so misunderstandings could be cleared up and further questions asked. Further, the results from the study have been considered as reasonable by the experts we have consulted, which indicates that the validity of the tools used has been sufficient.

2.4.3 Reliability

The reliability of this type of in-depth, qualitative case study is often rather poor (Lekvall, Wahlbin, & Frankelius, 2001). The questions we ask the interviewees would likely be answered slightly differently by other interviewees in the same situation, and possibly very differently depending on the perspective and position of the person interviewed. Experts at Entiros and Linköping University have been questioned about the study’s results and found them both reasonable and within the realm of the expected. This indicates that the reliability of the study is at least good enough in the sense that the results are not skewed far from what experts in the field would expect. Further, the findings are of a quite broad and general nature, which makes it easier to see if we had gotten strange answers.

3 Theoretical Framework

In this chapter the results from the literature study are presented. The framework is intended to both give the reader a basic understanding of the issues and topics to be analysed later on and to provide the theoretical basis for said analysis.

First the basics about application integration will be explained, followed by a review of value theory, especially in the IT case, and the value and benefits of AI reported in academic articles. Following this comes two sections on openness and pricing, handling these areas mainly from an integration perspective.

3.1 Integration

Since an understanding of integration and its characteristics will be necessary to grasp the value aspects of an integration system, the following section will aim to establish just what integration is in the context of this thesis.

3.1.1 Concerning Terminology

There are many terms used to describe integration of IT systems, some of which are presented in List 3.1 below:

- Application Integration (AI) (Sprott, 2000)
- Enterprise Application Integration (EAI) (Linthicum, 1999)
- Extended Business Integration (EBI) (Markus, 2000)
- Systems Integration (SI) (Hasselbring, 2000)
- Supply Chain Integration (SCI) (Linthicum, 1999)

List 3.1 – List of sample integration terminology

Each of these terms refers to different kinds of integration, often overlapping with other terms, though different academic authors can use the same term in different ways (Themistocleous & Irani, 2002). According to Sprott (2000), EAI used to be the general term for integration during the 90s, while the field was still evolving. Since then, the field has stabilised and AI has become the generic term for the field as a whole (Sprott, 2000).

Themistocleous and Irani (2002) propose a taxonomy for integration, where they use AI as the general term for all kinds of integration. AI is then categorised into intra-organisational, inter-organisational, and hybrid AI. Terms like EAI and SI refer to intra-organisational AI, i.e. integration of applications within the organisation, while SCI, EBI, and similar terms refer to B2B inter-organisational AI (Themistocleous & Irani, 2002). Hybrid AI refers to B2C AI, i.e. different kinds of integration with consumers, such as certain e-stores and online services.

As can be seen above, there is no clear consensus within the field regarding the exact usage of the different pieces of terminology surrounding integration. Nonetheless, in order to keep the terminology of this thesis more consistent across many sources, AI will be used as a general term for middleware-based integration of IT systems, as described by Sprott (2000) and Themistocleous & Irani (2002). If there is a need to be more specific about what kind of integration is being discussed the terms EAI, Hybrid AI, and EBI will be used to respectively describe intra-organisational, B2C, and inter-organisational AI. If a referenced author uses a term that differs from these descriptions; that term will be generalised into one of these unless there is some special reason to keep it as is.

3.1.2 Defining Application Integration

According to Hasselbring (2000) an organisational unit of an enterprise can be structured into three architectural layers: business architecture, application architecture, and technology architecture, as described in **Fel! Hittar inte referenskälla.**Figure 3.1 below. In this model business architecture is a combination of organisational structure and business process workflows, technology architecture is the IT and communication infrastructure of the organisation, and application architecture is the applications that realise the concepts from the business model and connects the business process models to the actual technological solutions (Hasselbring, 2000).

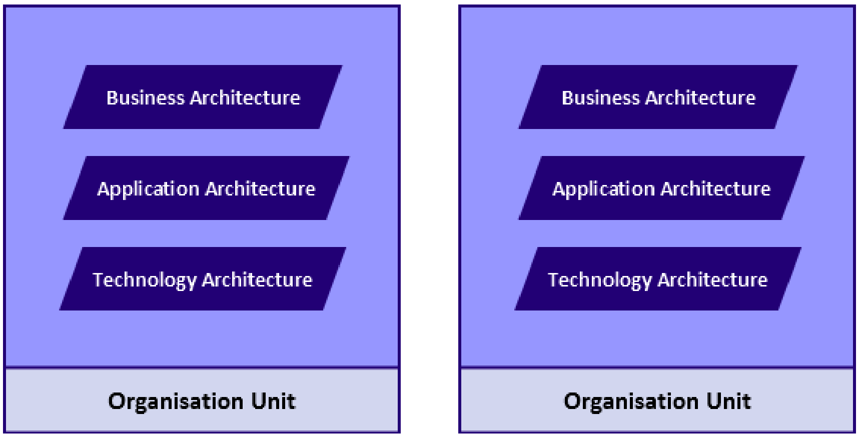

Figure 3.1 - Layered model of an organisational unit (Hasselbring, 2000)

Just looking at units like isolated pillars, in the manner of Figure 3.1, does however not reflect reality very well; real world units in an organisation do not operate in a vacuum, separate from each other, but are rather interacting and relying on each other in many different ways. This means that the three architectural layers of the different organisational units must be connected, integrated, to properly support the business processes of the units, as shown in Figure 3.2 below. According to this description, AI is just one layer in an organisation's process integration support structure, whose purpose is to connect applications and share data over an organisation's extant IT infrastructure. (Hasselbring, 2000)

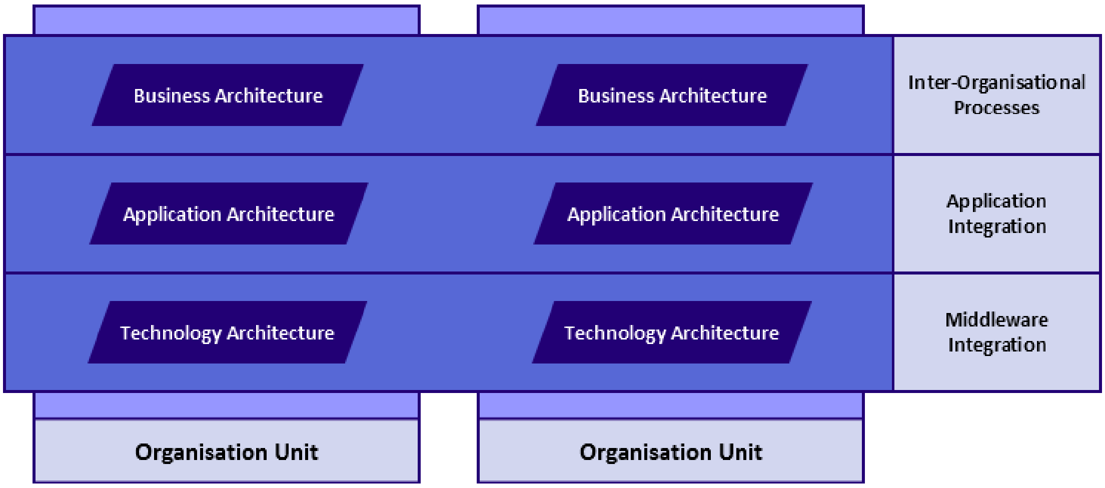

Figure 3.2 – Different layers of integration (Hasselbring, 2000)

David Linthicum defines EAI as follows:

> *... the unrestricted sharing of data and business processes among any connected applications and data sources in the enterprise. (Linthicum, 1999, p. 3)*

This description captures the same aspects of integration of data and business process sharing as Hasselbring (2000) mentions, but both of these definitions are centred around intra-organisational integration and thus feel too narrow to properly define what AI is in the broader meaning used in this thesis.

Another description of AI, with a broader scope, comes from Themistocleous and Irani (2002), who say that it has the goal of facilitating the integration of inter- and intra-organisational IT systems by bringing together functionality from different applications in a unified manner. Hasselbring's (2000) model can be adapted to encompass this broader view of AI, by realising that his architectural layer model of an organisational unit can be used to describe an entire organisation as well. To cover all three sub-categories of AI mentioned in Themistocleous' and Irani's (2002) taxonomy, Hasselbring's (2000) model can quite easily be expanded by considering companies around the organisational units in much the same way as the units are considered in the current model.

To be considered proper AI, an integration system has to achieve the goals of AI in a certain way. One older method of integrating IT systems was to build custom interconnections between each of the systems to be integrated, but maintaining these interconnections quickly becomes complex, since *x* applications will require *x*(x-1)/2* connections to be fully integrated (Themistocleous & Irani, 2001b) Interconnecting applications in this way is not considered AI by Themistocleous and Irani (2001b), since it is too inflexible and it has to be hard-coded into the applications in question. Proper AI is instead achieved via a middleware layer that handles communication and translation of data between connected applications (Themistocleous & Irani, 2001b; 2002; Linthicum, 1999; 2003; Hasselbring, 2000) The applications to be integrated are then connected to this middleware instead of each other, thus eliminating much of the need for code changes in the applications themselves (Themistocleous & Irani, 2002). One key difference compared to traditional middleware approaches to the integration problem is that AI focuses on integration of both business-level processes and data, instead of just focusing on data integration (Linthicum, 1999).

3.1.3 Integration Architectures

The integration middleware mentioned above is an implementation of the integration architecture that the organisation has made to allow its business processes to flow as intended. Since each organisation and trading community has its own set of integration issues and priorities to be solved, there is no universal technological solution that can apply to each and every one of these organisations' needs (Linthicum, 2003). Instead, each organisation must have an integration architecture that can handle its need. This integration architecture can be seen as the blueprint to a system that will be capable of implementing the organisation's intended business process model. There are, however, some architectural elements that are very similar across many organisations' architectures. Linthicum (2003) talks of four general categories of approaches to creating integration architectures, which are briefly explained below:

Information-Oriented

Of all approaches, this is the most typical in a traditional integration solution. Just as the name implies the approach means that information sources, i.e. databases are integrated to give applications access to larger data sets and thus more information. This approach has in turn three major categories, described in List 3. below. (Linthicum, 2003)

- **Data Replication**
  This category is the simplest form of integration, which involves simply moving data between two or more databases. The main problems to be handled by integration middleware are to account for differences between different database schemata. (Linthicum, 2003)

- **Data Federation**
  This category involves the integration of multiple databases, creating a single, virtual enterprise database that is the aggregation of multiple real, physical databases. Linthicum (2003) describes this category as the most elegant solution to the information-oriented application integration problem. (Linthicum, 2003)
- **Interface Processing**
  This category is focused on using application interfaces to integrate custom applications with packaged solutions. The abundance of packaged solutions, such as Enterprise Resource Planning (ERP) systems, made this category especially interesting. (Linthicum, 2003)

List 3.2 – Categories of information-oriented integration approaches (Linthicum, 2003)

Business Process Integration-Oriented

This kind of integration, often called Business Process Integration (BPI), involves managing the movement of data and correct execution of processes in other applications. By bringing together relevant processes from different applications more complex business processes in the organisation can be automated, increasing the value of the applications. (Linthicum, 2003)

Service-Oriented

This kind of integration builds on the SOA principles, meaning applications are able to access services in other applications to share business logic and methods. By using technologies such as Web Services, applications can communicate and work together in the integration architecture. Although this is most useful, due to the invasive nature of the integration it is also the most difficult to accomplish. (Linthicum, 2003)

Portal-Oriented

This kind of integration avoids the back-end integration problem entirely, by simply creating a common interface through which all necessary applications can be accessed, often based upon a web browser. By aggregating the user interfaces of the different applications, the users themselves do not have to switch between different applications to perform their tasks, but the applications themselves are not actually integrated. This means some other information-oriented integration is needed to avoid having to manually transfer data between systems. (Linthicum, 2003)

3.2 Value

There are many ways of looking at value and value creation, not least in such a broad area as Application Integration. After defining what is meant by value, this section will delve into topics interest, such as the value of IT and the benefits from integration.

3.2.1 Defining Value

In order to be able to analyse and discuss value, it is first necessary to define what we mean by the term. There is, however, no consensus among researchers of exactly what to include in the term, so there are many potential answers to the question “What is customer value?”. Below, we cite three different ways of looking at value:

> *“… perceived value is the maximum price the customer will pay.” (Dolan & Simon, 1996, p. 9)*

> *“Value equals quality relative to price.” (Gale, 1994, p. 29)*

> *“... value is the expression in monetary terms of what the customer firm receives in exchange for the price it pays for a market offering.” (Anderson, Kumar, & Narus, 2007, pp. 26-27)*

What the first two explanations have in common is that they see price as something that is intrinsically linked to value. Dolan and Simon (1996) say maximum price and value are one and the same, meaning an offering with a higher maximum price will be of greater value to its customers than one with a lower. This feels strange, since customers are happier paying a lower price. Gale (1994) defines value as the difference between the (perceived) quality and the price of an offering. This way of thinking also feels strange, since this means price is a kind of negative quality for an offering. Thinking of value as quality minus price also creates the problem of how to measure quality. Dolan’s and Simon’s (1996) maximum price approach, although lacking in certain regards, has the merit of being rather easily measureable. How to combine different measures of quality in an offering into a single value expression as per Gale’s approach seems problematic at best.

The third explanation of customer value is based mainly on the two earlier and compensates for many of the deficiencies they have. Anderson, Kumar, and Narus (2007) have been researching value propositions and found it necessary to create this explanation as their own way of defining customer value, since previous attempts have had issues, mainly concerning how to handle price and how to measure value. This definition of value is what will be used in the thesis.

One main benefit from using this definition is that it clearly separates the value and the price of an offering; the price of an offering does not alter its value. Instead, value and price considered jointly creates the attractiveness of the offering. Thus, if a customer is considering two offerings of exactly the same good or service that are priced at different points, both will have the same value, but the cheaper one will have a higher attractiveness. Value is also more clearly measured in this case; by calculating the money gained from cost savings, efficiency increases, and similar economic factors, the value of an offering is derived. This handles the issue of how to combine different quality measures, by instead calculating their impact on the customer’s bottom line, although this can be quite problematic in many cases. The main drawback is that this definition of value is very firmly grounded in a B2B world, but since this thesis concerns enterprise usage of integration, that will likely not cause any problems. It also cannot handle benefits that have no connection, direct or indirect, with some kind of monetary value for a company, but we expect such benefits to be a very minor part of an integration solution. (Anderson, Kumar, & Narus, 2007)

3.2.2 The Value of Information Technology

Davern and Kauffman (2000) talk of the value of IT in two terms: potential value and realised value. The distinction they make is that the potential value of an IT system is the theoretical maximum value that the system could deliver, if implemented and utilised perfectly, while the realised value is how much of the potential value that could be captured during the implementation. The factors that impact the amount of potential value that can be captured are called *conversion contingencies*, which include e.g. IT management, IT project expenditures, and organisational flexibility. In order to capture as much as possible of the potential value of an IT investment a company needs to be skilled at implementing and adopting systems; simply acquiring new IT systems will not provide the value the company hopes to gain. (Davern & Kauffman, 2000)

This can be related to Smith's and Nagle's four types of value: Value in use, Value in exchange, Perceived value, and Willingness to pay (Smith & Nagle, 2005). In this case, the value in use is the maximum value that a system can confer in an ideal situation, i.e. the potential value from Davern and Kauffman (2000). The value in exchange, on the other hand, is the maximum value that a certain company can gain from a system, which will be different depending on how the company and its situation differ from the ideal case. Value in exchange thus becomes the maximum value potential that can be realised by a company. The perceived value is the same as the actually realised value, i.e. how large a part of the potential maximum (value in exchange) that a company captures or believes itself capable of capturing. Finally, the willingness to pay is where a vendor has to put its price for the offering to be attractive to the customer. Previous case studies also indicate that there exists certain factors affecting the successful implementation of an AI solution, e.g. Lam (2005), who amongst other things mention how preparations affect implementation success.

While IT certainly can be valuable to some degree, the question is whether IT can be of strategic importance for a company or not. Carr (2003) argues that IT is a commodity, since everyone can get it, and thus is a hygiene factor rather than something strategic, in the sense that a company cannot gain a persistent competitive advantage from having it. His reasoning is that if a company does something to gain an advantage using IT, all their competitors can get the same technology and gain the same benefits, thus nullifying the first company's advantage. Other authors (Brown, Hagel, & Varian, 2003) argue that while IT certainly is a ubiquitous commodity, the skill and experience required to get the most out of a complex IT system is not. Simply having IT is never going to confer any competitive advantage, but being very skilled at applying and using IT systems can.

3.2.3 Benefits of Application Integration

In the expanded model of AI based on Hasselbring (2000), AI aids an organisation by helping integrate the application layer, which is the glue that connects the abstract process model with the existing IT infrastructure. There is, however, little mention in more specific terms of the benefits that an organisation can expect from an AI solution. The only benefit that Hasselbring (2000) mentions explicitly is being able to keep legacy systems longer and running them together with newer systems easier. Beyond this, there is no mention of any specific benefits an organisation can expect to gain from an AI system. Hasselbring does of course mention that having an AI system is beneficial, but does not specify how these benefits will manifest more specifically.

Linthicum (1999, p. 3) rather bluntly states that the value of AI should be obvious once the core concepts have been explained, but there is once again little explicit mentioning of the benefits an organisation can expect to gain. Overall, many authors, e.g. Linthicum (1999), Hasselbring (2000), and Sprott (2000), consider the value and benefits of AI as obvious, or at the very least easily understood. The potential value of the technology might be obvious, but as seen by Davern and Kauffman (2000) that does not mean every company can realise that value. The case is also that there might be some conditions that are necessary for a company to be able to even have the possibility of realising the full potential (Smith & Nagle, 2005).

In order to analyse the benefits from an AI system, it is necessary to have some framework to categorise them. Shang and Seddon (2000) have made a framework for classifying benefits of enterprise systems, based on benefits reported in success stories from three different ERP vendors. In order for a benefit to be included it had to be included in at least three cases from at least two vendors. In all the authors went through 470 cases, of which 233 were useable in the analysis. These benefits were classified into five dimensions based upon previous IT benefits research. (Shang & Seddon, 2000)

> An issue with the model is that there is quite the potential for overlap and correlations dimensions. For example it seems reasonable that there might be a connection between costs, an IT infrastructure benefit, and lower costs overall, an operational benefit. Further, organisational coordination, an organisational benefit, might lead to improved productivity, operational benefit. It could also be hard to e.g. differentiate between some resource benefits and the cost savings they might bring, making the line between managerial and benefits blurrier. The five dimensions, hereafter referred to as benefit types, are 3.3This type of benefits concerns improvements in internal coordination, process flows or improved abilities to outsource. An example of an organisational benefit in the AI case could be better support for business processes or improved organisational learning.

List 3. below.

1. **Operational**
   This type of benefits concerns direct cost savings and productivity increases. An example of an operational benefit in the AI case could be cost savings thanks to automation or improved productivity thanks to faster access to relevant data.
2. **Managerial**
   This type of benefits concerns improved allocation of resources and access to better data for decision-making. An example of a managerial benefit in the AI case could be to ability to gather more data for a Business Intelligence system or improved logistics planning.
3. **Strategic**
   This type of benefits concerns improved abilities to cooperate with others, grasp opportunities in the market, and improved strategic goal fulfilment and planning. An example of a strategic benefit in the AI case could be improved coordination with suppliers or access to new market channels.
4. **IT Infrastructure**
   This type of benefits concerns improvements in IT related areas, such as improved implementation times or lower IT costs. An example of an IT infrastructure benefit in the AI case could be a more flexible infrastructure or better IT systems visibility.
5. **Organisational**

This type of benefits concerns improvements in internal coordination, process flows or improved abilities to outsource. An example of an organisational benefit in the AI case could be better support for business processes or improved organisational learning.

List 3.3 – Framework for classifying benefits of IT systems (Shang & Seddon, 2000)

Themistocleous and Irani (2001a) have made a case study of AI benefits, including 15 companies using one of four different integration strategies. These strategies involved which kinds of systems were being integrated: custom systems only, custom and packaged applications, custom systems and e-business software, or custom, packaged, and e-business systems. In all, the study identifies 28 benefits that the case companies have gained from their AI systems. In **Fel! Hittar inte referenskälla.** below we have collected the benefits that at least 20 per-cent of the case companied gained, leaving out the other 19 benefits that were only present in one or two cases. Of the nine benefits in the table, three have been reported in more than half of the cases: *Reduces cost*, *Increases flexibility*, and *Provides flexible, maintainable, and manageable solutions*. That an AI system provides cost savings and increased flexibility in both the IT infrastructure of an organisation and the organisation itself matches what other authors say about AI well (Hasselbring, 2000; Linthicum, 1999; 2003).

That operational and IT infrastructure benefits are common in Table 3.1 below match the Shang and Seddon study on ERP systems (Shang & Seddon, 2002). Organisational benefits do, however, seem way more common than in that study. The inflexibility of ERP systems has been commented on in other articles, e.g. (Themistocleous & Irani, 2001b; Hasselbring, 2000), so this might be an indication that an AI solution is better at supporting organisations in this way than an ERP system is.

Table 3.1 – Main benefits from AI solution (Themistocleous & Irani, 2001a)

Benefit	Category	% of cases
Reduces cost	*Operational*	***53.3% (8/15)***
Achieves return on investment	*Managerial*	***33.3% (5/15)***
Allow organisations to do business more effectively	*Managerial*	***20.0% (3/15)***
Increases data analysis	*Managerial*	***20.0% (3/15)***
Achieves customer satisfaction	*Strategic*	***26.7% (4/15)***
Faster and cheaper implementations than bespoke solutions	*IT Infrastructure*	***20.0% (3/15)***
Results in reusable systems, components, and data	*IT Infrastructure*	***33.3% (5/15)***
Provides flexible, maintainable, and manageable solutions	*IT Infrastructure*	***66.7% (10/15)***
Increases flexibility	*Organisational*	***53.3% (8/15)***

Something that is surprising about the benefits reported is that there were relatively few mentions of strategic benefits. Strategic benefits are significantly more common in Shang's and Seddon's study of ERP system benefits (Shang & Seddon, 2002), perhaps also indicating a difference between ERP and AI systems. However, strategic benefits are by their nature intangible (Shang & Seddon, 2000), so the problem might lie in the identification of benefits hailing from the AI system in the Themistocleous and Irani case study (Themistocleous & Irani, 2001a). Another thing that is missing is the benefit of being able to keep legacy systems around for longer with more ease, as spoken of by e.g. (Hasselbring, 2000; Markus, 2000). Most likely this has been included in one of the other benefits, but since it is said to be very important it would have been interesting to see how many companies actually gained the benefits promised. For a full list of benefits from the case study, see Appendix II.

3.3 Openness

This section describes open source software and its potential benefits. It starts with the definition of open source and then looks into some of the aspects that open source bring to a software offering.

3.3.1 Open Source Software

When talking about openness in the software business, one term that is often brought up is open source. Today it is a widely used concept for developing and distributing software and the implementations range from complex operating systems like Linux to simple applications. Much of the literature regarding openness in software looks into open source solutions in particular instead of openness factors in general. Thus, we have chosen to use open source research as a base to describe openness in this study.

According to the Open Source Initiative (OSI), open source software can be described as:

> *Software that can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition. (The Open Source Initiative, 2014)*

The Open Source Definition is a set of criteria that is maintained by the OSI, acting as a standard body for the open source community. If you want to distribute your code as open source, the license of choice must comply with the criteria.

Not everyone in the communities of open/free software agrees with the criteria stated by the OSI. The term open source software itself can be too restrictive for some licensees, and the Free Software Foundation instead uses the term free software. By free they do not mean it in terms of price but that the license respect's the users' essential freedoms: the freedom to run, to study, to change and to redistribute the software. In practice, the open source criteria are a little weaker than the criteria for free software. All software licenses that are recognised as compatible with the free software criteria will be able to pass the open source criteria as well. Open source software is also free software in most cases but there are some exceptions, sometimes the open source license is too restrictive to qualify as free software. However, these kinds of licences are rarely used in programs. One example of projects that are free source but not open source is the GNU project. (Stallman, 2009)

3.3.2 Open Source and Integration

On the Swedish market there are many different types of integrators present, which deliver different kinds of integration solutions. The smaller ones are just consultants and resellers of systems developed by large software vendors while the bigger companies both develop and sell their own systems. There are both proprietary and open solutions available on the market today. Apart from Entiros there are other integrators who are partners with MuleSoft and sell their Mule ESB. One of these is Callista, which like Entiros also use the openness as part of their marketing. According to them, they base their integration solutions on open source code, which are both cost effective and prevent lock-in effects (Callista Enterprise, 2014). Another of the smaller players is Redpill Linpro, which also bases their offering on open source. According to their website, open source leads to better products and services, greater flexibility, lower costs and brings the freedom of choice to their customer (Redpill Linpro, 2014).

When looking at the larger players, which develop their own systems, wordings of open source or other big marketing words related to openness aspects are harder to find. Many that provide proprietary offerings seem to focus more on the benefits and functionality of their systems instead.

3.3.3 Benefits of Open Source

To get a better understanding of why open source software is used by some system integrators in Sweden , we will in this part of the chapter write about some of the factors that is often brought up as benefits of open source by its proponents and has been look upon in different studies. Apart from these factors, other potential benefits exists but are not as frequently mentioned in the literature, which is why we will concentrate on five benefit factors. The factors that we use in this study are *vendor/customer lock-in*, *cost*, *security*, *flexibility/modifiability* and *community support and maintenance*.

Lock-in

A concept that has been around for decades in many different markets and businesses is lock-in, where a service or product provider manages to create a situation where their customers would have a hard time changing to a competing solution. The term *switching costs* is one way of describing the barriers that build around these kinds of offers. If the cost for switching to another provider is high, the probability to actually make a switch gets low and the provider has successfully locked in their customer. In the IT industry the concepts of lock-in and switching cost get even more important than other sectors because of the rapid technological development, where new products, services and standards show up every year and the rules of the market are changing continuously. Decision makers have to make a lot of hard decisions concerning system upgrades because of these factors, decisions that often have great impact on the company's strategic options moving forward. (Shapiro & Varian, 1999)

In a survey by Computer Economics (2005) asking "*what is the most important advantage in the use of open source?*", the alternative *less dependence of vendors* was chosen as number one (44%). One reason for valuing greater independence from software vendors can be that the reduced flexibility when it comes to maintenance and service can add cost to the customers without adding any value. Software vendors can force their customers to upgrade their systems by phasing out older versions of the software and stop providing the old software with essential security updates and breaking compatibility. It is not always the customers feel that they need to upgrade their systems, as the new functionality is not necessary for the work they are using the systems for. One advantage with software that has an active and open community behind it is that even if the developers abandon a project, the community itself or a third party developer can continue to support the software if there is a demand on the market for it. (Computer Economics, 2005) The community aspect is not without its risks however. How well it handles task like continuing development is a question of the quality of the community and what types of members it has, argue AlMarzouq et al (2005). This view is explained more thoroughly in the section of Community Support and Maintenance.

Cost

One of the big trends in software pricing that is brought up by Cusumano (2007) is the fact that the initial license cost is becoming a less important source of revenue for the software vendors and many new price models are focusing on additional services and maintenance agreements. The rise of open source software is part of this development as many solutions are based on a low or zero initial license fee and alternative revenue streams. (Cusumano, 2007)

Supporters claim that low cost is one of the benefits of using open/free software solutions. The main argument is that the licensing fees are eliminated which reduces cost compared to buying proprietary software. Administrative overhead may also be reduced because of the lack of need of accounting for copies in use. The free contribution of the community to supporting the software, improving security and stability is also supposed to reduce costs for the users of the software. But if you consider TCO (Total Cost of Ownership), where total cost of training, acquiring and customizing is included, and it becomes less clear that a change from proprietary software to open source software is always worthwhile from a cost perspective. (AlMarzouq, Zheng, Rong, & Grover, 2005)

Open source software is sometimes available in two versions where vendors utilize a "dual-license" business model. One version is the community version that incorporates the ideas that are often heard about open source, like free of initial license fee, is free to use and rely on the community for support. The other version uses a model where the customer buys a licensee where the vendor provides access to their support team and/or extra functionality and extensions to the community version of the software. A majority of commercial open source solutions are based on this model and even if some large companies have development resources to support a community version of the software, many opt to buy the license to avoid bringing the risk of costly maintenance in-house. This model begs the question of how many of the ideas of open source that actually are left when a company buys these kinds of licences from software vendors. The difference between proprietary and open source solutions becomes less clear. (Brandel, 2010)

Security

One of the benefits of the open source software model according to its supporters is the increased security compared to proprietary software. In general many arguments circle around the assumption that if many people have access to the source code and read through it, the chance of finding bugs and security flaws increased compared to the situation where the source code is controlled by only a limited set of programmers. Eric S. Raymond, an advocate of the open source model, has summarized this in a statement called Linus law regarding software bug detection:

> *"Given enough eyeballs, all bugs are shallow."*
>
> *Raymond (2002), page Release Early, Release Often)*

Hoepman and Jacobs (2007) discusses the security aspects of open and proprietary from the perspective of Kirchhoff's principle, often cited in cryptology and states that a cryptosystem should be secure even if everything about the system is known, except the key. They argue that for the same reason as security through obscurity can be considered bad practice in the field of cryptology, using that argument when debating source code is equally bad. (Hoepman & Jacobs, 2007)

Even if the characteristics of open source software can be used to enhance security, individual software project can take very different approaches to security. While many free or open source projects do a good job in addressing security issues by working with them actively and have software security as a main focus, others do not. It can also be very hard to actually determine the level of security of a software project even if the source code is available, and this problem has resulted in a situation where much flawed code is in use. (Messmer, 2013) There is also the question of whether there actually exists a difference between proprietary and open source software in terms of security. In an empirical study of 17 well-known and widely deployed software packages they did not find any evidence that the particular type of software development is the primary driver of security. The study also indicates that the thing that actually matters when it comes to patching behaviour is the security policy of the individual vendor, which does not have to be connected to a particular style of software development. (Schryen, 2011)

Flexibility/Modifiability

The word "open" in open source is not just a term for describing a license model where the source code is available for reading; it also means that programmers should be able to do more things with the code. This is visualised by the open source definition itself. If software shall be able to have an open source compatible license model, it has to show the source code in a format that makes it easy for programmers to make modifications. The licensee also must allow for the code to be modified and allow for the modified code to be redistributed on similar terms as the original software. (The Open Source Initiative, 2014)

The open nature of the license model that encourages modification may also be put into use for increasing software flexibility. Flexibility can mean different things and two different kinds of flexibility related to open source is *flexibility of use* and *flexibility allowed by licenses.* The first kind is more of a technical benefit while the latter is a business benefit. The flexibility of use is beneficial because it makes it easier to change, customize and experiment with the software and allows freedom of choice. The flexibility allowed by licenses can have a significant impact of reducing capital spending in a company. However, there are also drawbacks of open source that can reduce the benefits brought by flexibility. On the technical side, a vast number of interfaces, lack of expertise and strategic roadmaps may reduce the willingness to adopt open source solutions. The business drawbacks are related to finding staff and developing competences as well as the possibility of lack of ownership and support. (Morgan & Finnegan, 2007)

Community support and maintenance

Many of the alleged benefits of open source are derived from the community that can surround open source projects. Just like in the case of software security were Linus Law (Raymond, 2002) says that many eyeballs are more likely to find security flaws, this way of thinking can also be applied to maintaining and support software projects.

An open source community consists of different kinds of members, the *core* that is the smallest group but the one that is responsible for the majority of the code, *co-developers* that support the core in the development and the *users,* who can be both active and passive. The active users contribute to the project with bug reports and ideas while the passive users just use the software as free riders. The quality of the software is determined on the motivation of the members in the community to contribute with development and support. A small community with a high degree of core and co-developers may create a healthy community while a big community with just idle users will not. The potential of using the community as a source of support while maintaining the software is big as the whole community is a potential helper. There are risks involved, however, as a low quality community would be of poor help when supporting the software. When using support services from software vendors there exist a clear role of responsibility for supporting the customer when needed, something that is not guaranteed when using a community. It is of great importance to evaluate the community of a software project before relying on it for support. (AlMarzouq, Zheng, Rong, & Grover, 2005)

3.4 Pricing

This section provides information about the characteristics of software pricing. It brings up some pricing theory in general, the unique aspects of IT, how software pricing is changing and different examples of price models from the software industry. It ends with the SBIFT model that is used in this study to identify current price models and to describe in what dimensions problems exist.

3.4.1 Pricing and IT

The pricing of a product or service offering has always been crucial for reaching the targeted customer. The price level is of great importance in theories in the field of marketing strategy, like in *Generic Strategies* from the book Competitive Strategy (Porter, 1980) where the price of your offering will depend on your choice of strategy to reach a competitive advantage. If you try reach a cost leadership position on the market, you have to set your price lower than your competition to create a competitive advantage or to set the price at the same level as your competitors but getting better profit margins by a superior cost structure. The other type of strategy is where you try to take a higher price for your offering through differentiation. In this case, the price level is still important for the profit margin but is not the central aspect in creating an attractive offer for the customer. Instead, creating an offering that delivers superior value becomes imperative, to give customers a reason to pay your higher price.

Another tool often used in marketing where price is an important factor is the marketing mix, often described with the 4 Ps, product, price, promotion and place (McCarthy & Perreault, 1991). The 4 Ps represent variables that are used to satisfy a target customer. The product represents the physical goods, a service or a blend of both that is "right" for the targeted market. The price refers to the price of the product and it does not have to be just a monetary value. Promotion is how the marketer channels information about the product to the different parties and place is where in the market the customer gets in touch with the product.

Even if the cost structure of IT goods is special in many ways, the market for IT does not differ that much from other markets and the mechanisms that Porter (1980) states still apply. Both of Porter's (1980) generic strategies, differentiation and cost leadership, are used by companies struggling for dominance on the market. But even if classic strategies can be reused, the unique properties of IT become important when it comes to pricing. IT can be used to personalize both the product itself and the pricing depending of the customer's needs and ability to pay for them. It also leaves implications on marketing as in the case with *experience goods*. Information goods have to be experienced for the customer to see the worth of it, and one way of promoting software can be to simple give away samples of the software. (Shapiro & Varian, 1999)

3.4.2 Examples of Software Pricing

Even if IT has some unique characteristics that can create new kinds of pricing of products and services compared to other markets and technology, the pricing of software is not something that was set in stone when the first IT investments were made decades ago. IT is still evolving at a rapid pace and with the changes in technology come new business models and new ways of pricing software to mirror new usage patterns and functionality.

Cusumano (2007) describes three ways that software vendors use to create revenue. The first one is an up-front license fee that the customer has to pay for the perpetual right to use that version of the software. The second one that software companies often sell to their corporate customers is a separate maintenance agreement, which often consists of an annual fee that the consumer pays to get access to patches and updates to the product. The third type of revenue comes from services required to install and integrate the software, train users and customize the software for specific customer needs. This type is the most commonly used among integration consultants. In the early nineties, the up-front license fee was the most important source of revenue for software vendors but since then, maintenance and other services have overtaken that role. Today, it is not uncommon that a software vendor uses a revenue model where one third comes from license fees, one third from maintenance and one third from services. This shift in software pricing can at least partly be explained by the fact that it is getting harder to have large initial fees when competing solutions are pushing this types of fees down to zero, like different open source solutions. (Cusumano, 2007) For much software, initial license costs and annual fees are just a fraction of the total cost of the investment. The big part of the IT investment consists of the cost for support and administration of the system. Even if the initial investment is zero, the total cost will be determined by how much resources the company has to invest in administrative tools, training and people with the right skills to maintain the system. (Computer Economics, 2005)

In response to a declining interest in paying for software by initial license fees, Cusumano (2007) brings up some examples of models that try to answer this problem. One type of model that has been adopted is a term or subscription license, where the customer buys the right to use the software for a predetermined period and pays in instalments, like a monthly fee. When the period has ended, the user has to stop using the software or renew the license for an extended period. For enterprise software applications it may be hard to change into another vendor when the period has ended because of the large investments that would require, thus most customers renew their licenses. The big value impact for this kind of model is that it breaks up the big initial costs into monthly payments rather than increasing the flexibility in how the company uses software. If the vendor bundles together maintenance with a short-term usage license and hosting, you get a model that can be described as Software-as-a-Service (SaaS). This reduces the need for complex installation and integration processes for the customer and gives complete control to the vendor that can add training programs and software customization to the offer. A third take on the term-based models is to pay for actual usage of the software and base the fee on some metric. (Cusumano, 2007)

Another aspect of pricing software is the assessment base of the price model. This can be divided into usage-dependent and usage independent bases. With a usage dependent assessment base the price will vary with the usage. Some examples of these kinds of bases are paying per transaction, memory requirements or time. In such a model the software vendor must do some kind of monitoring to set the price. Examples of the usage-independent bases are when the software vendor pays a fee per named user, concurrent user, machine, locations, produced amount or key performance indicators. (Lehmann & Buxmann, 2009)

3.4.3 Pricing of Integration

The pricing of integration solutions is of course different from consumer and less complex business software. To implement platform solutions, create new integration points and make the software fit internal and external processes of the customers; high initial implementation cost will always be an issue. For example Lam (2005) points out that there will always exist some need for custom coding when integrating software, which will add costs to the project. The implantation costs can be of different nature, either the customer has to pay for consulting hours or use their internal IT resources.

When the implementation is done however, there is room for different kinds of price models for the different systems connected into an AI solution. Some functionality may benefit from the new SaaS and cloud models. Many companies in the integration business have started to offer cloud solutions.

Small players like Entiros offer cloud solutions to provide services like maintenance at a distance and big players like Infor, which provides cloud services to their ERP system M3. With new service models, pricing aspects as subscription and volume based pricing become relevant.

3.4.4 Problems with Price Modelling

Software can take many forms and has vastly different use cases and functionality, and thus it is unlikely to find a price model that fits all software. According to Bontis and Chung (2000), because of the fact that there is no generalizable formula for valuation of intellectual capital, which act as a base for all software development, there is no generalizable model for pricing software. In a case study of three companies with different price models, they found support to the proposition that software pricing is a complex and subjective process. They did however find some common characteristics between the different models and propose a process flow for pricing software. The process of pricing should consider both the business objectives of the vendor and the buyer's value realization. The vendor's business objectives can be to grow the user base, maximize revenue, gaining market share or something else. The buyer's value realization is based on how well the provided software and the usage of it offer a solution to problems in the buyer's business processes. It is important to design the pricing structure based on variables that the buyer will use while measuring value, as the source of these variables is the nature of the buyer's business. To find the right variables, the vendor has to understand its customer and that is the starting point for all price modelling. (Bontis & Chung, 2000)

In a study by Mathew and Nair (2010) where they look at the pricing of the SaaS model, they find that the perspectives of service providers and clients when it comes to pricing can be very different. When the software being distributed increases in complexity, the costs of delivery, maintenance, support, manpower, and resource appropriations also increase for the software provider, which has to increase the price to the customer. However, the study does not find a relationship between IT requirements and willingness to pay for the service that may result in a situation where the software provider adds a lot of costs to itself without visualising the added value to the customer to motivate a higher price compared to a simpler service delivery. The authors suggest that this may be a problem that arises because of lack of knowledge of SaaS among the customers, who fail to see the benefits of this model compared to traditional software delivery. (Mathew & Nair, 2010)

Iveroth et al. (2013) write about the strategic importance of pricing, and that the price model itself can be used as a tool to create differentiation in the market. Strategies should fit both external factors from the competitive arena, like opportunities and threats, and internal factors, like the strengths and weaknesses of the organisation. The balancing of the external and internal factors is an on-going process and one of the tools for managing this is to know how pricing and pricing conditions are expressed. In short it is of great importance that the price model match the risk sharing between seller and buyer. In contrast to other parts of the marketing mix, pricing provides the opportunity to create immediate effects on the offering without any significant expenditure. But at the same time, using pricing as a tool can be very difficult. (Iveroth, et al., 2013)

3.4.5 The SBIFT Model

Iveroth et al. (2013) propose a tool that can be used when working with price modelling, the SBIFT model. Price models can be described as systems of price-related aspects of the agreement between a seller and a buyer. Any agreement between a seller and buyer translates into a particular price model and the SBIFT model consists of five dimensions that describe all price models. In this study the model has been used both for describing how the pricing of integration solutions in the case companies work and for analysing in what dimensions possible problems exist in the current models, from a customer perspective. By making changes in the problematic dimensions, the perceived value of the offering may change. The five dimensions are:

Scope

The first dimension describes the scope of the offer. The two extremes of the scale that represents the dimension are *package* and *attribute*. A complete package means an offer that consists of a whole set of products and services that are priced together. Attribute means that the offer is split into small components and individual products and services are priced separately. (Iveroth, et al., 2013)

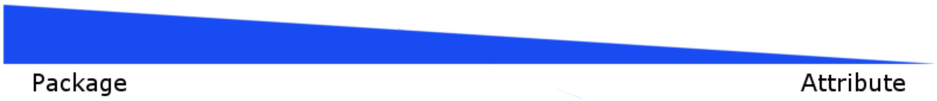

Figure 3.3 – Scope SBIFT Dimension

Base

The second dimension describes the information base of the offer. The classical base of pricing that is still widely used and accepted by both sellers and buyers is *cost*. The costs of developing, producing, distributing and selling products and services are often an easy way to define a price floor. One alternative to cost based pricing is setting the price by *competitor's price*. That means that the price is set with the price level of comparable products and services provided by competitors as a base, in other terms a market based price. The last base for the offering is *customer value*. The price level is defined as the balance between customers' perceptions of what they obtain and what they sacrifice in order to acquire or use it. (Iveroth, et al., 2013)

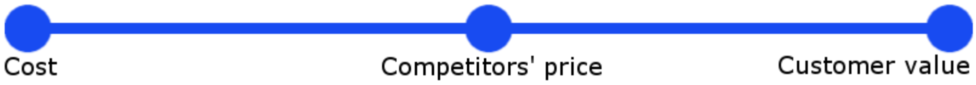

Figure 3.4 – Base SBIFT Dimension

Influence

The third dimension concerns the extent to which the seller or the buyer can influence the price, describing the power balance between the negotiators. The most extreme situation from which the seller has a powerful position in the price negotiation is the *pricelist*, where the seller can set the price of the product or service according to its own criteria and the buyer can either accept the price or refrain from taking the offer. The next situation is called *negotiation* and describes a more even situation for the two parties, and act as a price list where prices are negotiable to some extent. The next type of influence is *result-base price*, where the price is set based on the result of the use of the product/service. In *pay-what-you-want pricing*, the buyer gets the opportunity to translate their value of using the product/service in monetary terms by him/herself. The next type is to let both the seller and different buyers determine the price through an *auction*. The last type of influence is *exogenous pricing* which describes a situation where neither the seller nor the buyer can influence the price by himself or herself; the price is instead determined by an outside factor like an index. (Iveroth, et al., 2013)

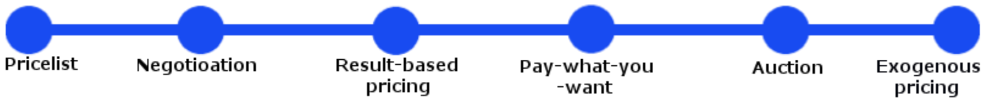

Figure 3.5 – Influence SBIFT Dimension

Formula

The formula dimension of the SBIFT model takes up the aspect of connecting price with the volume of usage/consumption. At one end of the axis is *fixed price regardless of volume,* where the amount paid is always the same regardless of the usage of the product/service. The next thing on the formula axis is *fixed fee plus per unit rate* that combines a fixed price component with a variable one that depends on quantity. In the middle of the axis is *assured purchase volume plus per unit rate*, where the seller is guaranteed to deliver a certain volume for a fixed price, even if the buyer does not use it. With *per unit rate with a ceiling* means that the seller is pricing the product or service per unit but after a certain quantity of units the seller is not charging for the units above the ceiling. The last one on the axis, *per unit price*, is where the buyer simply pays per unit. (Iveroth, et al., 2013)

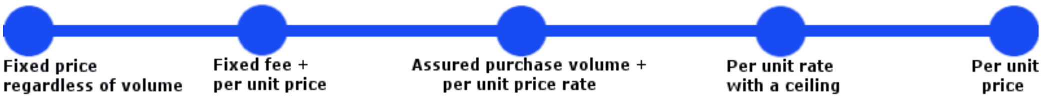

Figure 3.6 – Formula SBIFT Dimension

Temporal Rights

The fifth dimension is about the right to use the offer. On one side of the axis is the *perpetual* temporal rights, where the buyer has access to the offering forever. The buyer only buys the right to the current offer and not to future versions. The second type of temporal rights is *leasing* and has both a perpetual and a time limited component. After the leasing period, the buyer has the right to buy the offer for a predetermined price and use that version forever. The next type is called *rent* where the buyer buys the right to use a specific offer for a period of time and after the period has ended, the buyer has the return it to the seller. The fourth type, *subscription* is similar to rents but also includes the right to use new enhanced versions and future updates to the offer that arrive within the agreed period. The final type of temporal rights is *pay per use,* which describes the situation when a buyer pays for an offer at the time of its consumption. (Iveroth, et al., 2013)

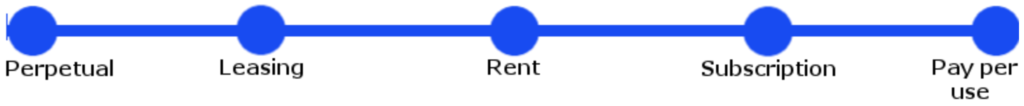

Figure 3.7 – Temporal Right SBIFT Dimension

4 Analytical Model

In this chapter the different parts of the theoretical framework will be condensed into an analytical model, which will form the basis for the analysis and conclusions. The model will have three parts: benefits, openness, and price models, which will be explained in turn. Finally, the model will be explained in its entirety.

4.1 Value Creation Model

This section will explain how the different parts of the theoretical framework relate to the definition of value.

4.1.1 Integration Benefits

With the definition of value from Anderson, Kumar, and Narus (2007) presented in section 3.2.1 of this thesis, the value gained from having an integration solution is based around how much money can be saved and earned from the benefits the system confers. When using Shang's and Seddon's (2000) framework for benefits from enterprise systems, benefits are grouped into five broad categories: organisational, managerial, strategic, IT infrastructure, and operational benefits. Below, each of the five categories is related to what kind of value it can bring to a company, with an extra focus on the kinds of value such benefits from an integration solution might bring:

- **Operational**
  These kinds of benefits are based around cost savings from automation and productivity improvements. Examples of value gained from such benefits could be reduced administrative costs from automating certain tasks or reducing the need for duplicate work, or increased earnings thanks to staff being more productive.
- **Managerial**
  These kinds of benefits are based around improved allocation of resources, better information to support decisions, and better organisational performance. Examples of value gained from such benefits could be cost savings from more efficient logistics, or improved earnings stemming from decisions made possible by better information.
- **Strategic**
  These kinds of benefits are based around supporting strategic decisions, improving the ability to cooperate with others, and being able to make new offerings or accessing new channels. Examples of value gained from such benefits could be reduced costs from better cooperation with suppliers or improved earnings thanks to new channels to customers.
- **IT Infrastructure**
  These kinds of benefits are based around improved flexibility and capabilities in the IT infrastructure and reduced IT costs. Examples of value gained from such benefits could be cost savings from a more efficient IT department or cost savings thanks to faster implementations of new software.
- **Organisational**
  These kinds of benefits are based around improved process overview, better internal cooperation, and improving work processes. Examples of value gained from such benefits could be cost savings from more efficient processes or reduced administrative costs thanks to a more transparent organisational structure.

List 4.1 – Value from Application Integration benefits, from Shang and Seddon (2000).

In the end, the value that a company can gain from an integration solution is limited by its *conversion contingencies* (Davern & Kauffman, 2000), factors affecting the capture of potential value from the system. To capture as large a part of the potential value of the system as possible, the company needs to assess both the system's capabilities and their own position thoroughly, so that they can gain as many of the system's potential benefits as reasonable. Enjoying a high value from an integration solution thus boils down to being well prepared for the system and working hard to be able to gain as many benefits as reasonable from the system.

4.1.2 Openness

Openness is a term that is used by some integrators to promote their software by pointing out different benefits that come with open solutions. If these benefits actually are something that customers care about and make software solutions more valuable from their perspective is however unclear. The openness benefits that are promoted are often related to benefits of open source software in general, and to make an analysis of the value of openness in integration solutions more structured, some commonly used open source benefits will act as a base. The analysis will take a look at in what way the benefits are present in the different case companies and then how the individual factors can connect to the Anderson, Kumar and Narus (2007) definition of value. The open source benefits of interest are:

- **Lock-in**
  One of the most commonly claimed benefits of open source software is the potential of reducing vendor/customer lock-in. Shapiro and Varian (1999) talks about how an ever changing IT environment enhances the problems with vendor lock-in and high switching costs. Using solutions that mitigates the risks could be valuable for customers. This is also reflected in a survey by Computer Economics (2005) where lock-in ranks higher than for example direct cost savings from low or free software licenses. To use open source software to decrease the risk of vendor lock-in may be perceived as valuable to the customer.
- **Cost**
  Another potential benefit of using open source software is cost savings. Cusumano (2007) claims that one important driver for changing price models with low or zero initial license cost is the rise of open source. By implementing the benefits of the community surrounding the open source project, further cost saving may be possible. (Brandel, 2010) If open source software can reduce the costs for the company, this aspect would connect to our definition of value.
- **Security**
  By using open source software, one benefit could be enhanced security. Keeping the source code open prevents security through obscurity (Hoepman & Jacobs, 2007). There are however big differences regarding security among open source projects. (Messmer, 2013) If open source software can increase the security in a way that reduces the risk of costly security flaws for the companies, this factor may be valuable.
- **Flexibility/Modifiability**
  The open source license model promotes modifiability of the source code (The Open Source Initiative, 2014), which can be beneficial in cases where standard solutions are not enough for a customer with special needs. Open source may increase both the flexibility of use and the flexibility allowed by licenses (Morgan & Finnegan, 2007). If the software can be made to fit the companies’ internal processes and workflow in a cost effective way, this factor may be of value for the case companies.
- **Community support and maintenance**
  The last benefit of open source offering is the aspect of the community. By having a community surrounding the software project, that community can be used for support and maintenance of the software, removing the need for expensive support and service agreements from software vendors. However, as AlMarzouq et al. (2005) bring up, the quality of the community will determine the quality of the support. This aspect may be valuable to the case companies if the costs for support and maintenance can be reduced with the help from communities.

List 4.2 – Potential Value of Openness Aspects

4.1.3 Price Model

The third aspect that this study uses to look at the value of an integration offering is the price model. Pricing in general is an important area when creating competitive offerings and finding the right segments of doing business, and is a central part in both strategic theory (Porter, 1980) and in marketing (McCarthy & Perreault, 1991). The price level by itself is essential for how attractive an offering is but some argue that how you pay also can be a way of differentiating from the competition (Iveroth, et al., 2013). This makes the concept of price models interesting and one part of this study is to explore whether the payment for a certain offering may by itself create value.

IT has some special characteristics that make pricing different from other types of services and goods. One aspect mentioned by Shapiro and Varian (1999) is the vast opportunities of personalised pricing IT creates. Different kinds of pricing work for different customers and price models can be very different depending on the service. The pricing of software is changing and different vendors test new models. Cusumano (2007) brings up how the landscape of software price models is changing, moving away from initial license fees and the rise of new types of offerings like SaaS cloud solutions.

To explore the potential value of the price model in this study we need a way of defining the price models and how customers of integration solutions see different aspects of the models. To do this in a structured way, we have used the SBIFT model (Iveroth, et al., 2013) to identify the types of price models in use in the case companies. The dimensions are:

- **Scope**
  The first dimension looks at the scope of the offer, is it priced as a whole package or is the offer split into smaller components that are priced separate?
- **Base**
  The second dimension looks at the information base of the offer. Is the price based on the cost of creating the offer, the competitor's prices or the more vaguely defined customer value?
- **Influence**
  The third dimension looks at the influence of the seller and buyer. Is it an uneven situation where one of the parties can influence the price more or are they equal in how much or how little they can do?
- **Formula**
  The fourth dimension looks at the price formula—on what basis the price of the offering is set. Do you pay a fixed price regardless of volume or with a variable component in the formula that increases with volume?
- **Temporal Rights**
  The fifth dimension looks at the temporal rights of the offer—for how long you can use the offering. Is it unlimited access, is it full access for a specific period of time or are all rights revoked at the time of consumption?

List 4.3 – The SBIFT Model (Iveroth, et al., 2013)

After identifying the price models in use, the SBIFT model dimensions will be used to identify where in the current models that problems exist and in what way the interviewees suggest how the dimensions should change to create a better fit for the companies. These changes will then be related to price models in use and how changes in current models could contribute to an increase in value of the type defined by Anderson, Kumar & Narus (2007).

4.1.4 Final Model

This analytical model is based on the three areas previously mentioned: benefits, openness aspects, and price models. The idea from the theoretical standpoint is that all of these factors contribute to the customer value of an integration offering. By comparing the data from the case study with what the theory and previous case studies say it will become apparent what relations to customer value the different parts of the model has. By analysing which parts of the models that confer more value and which part that confer less, it will be possible to form some interesting ideas regarding the nature of value creation in an Application Integration solution.

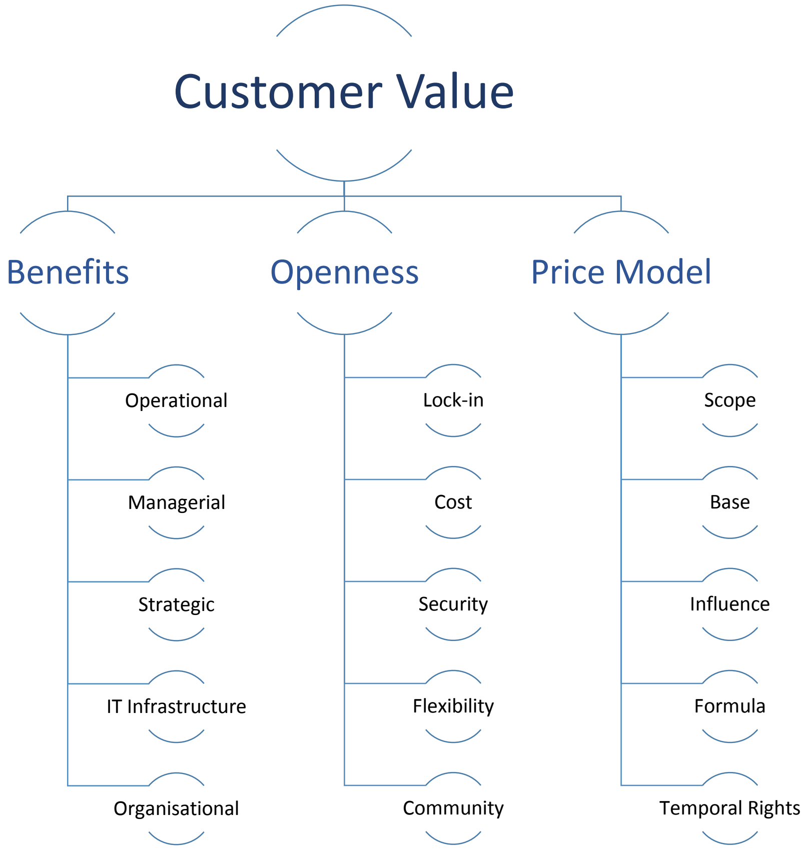

Figure 4.4 – Value creation model

5 Empirical Data

In this chapter the empirical foundation of the thesis work will be presented. The first section will provide background information regarding the companies that have participated in the study. Do note that all companies have been anonymised. Then the information for the areas of benefits, openness, and price models will be summarised. Finally, the relevant information from the expert interviews is presented.

5.1 Background Information

Here we present the necessary background information regarding the case companies that were studied. Do note that all companies in the study have been anonymised.

Office Supply Company

Office Supply Company sells office products and services. They have a revenue in the middle range of the study. They sell their products through several different channels: physical stores, directly to the customer via web-based store or through service deals where equipment is delivered to a locker at one of the customer's sites. The Swedish company that was interviewed is a subsidiary in an international group. They are using Infor's M3 Enterprise Collaborator (MEC) integration platform together with an internally developed platform to handle their integration needs. The internally developed platform is quite old and Office Supply Company is in the process of evaluating alternatives to replace it. They have tens of internal integration points to different systems and hundreds of external integrations points, mainly to corporate customers. Integration is not a post in their IT budget and the interviewee did not know how large a fraction was made up by integration costs. Some of the IT has been outsourced and much of the IT costs are attributable to support costs from consultants.

Industry Tools Company

Industry Tools Company sells professional tools, machines, and clothing, and provides services like education, service and product refilling to their customers. They have a revenue in the middle range of the study and sell their products and services through physical stores in Sweden and through their web shop. They use Infor's MEC to interconnect different internal ERP systems. In addition to the internal systems they also integrate with external service providers, e.g. Strålfors, which handles invoices. The company has made modifications in many of their systems, so they can get data in the right formats for the M3 engine. Currently they integrate seven major internal systems and many others externally, though the interviewee did not know the precise number of external integration points. The interviewee did not have an exact number for how much of the IT budget that is spent on integration, but thought it could be around 10%, depending on what to count as integration costs. For customers of a certain size they provide a special service of creating integration points manually, to make the standard system work and get the right data into the MEC.

Clothing and Design Company

Clothing and Design Company sells clothes, fashion accessories, and articles for interior decoration to consumers. They have a revenue in the middle range of the study and sell their articles through physical stores in Sweden, Norway and Finland as well as a web shop. The offer a wide range of products in many potential configurations, making their number of articles very large. Handling optimization of logistics, refilling levels of store stocks, and sales planning are very transaction heavy work, putting great pressure on their integrations. To handle all these tasks, the company relies on many different IT systems, approximately 20 big systems and several additional smaller ones, all of which are integrated in some way. The heart of the system setup is the ERP system Oracle Retail, which contains the integration module that is used to make connections between the different systems. Many of the transactions are processed overnight by FTP servers. The interviewee considers their transaction system architecture appropriate to deal with heavy workloads, concurrent processing, and assuring data consistency. Much of the IT infrastructure is outsourced to Asia and the IT work in Sweden is done on a higher abstraction level, focused on strategic rather than operational considerations.

Furniture Company

Furniture Company sells furniture and home decorations through physical stores and a web shop. They have a revenue in the upper range of the study. The stores are a mix of owned stores and franchise stores. Furniture Company's headquarters provides concepts to the different stores and delivers services like selection of products, marketing, logistics, accounting, and IT. The company uses a heavily modified version of Infor M3 as the transaction engine for their integrations. A few years ago they started an evaluation process for buying a new integration platform, which was quite time-consuming because of the complex organisational and legal structures of the company. These complex structures posed significant challenges for the available solutions on the market; so several different software vendors were contacted to submit their offers for a new system that would fit the company's needs. Because of the size of the company and its special niche position and structure, the software vendors were interested in having Furniture Company as a use case to reach new customer segments. This rendered the Furniture Company's position in negotiating rather strong. Now, the company has chosen a system and has begun implementing it, although the process is not yet complete. One aspect that adds to the complexity of the IT systems is that the company relies on many small suppliers for their products. These small companies often have a very low level of IT knowledge and utilisation, which makes it difficult to integrate them into Furniture Company's systems. The result is that many transactions from these suppliers have to be manually handled.

Shoe Company

Shoe Company sells shoes in the Nordic countries through a large series of stores. They have a revenue in the upper range of the study. These store are handled by several separate entities in the group, but are administrated centrally through the company's headquarters. The headquarters manage e.g. product development and logistics for the whole business group. The company is using many integrated IT systems. In addition to a large number of internal integrations, there are several systems that are integrated to external points. Overall, the interviewee says, integration is what makes the other systems work; without the data from the integration solution other systems cannot handle their tasks properly. Shoe Company is a bit different from the previous four companies in this study because of the fact that they do not have a dedicated integration platform; instead their solution is based around point-to-point integrations of different kinds. The interviewee said that sooner or later they were going to implement a platform, but that the current solution was good enough for now. Integration costs are not a separate entry in the IT budget, but the costs are often in the form of consulting fees.

Leisure Product Company

Leisure Product Company sells leisure products in Sweden through both physical stores and a web shop. They have a revenue in the lower range of the study. Some stores are owned and operated directly by the company while others are operated by franchisees. The company is currently integrating approximately seven internal systems and a few more external ones, like invoice handling. The company does not use a dedicated integration platform, but has built a custom point-to-point solution with their Microsoft Dynamics ERP system as a central hub, which is used to channel information between their different IT systems. This solution is new and has been operational since early summer 2014. They do not have a separate entry for integration costs in their IT budget and the interviewee did not have a clear view of what the total costs might be, but they do have integrations costs as a factor when planning IT projects.

5.2 Benefits of Integration

Here the empirical data for the benefits of integration are compiled.

Office Supply Company

The interviewee from Office Supply Company feels that having AI in the company is absolutely vital for achieving sufficient automation and efficiency to be competitive. Automating the data flows between applications saves a lot of work hours, allows more efficient data transfers and increases productivity per working hour in the IT department. It was estimated that they would need at least twice as many employees in the IT department to handle the work manually. With the AI system, manual transfers and such are only made if a major problem arises that demands it. The system also provides many indirect efficiency gains, since it enables a wider and more continuous collection of data into Business Intelligence systems. The company uses QlikView to analyse e.g. sales data, making it possible to handle their wide range of products and react quickly in making decisions about problems and discrepancies that arise. Another gain from the improved data collection is improved logistics planning, enabling fewer and more efficient transports, which is great for both the environment and from an efficiency point of view.

From a strategic point of view, Office Supply Company is not actively using the integration platform to find new possible advantages that may be exploited, but it is used to deliver new services that customers demand. The integration capabilities are taken rather for granted by those who formulate the strategies, so while the platform is certainly included in plans requiring integration no strategies are formed around the specific capacities the platform can deliver. Integration is still, however, tacitly considered a large and important part of the capabilities of the organisation. One strategic development is that the group provides multinational customers the option to have contact with one main subsidiary that handles all of their business needs internationally. This is possible thanks to integration throughout Office Supply Company's group. Infrastructure-wise, the main benefit the platform delivers is the ability to keep legacy systems running easier and to be able to feed those systems with new information, in order to develop new organisational capabilities. This does not remove the problem that legacy systems are often slower than new ones and so forth, but it mitigates those problems. The most important benefit that the solution provides is a combination of the efficiency improvements and the ability to meet customer demands for integration, without either of those, the company would not be able to compete.

Industry Tools Company

Industry Tools Company did not see any special cost saving or efficiency increasing benefits for the main business areas from having an integration platform compared with point-to-point integrations, but overall saw it as essential to have some kind of integration to have the efficiency to be able to act competitively in their market. The platform is, however, a tool that increases the efficiency of the IT department, since it makes setting up new integrations less work-intensive and provides structure to the integration tasks. Their platform is extensively used to funnel relevant data into several kinds of decision support and analytical systems, providing definitive benefits to the different parts of the organisation utilizing those systems. Strategically the platform itself is of little interest, decisions are made expecting certain functionality and it is up to the IT department how it is delivered. Integration is put to a strategic use when customers make demands to receive data in certain formats, or when certain IT-related tasks can be out-sourced while remaining connected to internal systems. According to the interviewee, the trend points towards an increased demand from customers for different kinds of services that require integration.

As mentioned above the integration platform does increase the efficiency for the IT department when working with integration tasks. The interviewee claims that the platform does somewhat increase the complexity of handling the integrations for the company, but that the improved handling ultimately was more beneficial than the increased complexity was detrimental. The main benefits of having the platform from a technical point of view is that the process of setting up new integrations is sped up greatly and that less technical work is required. However, the increased complexity of handling the platform necessitates an increased corporate competence at handling integration issues. With regard to organisational benefits the platform does provide the company with the capacity to interconnect systems, but it is up to the different departments to properly realise this potential capacity for cooperation and process handling. The most important benefit from the solution is the ability to cooperate with customers, both to meet their demands and to more efficiently coordinate activities like logistics. This capability is seen as necessary for the company to be attractive to customers.

Clothing and Design Company

Before the integration platform in Oracle Retail was used, Clothing and Design Company had to do a lot of routine work manually, e.g. filling out certain forms properly. Now those tasks have been outsourced to a country in Asia, greatly decreasing labour costs, which is made possible by the integration platform, connecting the Asian and Swedish systems together. This enables the Swedish IT department to handle more critical analytical tasks, since purely administrative tasks have been minimized, thus increasing the productivity of the more expensive staff in Sweden. One benefit that arises from this is that the time zone difference between the Asian country and Sweden makes it possible for the Swedish workforce to arrive at work in the morning to find all necessary updates and data transfers already made by the Asian workers, also increasing operational efficiency. Overall, the integration platform has automated many simple tasks and enabled efficiency-increasing outsourcing decisions to be taken, thus reducing costs and improving efficiency for the company. The automation has, however, increased the need for data validation, since many problems can arise if data is inconsistent across systems that need it.

From a management and strategic point of view the platform provides some great benefits. The interviewee feels that the integration allows for collection of more detailed data from different systems, thus enabling better decision support for e.g. putting on sales or updating the range of articles in certain stores. Because of the automated integration with international shipping companies, Clothing and Design Company has a Swedish warehouse where goods can be imported from Asia for relaying to other Nordic countries, without the goods being considered to be imported to Sweden. This agreement with the Swedish Customs Agency is only possible thanks to the integrated data flows, making it possible to see that there are no discrepancies in the goods flows. Infrastructure-wise, the platform has increased complexity and costs, while also increasing the number of places to search for any errors that arise, since it can be hard to track where the root of a problem lies. The outsourcing of IT administration tasks to Asia is an example of how the integration platform improves the company's ability to handle more complex process flows. The most important benefits from the solution have been the ability to outsource the IT administration to Asia, thus cutting a lot of costs, and the ability to gather more and better data for decision support systems, which is very important for e.g. efficient logistics, having the right articles on stock in stores, and planning product launches.

Furniture Company

The main operational benefit that Furniture Company has gained from their integration platform is the ability to automate certain tasks that previously had to be made manually. Operational benefits are, however, not the most important benefits the company gains from the system; most important is the specialised handling of data for decision support systems and the process flow support and overview. The system gathers complex data from a multitude of systems across the company; the different headquarter departments, stores, subsidiaries, and franchisees. This data is adapted to certain structures that allow for advanced analysis, which the interviewee says would not be possible without the integration system, due to the size of the task. This data can be used for timely analysis of e.g. discrepancies in any routines, comparing stores with very different circumstances by standardising their figures, and being able to perform detailed benchmarking and setting goals for the different parts of the chain. Due to the complex organisational and legal structures of the company, the integration capacity of the customised platform is almost necessary to be able to gather the required data for the management.

As for strategic benefits, the interviewee says that customers are making increased demands for information from the company e.g. delivery status of their ordered furniture. The integration platform is a tool that allows the company to gather such information from various sources and provide it to the customers. The company has also started connecting with their suppliers using the integration platform, but the interviewee says that many suppliers are still not technologically developed enough to be able to connect in this fashion. Infrastructure-wise, the company prefers to have a larger internal IT department to handle both running their own server halls and certain development tasks, which they feel works well together with the customised platform they got. When the platform was being customised, they put a lot of effort into identifying and mapping their process flows, all so the system would be tailored to their needs. This adaption provides greater visibility of processes across the entire organisation. The most important benefit from the solution is the ability to handle the company’s complex structures and the ability to create the advanced decision support data, which is seen as extremely important for the company.

Shoe Company

According to the interviewee at Shoe Company, integration is an enabler for many IT systems at the company. Having integrated capabilities is vital for the business, e.g. since they handle logistics for so many stores, it would not be possible to handle the flow of goods if the stores and suppliers were not integrated. Integration is also necessary to gather data for business intelligence and other decision support systems. The collected data enable the top management to get an overview of the group and provides data support e.g. to handling a just-in-time delivery model. Overall, having integrations with transportation companies, stores, article databases, and central stock figures is necessary to plan logistics and store supplies. The integrations also allows the company to provide e.g. accurate stock figures in different stores for a certain shoe model to customers browsing their web store. The web store is seen as a very important channel to the customers, even though it still has rather small sales figures, because of the impact it has on the company’s brand and image. To get the web shop to work properly as an information platform like this requires integration and is an example of a strategic benefit.

Infrastructure-wise, the interviewee says they are somewhat suffering because they lack of a dedicated integration middleware. They do not think the functionality of the current solution is lagging behind that of a platform. However, using point-to-point integrations means creating new integrations is a time-consuming task and the resulting solution also lacks overview of the integration situation. The lack of a platform could slow them down and grow unmanageable in the future, the interviewee fears, but for now the non-platform solution is sufficient to reap the benefits from integration. The integration solution has also helped Shoe Company handle their invoice handling to manufacturers in China, where they are integrated with just one Swedish service provider, who in turn has integrations with all the other parties in China. This means the integration solution effectively has helped Shoe Company outsource parts of their integration work, which is an example of an organisational benefit. The most important benefits from the solution are that it makes other systems work properly, since they can get the data they need, and the ability to coordinate individual stores and suppliers, greatly improving logistics handling.

Leisure Product Company

The main benefit that the interviewee from Leisure Product Company sees from their integration system is a drastic reduction in the amount of duplicate work that needs to be done at the company. Previously, many divisions had to do the same work since it could not be shared properly, but now they have gained that capability. Thus the reduction of duplicate work is both an operative benefit, since it reduces costs, and an organisational one, because of the improved capacity to share work and have processes flow between divisions in the company.

Regarding strategic benefits, the interviewee said that the capabilities gained from the integration system are important for strategic goal fulfilment and for following strategic plans. For example, their web shop is integrated with other systems to be able to deliver information demanded by the customers, which is important for the web shop to be able to fill its role as a showroom for the company's products. Overall, the interviewee says, the ability to connect things is considered from a strategic point-of-view, although the integration solution itself might not be considered per se. According to the interviewee, the integration solution did not help implementation of new software, although it does bring more structure to some of the implementation work, since the integration format is pre-defined. The most important benefits from the solution are the reduction of duplicate work, with corresponding cost savings, and the support for strategic goal fulfilment.

5.3 Openness

Here the gathered data regarding openness and open source are presented.

Office Supply Company

Some of the IT systems used by the company today are open source. They are used in conjunction with the in-house developed integration platform and proprietary systems. One of the main points that was brought up during the interview with the company representative was that the attitude towards open source systems had changed compared to a couple of years ago. At first, they were positive about open source, and the enthusiasm was mainly driven by the prospect of cost savings: "*Free tastes good*". They liked the idea of not having to pay expensive license fees when introducing new IT systems into the organisation. However, as time went by they started to run into problems with the open system and the "free" aspect of the open source license model became increasingly questionable. Problems arose when developers in the communities of open source projects abandoned them, and the company had to use their own resources to handle the maintenance and service of the software and thus adding costs to the projects. The situation is manageable but the costs are increasing.

This experience with open source solutions in IT systems has made them more reluctant to open source when acquiring new systems. Now it is also clear to them that the initial cost for a license is not the most important aspect when buying a system. But even if some of their open source software experiences have led to a bitter aftertaste, the interviewee was not entirely negative about open software and if the questions of ownership and responsibility for maintenance in the long term can be handled with service agreements, they may be interested in open solutions in the future.

Problems with customer lock-in do exist when buying software from vendors much bigger than themselves. It is hard to know how about the priority you have as a customer and how much negotiation power you have. The interviewee did however not know if open source was the solution to mitigate the risk of being locked to a specific vendor. Instead that may be mitigated by new service delivery models, like cloud solutions, that do not have to be open if adequate functionality and performance is delivered at the right price point and at an acceptable level of reliability.

In terms of flexibility and modifiability, they need flexibility in the integration platform to make it fit processes and workflows in the organisation. They are however less interested in modifying the code themselves, they rather want adapters to connect systems and be able to buy those final solutions from an outside source instead of using internal manpower to code them. The company did not see any value in harnessing the community of an open source project for getting better stability or security compared to a proprietary solution. However, the interviewer said that one useful aspect of the communities surrounding open source projects was getting free information on how to use new functionality and how others have solved problems, both of which can be of use for Office Supply Company. When buying a new major IT system, like an integration platform, open source is not attractive, but it is more so when looking at supplementary systems.

Industry Tools Company

The attitude towards open source solutions is that they can be useful in certain systems in the organisation, but not in the main integration system. In supplementary systems the properties of the open source model can add value the solution. According to the interviewee, it would be dumb to not choose the open alternative if the decision stands between a proprietary and an open solution, as long as both can handle the requirements.

The most important aspects of open source are to prevent lock-in and enable the opportunity to make use of external consultants when solving problems and adding functionality to the systems instead of just having to rely on a single vendor. When it comes to costs, it depends on the type of the system if the open source model is beneficial or not. The community aspect of open source is not very important when it comes to stability, security or the overall quality of the code itself but there may be some use when it comes to support, as developers can get access to information regarding known problems within the project and get ideas on how to solve them.

Even if they do internal development and coding, the company wants to minimize the amount of time they invest in modifying the code themselves and maximize the use of standard solutions. They are however not always successful in managing the code without adding things of their own. Starting open source projects, with more opportunities to make modifications, can actually oppose the policy of doing as little coding as possible, since some developers become tempted to create their own solutions.

Clothing and Design Company

Some open source systems are used but not that many. Instead they rely mostly on the integration platform from Oracle and buy extra functionality when needed and as Oracle develops new functions to the system. There are however some small systems that are based on free and open software that are used in the organisation, but these are of little strategic importance. Most IT services that are used by the organisation are operated from the IT department abroad; they are not performing any coding themselves at the headquarters. This division of management and operations makes the flexibility and modifiability of open source solutions rather uninteresting for the company. They want functionally and are not very interested in how the systems are built, and that makes it a better alternative to add modules to the standard platform when needed rather than creating their own solutions to problems using open source.

They have recognized the potential problems of customer lock-in because of the fact that they are using platforms from very big vendors. But instead of relying on open alternatives to mitigate the risk of lock-in situation they are trying to keep the pressure on the vendors by talking to other customers in the market of IT systems and discussing the platforms and the quality of the service.

Furniture Company

In the process of buying or developing new systems, they do not have any preconceptions of either proprietary or open solution. What is important is that the new system will match the specifications that have been set prior to the project start. The IT department then makes the choice of whether it is better to buy proprietary software or to develop it internally, sometimes with open source software as a base, based on the alternatives at hand. But the software does not have to be open source for being bought and modified by the programmers in the IT department, in some cases they buy proprietary software and the right to modify it when it is suitable.

Today, some of the software used by the company is open source. For certain systems, e.g. their website, the flexibility provided by an open source project is seen as useful, while other systems like the main integration platform is proprietary. The decision to use open source or to buy a proprietary solution is mostly based on the projected cost. In some cases it is cheaper to build it by themselves using an open platform and in other cases, such as when the development time is scarce, buying a proprietary solution is the better choice. The crucial question is how to reach the planned goals of the IT projects in the most efficient way. What is important is to not be locked in to specific vendors; by being open minded when it comes to using proprietary and open solutions, more software solutions become available.

Shoe Company

They use many open source systems today and always look at many different options, some of which open source, when acquiring new systems. In general, the interviewee seemed to have a very positive attitude towards open source. Many of the systems they use have to handle heavy loads and the stability of the system is a key issue. These systems are built on Linux and one of the reasons for that is that they are more stable compared to the proprietary solutions they have tested. When adding a new system or upgrading an old one, there is always the choice between open and proprietary systems to take into consideration. One of the key points is to find consultants that are experienced with the particular software, regardless of whether it is open source or not. If you do not find the right competence to implement the software, it would be useless to the company.

The community aspect of open source can be interesting as long as the community is active. The main use of the community is to look for solutions when problems arise with the software. One of the main benefits is that support through the community is always available, whenever the problem arises. This is often not the case for proprietary offerings, unless you pay for 24/7 support agreements, which can be quite expensive. Using the community from the perspective of solving bugs is a bit overrated according to the interviewee because of the fact that many open solutions are already very stable and very little time goes into finding and solving bugs. But when bugs and security flaws actually appear, many software vendors with proprietary solutions can be slow in patching the flaw and are not always transparent in the process of doing so. Some proprietary software that they have tested has also proved to be fuller of bugs than open alternatives.

Looking at the problem of lock-in, using an open solution can prove to become useful of reducing the dependence of the bigger software vendors. The interviewee was also worried about many other aspects of handling the larger vendors, e.g., which could access sensitive data in a cloud storage and where the data is actually physically stored.

Leisure Product Company

Overall the company is positive about open source, especially regarding smaller systems like their web platform. Like in the other companies however they are less positive to use open source solutions in their main platform. One of the benefits open source brings is increased flexibility when they choose new IT systems because it increased number of vendors and software solutions. The community aspect can be of big use for the flexibility when software projects are moved between different consultants. The fact that the source code is visible and that members of the community can provide valuable information is useful in getting the new programmers up to speed in implementing the software.

The flexibility to choose from a larger number of vendors can also contribute to decreasing the risk of vendor lock-in. The interviewee however also added that this could also be said about proprietary software. The software they use from big corporations like Microsoft also has a big number of implementation consultants surrounding them and thus they can choose from many different when starting new projects. They also believe that open source can increase the security and stability of their software.

5.4 Price Model

Here we present the empirical data on how price models are used and considered by the case companies.

Office Supply Company

The organisation uses different price models for different software, but the most common one is based on an initial license fee and a service agreement. Some IT services have been outsourced, e.g. electronic billing, which uses a transaction model for payment. With this model you pay per transaction, meaning the costs increase linearly with the usage of the system. The transaction based price model is something the company is interested in and would like to use in other areas when paying for IT services. Even if they do not know to what extent such a price model would be feasible, it could be one way of putting IT costs more in relation with the revenue. When the revenue increases, so do the costs, which in some cases could be a fitting model for the organisation. Individual business units are held accountable for their own costs and another benefit of using a cost per transaction is that it becomes easier to place the IT cost on the unit that uses a specific service. This makes the internal bookkeeping more efficient compared to using other price models.

In the process of buying a new system there is much negotiation between the company and software vendors to reach agreements on price levels and price models. One big problem for the company is that it often is unclear how much the software will cost in the end. This can be connected to the fact that an integration platform is a big and complex system with many components and also that the vendors often are not very specific on the price. This makes comparisons between different alternatives difficult for the IT department. From the company's perspective they want to test the software before committing to huge investments, starting with a limited rollout and scale up if proven successful. This could be accomplished by using a price model based on some kind of trial license with a low initial cost that would increase with the usage. If the system is good and is used a lot, the vendor would get more money for it. This is however not a model offered by the software vendors that the company has been in contact with. The interviewee said that with the current price models, the cost of software licences tend to pile up with their current vendor without the company knowing how much use they actually will have of the software.

Bigger vendors tend to be more resistant towards flexibility in their price models, which have led the company to use smaller consultants when functionality is needed instead of buying a new big platform to replace the in-house integration platform currently in use. Even if this solves the problem of being locked into a vendor with a price model they do not like, the company does not want to be dependent on small consultants forever. Because of the fact that they use an older, custom built integration platform they do not have a price model that is based on packaging. Instead they just buy systems for specific functionality that can be added to the platform. In general, having to pay for additional functionality that is not needed when buying software is not a problem for the company according to the interviewee.

Industry Tools Company

When buying IT systems and services, the company uses business cases to estimate the value of the new functionality those systems would bring to the organisation. Even if the price models for different systems in their integration platform are different, the most common one is based around a service and support agreement for the software combined with a license fee for every user of the system. When buying whole systems it is often hard to estimate the cost because of the complexity of fitting big systems with many components into the processes and workflow of the organisation. When it comes to adding functionality to already existing systems it is not as difficult as buying whole systems, because the added cost can be calculated more easily without the need for complex business cases. Different software vendors are using packaging with their software offerings and the interviewee admits that they sometimes get more functionality than they are actually using. At the same time, buying functionality separately would most likely not be cheaper for them and it is not a very big problem.

Offerings with a transaction based price model are becoming more common, which is a development that the company welcomes. With a transaction based price model it becomes easier to do internal bookkeeping and often becomes cheaper for the company because of the way they use the systems. With the current model, they have to add a new license fee when a new user shall use the systems, which can be expensive. They really want to get away from the user based license fees and pay for actual usage regardless of the number of users. That kind of price model would be a better fit for the company's way of doing business and would increase flexibility and maybe decrease the cost of using some of the software they have.

Clothing and Design Company

The most common price model for software used in the organisation is a license fee per user in the system. Because of the fact that most of their IT is outsourced and managed from abroad the company mostly pays for services, with little interest in the actual technology delivering the results. They pay for services like information regarding warehouses, logistics and shipping, at an order or package basis. They use some transaction-based services inside their integration platform but these are few at the moment. The interviewee would like to see more volume-based or result-based pricing within the platform and thinks that the model with user based license fees are dated and many software vendors have not been able to keep up with the development of new functionality and performance when developing new price models. The organisation is stuck with some ancient license models that are very hard to understand, with complex terms and conditions that fail to provide information of the actual price of using the services. This makes calculation of the cost of new IT investment difficult when functionality is added to the platform.

The integration platform is module based and the organisation pays for the modules they are using and they do not feel they have to pay for software they do not use. Because of the fact that they do not develop software themselves, they wait for the software vendor to develop new functionality and then they evaluate if they will pay for new modules.

Furniture Company

When the integration platform was acquired, the price model was a crucial part in the choice of software vendor. For the most part, the company pays a license fee per workstation. In some systems inside the platform, they have bought licenses that they own for an unlimited period after paying a one-time fee. What type of price model is preferable is very dependent on the type of system considered.

Maybe the most important part of pricing of IT systems, according to the interviewee, is to get the license model right in the first place, especially when you buy big and expensive systems like an integration platform. It is important that the formulation of the agreement is done right, so the pricing of the services can match the processes of the organisation. When the company acquired their integration platform, they negotiated with the vendor and got a specific licensing model that matches the special conditions in the organisation well. The model was based on per terminal license and was not part of the original price models offered. The company had a very good position in the negotiation because of the fact that several software vendors wanted them as customers. The attractiveness of the company originated from the unique organisational structure of the company, and the software vendors saw an opportunity to create new solutions for different kinds of customers not well served by their current offerings. Because of the fact that the company used a lot of man-hours to get the agreement where they wanted it, today they are content with how they payment works.

Even if there are some systems that may benefit from transaction based price models, they are in general not very interested in that kind of model. There are potential problems with paying for usage, like less control of costs and the risk of high costs when lots of transactions has to be made. For the types of systems that are using in their work, other models work better. The added benefit of better internal bookkeeping that other interviewees had mentioned was of little use to them as it is not a problem they have in their organisation.

Shoe Company

Different systems use different price models in the organisation. Many of their systems use some kind of traditional license model where they pay per user, per simultaneously active user or the number of connected devices. One way of getting away from these kind of licence models has been to use open source software. Approximately 30% of their IT budget goes into payment of license fees, like initial fees, usage fees and fees for updating and maintaining systems. The interviewee said that they felt stuck with some of these agreements and would rather use another price model instead. In general, they are tired of the overly long, complex licence agreements that are common with larger American vendors. They also feel that it can be quite hard to actually understand what the terms for buying and using some software actually are, due to the complexity of the license agreements. The agreements by themselves are a tiresome part of the business and take time that they would rather use to do something else. The software itself however often works fine so one way of increasing the attractiveness of many software offerings would be to use less complex, “American”-style licence agreements.

Even if they have traditional price models for a lot of software they use, they also use other models. For certain kinds of external partners they use portal based integration, which is paid per transaction. This effectively means the integration platform is bought as a service from an external company, rather than Shoe Company buying or building a platform of their own. The interviewee said that these kinds of new, intuitive software offerings are interesting and maybe a way of getting rid of the old price and business models currently in use by most software vendors.

Leisure Product Company

The most common price model for the company is a yearly license fee per user with an initial investment. Although they are not very satisfied with that model, they do not have a particular model they prefer in any given situation. They see the payment as a necessary evil and the only really interesting thing about the price of an offering is its total size. Overall they prefer models with a fixed cost, as then the costs are easier to calculate and prognosticate. But besides that, the only really interesting thing about a price model is the total cost for the company.

When starting an IT project they feel they can get a fairly accurate figure for the costs rather early on, given a well-made requirements specification, even though the real costs often become a bit higher than initially estimated. According to the interviewee it is often a bit problematic to get a larger software vendor to actually cite a price or cost for a given implementation project, especially in an early phase.

5.5 Expert Interviews

In this section the three expert interviews are summarised.

5.5.1 Thomas Rosenfall – 2014-08-12

Thomas Rosenfall is a junior lecturer in industrial marketing at Linköping University and the interview was mostly focused on open source because of his previous experience and knowledge on the subject.

He thinks that answers will likely differ depending on the role of the interviewee, a CIO will have a different view from an IT department manager when talking about open source. From a theoretical standpoint, an open source solutions should always be preferable as long as it follows the requirements. In real life however, this is not the case and organisations will have different views.

Small projects have limited lifespans and there is always the risk of dying communities. One way of mitigating the risks is to use a partner when starting using an open source software that agrees to continue development if something like that happens. If dying communities and projects make people wary of using open source, Rosenfall makes the argument that if the vendor of proprietary software disappears, the source code may well be gone forever. It should also be noted that just because a software solution is open source does not mean that large vendors cannot back it. Many big open source projects today have a company or organisation that actively develops the software and provides services to their customers. Regardless of open or proprietary and if they act on a global or regional scale, when buying and implementing software you often use a local consultant, and those are exchangeable.

It will be impossible to completely avoid lock-in in these kind of complex and important systems. Having the source code available is however important if you want to have the ability to choose from many consultants, and from the programmer's perspective the open source model is rarely disputed.

5.5.2 Daniel Kindström – 2014-08-19

Daniel Kindström is an associate professor in industrial marketing at Linköping University and the interview was focused on the value and price model aspect of the study.

He thinks that it is important to think about what potential value really is, as the potential to realise the value of an offer is very dependent on the customer. Different customers have different perspectives on value and it is important to understand that potential value is relative. When using value in the way we do in this study you can think both in terms of cost savings and potential of increasing revenue. It can be good to have a discussion about both direct and indirect strategic value of IT systems as many customers may have a hard time seeing the value of an offering, even if they are using the functionality and are dependent on the running system.

A price model can be attractive to the customer even if it is bad at showing the actual value of the offering if it fits well into internal structures of an organisation, which is important to remember when analysing the value of price models. When thinking about price models from a vendor perspective there is always the factor of managing risk, since many new price models tend to move risk from the customer to the vendor.

5.5.3 Johannes Bynke – 2014-08-22

Johannes Bynke is the sales manager at Entiros and the interview was focused mainly on the benefits from Application Integration part of the study.

He thinks that one reason why some CIOs think that a middleware solution for integration increases complexity without giving a good overview of systems and connections is that many use older integration solutions. Several use an integration module in their ERP as the centre of the integration platform, like the MEC in M3, and those systems lack many of the benefits of a more modern system, such as an Enterprise Service Bus.

The fact that CIOs are not always part of the senior management of the corporation may inflict on answers that relates to strategic benefits of integration. Another aspect is that much of the IT work in a company consists of administration and solving technical and organisational issues rather than putting time into strategic work. There should however be a difference between manufacturing companies and companies that mostly sells products to consumers. It is remarkable how little strategic thinking companies put on integration. According to Bynke, many companies work with a 70/30 model, where 70% of the time is spent on getting things to roll at all and 30% is for planning and developing capabilities.

One way circumventing huge costs and “big bang” projects when it comes to integration is to work in an iterative way, with smaller projects that are phased into the organisation one at a time when replacing an old platform.

6 Analysis

In this chapter the empirical data from the case study is analysed using the analytical model. First the benefits aspects of an Application Integration solution are considered, followed by the openness aspects and open source, and finally the price model. In all of these areas we will try to find what connections there are within the area and how the area connects to value for the companies.

6.1 Integration Benefits

This section contains an in-depth analysis of the benefits the companies report having gained from their different integration solutions. First, the importance that the integration solutions have for the companies will be analysed, followed by an analysis of the five benefit types from a value perspectives, and finally a summarisation of the section's findings.

6.1.1 The Importance of Integration

When looking at the six case companies it is apparent that all of them consider integration to be important in some way or another. They do, however, consider integration to be important in different ways and to different degrees. That all companies have this positive view of integration is a rough indication that it creates value for them. However, most of the contacted companies were not interested in participating in the study, which could indicate that they do not consider integration to be all that important. Our sample size is simply too small for us to be able to draw any wider conclusions from the fact that all of them consider integration to be important. We can, however, see if any factors seems to affect how important integration is to the companies. In Table 6.1 below the companies' opinions on the importance have been summarised.

Table 6.1 – Importance of integration

Case	Type	Size	Solution	Importance
Office Supply Company	B2B	Medium	Standard and custom platforms	Vital
Industry Tools Company	B2B	Medium	Standard platform	Vital
Clothing and Design Co.	B2C	Medium	Standard platform	Necessary
Furniture Company	B2C	Larger	Modified platform	Necessary
Shoe Company	B2C	Larger	Point-to-point	Necessary
Leisure Product Co.	B2C	Smaller	Point-to-point	Beneficial

First, we notice that both of the B2B companies consider integration to be more important to their business than the B2C ones do. Further, both of them reported most important capabilities related to connecting with customers in some way, which none of the other companies did. Johannes Bynke (2014-08-22) says that corporate customers, especially larger ones, quite often demand some kind of integration with their suppliers. This could concern e.g. electronic invoices, delivery and stock information, or handling of warranty issues. Overall, serving larger customers seem to increase the perceived importance of the integration solution.

When looking at the size of the companies it would seem like the larger companies consider integration to be more important. A larger company usually has a more complex organisation, which leads to greater difficulties in coordinating work and a larger number of IT systems, all of which reasonably increases the importance of integration. Johannes Bynke (2014-08-22) confirms that larger companies are more likely to have a need for integration. Overall, having a larger organisation seems to increase the importance of integration.

Finally, we notice that companies with some sort of proper AI solution seem to consider integration more important than those with a solution built around point-to-point integrations. This is, however, not the right way of looking at the data; it feels far more reasonable to say that the companies that consider integration more important have AI solutions, while the others have simpler point-to-point solutions. Thus, the type of solution used probably cannot be said to affect how important integration is perceived to be, but the reverse seems to be true.

6.1.2 Value from Different Benefit Types

In general the benefits reported by the companies fit the benefit framework by Shang and Seddon (2000) rather well, but there are some cases of overlap between two or more of the categories and some benefits that give rise to other benefits in other areas. Below, the five main types of benefits will be analysed.

Operational

Most of the companies interviewed in the study were of the opinion that operational benefits were one of the two main types of benefits they gained from their integration solution. When asked for examples, most interviewees talked about different kinds of cost savings, e.g. from automation of tasks that previously had to be done manually, reducing the need for duplicate work, decreased payroll thanks to improved productivity, or more efficient handling of logistics and stocks. That operational benefits is a type that is commonly found fits what Themistocleous and Irani (2001a) saw in their study of benefits from Application Integration. In that study one of the most commonly reported benefits was cost reductions, which matches the answers from our empirical study well.

With the definition of value from Anderson, Kumar, and Narus (2007) being based upon monetary aspects, it is relatively simple to connect these cost savings to value creation. Thus it can be said that the cost savings reported from using the integration solutions are clearly indicative of operational benefits and thus value-creating for the companies.

Managerial

This type of benefits is the other of the two main types of benefits gained from AI solutions that were mentioned by the interviewed companies. The most common benefits of this type that were mentioned during the interviews were improved abilities to gather data for decision support systems, e.g. Business Intelligence tools like QlikView, and improved abilities to plan logistics and stocks, leading to a better allocation of resources. When comparing these results to those from Themistocleous' and Irani's study (2001a) there are some similarities and some things that differ. Our finding that companies report increased abilities to gather and analyse data match theirs, but in their study there was no specific mention of improved resource allocation capabilities. This might be because resource allocation improvements are instead seen as cost reductions. Their study also found that "*Allows organisations to do business more effectively*" and "*Achieves return on investment"* as important benefits, while we have no direct mention of such benefits. However, many of the interviewees did hint at the importance of integration for their business, which can be seen as an indication that integration improves business efficiency. We also did not hear any mention that the integration solutions had not been worth their cost, so this difference might come from our choice of questions.

Linking these kinds of benefits to our definition of value is less straightforward than for operational, but the connection is still quite clear. For the resource allocation benefit the improved logistics and stock planning give rise to cost savings, since more efficient routes and fewer deliveries can be used and there is less capital bound in large stocks, which is clearly value creating. Better decision support data is not as immediately value-creating as resource allocation, but there is still a link. Decisions that are based in better data have better conditions to be successful. Better decisions most likely gives rise to cost savings and earnings that are larger than for less well-grounded decisions. This means the improved data access and analysis also can create value for the companies. Overall, this indicates that managerial benefits from the integration solutions are value creating for the companies studied.

Strategic

The strategic benefits reported roughly come in three types: handling customer demands, running web shops, and aid in strategic goal fulfilment. The first category is the one most commonly reported, with the interviewees from Office Supply Company and Industry Tools Company saying that it is one of the most important benefits from the integration solution. Shoe Company also mentions that their solution is an important part of getting information to customers. These findings match those by Themistocleous and Irani (2001a), whose study found that the only strategic benefit reported by several companies was "*Achieves customer satisfaction*", which can be compared with our handling of customer demands. The web shops both act as a new market channel and as a platform for delivering information regarding e.g. stocks of certain products or new products to be launched. All six companies set up their own web shops, but Shoe Company specifically mentions it as an important part of their marketing. Finally, Leisure Product Company says that their integration solution helps with strategic goal fulfilment, but does not give any specific examples for this.

It is a bit surprising that the new market channels the web shops constitute are not seen as benefits by more companies. For most companies, the web shops seem to be there more as a channel for marketing and providing information for customers rather than as an important sales channel. This could be explained by the fact that many people still are a bit reluctant to shop for certain things, like clothing or furniture, online. Since all companies still have web shops, however, it seems they still think it is worthwhile to keep these channels up, indicating they create a net value. Since some kind of integration solution is necessary to get a web shop up and running, the companies' solutions are thus creating some value, although it is as always hard to quantify the value of marketing.

There could be many reasons for the lack of reported strategic benefits. One possible answer could be that the solutions themselves are not considered strategic, but the capabilities they actually confer are. This would mean the companies strategically consider the ability to connect IT systems both internally and externally, but do still not think of the solution itself as strategically important, since it is the connected systems that handle the important tasks, not the integration solution as such. Comments from Office Supply Company that integration is tacitly considered an important part of their capabilities does indicate this could be the case. This connects to what Daniel Kindström (2014-08-19) calls an indirect strategy, when e.g. an IT system is strategic not because of an explicit plan but rather because of how it is used. If this is the case, the integration solution might not be intended to be strategically important, but its role as enabler for many other IT systems the company uses might render it strategic, which indicates the solution provides more benefits than the company expects it to.

It could also certainly be the case that the integration solutions simply do not provide that many strategic benefits and opportunities beyond what has been reported. As pointed out by Carr (2003), simply having an IT system is never going to confer a competitive advantage, since the competitors can also get the same systems. Thus it is possible that the scarcity of reported strategic benefits could stem from the fact that the integration solutions are not strategically important; they are important for e.g. cutting costs and gathering decision support data, but are not on a strategic level. However, it could also be that there are more benefits to be reaped from the solutions, the companies have just for some reason not been able to do so. Since an IT system could be strategic if the company using it has the right expertise (Brown, Hagel, & Varian, 2003), the problem could be that the companies do not use the systems in the right way for gaining some potential strategic benefits. This expertise that Brown, Hagel and Varian (2003) talk of could also be likened to a conversion contingency, in order to be able to realise the value potential of the solutions the companies must have certain capabilities (Davern & Kauffman, 2000).

For example it is possible that the system's capabilities have not been examined and analysed in-depth, so the company is not aware of what the solution could actually do on a strategic level. Most likely the IT departments are well aware of how the systems work and what can be done with them from a technological point-of-view, but it is still possible that the management level is not fully aware of the full extent of the capabilities of the systems. In this kind of situation the company would gain more benefits from their system using an inside-out view of strategy, where they study the system's capabilities and then see what opportunities in the market could be captured using that capacity. Johannes Bynke (2014-08-22) says that companies often are too busy with the operational and administrative sides of IT systems to stop and fully consider their strategic potential.

The connections between the strategic benefits and value creation are very indirect, but these benefits are also among those that have the potential to create the most value. A benefit such as opening up a new market channel could potentially lead to greatly increased earnings, better handling of certain customer demands could improve competitiveness and earnings, and better strategic goal fulfilment might both improve earnings and reduce costs, just to mention a few. The value that stems from this type of benefits are hard to quantify, meaning it is hard to actually calculate what value comes from them (Anderson, Kumar, & Narus, 2007). As indicated by Brown, Hagel and Varian (2003), conversion contingencies in the form of demands on expertise in handling integration are especially important for realising the value potential of an integration solution. The maximum value potential a company can realise seems to be at least partially related to how willing customers are to purchase their goods from a web shop, which affects whether the web shop is a major sales channel or more of a marketing channel. This relates to Smith's and Nagel's (2005) value in exchange, where the company with customers more accustomed to shopping over the web have a larger value potential to try to realise.

IT Infrastructure

Most interviewees mention some benefits in this area, especially the ones who use proper AI solutions. The earlier study by Themistocleous and Irani (2001a) found that more than 20 per-cent of companies reported the benefits "*Results in reusable systems, components, and data*", "*Faster and cheaper implementations than bespoke solutions*", and "*Provides flexible, maintainable, and manageable solutions*". Our interviewees reported benefits such as improved capabilities to continue using legacy systems, improved efficiency in the IT department, and faster implementation of integration connections. Shoe Company and Leisure Product Company, who do not use integration platforms in their integration solutions, reported decidedly fewer IT Infrastructure benefits. The interviewee from Shoe Company spoke of how the lack of a dedicated middleware was detrimental to their visibility of the solution and that they would probably have to update sometime, but that for the time being they were satisfied with the solution they had.

From discussions with Entiros it was expected that benefits like legacy systems use, improved visibility and reduced complexity would be reported more often. Improved handling of legacy systems was only explicitly mentioned by Industry Tools Company, but is likely important for others as well. Many companies were not satisfied with the visibility their solutions provided, especially those who do not have AI solutions, which Johannes Bynke (2014-08-22) thinks could be related to the fact that most have systems that use older technology as their base. Further, the companies with AI platforms often said that the solution had increased the complexity in handling integration, e.g. Industry Tools Company and Clothing and Design Company, but that the benefits from the system outweighs the drawbacks. It should also be noted that Shoe Company and Leisure Product Company, who do not have AI platforms, still consider their integration to be rather complex to handle.

IT infrastructure benefits are often quite straightforward to link to value-creation. Benefits that improve flexibility, implementation speed, and efficiency all lead to lower IT costs, which is immediately valuable. Retaining legacy systems postpones the need for new investments, also saving costs, but could be detrimental if the company keeps using old systems that do not quite handle their requirements for longer than they otherwise would. Visibility helps the IT department manage the IT systems and enables them to handle problems that arise quicker, since they are found earlier. Overall, IT infrastructure benefits have a quite clear connection to value, but the companies studied report only modest value in the area.

Organisational

Regarding this type of benefits there were broadly speaking two groups for the companies: those who reported many or big benefits and those whose benefits were more modest. The first group consisted of Fashion and Design Company and Furniture Company and mentioned several benefits of an organisational nature from their integration solutions, e.g. being able to outsource vast parts of the IT administration, special import agreements with the customs agency, handling of a very complex legal and organisational structure, and great process support. The other companies also reported benefits, e.g. reduced need for duplicate work, improved support for internal coordination, and process support, but the benefits were fewer and less important. The only organisational benefit from Themistocleous' and Irani's study (2001a) was *"Increases flexibility"*, which is comparable with the mentions of process and cooperation/coordination support.

What sets the two groups apart is that the first group seems to have spent more time and effort either adapting the systems to their situation or adapting themselves to the system. Furniture Company had their system heavily modified by their ERP vendor Infor as a part of the procurement process, customising it to their special needs and situation, and especially mention how they made a major process mapping to make sure the system would support their workflows and business processes. Fashion and Design Company are on the other hand dedicated to only using standard systems without modifications and instead adapt their organisation and processes to those systems' requirements. The other companies have not mentioned anything special regarding how they implemented or adapted their solutions, but have also reported fewer organisational benefits. From this it seems like the companies that put more effort into using the systems, whether through adapting the system or themselves, are better at gaining organisational benefits than those who do not put as much.

From a value perspective, organisational benefits can be a bit tricky to link to value creation. Certain benefits, like reduced need for duplicate work, can easily be linked to cost savings, but others, like process or coordination support, can be seen as improving productivity which in turn can either save costs or increase earnings, but the link is more indirect. Capturing the potential value of the organisational benefits from an integration solution seems to place higher demands on the company in question, but most will be able to capture some value in this area.

6.1.3 Summary of Benefit Types and Value

As has been seen in the above analysis of the benefits types, each type seems to have a special relation to value and the types seem to be differently hard to gain. The attributes of these relations with value have been summed up in **Fel! Hittar inte referenskälla.** below.

Table 6.2 – Benefit types and value

Benefit type	Relative Effort	Link with Value	Potential Value	Value Realisation
Operational	Lower	Direct	Medium	High
Managerial	Lower	Direct & Indirect	Larger	High
Strategic	Higher	Indirect	Larger	Low
IT Infrastructure	Medium	Direct	Medium	Middle
Organisational	Higher	Direct & Indirect	Medium	Middle

The relative effort is our estimation of how much effort a company must expend in order to realise a large part of their potential value for the area. This estimation is based on how many benefits a company reports for a type in relation to how much they say they have focused on the type of benefit. For example, all companies have focused on operational and managerial benefits and report large benefits in both areas, making it hard for us to estimate how much effort they had to expend to gain these benefits. However, only Furniture Company and Clothing and Design Company have explicitly focused on organisational benefits and are also the only companies to report great benefits of that type, indicating organisational benefits require more effort to gain. Note that lower relative effort does not mean it is a low effort in an absolute term; it is just for the sake of comparison.

The link to value refers to whether we estimate the benefits to be immediately impacting economic figures, the basis of our definition of value, or to give rise to circumstances that then create value. For example, the improved data for decision making, a managerial benefit, does not immediately mean lower costs or improved earnings, but can be used to support decisions that result in cost savings or stronger earnings. This would mean the benefit has helped create value, but the connection is indirect.

The potential value column contains a rough estimation of how large the potential for the benefit type is. This estimation is based on discussions with the companies and Entiros and readings of the theoretical framework (Hasselbring, 2000; Linthicum, 1999; 2003; Sprott, 2000). For example, Hasselbring (2000) speak of how connecting legacy systems, an IT infrastructure benefit, is of value for companies, adding potential value for that category. By going through each category like this we have arrived at our estimation of its potential value. It should also be noted that the maximal amount of value potential that can be realised will differ from company to company, due to their different circumstances (Smith & Nagle, 2005).

Finally, the value realisation column contains our estimate for how much of the potential value of the solutions that the companies manage to realise. This estimation is based on how many and the magnitude of the benefits the companies report for each type, in relation to our estimation of the value potential for the types.

We would like to end with a big caveat that all of the above is a rough estimation of what the connections look like based on a small number of cases and should be seen as hypotheses rather than well-grounded descriptions of reality.

6.2 Openness

This section contains an analysis of the answers from the case study regarding benefits of open source and how they relate to value. It starts with an overall analysis of the importance of openness and then the value from open source benefit factors are analysed one by one. The analysis is then summarized at the end of the section.

6.2.1 The Importance of Openness

When talking about openness factors with the interviewees it became clear that there are different attitudes regarding open source among the case companies. They are using open source systems to a varying degree and are finding the benefits valuable in different ways. In general, the interviewees closer to the technological side and the actual implementation of the systems seem to be closer to having the theoretical benefits of open source than the ones with more managerial roles. Some of the scepticism of open source may be related to the fact that they do not actually know what such solutions can be like. It may also be a matter of how the organisations are structured and how they work with IT. In some cases the benefits of open source may be more obvious, like if you as a manager are interested in the actual implementation and building of the IT infrastructure. In other cases where functionality is the main priority, standardised systems, SaaS or outsourcing can be used and open source may be seen as less beneficial.

No company in the study with a middleware solution for integration uses an open source solution for the main integration platform, and several has directly expressed that they do not want one either. However, all of the companies use open source software in some of their smaller systems in their integration solutions. The cause of this may be related to the fact that the more important the systems are, the more the company representatives need assurance that it will be operational at given times. Having a big vendor behind the system provides a clear role of responsibility. By using an open source integration platform, they may be wary that they are moving a lot of risks into themselves and would be stuck with a lot of costly problems if something goes wrong. The question of responsibility and ownership as a potential drawback of open source is brought up by different authors who connect it to different aspects. This can be a drawback from having increased flexibility (Morgan & Finnegan, 2007) or reliance on the community for support (AlMarzouq, Zheng, Rong, & Grover, 2005). But not all open source software is run by an abstract community of programmers with unclear responsibilities. Today there are big companies that stand behind open source projects and can deliver services of the same type as big vendors with proprietary solutions, offering models like dual-licencing where openness and clear vendor responsibility combine (Brandel, 2010). Regardless of the reason, it seems that the potential benefits for using an open source solutions for an integration platform are not something that the case companies perceive as bigger than the potential drawbacks. However, in the broader sense of integration solutions, where proprietary integration platforms can act together with different supportive open source systems, openness aspects can be important in creating valuable IT solutions for at least some organisational types.

6.2.2 Value from Open Source Benefit Factors

In this section, the benefit factors of open source from the analytical model are used to analyse the answers from the case study regarding the value of openness in integration solutions.

Lock-In

In the case study, all of the interviewees express that vendor lock-in is something that they all have recognized. Mitigating the risks of being locked in into a specific vendor with high switching cost, as Shapiro and Varian (1999) describe, is of importance when planning and building their IT infrastructure. How the companies mitigate the risk of lock-in and what role open source software play in doing that varies considerably. In some cases open source is a valuable tool in keeping options open when choosing the systems while others use different kinds of strategies to mitigate the risks.

Industry Tools Company thinks that preventing lock-in is the most important aspect of open source since it increases the available choices of external consultants when solving problems and increasing functionality. Furniture Company believes that it is important to be open minded when choosing software because open and proprietary solutions can be useful in different situations and the important thing is to have a large selection of options. Using open source as a way of increasing the choices of available consultants goes in line with what Thomas Rosenfall (2014-08-12) said in our expert interview. Leisure Product Company also has the opinion that the flexibility of choice is important to mitigate lock-in but added that it does not have to be open source for this to apply, as widely used proprietary software solutions also have lots of consultants surrounding them. This implies that open source does not have to be the answer on how to get many consultants to choose from; it depends on the software platform in use. However, if you are using a proprietary solution that is more obscure and lacks any local partners for integration, you may run into problems if you want to choose from many different consultants. If you have the source code available, even programmers with no previous experience of the specific software will have more options for understanding the functionality than just the vendor's documentation, which may increase possible options for the customer. But just as Messmer (2013) points out when it comes to software security, different software projects have different views of what to focus on and what are proper coding standards. Some projects are very good at writing well-documented and easily read source code while others are not. The answer to the question if open source software is beneficial for transferring software projects between developers will be dependent on the specific open source project.

In the analysis of the survey made by Computer Economics (2005) they brought up the point of using open source as a way of reducing dependence on big software vendors. Office Supply Company thinks that it can be problematic to negotiate with vendors that are much bigger than themselves because they have very limited ability to make any demands, and that is a relationship they do not want to be locked into. They were not sure however that open source is the way of preventing that; instead new types of service delivery models may be the way forward. Also Clothing and Design Company and Shoe Company say that reducing the risk of being locked into an unfavourable position against a big vendor is the main issue when discussing lock-in. They do however have different ways of working with the problem. Shoe Company is one of the more positive regarding open source and is using many open source systems and, according to the interviewee, this helps them to reduce unwanted lock-in into large software vendors. Clothing and Design Company instead uses dialog with other companies in the same business as them to keep the pressure on the big vendors and get a better negotiation position by the number. This stance may be explained by the fact that they use standard solutions when they can and are heavily invested in a proprietary platform that they do not want to change. They are also not dependent on external consultants in the same way as the other companies in the study because of the fact that they have outsourced their IT department abroad.

The different views of how useful open source is to mitigate the problems of lock-in are summarized in Table 6.3.

Table 6.3 – Views on open source and lock-in

Case	Lock-in is a potential problem	Major source of lock-in	Open source software is useful in reducing lock-in
Office Supply Company	Yes	Weak negotiation position	Uncertain
Industry Tools Company	Yes	Lack of options	Yes
Clothing and Design Company	Yes	Weak negotiation position	No
Furniture Company	Yes	Lack of options	Yes
Shoe Company	Yes	Weak negotiation position	Yes
Leisure Product Company	Yes	Lack of options	Uncertain

In general, the companies that think that lock-in is a problem of lack of options seem to think that open source software can be useful in increasing those options, both in the way of increasing available consultants and making the creation of own solutions easier. The ones that think that the major concern with lock-in comes the uneven negation position when using software from big vendors are less clear in the usefulness of open source and think that it exist other options of mitigating lock-in.

It should be noted that much of the lock-in discussion surrounds the smaller systems and sub-systems that provides specific functionality inside the integration solutions of the case companies. Complex and important IT systems, for example different kinds of integration platforms, will always have a big lock-in built into them regardless if they are open or not, as noted by Thomas Rosenfall (2014-08-12). The more dependent a company is of their integration platform, the higher the switching costs. Open source arguments for an easy way of changing platforms are not very relevant but the increased ability to choose between consultants for maintenance and support as well as reducing dependence on large software vendors may still apply.

When looking at the lock-in aspect from the perspective of the value defined by Anderson, Kumar and Narus (2007), it boils down to minimizing risk. If a company is stuck with software that is difficult or expensive to maintain and support because of the lack of competent consultants, it will increase costs. If the software in question also lacks vital functionality for supporting the business model of the company, the potential increase in revenue will decrease. Having the ability to choose software and how to maintain it will in those cases be of value for the customer. This also applies to minimizing the risk of getting stuck with large vendor where a bad negotiation position can jeopardise both an efficient cost structure and increased future revenue. The case study also implies that the value of mitigating vendor lock-in can be captured in different ways and do not have to involve open source.

Cost

Cusumano (2007) argues that open source has contributed to diminishing the importance of initial license fees and the transfer the costs to maintenance and support. AlMarzouq et al. (2005) say that removal of licence fees and moving support into communities can decrease direct costs even if Total Cost of Ownership (TCO) may be difficult to decrease. Looking at the different case companies it seems that they in general agree with the TCO argument as initial license fees for much software is of little importance for their IT budget. Instead most of the cost consists of maintaining the software and they either have to pay for internal IT maintenance capacity or buy service agreements from vendors and consultants. In solutions where open source software projects are backed by some kind of organisation that supplies complementary services like Brandel (2010) give examples of, the difference in cost structure between proprietary and open source solutions becomes minor.

Office Supply Company has a complicated relationship with open source because of the cost aspect. By reading too much into the "free" aspect of open source, they became stuck with a "dead" project and now they have to put more resources than expected into maintaining the project themselves. This points to the importance of looking at software costs from a TCO perspective rather than just initial costs. Apart from Office Supply Company that made this lesson the hard way, the rest of the companies in the case study seem to comply with the TCO view and uses business cases to get a more complete picture of the software costs. Thomas Rosenfall (2014-08-12) says that one way of reducing the risk of dead communities is to use a partner in combination with an open source project. A partner can be a consultant or a software vendor that is backing the project and provides some kind of insurance that the project will live on. If you want an open solution and neither the partner model nor some kind of commercial open source project (Brandel, 2010) are interesting, the need to evaluate the project and the community surrounding it becomes even more important. (AlMarzouq, Zheng, Rong, & Grover, 2005)

For companies like Industry Tools Company, Furniture Company and Shoe Company that develop much software in-house, an open source solution can sometimes be the most cost-effective solution. It can be easier and also cheaper to start from an open source project when developing new systems that adds functionality to their integration solutions.

When looking from the value definition of Anderson, Kumar and Narus (2007), the value of openness for reducing costs is very dependent on how the IT is structured in the organisation. For organisations that build a lot of systems themselves, there exists an opportunity to decrease developing costs by using an open source project as a base, and in these cases potential value is high. For organisations that rely on standard solutions and do not have or do not want to do a lot of coding themselves and want the functionality more in a service-like manner, the potential value is low for open solutions. The conversion contingencies (Davern & Kauffman, 2000) between the case companies are different and the organisations that have the capabilities and resources to capture the potential cost savings will see more value in open source from a cost perspective.

Security

According to some views, security is another argument for using open source, and is mainly based around the fact that more eyes will find more flaws (Raymond, 2002) and that security by obscurity is something to be avoided (Hoepman & Jacobs, 2007). In the case study however, the aspect of software security was not a big issue for any of the companies. The only real security concern that we heard about during the interviews was from Shoe Company that had concerns about to what degree they can trust service providers when using things like cloud services. When choosing between open source systems and other alternatives, security does not seem to be a factor according to the interviewees.

One aspect that several of the case companies seem to think is more important than security is stability which in some ways also can be related to similar arguments like the one from Raymond (2002), more eyes will also detect more stability bugs than a few eyes, not just security flaws in the code. Some open source projects also focus on writing and documenting the code in a way to enhance security and stability (Messmer, 2013). However, even if stability is brought up as important in some interviews, no one seems to have any big issues with stability bugs and this is true for both open source and proprietary software. The fact that the problems in that area seem to be few may explain why they see little value in open source from a security standpoint.

That software security seems to be such a non-issue can seem a bit strange from our perspective because there is no lack of sensitive information handled by many of the systems in their integration solutions. There are many transactions performed every day that contain data from customers, suppliers and other external and internal parties and all of the companies in the case study most likely wants to perform these transactions in secure way. An explanation for the lack of interest may be our choice of interviewees. From a management perspective this questions may be of less importance than functionality, stability and cost. If we had interviewed persons with more technical roles closer to the implementations we may have gotten different answers. There is also the possibility that they would not consider using open source in systems where security is a high priority but that was not anything we heard in any of the interviews.

From a theoretical standpoint there could exist connections between the security benefit of open source and the Anderson, Kumar and Narus' (2007) definition of value. Security flaws may create problems for companies both when it comes to increased costs and loss of potential revenue. But because we did not find anything in that direction when talking to the interviewees we cannot say that the security factor of open source creates any value for the integration solutions.

Flexibility/Modifiability

Just like the cost factor, the usefulness of the aspect of flexibility and modifiability is very dependent on how the organisation works with IT. Office Supply Company needs a flexible integration platform to make it fit the organisational structure. They are however not interested in modifying the code themselves, instead they rather buy final solutions like adapters to different systems. This view is shared with Industry Tools Company and Clothing and Design Company, where they work to minimize the amount of code they write themselves and want to use standard solutions as much as possible. In both these cases the value of modifying the code seems rather limited. In the case of the Industry Tools Company, the interviewee goes as far as saying that open source software may create problems with keeping the policy of using standard solutions because programmers that get the opportunity to change and improve code may fall for the temptation of doing so. The main reason for wanting standard code in this case is to make the process of updating the software more efficient. In the case of Clothing and Design Company they are more interested in functionality than having to code themselves, which once again can be related to their outsourced IT department.

For the companies that do much coding themselves the picture is different. Both flexibility of use and flexibility allowed by licenses (Morgan & Finnegan, 2007) can be useful for these companies. Instead of having to change their own workflow and processes to fit standard solutions they can modify the software to work according to their specific needs. And with the flexibility allowed by the open source licenses it is possible to create more cost effective solutions as noted in the cost analysis. There are of course proprietary platforms that can be very flexible solutions as well and can be configured to fit different organisational structures. Integration platforms are often made with a modular design that allows for flexibility even if the source code cannot be modified.

As in the cost factor the potential value of flexibility and modifiability depends on the company. Those who build processes and workflow to work well with standard solutions and want to make things like upgrading systems as straightforward as possible to maximize efficiency may benefit from less modification of the systems. In these cases the modifiability and flexibility of open source would not add much value. In the case where the organisational structure makes it hard to change workflow and processes to fit standard solutions, this aspect of open source may be valuable for the customer.

Community Support and Maintenance

The community aspect of open source is the one factor that has had most diverse opinions among the case companies. Some of the companies, like Clothing and Design Company and Furniture Company did not have much to say about the importance of communities around the open source systems they are using. Office Supply Company and Industry Tools Company use the communities to get information about new functionality and as a source for support for the IT departments. The problems that Office Supply Company faced with open source discussed in the cost section can be related to community support. As AlMarzouq et al. (2005) bring up, it is crucial to evaluate the community before committing to an open source project, something that was not done enough in that case. The Shoe Company was in general more positive about the benefits that a community can bring and mainly use it for support. A strength of support from the community is that it is always open; you do not have to pay for expensive 24/7 service agreements from a software vendor. However, it is very important that the community is active for it to be useful. The interviewee also thought that even if the community can be used for maintenance, like solving bugs and keeping the systems stable that aspect is also a bit overrated because most systems they use are very stable to begin with and not much bug solving is needed. One benefit compared to proprietary solutions is that when a security or stability issue actually appear the process of solving the problem can often be more transparent because of the community. The benefit that Leisure Product Company brings up is also related to support, as the community can be helpful when moving between consultants as a complement to the source code.

In general, it appears that the companies with a more positive attitude to open source also seems to make more use of the communities than the companies with a more negative attitude. One exception seems to be Furniture Company that have seen benefits of open source in other areas but does not have an opinion of communities. Another aspect that apply to all of the companies, even those who make use of the benefits of the community, is that they only seem to use it as a source of information, being idle users as AlMarzouq et al. (2005) put it. This is however not an aspect that was discussed thoroughly during the interviews and it is possible that the companies or individual programmers in the IT departments contribute to open source communities.

The value (Anderson, Kumar, & Narus, 2007) of the community seems to relate to how much the company believes in the open source model. It can be a source of support and maintenance that can contribute to reducing costs of some systems if they do not need separate support agreements. If it actually is a cost reduction will of course depend on how much time the developer needs to put into searching communities for answers in comparison to a paying for a support license from a software vendor. If the IT department work well with this kind of community support it may add value to the offering. Just in the cases of the cost and flexibility/modifiability factors of open source, the potential value (Davern & Kauffman, 2000) will depend on the organisations internal capacities of making good use of open source.

6.2.3 Summary of Openness and Value

When looking at how the different aspects of open source can be perceived as valuable for the case companies, it becomes apparent that the value of openness is dependent on how the organisations are using and managing IT. There seems to be two different kinds of companies that benefit differently from open source. The ones that do much coding themselves and want to create software solutions that fit them better than standard offerings can see potential value in several open source factors, like reducing direct costs, flexibility in developing software and use communities to support and maintain the software. The situation is the opposite for companies that want to do as little coding as possible and rely on standard solutions for increased efficiency in workflow and processes. For these companies the increased flexibility and choices may be considered drawbacks as they may increase their complexity in handling IT and lead to added costs.

For two of the factors, lock-in and security, it is more difficult to see a special distinction between the two company types. When it comes to security, none of the case companies seem to value that aspect of open source at all. For lock-in, they all consider it as something important to take into consideration when buying software but they are not all in agreement that open source software is a good way of mitigating the risks of lock-in. Lock-in in the case companies can be divided into two types, the lack of influence that buying software from large vendors brings and the lack of software and implementation options. The ones that think that the way of mitigating lock-in is by having many options seem to value open source to a higher degree than the ones that want to increase their influence. A summary of how the open source benefit factors relate to value can be seen in Table 6.4.

Table 6.4 – Summary of benefits of open source and value

Benefits of open source	Companies that code solutions themselves	Companies that want standard solutions
Lock-in	Depends on type of lock-in	Depends on type of lock-in
Cost	Potential value	No potential value
Security	No potential value	No potential value
Flexibility/Modifiability	Potential value	No potential value
Community support and maintenance	Potential value	No potential value

6.3 Price Model

This section contains an analysis of the connection between price models and value. First the general importance of the price model among the companies are described, then aspects of the current price models are identified and the value is analysed with the help of the dimensions of the SBIFT model. The section ends with a summary of price model dimensions and value.

6.3.1 The Importance of Price Models

When observing the mix price models inside the integration solutions it soon becomes clear that there will not be one type of price model that works for all systems that are integrated in a large company. Different kinds of software are used differently and it is often necessary to pay in different ways. Bontis and Chung (2000) argue that there is no generalizable price model for all software, but it is important that how you price your software offering is connected to how well it solves the problem for the customer. In the case study all the companies use many different systems that are integrated in different ways and how they pay for services and systems vary.

Among the case companies, several were not entirely satisfied with the current price models in their integration solutions. One common opinion was that many software vendors are using out-dated price models, like the traditional model with initial license fees and inflexible terms when it comes to usage. Cusumano (2007) says that this old model with up-front license fees that you use to pay for the perpetual rights of the software is becoming less important as a source of revenue for software vendors, but according to the case companies, this type of price model is still common. Because of the complex nature of integration solutions, much IT costs also go to implementing, integrating and maintaining the different systems, by the use of consultants or internal IT resources. Support agreements from integrators and software vendors seem to be common as well. Even if old price models still exist, some of the companies use specific systems with newer models, often by paying for IT functionality as services instead of setting up systems themselves, like in invoice handling and managing information flows to and from partners and customers. There is also some of examples of out-sourcing among the case companies where whole IT departments or part of them are moved outside the company for different reasons.

The majority of the case companies seem to think that how they pay for software is of importance and is a factor when they plan their IT infrastructure. How and to what degree they find solutions with price models that match their organisations vary between the companies. The negotiation positions are very different and affect the influence they can have over the price models in their integration solutions. Even if it can be difficult as a customer to get the right price model, there are several examples in the case study that points to the importance of finding software offerings with price models that can contribute to making the work with IT in an organisation more manageable and cost effective. Looking at price models from an Anderson, Kumar and Narus (2007) perspective, the question is if a system offering can be more cost efficient or contributing to future revenue by using a certain kind of price model, and by that be more valuable.

6.3.2 Value from Price Model Dimensions

To make the analysis of the potential value of price models, the SBIFT model (Iveroth, et al., 2013) is used to divide different aspects of the current price models in use into dimensions. The problems that some interviewees have with their models are described in the dimensions and how changes can have an impact of the value of the models.

Scope

When looking from the scope dimension the integration solutions used by the companies are in most cases priced more at an attribute level, where added functionality is priced separately, often by buying or developing new systems that are integrated into the solution. The custom-built application platform in Office Supply Company is working this way where separate systems are bought when needed. The ones using different middleware solutions based as integration modules in the ERP systems from Oracle and Infor are more towards the package side of the scale as they have to have the base offering to add modules if they need more functionality. What types of costs that appear when adding new integrations points in the middleware solutions were not mentioned by the interviewees. The companies that lack a middleware solution for integrating their systems, Shoe Company and Leisure Company, are even more to the attribute side of the scale because they just add integration points when buying new systems.

Individual systems in the integration solutions are using both package and pure attribute models. In some cases the companies have to pay for some software that they do not need because of some package models, but as mentioned by Industry Tools Company, buying software separately will most likely not be any cheaper. In general, no one seems to think that having to buy more software than needed because of packaging models is a problem.

When relating this dimension to the value definition of Anderson, Kumar and Narus (2007), it is not clear that any changes would change the perceived value of the offering by cost savings. Most companies in the study seem to think that paying for functionality they actually use is preferable to big packages of functionality that may be useful in the future. The fact that much software in the integration space is module-based seems to fit the companies quite well, and that paying at an attribute level seems natural. There will always be some level of packaging but the current level does not seem to be problematic for the case companies.

Base

The base dimension of integration solutions is a bit hard to evaluate because it is often seen from a seller perspective and we do not have insight in what vendors use as base for their offerings. When looking at complex systems like integration platforms it will be difficult to compare different vendors because each offering will be unique, which make market based pricing challenging. It gets easier when looking at the base of the price when implementing systems with consultants, as the base in most cases will be the amount of consulting hours required. In the case study we did not see much indications of customer value as a base of the price, most likely that value-based pricing in these cases is very hard to do.

In general, it seems that from a buyer perspective the pricing process is far from transparent. Office Supply Company, Industry Tools Company, Clothing and Design Company and Shoe companies all have difficulties estimating the price and comparing software projects because the base of the price can be unclear from the seller' side. Increasing the value (Anderson, Kumar, & Narus, 2007) in the base dimension would not be question of moving between the different alternatives on the axis as much as making the base more obvious for the buyer. If the seller can inform the buyer of on what bases the price will fluctuate in more detail than in current offers, that may help the processes of prognosticate the cost of software projects and shortening the planning phase when buying new systems. That would increase the value of the offering for the buyer. The problem lies in what the can be done to make the pricing process more transparent.

One source of the difficulties may be derived from the complexity of many systems, especially integration platforms. If this is the case, then one way of mitigating the problem is to reduce the complexity of the software project. Johannes Bynke (2014-08-22) suggests that one solution is to do small iterative projects when implementing software, instead of big bang projects where everything is changed at once. Office Supply Company has another suggestion and would like to see some kind of trial period when they buy software, and when if it works well they can increase the use and pay more money for it. It is easy to see the benefit for the buyer with such a model but also why software vendors may be reluctant to such a model, as they risk spending resources on a customer that may not buy the software solution in the end. However as Shapiro and Varian (1999) point out, sometimes the best way of creating interest and understanding for the value of a software offering is through the experience of using it. In cases where potential customers are hesitant if the software is worth the price, giving them the opportunity to test the software in smaller scale with the possibility of ramping up the usage and price in the future may be beneficial for the sellers as well.

Another problem related to the lack of a transparent pricing process lies with the license agreements. They often have long and complex terms witch take much time for the IT deportment to handle. Clothing and Design Company feel like they are stuck with some old licenses that are hard to understand and makes calculating the costs of some systems hard. Shoe company feel like they are often content with the software itself but terms and conditions can be difficult to understand and it is hard to know what you are agreeing to.

Influence

In the influence dimension of the integration solutions, different systems will fall under different positions on the axis but the most common ones seem to be pricelist and negotiation. In smaller systems that are integrated, the price list is used and you can straightforwardly pay what the vendor is asking. The more complex systems, like integration platforms, fall under negotiation in the axis, where the customer have a saying in how much they want to pay and what to pay for. The negotiation power of each of the companies are different, and how some of the companies mitigate the risks that come with a weak position in the negotiations have been brought up in the lock-in section of the openness analysis. In most cases the negotiation position of the companies is weak they are much smaller than the big software vendors they negotiate with. One interesting exception from this is Furniture Company that was able to play out different vendors against each other and got a favourable deal in the end. In their case, an unusual organisational structure gave them the status as a valuable customer to several software vendors, which brought them negotiation power. In this case, the influence dimension of Furniture Company looks a bit different as they move more to the right where the buyer influence increases, even if it does not go as far as to result-based pricing as it still was negotiation that set the price. Result-based pricing in general seems to be absent in the price models of the case companies, and only Clothing and Design Company mentioned result-based pricing as something they are interested in without going into further details.

The influence dimension is important when analysing the value of price models as it reflects the ability the companies have to get the price model that fit them. Most of them are able to modify the software itself or change their internal workflow and processes to work with the IT systems in an efficient way, but when it comes to price models they do not have the same options. If a company finds a software solution with a price model that match the organisation, the price model may be able to increase the value (Anderson, Kumar, & Narus, 2007) of the offering. One indication that this may be the case is that the company that seems the most content with their integration solution is Furniture Company, the company with the best negotiation position for forming both the functionality of the software and the price model. They have in general very little to complain about their solution and that specific price model seems to create value for them, as different solutions would be less optimal for their organisational structure and by that more costly.

Formula

In formula dimension of integration solutions, how much the price is connected to usage is like in the other dimensions very dependent on the type of software. Many systems use the traditional license model where you pay a licence fee per user, concurrent user or workstation. This can be connected to a per unit price formula or in some cases a fixed fee + per unit price where you pay an initial fee for the software regardless on how many that uses it. There are also systems that use a fixed price regardless of volume and transaction based cost. Some new software is using newer pricing models like different SaaS solutions and subscriptions that change the formula, but overall the traditional software license model is still very common.

Some of the case companies are interested in increasing the amount of volume-based pricing in their price models, paying per transactions may in some cases be a better fit for their internal processes and workflow and make it easier to manage IT costs. Office Supply Company is one of the companies interested in more transaction based pricing and think that paying per transaction would be a way of scaling IT costs linearly with the revenue, which would be a good way of keeping control of the costs in their organisation. Another benefit this model would bring is making the internal bookkeeping easier for the IT department, where individual business units would be accounted for their usage directly. Industry Tools Company agrees with many of these points and also thinks that a more transaction based price model would fit their organisation and increase the flexibility of their business. Clothing and Design Company also think that a more volume or result based price model would be better in some cases where user-based licenses do not work that well for them. Shoe Company is using transaction-based price models in some of their portal-based integration services and are interested in the possibilities of new software service models, both from a pricing and functionality standpoint.

Furniture Company however sees potential problems with using a transaction based price model in many of their systems as that would rather decrease the cost control rather than strengthen it. They interviewee think that there are some types of system that may benefit from a more volume-based model, but in most cases that would only add costs because of the way they work. Systems that do not bring any revenue to the organisations will only add costs the more they are used with a volume-based model, and many of their systems seem to be of that type. They neither have any interest in using a transaction-based model for doing any internal bookkeeping witch also may decrease the benefit from their point of view. Leisure Product Company prefer fixed cost models because it makes it easier to calculate and prognosticate their IT costs. The total IT cost is the only thing that matters to them and if every software project just add a fixed cost it becomes easier for them to overview the size of it. It should be noted that Leisure Product Company is the smallest company and their size may contribute to their view on IT costs.

Among the case companies, there seem to exist two major views in the formula dimension. The first one exists among the companies that see benefits in having distinct connections between price and usage, and how this may contribute to a more flexible way of paying for software that may lead to cost savings for them, which lead to increased value (Anderson, Kumar, & Narus, 2007). This can also lead to organisational benefits like making the distribution of IT costs between different parts of the organisation easier. The second one exists among the companies that instead have little interest in connecting the price to usage and think that fixed cost is more beneficial when calculating total cost of IT in the organisation. Because they are more interested in total cost, they see little benefit of using the price model to track the usage and place the cost on the corresponding organisational unit.

The formula of calculating the price seems to affect the value of the price model, but the kind of formula that is beneficial depends on how organisations handle IT.

In this dimension it becomes clear that there is a connection between price models and the management of organisations, as they can influence aspects like who is responsible for IT costs that appear, how managers keep track of the costs and how this affect the IT use among the employees. Even if this is a part of the price model concept that certainly is interesting, it is not something we have focused on in this study and may be a subject of further research on price models and integration solutions.

Temporal Rights

In the temporal rights dimension of integration solutions it is also very dependent on the type of software. As most software vendors uses some kind of licence model, most of them falls in the rent and subscription category. A few also uses a perpetual model where software is bought once without any more licence fees and some SaaS services use a pay per use model. The most usual model seems to be to pay per year for a software license.

In the consumer software space we are seeing a change in this dimensions from perpetual use to more subscription-based price models in many cases. More business-oriented software have had support, service and upgrade agreements with rents and subscriptions far longer and among the case companies there seem to be quite small movements in the temporal rights dimension. Some of the case companies do use some IT services and cloud solutions like SaaS that Cusumano (2007) describes, often connected to invoice and billing. This may be seen as a form of outsourcing, where the companies increasingly pays for having services done instead of handling this with their own IT systems. Even if the value of changing the temporal rights dimension is not very clear among the case companies right now, the new price and service models may change that in the future. Cloud offerings have become more common among all types of vendors and integration is not an exception. There exist problems with SaaS as Mathew and Nair (2010) point out, some internal costs rise for the software provider and it is not clear that customers can see the added value compared to traditional software delivery, and one of the challenges for cloud providers will be to motivate the price.

6.3.3 Summary of Price Models and Value

As stated in the beginning of the price model analysis there exists no price model that fits all solutions and what model that will work well will depend on the type of software and how the IT use is structured in the organisations. When looking at the price models of the integration solutions in the case companies through the SBIFT model (Iveroth, et al., 2013), it seems that some of the dimensions can affect the potential value of the software offerings more than others. In the scope dimension, most software is priced at the attribute and the case companies do not have any direct problems with this arrangement and there are no clear changes that may affect the value. In the base dimension the price can be set by the amount of consulting hours used, there are however unclear how software vendors set their prices in other cases which leads to problems for the companies.

Changes that may affect the value of the offering will consist in making the pricing more transparent and there are a few suggestions in how to do this. Less complex terms, smaller and less complex projects and creating an option to try the software may be able to increase the value. In the influence dimension negotiation is the most common type for complex software offerings, and lack of a good negotiation position is a common problem. With increased influence comes increased value of the price model. The formula dimension consists of different software licenses with some fix and some variable parts in general. There are two major views of how to change the models to make them more valuable, either to have more volume- or transaction-based prices or to have more fixed prices. What type that is more beneficial is related to how the organisation handles IT. The temporal rights dimension uses mostly rent and subscription-based payment but there are some software with perpetual and pay-per-use models. In this dimension it is unclear what changes that could provide more value to the customer. There is a summary of the dimensions and how they can effect value in Table 6.5 below.

Table 5.5 – Summary of SBIFT dimensions and value

SBIFT dimension	Common placements on the axis in the dimension	Changes that may affect the potential value
Scope	Most software is priced at the attribute level, varying level of packaging.	No clear changes.
Base	Priced at cost when using consulting hours, unclear from a buyer perspective in other cases.	Make the pricing process more transparent. Less complex terms and agreements. Less complex projects. Option to try the software.
Influence	Negotiation in complex software offerings, pricelist in simple software offerings.	Increased buyers influence on the price model and software offering.
Formula	Per unit price formula and fixed fee + per unit price. In some cases fixed price regardless of volume and per unit price.	Higher degree of volume-based pricing in some cases, higher degree of fixed pricing in other.
Temporal rights	Mostly rent and subscription, some perpetual licenses and pay-per-use SaaS solutions.	No clear changes.

7 Conclusions

This chapter will sum up the conclusions from the analysis and relate those findings to the research questions. Its purpose is to both give a summarised version of the most important things from the analysis and to use that information to answer the research questions as clearly as possible. The chapter will be divided into three sections, one for each research question.

7.1 Benefits and Perceived Value

What connections are there between the benefits and the value of a solution?

Overall, all companies interviewed were of the opinion that the integration solutions they used, and the benefits they gained from them, were valuable to them. All companies studied regarded integration as important for their business, although to different degrees, and reported benefits of all types from the analytical model. When analysing which companies gain which benefits, we see that certain types of benefits are related. It was common to see that a benefit like improved abilities to allocate resource gave rise to productivity gains, which is another type of benefit. As the analysis continued, a link was established where managerial and organisational benefits often gave rise to further operational benefits. Thus, reporting of managerial and organisational benefits also makes it likely that a company has gained operational benefits from their integration solution.

There was also a connection between the degree of benefits reported and both the type of customers served and the complexity of the organisation. Both of the B2B companies reported that the integration solution was vital for them staying in business, which is obviously quite valuable, and this was mostly because of the fact that their customers had much greater demands for information and direct connections to the company's systems. The more complex organisations, Furniture Company and Fashion and Design Company, were also reporting more benefits than those with less complex ones. These answers indicate that these companies are either better at reaping benefits or have a higher potential value to capture, as per Smith's and Nagle's value in exchange (2005).

Further, we can see that types of benefits were not reported equally. Operational and managerial are the most common, followed by organisational, then IT infrastructure and strategic benefits. We heard no mentions that companies were not interested in gaining certain types of benefits, but there seemed to be a greater focus on the operational and managerial aspects when we discussed the solutions during the interviews. It does overall seem to be harder for organisations to gain organisational, IT infrastructure, and strategic benefits than operational and managerial. When looking at the companies that report more benefits from these less common types, we can see that the answers come from those who have put more effort into developing their systems. This fits the theoretical view that the way in which companies implement systems has an impact on how much of the potential value they can realise (Davern & Kauffman, 2000). This could thus be seen as an indication that especially organisational and strategic benefits are harder to gain, which means it is harder for the companies to be able to realise a large part of the potential value.

When looking at our empirical data, it seems to match the theory rather well. All five types of benefits provide benefits and potential value for companies. Different companies will have different maximum value potentials from systems, because of their different circumstances (Smith & Nagle, 2005). For example, we see that the complexity of the company's organisation and the demands for integration from their environment seem to affect how large a part of the potential value that can reasonably be realised by a company. Certain factors, e.g. effort during the implementation to adapt the system to the organisation's needs, affect how much of the potential value will be realised (Davern & Kauffman, 2000). For example, strategic and organisational benefits seem to put higher demand on companies to be fully realised.

7.2 Openness and Perceived Value

What connections are there between openness aspects and value?

During the analysis we saw that there were many different attitudes regarding the openness and open source aspects. Those attitudes can be said to fall in a range between two extremes: the ones who only want to purchase functionality and do not really care about the software, and those who want to modify and develop their own software. Depending on which end of the range a company is closer to, the value potential of open source will be different, which can be related to Smith's and Nagle's discussion of Value in Use (2005); those companies who are more likely to use all aspects of open source have more value to gain from it. We will now go through all five openness aspects and summarise the value potential.

For lock-in, both types have similar value potential. Both open and proprietary software lock-in are based on how many vendors or consultants are available for a given piece of software. This tends to be higher for open source, according to Thomas Rosenfall (2014-08-12), but as Leisure Product Company mentions, it is not necessarily the case that proprietary software has fewer consultants to choose from. The other aspect of lock-in is the relative power of the company versus the vendor or developer, which could point towards smaller open source vendors. However, as Fashion and Design Company and Furniture Company have shown, it is possible to get in a good power situation vis-à-vis proprietary developers as well. Lock-in in general is one of the most important openness aspects to consider from a value perspective, since the potential costs of getting stuck in a bad deal are high, which is true for both ends of the range.

For the cost and flexibility/modifiability aspects there are big differences between the ends of the range. Those companies that have a greater interest in developing or modifying software, on their own or with consultants, have a greater interest in these aspects and thus also a greater value potential from software that offers these capabilities. The TCO does not change much between license-less open source and proprietary software, but if the open source-code is used as a basis for developing a new piece of software or is modified heavily, it is instead a relatively inexpensive way of getting a code base so the development does not have to begin from scratch. Thus, those companies that want to modify or develop their own software have a larger value potential and perceived value in open source software than those that do not.

None of the companies we interviewed were very interested in the security aspects of open source, no matter which side of the range they were, which is quite surprising from an academic point of view. There are many potential reasons for this disinterest, e.g. the fact that security and stability is seen as quite good already, but no matter which these opinions indicate that security aspects of open source do not affect the perceived value of the solutions.

The community aspects of open source are not as clearly reflected on the opinion range, but rather have to do with the work processes around the software administration in the company. Those companies that handle their own software and have developers employed to stay up to date on stability and functionality concerns have a larger value potential from open source software with a vibrant community than those that do not. The community can be used to access opinions, solutions, and advice that would otherwise cost the company money to get, which is certainly valuable. If on the other hand the company is not interested in such aspects but rather just wishes to keep things running smoothly, it could be better to have someone else handling these administration tasks, which means the potential value is lower. Overall there seems to be some correlation between the in-house developers and a preference towards community support, but it is not perfectly clear.

7.3 Price Models and Perceived Value

What connections are there between price models and value?

To explore the connections between the price models and value in integration solutions, we have looked at the price models from the perspective of the SBIFT model (Iveroth, et al., 2013). The answers from the case study have been analysed from the perspective of the different dimensions in the SBIFT model and how changes in the dimensions can change the value of the type defined by Anderson, Kumar and Narus (2007). Not every company was entirely pleased with the price models and the complaints are mainly located in the dimensions *base*, *influence* and *formula.*

The problems in the base dimension are related to the fact that the pricing process lacks transparency, the companies feel that they are not sure what the price is based on; software vendors can be poor at informing about the price and give quick estimations. In the influence dimension the more complex IT systems, like integration platforms, are priced through negotiation, and the majority of the case companies feel like they have a bad negotiation position when dealing with big software vendors. In the formula dimension we see more clearly that problems come from the placement on the axis in the model. There are two groups among the case companies, the ones that want more volume- or transaction-based price models and the ones that want more fixed cost. The first group expresses an interest in using the price model as way of breaking up IT costs between organisational units while the other group is interested in the total IT cost.

The three dimensions mentioned are connected when it comes to explaining the perceived value of the price models. The influence dimension brings up the problem of uneven negotiation positions, witch limit the companies in forming price models that match their internal workflow and processes. The case study implies that companies that have higher ability to choose how they pay for software have a better chance of getting a price model that is a good fit for how the organisation work with IT, and by that becomes more valuable than other models. The base dimension brings up an aspect that is a source of irritation among several of the case companies. Without getting to know what the price for IT projects may be in a fast and efficient way, the work for the IT department gets hard and the process of buying new software more time consuming and costly than necessary. Software vendors that can create offerings that are more transparent in what the price is based on, may be able to create more valuable offerings. If the influence dimension is most important for changing into more fitting price models and the base dimension as an example of a direct flaw in current offerings, the formula dimension is the place where the actual model of pricing software can be altered to better fit the customers and increasing the value of the offering.

There are some suggestions to changes in the current models that may increase the potential value. The unclear situation in the base dimension may be mitigated either by working with smaller, less complex software project that are easier to estimate, creating less complex terms and conditions or by introducing a trial period for the offering. In the formula dimension the companies more interested in mapping cost to usage may be interested in some of the new price models brought up by Cusumano (2007), like different SaaS models that use pay-per-use or a big variable part of the price. These kinds of model are already in use in smaller systems among some of the case companies. For the ones caring about total IT costs, there may be better models that are based around fixed costs. The main takeaway is the importance of finding a model that fits the organisation. With the right price model, the IT department may have the ability to work more efficiently and reduce the costs of using their integration solutions, which would be of value for the customer.

8 Discussion

In this chapter we will discuss our thoughts on and the implications of the results, how further research can pick up on our results and continue studying the issues, our recommendations to companies using and selling integration, and finally, the generalizability of our results.

8.1 Thoughts about the Results

Most results we got in the study have matched our expectations, with one notable exception: the lack of interest in the security aspects of open source. The results from the five benefit types, operational, managerial, strategic, IT infrastructure, and organisational, did not exactly match what was found in Themistocleous' and Irani's previous study (2001a) or the expectations from the theoretical side of the subject, but were hardly that surprising. Strategic and IT infrastructure benefits were those that differed the most from the expectations, both being seemingly less important and valuable than expected from a theoretical and technological point-of-view. The benefits framework by Shang and Seddon (2000) has not been excellent, but certainly works for this kind of study. Its main problems are that the dimensions are not that clearly defined, so many benefits are rather ambiguous as to where they are supposed to go, and that there seems to be rather significant correlations between certain dimensions, such as organisational benefits giving rise to further operational benefits. Overall, the framework could earn a lot by having its dimensions described more clearly and giving more examples of what kinds of benefits goes where.

Regarding open source, we also mostly found results that matched our expectations, except for the open source security. We were expecting to see varying degrees of interest in open source software and the openness aspects we studied, but only Leisure Product Company had any interest at all in the security aspects. The theoretical material we based our five openness aspects of lock-in, cost, flexibility/modifiability, security, and community on has been very relevant for our study and has given us a firm ground to stand on. The discrepancy between reality and theory for security has been interesting to note and seems to indicate that either the companies are not fully aware of what open source entails, or that the theoretical expectations are not fully realised in many pieces of open source software. Basing our openness aspects on open source factors has been a good way of approaching the area of openness in our opinion, but there is a risk of us missing important openness aspects that are not immediately related to open source code.

As for the price models, most results matched our expectations, but it was interesting to see the indications that a price model that was more tailored to a company's internal circumstances could actually create value by itself. The SBIFT price model framework (Iveroth, et al., 2013) has overall been an excellent tool for breaking up and analysing price model aspects. It would have been interesting to actually use the framework on one specific offering to identify the type of price model used, and in more detail see how changes in certain dimensions would influence the value of the offering. In this study, it is used more as a tool to ask questions and do the analysis in a more structured way, looking at whole sets of software offerings and price models. The opinions regarding the benefits and drawbacks of volume-based pricing in the formula dimension were not something we thought about before the study, and they indicate that it exists potential to change price models in different ways to better fit certain customers.

We feel that our method for handling this study has overall been working well. The main problem was that our method for contacting companies to ask if they wished to participate did not give many case companies to study. Some other method, or just putting even more work into it, could perhaps have given better results, but since the study is explorative we still feel we got a sufficiently varied range of case companies to be able to fulfil our purpose for the report. During the interviews, it is possible that we were putting words in the interviewees' mouths, so to speak, and that that is why our results match our theoretical framework so well. We have however been aware of this risk and continuously worked to try to ask our questions as neutrally as possible and to be careful with our suggestions.

8.2 Directions for Further Research

As this is an explorative study, one of the objectives has been to find interesting parts that can serve as stepping-stones for further research.

The most interesting matters from the integration benefits analysis and conclusions to use in further research would be the connections between different benefit types and value and the correlations between benefit types. We have seen indications that there are different levels of value potential and different demands on companies to be able to realise a large part of the value potential for the different types of benefits. Studying these matters further, ideally in a more quantitative cross-sectional study, would allow researchers to see what value is actually created by integration and in which ways companies gain it. Further, there seems to be certain connections or correlations between types of benefits, e.g. organisational benefits giving rise to operational benefits, which could also be studied further. Doing so would allow researchers to improve the benefits framework, e.g. by being able to more clearly define the different dimensions.

When it comes to open source, this study has had the CIO perspective, which has given quite diverse opinions of the value of open source. This managerial view could be complemented with a more technical perspective, talking with people in organisations who are more active in developing, implementing and integrating systems and see if there is a difference in attitude towards open source. This would be interesting because there are indications, both from the case study and the expert interviews with Thomas Rosenfall (2014-08-12) and Johannes Bynke (2014-08-22) that people that are closer to the implementation are more positive towards open source in general. One of the conclusions from this study is that some of the benefit factors of open source are valuable for certain kinds of organisations, those who do much coding themselves and need the flexibility that open solutions can provide. Looking further into this topic may be of interest for further research. It would also be interesting to see what other members of senior management, even further away from the technical perspective, think about open source.

Lock-in was a factor that all the interviewees in the study had recognized and mitigating the risks of lock-in was of importance for most of them. We went into the study with arguments connected to open source regarding how to handle lock-in in integration solutions, but in several cases they had other solutions to the problem with no relation to open source. For example, commonly used proprietary software can also be a way of increasing available options of consultants, as Leisure Product Company brings up. Even if lock-in as a concept is not very new and there certainly exist research that look into it from different angles, we think that there may be room for more studies regarding how companies tackle lock-in. There may be clever ways of doing it not currently described in academic works.

One interesting thing that came up in the price model discussion is how the price model can affect the management control of organisation and specifically the questions of responsibility for IT costs. With some transaction-based models it may be easier to connect specific usage to business units or individuals in the organisation, and how that may affect IT usage may be of interest for further studies. We see that there are two major views when it comes to the formula dimension of the price model, either a model based on transaction or on fixed cost. Interestingly, the interviewees in this study have a rather risk averse view on how to change price models. Instead of talking about new ways of pricing software in terms of strategic choices where a fixed price can become economically beneficial if they think that volumes will rise or volume-based if they think that volumes will slump, they rather talk about the benefits of more cost control and making the work of cost allocation in the organisation easier. Other types of managers may have different views on that subject.

8.3 Recommendations

One of our main takeaways from the study, when talking with experts and when reading the literature, is that the success of an integration solution greatly depends on the effort a company puts into an implementation project prior to its start. For example, the companies that put more time into understanding their processes and adapting their systems before the implementation are likely to gain more organisational benefits. It is also crucial to make sure the license agreements and contracts are fully understood, so the company does not step in any unforeseen pitfalls. If the company is choosing between open and proprietary solutions for their software needs, they must make sure they have a good idea of what open source actually means and the benefits and pitfalls that come with it.

As we saw in the analysis, many companies seem focused on gaining operational and managerial benefits from their integration solutions. This is a good place to start, as those bring tangible benefits that are certainly valuable for the companies. There is however a lot more to an integration solution than those types of benefits; it seems like more potential can be realised if the company puts more effort into the other areas. For example, organisational benefits have the potential to bring about many benefits in the form of e.g. process improvements, organisational learning, and internal coordination, but to realise that value potential the company needs to make sure they know their processes and that the integration solution fully supports and improved them. Strategic benefits are another area where companies must put in more active effort to fully realise the potential value. Thus, we advise companies to make an assessment of their current integration solution to get a better idea of how those capabilities could be matched with market opportunities or customer desires, which ultimately can create a lot of value for the company.

Regarding open source we advise companies to read up and truly understand what open source really is about, so they can make better decisions regarding the matter. For a company interested in modifying their software, or perhaps even developing their own, open source can offer a lot of potential value. When deciding whether to use open source or proprietary software for a particular task, companies should assess how many vendors are available for different solutions, what bargaining position they have, and the status of the open source community for the software. If an open source project should be used for more than just a starting point of in-house software development, the quality of the community should be examined thoroughly. With a high quality community behind a software project, companies have a higher change of capturing the value of open source and may be able to reduce costs of maintenance and support and get the opportunity to create the software solutions they want. Getting stuck with software projects with dying communities may instead create costly problems and by evaluating software projects or use some kind of partner for risk sharing, these problems can be avoided.

The case companies have suggested different modifications of current price models that may be more or less feasible. From a vendor perspective they have to take their own capabilities and cost structure into consideration when creating new price models that can both add revenue and be attractive to the customer. The risk sharing aspect must come into consideration when creating price models, if risk is transferred to either the seller or the buyer there should be some kind of compensation for it. There is potential in making the base of the price more visible to the user, as the lack of clear price estimations and comprehensible terms lead to much frustration among the customers. Trial periods for some offerings may be one way of making the benefits of the systems more clear to the customer, and with the rise of cloud solutions there may be efficient ways of doing this without complex installations at the customer's location.

8.4 Generalizability

Due to the fact that we only studied six companies, only interviewed one person at each company, and also had a large number of target companies that did not wish to participate, the achieved generalizability of this study is low. The problem with the large number of target companies that did not wish to participate is that there is no way for us to confidently say that our companies reflect the target group well. Most likely, our case companies feel integration is more important than many of those who did not participate, which skews the study to the more positive side. For example, one company that did not wish to participate cited that they did not feel they had anyone with the right expertise to properly answer our questions, which could be seen as an indication that integration is not all that important to them. Other companies we contacted did not wish to discuss such things as a matter of policy, which could indicate that they believe integration is important and that how it is handled should be kept confidential.

With all of this in mind, the ability to draw conclusions about the actual state of integration for retail companies in general is very restricted. However, since the study has an explorative direction and is more concerned with finding potential connections and hypotheses to be tested in further research, that type of reduced generalizability is not that large a problem. With regard to the ability to formulate hypotheses about connections and patterns, the generalizability is much better. The companies we studied had similar backgrounds and were mostly interested in the same kinds of questions and saw similar problems. Even though we likely have not found all possible hypotheses in the area, those that have been found are likely to be valid as interesting areas of further research. This view is supported by the fact that all experts and professionals that we have spoken with regarding our results have found them to be interesting and reasonable. In the end, the results from this study should not be generalised to a larger population or used as an accurate description of reality, but are suitable as a starting points for further research.

References

AlMarzouq, M., Zheng, L., Rong, G., & Grover, V. (2005). Open Source: Concepts, Benefits, and Challenges. *Communications of the Association for Information Systems, 16*, pp. 756-784.

Anderson, J., Kumar, N., & Narus, J. (2007). *Value Merchants: Demonstrating and Documenting Superior Value in Business Markets.* Boston: Massachusetts: Harvard Business Review Press.

Bontis, N., & Chung, H. (2000). The evolution of software pricing: from box licenses to application service provider models. *Internet Research: Electronic Networking Applications and Policy*, Vol. 10, No. 3.

Brandel, M. (2010, May 10). Hidden Snags in Open Source. *Computerworld*, 18-22.

Brown, J. S., Hagel, J., & Varian, H. (2003). Does IT Matter? *Harvard Business Review*, Vol. 81, No. 7, pp. 109-112.

Callista Enterprise. (2014). *Erbjudanden: Systemintegration*. Retrieved from Callista Enterprise: http://callistaenterprise.se/erbjudanden/systemintegration/

Carr, N. G. (2003). IT Doesn't Matter. *Harvard Business Review*, Vol. 81, No. 5, pp. 41-49.

Computer Economics. (2005, May). *Key Advantage of Open Source is Not Cost Savings*. Retrieved from Computer Economics: http://www.computereconomics.com/article.cfm?id=1043

Cusumano, M. A. (2007). The Changing Labyrinth of Software Pricing. *Communications of the ACM*, Vol. 50, No. 7.

Davern, M. J., & Kauffman, R. J. (2000). Discovering Potential and Realizing Value from Information Technology Investments. *Journal of Management Information Systems*, Vol. 16, No. 4.

Dolan, R. J., & Simon, H. (1996). *Power Pricing - How Managing Price Transforms the Bottom Line.* New York City: New York: The Free Press.

Eisenhardt, K. M., & Graebner, M. E. (2007). Theory Building from Cases: Opportunitites and Challanges. *Academy of Management Journal*, Vol. 50, No. 1, pp. 25-32.

Gale, B. T. (1994). *Managing Customer Value: Creating Quality and Service That Customers Can See.* New York City: New York: The Free Press.

Hasselbring, W. (2000). Information System Integration. *Communications of the ACM*, Vol. 43, No. 6.

Hoepman, J.-H., & Jacobs, B. (2007). Increased Security Through Open Source. *Communications of the ACM*, Vol. 50, No. 1.

Iveroth, E., Westelius, A., Petri, C.-J., Olve, N.-G., Cöster, M., & Nilsson, F. (2013). How to differentiate by price: Proposal for a five-dimensional model. *European Management Journal*, Vol. 31, No 2.

Lam, W. (2005). An Enterprise Application Integration (EAI) Case-Study: Seamless Mortgage Processing at Harmond Bank. *Journal of Computer Information Systems*, Fall 2005, pp. 35-43.

Le Duc, M. (2007). Metodhandbok som tankekarta. Retrieved from http://www.leduc.se/metod/Induktion,deduktionochabduktion.html

Lehmann, S., & Buxmann, P. (2009). Pricing Strategies of Software Vendors. *Business & Information Systems Engineering*, Volume 1, Issue 6 pp. 452-462.

Lekvall, P., Wahlbin, C., & Frankelius, P. (2001). *Information för marknadsföringsbeslut, 4:e upplagan.* Gothenburg: Sweden: IHM Publishing.

Linthicum, D. S. (1999). *Enterprise Application Integration.* Boston: Massachusetts: Addison Wesley Longman, Inc.

Linthicum, D. S. (2003). *Next Generation Application Integration.* Boston: Massachusetts: Addison Wesley Longman, Inc.

Markus, M. L. (2000). Paradigm Shifts—E-Business and Business/Systems Integration. *Communications of the Association for Information Systems*, Vol. 4, No. 10.

Mathew, M., & Nair, S. (2010, April-September). Pricing SAAS Models: Perceptions of Business Service Providers and Clients. *Journal of Services Research, 10*(1).

McCarthy, E. J., & Perreault, W. D. (1991). *Essentials of Marketing, Fifth Edition.* United States of America: Von Hoffmann Press, Inc.

Messmer, E. (2013, March 25). Security of open-source software scrutinized. *Network World*, 12,14.

Morgan, L., & Finnegan, P. (2007). How Perceptions of Open Source Software Influence Adoption: An Exploratory Study. *ECIS 2007 Proceedings*, Paper 118.

Porter, M. E. (1980). *Competitive Strategy: Techniques for Analysing Industries and Competitors.* New York City: New York: The Free Press.

Radar Group. (2010). *Marknadsrapport: integration.* Stockholm: Sweden: Radar Group.

Radar Group. (2011). *Sammanställning Integration 2011.* Stockholm: Sweden: Radar Group.

Raymond, E. S. (2002, August 2). *The Cathedral and the Bazaar*. Retrieved from The Cathedral and the Bazaar: http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/

Redpill Linpro. (2014). *About us: Open Source*. Retrieved from Redpill Linpro: http://www.redpill-linpro.com/about-us/open-source

Schryen, G. (2011). Is Open Source Security a Myth? *Communications of the ACM*, Vol. 54, No. 4.

Shang, S., & Seddon, P. B. (2000). A Comprehensive Framework for Classifying the Benefits of ERP Systems. *AMCIS 2000 Proceedings*, Paper 39.

Shang, S., & Seddon, P. B. (2002). Assessing and managing the benefits of enterprise systems: the busniess manager's perspective. *Information Systems Journal*, Vol. 12, No. 4, pp. 271-299.

Shapiro, C., & Varian, H. R. (1999). *Information Rules: A Strategic Guide to the Network Economy.* Boston: Massachusetts: Harvard Business Review Press.

Smith, G. E., & Nagle, T. T. (2005). A Question of Value. *Marketing Management*, July/August, pp. 38-43.

Sprott, D. (2000). Componentizing the Enterprise Application Packages. *Communications of the ACM*, Vol. 43, No. 4.

Stallman, R. (2009). Why "open source" misses the point of free software. *Communications of the ACM*, Vol. 52, No. 6.

The Open Source Initiative. (2014, April 9). *The Open Source Definition*. Retrieved from Open Source Initiative: http://opensource.org/osd

Themistocleous, M., & Irani, Z. (2001a). Benchmarking the benefits and barriers of application integration. *Benchmarking: An International Journal*, Vol. 8, No. 4.

Themistocleous, M., & Irani, Z. (2001b). ERP and application integration - Exploratory survey. *Business Process Management Journal*, Vol. 7, No. 3.

Themistocleous, M., & Irani, Z. (2002). Novel taxonomy for application integration. *Benchmarking: An International Journal*, Vol. 9, No. 2.

Appendices

In this final part of the thesis, all material important for either the empirical gathering of data or analysis thereof has been attached, to satiate the curious reader.

Appendix I: Interview Guide

This version is a translation of the interview guide that was actually used in for the interviews, which was in Swedish.

1. Background

1. Which were the main reasons for getting your integration solution?
2. What kinds of integration platforms do you use today?
3. Approximately how many systems do you integrate today?
4. Do you have an idea of how much money/how big a part of your IT budget goes into integration?

2. Benefits

1. What value do you gain from your system with regard to organisational benefits?
    a. Better process overview and control? Better organisational learning and knowledge management? Better organisational coordination?
2. What value do you gain from your system with regard to managerial benefits?
    a. Better resource management? Better quality of information for decision-making? Better management control capabilities?
3. What value do you gain from your system with regard to strategic benefits?
    a. New possibilities to sell your products? New services you can offer customers? Strategic goal fulfilment?
4. What value do you gain from your system with regard to technical benefits?
    a. More flexible IT infrastructure? Lower IT costs? Better IT operations overview?
5. What value do you gain from your system with regard to operational benefits?
    a. Cost savings from e.g. increased automation? Higher process quality? Increased productivity?
6. Are there any benefits you have experienced that we have not mentioned?

3. Openness

1. Is open source software important when you buy new systems?
2. Did you consider openness aspects when choosing your current system?
3. What value does the following open source aspects have:
    a. Free/low initial license cost
    b. Less lock-in to a single supplier
    c. Flexibility/easier to modify than proprietary systems
    d. Security/bug-fixes can be handled by a community
    e. Support can be handled by a community
4. With regard to openness, are you happy with your current solution?

4. Price Model

1. How do you pay for your integration solution today?
    a. Package price or separate services?
    b. How is the solution priced? (list price, result-based, negotiation, other factors)
    c. What is the final price based on? (fixed cost, variable cost, fixed + variable)
    d. What kind of agreement? (License, leasing, subscription, rented, pay for each use)
2. Are you happy with this mode of payment? Does the price model fit your business?
3. Was the price model a factor when choosing your current solution?

5. Wrap-up

1. How valuable do you think your integration solution has been in general?
2. What have been the worst things in your current solution?

Appendix II: Case Data

This appendix contains an aggregation of the case data from Themistocleous' and Irani's article "Benchmarking the benefits and drawbacks of application integration" (2001a). Note that the sorting of the benefits into types was made by the thesis authors.

Benefit	Category	% of cases
Provides more understanding and control of processes	Organisational	13.3% (2/15)
Improves management and supports decision making	Managerial	13.3% (2/15)
Results in more organised business processes	Organisational	6.67% (1/15)
Allow organisations to do business more effectively	Managerial	20.0% (3/15)
Improves planning in supply chain management	Managerial	6.67% (1/15)
Increases collaboration among partners	Strategic	6.67% (1/15)
Reduces lost sales	Operational	13.3% (2/15)
Increases productivity	Operational	6.67% (1/15)
Increases performance	Managerial	6.67% (1/15)
Achieves customer satisfaction	Strategic	26.7% (4/15)
Results in reusable systems, components, and data	IT Infrastructure	33.3% (5/15)
Reduces redundancy in of applications, data, and tasks	IT Infrastructure	6.67% (1/15)
Reduces cost	Operational	53.3% (8/15)
Achieves return on investment	Managerial	33.3% (5/15)
Faster and cheaper implementations than bespoke solutions	IT Infrastructure	20.0% (3/15)
Increases flexibility	Organisational	53.3% (8/15)
Quicker response to change	Organisational	6.67% (1/15)
Offers interfaces standardisation	IT Infrastructure	6.67% (1/15)
Provides flexible, maintainable, and manageable solutions	IT Infrastructure	66.7% (10/15)
Results in reliable data	Managerial	6.67% (1/15)
Process and systems scalability	IT Infrastructure	13.3% (2/15)
Provides portability	IT Infrastructure	6.67% (1/15)
Reduces development risks	IT Infrastructure	6.67% (1/15)
Achieves non-invasive solutions	IT Infrastructure	13.3% (2/15)
Achieves process integration	Organisational	6.67% (1/15)
Increases data analysis	Managerial	20.0% (3/15)
Improves data quality	Managerial	13.3% (2/15)
Supports efficient data sharing	IT Infrastructure	6.67% (1/15)

Category of benefit	Number of benefits	Average per cent
Operational	3	24.4 %
Managerial	8	15.0 %
Strategic	2	16.7 %
IT Infrastructure	10	16.7 %
Organisational	5	17.3 %