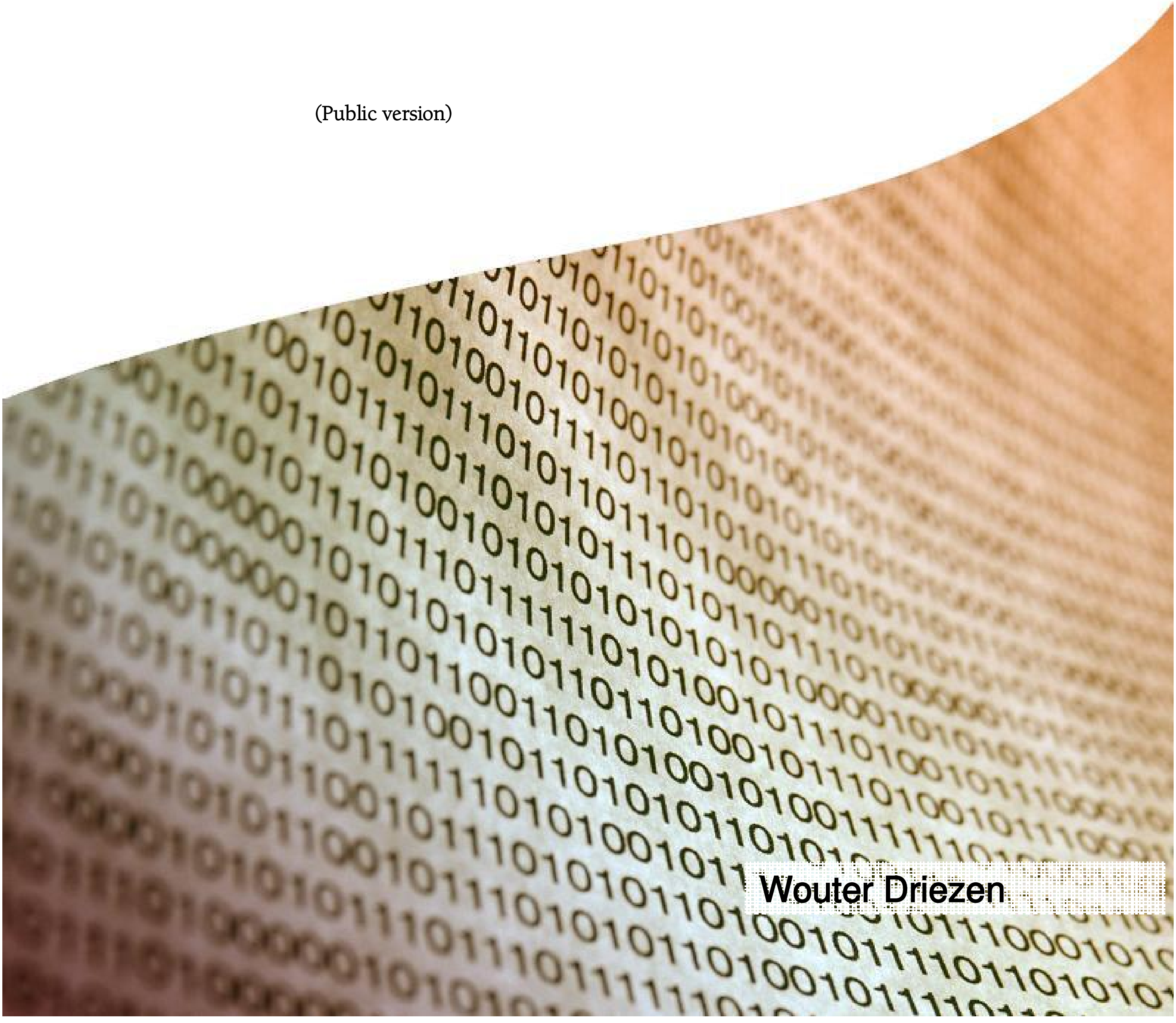
Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

Service-oriented and open-source software as potential markets for applications management

Analysis of the possible added value of Capgemini Outsourcing's Dutch Applications Management Service Center

(Public version)



Wouter Driezen

Service-oriented and open-source software as potential markets for applications management

Analysis of the possible added value of Capgemini Outsourcing's Dutch Applications Management Service Center

(Public version)

Author:	Wouter Driezen (University of Twente, The Netherlands)
Committee:	Prof Dr Jos van Hillegersberg (University of Twente) Prof Dr Hans E. Roosendaal (University of Twente) Ir Annet Harmsen (Capgemini Outsourcing)
Place:	Utrecht
Date:	January 8, 2008
Version:	2.1
Type:	Final

To my parents

Preface

Information technology (IT) develops at an astonishing pace. You probably know a relative or colleague who could not keep up with the perpetuating stream of developments and has now fallen hopelessly behind. Maybe even if you compare your proficiency at using the computer to that of youngsters these days… But the ongoing developments do not only affect the end-users of IT. There also exist a whole supporting industry that needs to keep up in order to stay in business.

One of those supporting industries is the outsourcing industry: those companies that take over the care for the IT of other companies in exchange for some remuneration. This research is about that part of the outsourcing industry that takes care of the software-side of IT, the so called applications management. The providers of applications management services are, too, confronted with the developments in IT. For example, newly developed technologies may offer them new sources of revenue, but they may just as well prove to be a threat to their current business. The applications management providers therefore need to reassess their competitive position on a regular basis to adjust their competitive strategies to a changing environment.

The rise of service-oriented software and the increasing attention that is paid to open-source software as a feasible replacement for proprietary software are two of such developments. In this research it is analyzed how the case organization, the AMSC, can add value to the applications management markets for service-oriented and open-source software.

The fact that only my name is mentioned on the title page does not imply that this report is the result of my sole efforts. Many have contributed to a greater or lesser extent to this research, some of which I would like to thank here. First of all, I would like to express my gratitude to Prof. Dr. Van Hillegersberg and Prof. Dr. Roosendaal, both associated with the University of Twente, for their help and critical comments during the research process. Your inputs have greatly supported me, and they have, so I believe, greatly improved the research itself. Also, I would like to thank Mrs. Annet Harmsen, my company supervisor, for her inputs and her great engagement. I am very grateful for the enthusiasm with which you have carried out your supporting task. Furthermore, I would like to thank Mr. Sjoerd Wittebrood and all other people who have contributed their insights, for they have helped me a lot by doing so. In this, I would also like to thank Mr. Ron Tolido, Capgemini's Chief Technology Officer for the Northern Europe and Asia-Pacific regions, for sharing his views. On a more personal note, I would like to thank Jan-Joost Stok for being such great companionship throughout my time at Capgemini. I have really appreciated our daily hot chocolate breaks during which we discussed our individual research assignments—or the garishly colored building across the freeway if some distraction was needed more. Finally, I owe special thanks to my parents and to Alyt for sharing with me the emotional peaks and lows I experienced during this research.

This research is written for the scientific community at large and in particular for Prof Dr Van Hillegersberg and Prof Dr Roosendaal. Both professors will use this research to judge whether my academic skills are worthy of a Master's title. Secondly, this research is written for the management of the case organization, the AMSC, as an aid in their effort to integrate service-oriented and open-source software in re-determining their competitive strategy.

I hope this research will be informative and valuable to you.

Wouter Driezen
Utrecht, January 2008

Management Summary

Research problem

Providers of applications management services operate in a dynamic environment and need to rethink their strategies on a regular basis. The case organization, Capgemini Outsourcing's Dutch Applications Management Service Center (AMSC), is one such applications management provider that is to re-determine its competitive strategy to adapt it to the latest changes in the applications management market. In response, this research is to answer the following research question:

> What value can the AMSC add to the changing applications management market?

The 2007 edition of the International Conference on Software Maintenance has identified the rise of service-oriented software and open-source software as the critical issues for the software maintenance community, of which applications management providers are industrial representatives. Therefore, this research will investigate, besides the current market and the current added value of the AMSC, the possible added value of the case organization to the new markets for applications management that these two types of software bring about.

Current applications management market and the AMSC

It is found that in the current applications management market demand is non-distinctive, demanded by clients seeking mostly operational efficiency. Furthermore, it is found that there are many equivalent providers, that there exists a considerable threat of entry by new players, and that there exists a potential threat that clients will have applications management performed in-house again. As this market offers little room for differentiation, the applications management providers compete on prices.

[Confidential content was removed]

Findings regarding applications management for service-oriented software

The analysis of service-oriented software itself shows that, although many benefits are attributed to it, none of those are supported by empirical research findings. Therefore, the benefits of service-oriented software for its users are considered to be uncertain.

Providing applications management for service-oriented software is found to require a number of new or different competences of the applications management provider, because service orientation affects the work in a number of ways. The amount and the complexity of applications management work may increase relative to traditional applications management. Furthermore, some aspects of the work itself may be different. Also, the value chain of which the work is part may be more complex.

The applications management market for service-oriented software is found to be attractive, because it grows rapidly (about 20% yearly), and because it may serve more strategic needs of clients.

[Confidential content was removed]

Findings regarding applications management for open-source software

Proponents of open-source software claim its superiority over proprietary software. However, the analysis of open-source software shows that such a claim is not irrefutably supported by empirical findings. The analysis also finds that open-source software is most viable for generic, context-independent applications, but not for—highly—specific business applications. Yet, these form the general subject of applications management services. Therefore, the relevance of

open-source software as a market for applications management services is considered to be very limited.

On the other hand, providing applications management for open-source software is found to require no to few new competences from the applications management provider, because managing and maintaining open-source software is little different from traditional applications management.

Besides the finding that the applications management market for open-source software is very limited in size, it is also found that it is an unattractive one. This is because its clients mostly seek operational effectiveness and because four of the five forces that form its competitive environment are less favorable than in the traditional applications management market (only rivalry is considered to be equal). Nonetheless, the market is forecasted to grow 5 to 10% yearly.

[Confidential content was removed]

Conclusions

[Confidential content was removed]

Recommendations

[Confidential content was removed]

Management Summary (Dutch)

Probleemstelling

Aanbieders van applicatiemanagementdiensten opereren in een dynamische markt. Als gevolg daarvan moeten zij op regelmatige basis hun competitieve strategieën opnieuw bepalen. De case-organisatie, het Nederlandse Applications Management Service Center (AMSC) van Capgemini Outsourcing, is zo'n aanbieder van applicatiemanagementdiensten die zijn strategie moet aanpassen aan de laatste ontwikkelingen in de markt voor applicatiemanagementdiensten. Naar aanleiding hiervan zal dit onderzoek antwoord geven op de volgende onderzoeksvraag:

Welke waarde kan het AMSC toevoegen aan de veranderende markt voor applicatiemanagement?

Op de International Conference on Software Maintenance van 2007 werden de opkomst van service-georiënteerde software en open-source software aangeduid als de kritieke ontwikkelingen voor de gemeenschap die zich bezighoudt met softwareonderhoud, waarvan aanbieders van applicatiemanagementdiensten industriële vertegenwoordigers zijn. Daarom zal in dit onderzoek onderzocht worden, naast de huidige markt en de toegevoegde waarde van het AMSC daarin, wat de toegevoegde waarde van de case-organisatie kan zijn in de nieuwe markten voor applicatiemanagementdiensten die door de opkomst van deze twee typen software ontstaan.

Bevindingen ten aanzien van de huidige markt en het AMSC

Dit onderzoek wijst uit dat de klanten in deze markt dezelfde diensten vragen en vaak om redenen van operationele efficiency. Bovendien zijn er veel gelijkwaardige aanbieders, is er een grote dreiging dat nieuwe partijen de markt betreden en bestaat er een gerede dreiging dat klanten hun applicatiemanagement weer zelf gaan doen. Omdat deze markt weinig ruimte biedt voor differentiatie concurreren de aanbieders van applicatiemanagementdiensten op prijs.

[Confidential content was removed]

Bevindingen ten aanzien van applicatiemanagement voor service-georiënteerde software

De analyse van service-georiënteerde software *an sich* wijst uit dat veel voordelen toegeschreven worden aan service-georiënteerde software of het gebruik daarvan, maar dat er geen empirische bevindingen zijn om deze uitspraken te rechtvaardigen. Daarom worden de voordelen van service-georiënteerde software voor zijn gebruikers zeer betwistbaar geacht.

Het aanbieden van applicatiemanagementdiensten voor service-georiënteerde software vergt een aantal nieuwe of andere competenties van de aanbieder, zo wijst dit onderzoek uit, omdat service-oriëntatie in software het werk op een aantal wijzen beïnvloedt. De hoeveelheid werk en de complexiteit daarvan kunnen groter zijn dan bij traditioneel applicatiemanagement. Ook sommige aspecten van het werk zelf kunnen anders zijn. Bovendien kan de waardeketen waar het werk deel van uitmaakt complexer zijn.

De markt voor applicatiemanagementdiensten voor service-georiënteerde software is aantrekkelijk, zo blijkt uit dit onderzoek, omdat hij snel groeit (jaarlijks ongeveer 20%) en omdat hij invulling kan geven aan meer strategische wensen van klanten.

[Confidential content was removed]

Bevindingen ten aanzien van applicatiemanagement voor open-source software

Voorstanders van open-source software claimen dat het superieur is aan in eigendom ontwikkelde software. Maar analyse van open-source software *an sich* toont aan dat er geen empirische bevindingen zijn die dit onomstotelijk kunnen onderbouwen. De analyse toont bovendien aan dat open-source software het meest geschikt is voor generieke, context-onafhankelijke applicaties, maar niet voor specifieke bedrijfsapplicaties. Terwijl juist voor die laatste groep applicatiemanagementdiensten bestemd zijn. Daarom wordt de markt voor applicatiemanagementdiensten voor open-source software als zeer beperkt gekwalificeerd.

Niettemin vergt het aanbieden van applicatiemanagementdiensten voor open-source software weinig tot geen nieuwe competenties van de aanbieder, zo wijst dit onderzoek uit, omdat het onderhouden en beheren van open-source software weinig anders is dan traditioneel applicatiemanagement.

Naast dat de markt voor applicatiemanagementdiensten voor open-source van zeer beperkte omvang is, blijkt dat het bovendien een onaantrekkelijke markt is. Dat komt doordat de klanten ervan voornamelijk kostenbesparing zoeken en doordat vier van de vijf krachten die de competitieve omgeving vormen minder gunstig zijn dan in de traditionele markt voor applicatiemanagementdiensten (alleen onderlinge competitie wordt als gelijk verondersteld). Desalniettemin, de voorspelde marktgroei bedraagt tussen 5 en 10% per jaar.

[Confidential content was removed]

Conclusies

[Confidential content was removed]

Aanbevelingen

[Confidential content was removed]

Table of Contents

1 Introduction

1.1 Research cause

In business the environment changes over time. Companies, having formulated strategies to position themselves in their environment, have to revise their strategies every once in a while to address such changes. Being able to properly adapt the organization to changing circumstances is a key to success; not doing so can lead a once fruitful product to commercial barrenness. On the other hand, being the first to tap a newly discovered source can be greatly rewarding for a company. Therefore, formulating strategy should be an ongoing process, constantly seeking to position the firm to the dynamics of customer demands, competitors' actions, or emerging technologies.

The applications management industry is a dynamic industry. In providing their services (managing and maintaining client organizations' business applications) the applications management providers continually need to assess the latest buzzes, trends, and hypes that are so common across all IT-related industries, and adapt their competitive strategies to the ones they consider significant. This continuous review of competitive strategies has incited this research.

1.2 Research questions

One such applications management providers is Capgemini Outsourcing's Dutch Applications Management Service Center (AMSC). The AMSC too is currently trying to determine how it can continue to add value as the applications management market changes. As such, it will serve as a case organization for this research. Therefore, the central question of this research is:

> What value can the AMSC add to the changing applications management market?

The most imminent changes in the applications management industry have been determined by analyzing the program of the International Conference on Software Maintenance (ICSM), the forefront of research in this field of practice. The ICSM describes itself as a platform where "participants from academia, government, and industry share ideas and experiences solving critical software maintenance problems". [1] The 2007 edition of the conference[2] is themed "Moving towards service-oriented architectures", stressing both the topicality and the importance of the rise of service-orientation in software. But it is not only the maintenance of service-oriented software that deserves special attention according to the conference program; among discussions on all sorts of maintenance techniques and organizing principles, a second kind of software is found that, evidently, is deemed critical enough to be discussed at this conference even though it is dedicated to service-oriented software. That second sort of software is open-source software. As a result of this analysis of the conference program, the rise of service-oriented software and the rise of open-source software are considered the most critical changes in the applications management market.

This research will focus on how the AMSC can add value to the new markets for applications management that emerge due to the rise of these two types of software. The research question will be answered by taking the current applications management market as a starting point and analyzing the two critical changes in the applications management market from a competitive strategy point of view, that is, analyzing the possibilities for the AMSC to add value to the applications management market for service-oriented software and that for open-source software.

[1] http://conferences.computer.org/icsm/

[2] http://icsm07.ai.univ-paris8.fr/

Scope

The markets under consideration in this research are the Dutch markets for applications management for the different types of software. Such geographical demarcation is justified because most major applications management providers operate from local offices. Furthermore, the Dutch market is considered a mature outsourcing market (Takahashi 2006). This means that it is not troubled by the teething troubles of a young market, making it a better unit for analysis.

1.3 Theoretical framework

Determining what value to add to a specific market is an exercise of strategy-making. But how is such an exercise performed? It is the academic field of strategic management that is devoted to answering that question.

The field of strategic management has produced a number of different perspectives. Mintzberg et al. (Mintzberg et al. 1998) divide the different schools of thought into two categories: prescriptive and descriptive ones. The prescriptive schools present a view as to how strategy-making should be done; the descriptive schools describe how strategies actually do get made. Because this research is about determining competitive strategies for the AMSC, the insights of the prescriptive schools will be used. The descriptive schools will not be used, because those look back on already established strategies, which not the case in this research.

The most influential prescriptive school of thought is the *design* school (Mintzberg et al. 1998). This school seeks to attain a fit between internal capabilities and external possibilities, which is reflected in its basic model: the SWOT-analysis. This simple model assesses the strengths and weaknesses of the organization in the light of the opportunities and threats in its environment (Mintzberg et al. 1998). This model will form the basis for this research.

Internal analysis

The first part of the process of seeking a fit as mentioned above is the analysis of the organization itself, in terms of its strengths and weaknesses. In this research this analysis will be based on the strengths and weaknesses that a number of employees of the case organization, operating in different functions and on different hierarchical levels, have identified.

External analysis

The second part of the fit-seeking process is to identify the opportunities and threats in the external environment of the organization. The external environment has been the determinant in the latest wave of another prescriptive school, the *positioning* school (Mintzberg et al. 1998). This school attempts to position a company in one of a set of predefined strategies, depending on the outcomes of an analysis of its environment. Although this school has been criticized for neglecting the creative aspects of strategy-making and reducing the infinite number of unique strategies allowed by the design schools to only a limited number, it has produced some useful techniques for analyzing the environment. While these techniques may not result in a strategy themselves, they may well provide supporting input for the actual strategy-making process. Therefore, in this research—among other things—two well-known techniques stemming from the positioning school—Porter's five forces framework (Porter 1980) (see: Figure 1) and his value chain concept (Porter 1985)—will be used to analyze the external environment of the case organization. The external environment in this research is limited to the two emerging markets of applications management for service-oriented and open-source software, but to have a standard for comparison the current—traditional—market for applications management will also be analyzed with the same techniques.

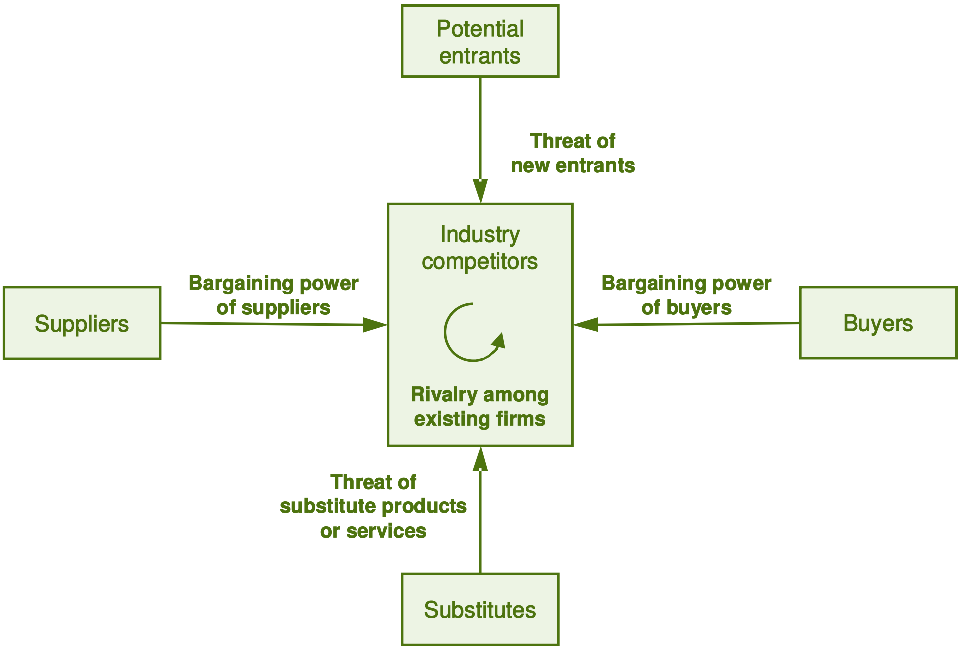

Potential entrants
Threat of new entrants
Industry competitors
Bargaining power of suppliers
Suppliers
Bargaining power of buyers
Buyers
Rivalry among existing firms
Threat of substitute products or services
Substitutes

Figure 1: Five forces framework (Porter 1980)

1.4 Sub-questions

To answer the research questions the following sub-questions have been formulated:

1. What is the current market for applications management services?
    a. What is the demand in this market?
    b. What does the value chain of this market look like?
    c. What does the competitive environment of this market look like?

2. What is the AMSC?
    a. What is its competitive strategy?
    b. What is its value offer?
    c. What are its competences?
    d. What are its strengths and weaknesses?

3. What will the market for applications management for service-oriented software look like?
    a. What will the value chain of this market look like?
    b. In what way will applications management be different?
    c. What new competences are required?
    d. What will be the demand in this market?
    e. What will the competitive environment of this market look like?
    f. What opportunities will this market offer and what threats will it pose?
    g. What value can the AMSC offer to this market?

4. What will the market for applications management for open-source software look like?
    a. What will the value chain of this market look like?
    b. In what way will applications management be different?
    c. What new competences are required?
    d. What will be the demand in this market?
    e. What will the competitive environment of this market look like?
    f. What opportunities will this market offer and what threats will it pose?
    g. What value can the AMSC offer to this market?

In the below tables these sub-questions are described in greater detail.

The current applications management market

		Research type	Research design	Research object
1.a	What is the demand in this market?	Descriptive	Literature research and analysis of statistical data	Description of the demand in this market in terms of: - Market size - Motives for outsourcing
1.b	What does the value chain of this market look like?	Descriptive	Document research and interviews	Description/depiction of the value chain
1.c	What does the competitive environment of this market look like?	Descriptive	Interviews, document and literature research	Description of the competitive environment in terms of the five forces framework (Porter 1980)

Table 1: Details of sub-questions regarding the current applications management market

The AMSC

		Research type	Research design	Research object
2.a	What is its competitive strategy?	Descriptive	Document research and interviews	Identification of its competitive strategy
2.b	What is its value offer?	Descriptive	Document research	Overview of its services
2.c	What are its competences?	Descriptive	Document research and interviews	Description of its competences
2.d	What are its strengths and weaknesses?	Exploratory	Interviews	Overview of its strengths and weaknesses

Table 2: Details of sub-questions regarding the AMSC

The market for applications management for service-oriented software

		Research type	Research design	Research object
3.a	What will the value chain of this market look like?	Exploratory	Literature research	Description/depiction of the value chain
3.b	In what way will applications management be different?	Exploratory	Literature research and interviews	Identification of differences
3.c	What new competences are required?	Exploratory	Previous analyses	Identification of newly required competences
3.d	What will be the demand in this market?	Exploratory	Literature and document research	Description of the demand in this market in terms of: - Market size - Motives for outsourcing
3.e	What will the competitive environment of this market look like?	Exploratory	Interviews, document and literature research	Description of the competitive environment in terms of the five forces framework
3.f	What opportunities will this market offer and what threats will it pose?	Exploratory	Literature and document research, the previous analyses	Opportunities and threats of this market
3.g	What value can the AMSC offer to this market?	Exploratory	SWOT-analysis	Overview of possibilities for the AMSC to add value to this market

Table 3: Details of sub-questions regarding the applications management market for service-oriented software

The market for applications management for open-source software

		Research type	Research design	Research object
4.a	What will the value chain of this market look like?	Exploratory	Literature research	Description/depiction of the value chain
4.b	In what way will applications management be different?	Exploratory	Literature research and interviews	Identification of differences
4.c	What new competences are required?	Exploratory	Previous analyses	Identification of newly required competences

		Research type	Research design	Research object
4.d	What will be the demand in this market?	Exploratory	Literature and document research	Description of the demand in this market in terms of: - Market size - Motives for outsourcing
4.e	What will the competitive environment of this market look like?	Exploratory	Interviews, document and literature research	Description of the competitive environment in terms of the five forces framework
4.f	What opportunities will this market offer and what threats will it pose?	Exploratory	Literature and document research, the previous analyses	Opportunities and threats of this market
4.g	What value can the AMSC offer to this market?	Exploratory	SWOT-analysis	Overview of possibilities for the AMSC to add value to this market

Table 4: Details of sub-questions regarding the applications management market for open-source software

1.5 Report structure

In the following the questions will be answered. First, the current applications management market will be analyzed and characterized. Then, a description of the case organization, the AMSC, and its strengths and weaknesses will be given. Subsequently, the applications management markets for service-oriented and open-source software will be detailed, each completed with a SWOT-analysis determining how the AMSC can add value to them and with recommendations regarding this matter. The report is ended with conclusions and some final recommendations.

2 Current applications management market

2.1 Introduction

This research will analyze the possible added value of the AMSC in the applications management markets for service-oriented and open-source software. The current applications management market will serve as a standard to which both markets can be compared. In the following, the current applications management market will be analyzed.

The current market offers applications management services to the business applications of client organizations. These services typically consist of maintaining, enhancing, and supporting business applications, resulting in a very simple value chain: services are delivered to the client, which remunerates the provider. The competitive environment of the industry is characterized by non-distinctive demand, many equal service providers, considerable threat of entry by new players, and the potential threat that clients will have applications management performed in-house again. With little room for differentiation, competition is price-based. Though the market is mature and still growing, the main motive for outsourcing remains to achieve greater operational efficiency.

First, the typical services offered will be described. Subsequently, the value chain of those services will be analyzed. Next, the size of the market and the main motives for outsourcing applications management it addresses will be described. After that, the competitive environment will be mapped.

2.2 Applications management services

Ever since the early days of machine-computing software applications have been supporting business processes. Back then, applications were written to perform single operations on a mainframe computer. Nowadays, enterprise resource planning packages can integrate a great part of all business processes in a company. Few businesses are left today that do not make use of some piece of software to support their operations.

But as with all tooling, acquiring such support is not a case of just a single deployment. Even though business applications are not subject to wear, they do need to be looked after. For example, the initial installation can be defective, containing bugs that need to be repaired. But also changes of a different nature can be needed to keep the software application up and running. Also, the end-user may need to be assisted with the operational use of the applications.

These issues are the ones addressed by applications management services. Levina and Ross (2003) defined applications management as the 'ongoing maintenance, support, and enhancement activities of all or part of a firm's application portfolio'. These activities will be described in more detail below.

2.2.1 Software maintenance and enhancement

As stated above maintaining software comprises different sorts of activities. Since the early 70s several attempts have been made to classify these different types of activities. An early effort by Swanson (Lientz and Swanson 1980) took the software system's owners' / users' intention in requesting the maintenance work as a starting point, resulting in a mutually exclusive and exhaustive typology. The three intentions he identified were, in summary (Chapin et al. 2001):

- To perfect the system in terms of its performance, processing efficiency, or maintainability ('perfective maintenance');

- To adapt the system to changes in its data environment or processing environment ('adaptive maintenance');

- To correct processing, performance, or implementation failures of the system ('corrective maintenance')[3].

Studies show that most effort is devoted to incorporating new user requirements. Lientz and Swanson (1980) show in their widely cited survey that about 75% of all maintenance effort is put in adaptive and perfective changes. Corrective changes accounted for 21%. Many subsequent studies resulted in similar figures (Bennet and Rajlich 2000).

While maintenance refers to maintaining the status quo of a software system (that is, ensuring that a piece of software keeps delivering the desired or agreed functionality), enhancement is associated with altering the functionality delivered by the application.

Development process

Managing an application typically involves developing a solution that changes the software in some desired way. This development process can generally be divided in a number of activities. Commonly, the first step is to analyze the desired situation. This can be to determine the impact of the desired change on dependent systems, or to find the cause of an error that needs fixing. Next, the change needs to be devised, after which it can be build. Subsequently, the built change needs to be tested, before it can, finally, be implemented. This process is not necessarily sequential, for not all proposed changes pass testing and need some rework or redesign (see: Figure 2).

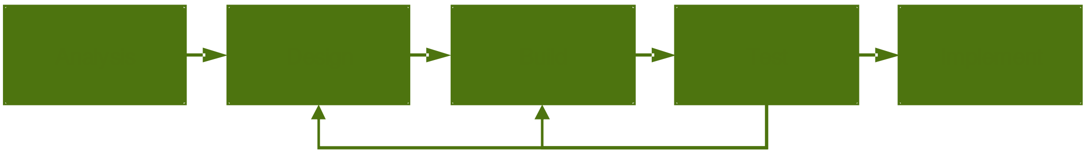


Figure 2: Activities in the development process

2.2.2 Support

Next to software maintenance and enhancement applications management services also comprise the support of the end-users of an application. This support can consist of answering a user's technical or functional questions about the software, or performing ad-hoc queries on the system. Typically, a help-desk is set up to this end.

2.3 Value chain

The applications management services delivered to the client are the result of an accumulation of value-adding activities that form the final product. To model the accumulation of added-value that forms the eventual service the concept of value chains (Porter 1985) is used, because modeling the value-adding activities as a chain makes identifying business opportunities very easy. After all, each activity represents some added-value and, hence, a potential business.

Typically, applications management services are provided to the customer, which, in return, compensates the service provider. Although this process consists of the delivery of services (as opposed to physical products), in some cases the service provider has to acquire some inputs to complete the service—for example, when the service provider also takes care of the software licenses, which it needs to acquire from the original software vendors. But if the applications

[3] Although the terminology ('corrective', 'adaptive', and 'perfective') is widely adopted among researchers and practitioners, few use it as Swanson defined it. Nowadays, in literature and in industry, a great variety of definitions exist (Chapin et al., 2001), making the terminology ambiguous. In this research no attempt will be made to resolve this issue.

management services provider is maintaining a custom-build application, such inputs generally are absent and the final product is the result of only the applications management provider's value-adding activities. The typical value chain for the applications management market is depicted below (see: Figure 3).

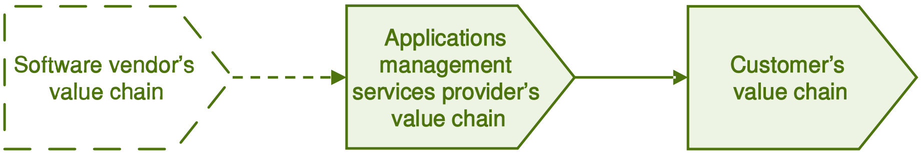

Software vendor's value chain
Applications management services provider's value chain
Customer's value chain

Figure 3: Value chain for the applications management market

2.4 Market size

The size of the market for applications management put up for outsourcing in the Netherlands, expressed in annual revenues, is approximately EUR 1.2 billion. For the upcoming years the market is forecasted to grow circa 12% yearly (Pierre Audoin Consultants 2006).

2.5 Motives to outsource applications management

Smith and McKeen (2004) identified three distinct but complementary approaches to outsourcing. The first approach dates back to the late 1980s and relates to outsourcing the "utility" functions of IT. The objective of this approach is to increase operational efficiency, because the outsourcer can bundle the non-distinctive IT functions of several companies and thus offer economies of scale.

The second approach uses outsourcing for tactical support, starting from the mid-nineties. By outsourcing mature IT functions companies can free their own IT personnel for new projects. Also, calling in an outsourcer is applied as a means of quickly obtaining new skills and technologies. The primary driver has shifted from cost reduction towards flexibility and responsiveness, though costs still remain important.

As from the turn of the century, the third approach arose. This approach applies outsourcing for strategic impact. Whole business processes that are considered non-core are being outsourced. In this way, companies can get full functionality without needing to develop the capabilities themselves.

Achieving operational efficiency is still by far the major driver for outsourcing in the Netherlands (Tramacere 2007). A recent Gartner benchmark study argues that maintaining and enhancing an application after its initial deployment counts for about 71% of the total cost of ownership (Duggan et al. 2005), thus offering a great savings potential.

2.6 Competitive environment

In strategic management literature one widely supported view is that strategy making should position a company in its environment. The careful analysis of its environment can provide valuable insights as to how to address the market. To gain insight into the competitive environment of the market for applications management services Porter's five forces framework (Porter 1980) will be used. Originally developed to assess the attractiveness of a market, this widely acknowledged framework identifies possible sources of competition in a market. Carefully analyzing these possible sources provides an understanding of the forces in the market.

To enlarge the empirical validity of this analysis several 'hands-on' experts have been consulted on the matter. These 'hands-on' experts were asked to assess each of the five forces on a five-point scale, ranging from very little to very great. Also, they were asked to give argumentation for their assessments. All this was asked in writing (see: Appendix A). The people consulted are all employed at the case organization, Capgemini Outsourcing Netherlands (OS), but they hold a variety of positions. Mr. Piet de Graaf is deputy manager of the Applications Management

Service Center (AMSC), and as such he is involved in agreeing deals with clients. Mr. Rob Bindt is responsible for supplying OS Sales with cost calculations for use in bid processes, and Mr. Marco van der Vet is OS Sales manager. Also Mr. Peter Schreurs was consulted. He is employed as OS Recruitment manager. All these people have direct experience with—aspects of—the competitive environment of the applications management services market.

The below analyses of each of the five forces are the outcome of their insights as well as insights from literature.

2.6.1 The bargaining power of buyers

Buyers will always attempt to receive maximum value at minimum costs. By forcing down prices, bargaining more or better services, or simply playing competitors against each other, buyers seek to attain the best offer they can get. Their ability to actually do so is called their bargaining power.

In the applications management market the buyers are typically large and medium-size companies or government institutions. Porter (1980) has identified a number of circumstances that cause the bargaining power of buyers to be great.In the below table the presence of these circumstances in the current market are listed (see: Table 5). A plus sign represents a strong presence, a minus sign a weak presence.

	Concentrated or large purchase volumes	Product represents a significant fraction of buyer's costs / purchases	Product is undifferentiated	Buyer faces few switching costs	Buyer earns low profits	Credible threat of backward integration	Product is unimportant to the buyer's product	Buyer has full information
Presence in current market	+		+	+		+		+

Table 5: Presence of circumstances that foster great buyer bargaining power in the current market

These findings are explained below. Also, it is noted how many of the 'hands-on' experts mentioned the presence of each of the circumstances, if applicable.

- Typically, the buyers acquire applications management services for several years at once, agreeing multi-year contracts with the service providers. By bundling their service requirements into a single acquisition the buyers artificially increase their purchase volume. With such bundled purchase volumes they are able to insist on better prices than when acquiring services in multiple smaller volumes. (Mentioned by 1 'hands-on' expert)

- The services they acquire are rather undifferentiated, especially now the value offers of the Indian and western service providers are converging (Moore and Martorelli 2007). This gives the buyers the assurance that they can always go elsewhere, thus being able to play suppliers against each other. (Mentioned by 2 'hands-on' experts)

- Because they acquire non-differentiated services, switching costs (the one-time costs the buyer incurs when switching from supplier) are rather low. This makes it easy for buyers to trade in their existing service provider when another one offers a better deal.

- The buyers pose a credible threat of backward integration. That is, they still hold the feasible option of doing the applications management themselves (Overby 2003). (Mentioned by 1 'hands-on' expert, albeit as a feasible substitute)

- The buyers are well-informed as a result of the sales process. Commonly, the buyers put out a request for quotation, to which the applications management providers respond. This provides the buyers with full information, making them greatly able to ensure that they get the best prices. (Mentioned by 2 'hands-on' experts, albeit once as a driver of rivalry)

This makes that the buyers of applications management services in the current applications management market find a number of circumstances in favor of their bargaining power. Moreover, two of the 'hands-on' experts rate the buyer bargaining power to be 'great', one rates it 'neither great nor little', one abstains from rating. Based on the foregoing the bargaining power of buyers over the applications management providers is considered to be great.

2.6.2 The bargaining power of suppliers

Just as buyers the suppliers to the applications management industry will try to maximize the reward they receive for their inputs. Being a service industry, the suppliers are found in the form of labor. The Dutch applications management service providers attract their personnel from the Dutch labor market as well as from several low-wage country labor markets. Porter (1980) has identified a number of circumstances that cause the bargaining power of suppliers to be great. In the below table it is shown to what extent these circumstances are present in the current market (see: Table 6). Again, a plus sign represents a strong presence, a minus sign a weak presence.

	The supplier group is dominated by few and is more concentrated than its client industry	The supplier group needs not contend with substitutes	The industry is not an important customer to the supplier group	The supplier's product is an important input to the buyer's business	The supplier group's product is differentiated or has built up switching costs	The supplier group poses a credible threat of forward integration
Presence in current market	-	-/+	+	+		

Table 6: Presence of circumstances that foster great supplier bargaining power in the current market

These finding will be detailed below. Also, it is noted how many of the 'hands-on' experts mentioned the presence of each of the circumstances, if applicable.

- In the applications management sector labor is hardly organized or unionized (Centraal Bureau voor de Statistiek 2006). Where a collection of organized personnel members can form a united front to their employer—the service provider—a single employee can hardly take a stand.

- Offshore economies offer vast amounts of skilled people, working for only a fraction of the costs of Dutch personnel (Parker 2004). This provides the service providers with an attractive alternative labor supply for the—simpler—tasks that can easily be outsourced. As English generally is an acceptable language of communication to both the applications management provider and offshore personnel, different native tongues do not have to form a barrier to the actual deployment of offshore personnel. (Mentioned by 1 'hands-on' expert)

- On the other hand, the Dutch labor market for highly educated IT personnel is rather tight (ICT~Office 2007), meaning that employers have few alternatives for these individual workers so that these can be somewhat demanding. (Mentioned by 4 'hands-on' experts)

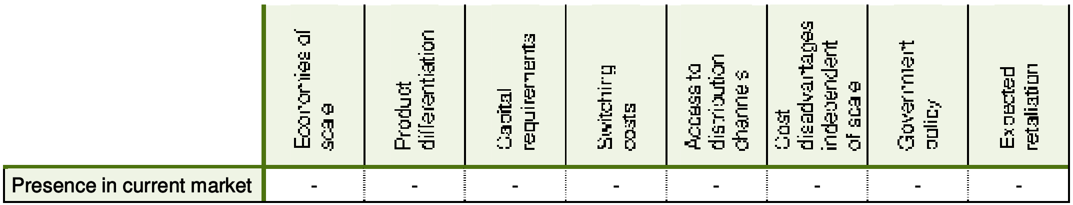
- The skills of the abovementioned highly educated IT personnel can be applied in different IT jobs, for example software development or at a non-IT company's internal IT department. This makes that the applications management services providers are not the only ones interested in 'buying' the services IT personnel can offer. As a result, highly educated IT personnel have several job alternatives. (Mentioned by 1 'hands-on' expert)

- The market for applications management services, obviously, is a services market. This means that the efforts of personnel are an important input to the end-product.

All in all, certain circumstances foster the bargaining power of the suppliers of labor in the applications management market, but others are not. The bargaining power IT personnel hold over the applications management services providers is considered to be 'great' by three of the 'hands-on' experts and 'neither great nor little' by one. Therefore, the bargaining power that the suppliers of labor have is considered to be great.

2.6.3 The threat of new entrants

When an industry is attractive and entry is easy new parties will take their chances on that market. New market entries increase the supply of services, thereby putting pressure on the prices. However, some market characteristics can provide barriers to entry, defending the existing firms from more competition. These are displayed on the x-axis in the table below (see: Table 7). As for the presence of these entrance barriers to the Dutch applications management market the following holds true:



	Economies of scale	Product differentiation	Capital requirements	Switching costs	Access to distribution channels	Cost disadvantages independent of scale	Government policy	Expected retaliation
Presence in current market	-	-	-	-	-	-	-	-

Table 7: Presence of barriers to entry in the current market

These findings are elaborated on below. Also, it is noted how many of the 'hands-on' experts mentioned the presence of each of the barriers, if applicable.

- The economies of scale that allow the existing firms to offer applications management services at lower cost than the client organizations could do themselves, are easily matched by service providers already operating in other geographic markets. Vivid examples of such abroad providers can be found in the so-called 'pure-players' from India, which use the economies of scales they have built up in India to operate and enter other geographic markets. (Mentioned by 2 'hands-on' experts)

- Existing firms do not have any advantages independent of scale (such as proprietary knowledge or favorable access to raw materials) that could inhibit the entrance of new players. Knowledge about programming languages or software design principles is available to anyone and personnel can flow freely among firms.

- Even knowledge of the Dutch language may not form a requisite, since English is generally accepted as language of communication as well. (Mentioned by 1 'hands-on' expert, albeit as a driver of rivalry)

- There exists little differentiation among the products, which limits the loyalty of clients to their current service provider. (Mentioned by 1 'hands-on' expert)

- Entering the market does not require large upfront investments, especially not for players that already operate in other geographic markets. All they would need is a front-office or a point of sale to enter the market. (Mentioned by 1 'hands-on' expert)

- Existing firms do not hold access to limited sales channels. When an outsourcing contract is put up for bidding, new players can simply step in.

- No restricting government policies exist.

- No harsh retaliation to entry is to be expected, because the market is still growing (see: Paragraph 2.4).

The above shows that very little barriers to entry exist, which would cause a great threat of new entrants. Furthermore, this threat is rated ‘great’ by three of the ‘hands-on’ experts, the fourth abstains from rating. Therefore, the threat of new entrants is considered to be great.

2.6.4 The threat of substitute products

A substitute product is a product that can perform a similar function as the original product. For example, for a trip from Amsterdam to Paris going by train or by airplane are substitutes to each other. Substitutes limit the prices that can be charged for a product, for as the prices for the one become too high customers will switch to a substitute.

	Threat of substitute products
Presence in current market	-

Table 8: Presence of substitute products in the current market

The applications management services industry faces little threat of substitute products. One way or the other, business applications need to be managed; there is no real alternative for that. There are, however, some alternatives to having the applications managed by an external party:

- Performing the activities in-house or by setting up a ‘shared services center’ with several other companies; (Mentioned by 1 ‘hands-on’ expert)

- Discarding the need for applications management at all, by, for example, outsourcing the business process supported by the application. (Mentioned by 1 ‘hands-on’ expert)

These alternatives are very little in number. Furthermore, two of the ‘hands-on’ experts rate the number of substitutes present to be ‘little’, one ‘great’, and one abstains from rating. Therefore the treat of substitute products is considered little.

2.6.5 Rivalry among existing firms

Next to bargaining with their buyers and suppliers and coping with the threat of substitutes or new entrants, existing firms in a market also compete with each other. Porter (1980) has identified a number of factors that fuel intense rivalry among existing competitors. In the below table the presence of these factors in the current market are listed (see: Table 9). Again, a plus sign represents a strong presence, a minus sign a weak presence.

	Numerous or equally balanced competitors	Slow industry growth	High fixed or storage costs	Lack of differentiation or switching costs	Capacity augmented in large increments	Diverse competitors	High strategic stakes	High exit barriers
Presence in current market	+	-		+		-		

Table 9: Presence of factors fueling rivalry among competitors in the current market

- In the Dutch applications management market numerous national, international, and offshore service providers all try to make a business. The competitors are relatively equally balanced, making them equal competitors to one another. (Mentioned by 2 ‘hands-on’ experts)

- The market for applications management is growing at a respectable rate (see: Paragraph 2.4), so competitors do not need to grow fully at the expense of the others.

- There is little differentiation among the competitors (Moore and Martorelli 2007) and switching costs are low. In such a situation buyers are likely to be very price-sensitive, as it is the only distinction between the service providers. (Mentioned by 2 'hands-on experts)

So, two of the factors that fuel competition among the applications management providers are present in the current market, as can be seen in the above table. On the other hand, also two other factors in the table show little presence. The 'hands-on' experts rate the intensity of rivalry among competitors twice as 'neither great nor little', once as 'very great', and one abstains from rating. Therefore, the rivalry among the competing applications management providers is considered to be moderate.

2.7 Synthesis

The current applications management market is characterized by moderate competition among equal competitors offering little diversified services. The value offer is rather simple: external applications management services providers offer services costing less than when they are performed by clients themselves. Also, the value exchange model is rather straightforward: the service provider performs applications management services, for which it is compensated by the client.

The current applications management market will serve as a standard to which the applications management markets for service-oriented and open-source software will be compared in the respective analyses. But preceding these market analyses, first the case organization will be described.

Value exchange	Applications management services for remuneration
Buyer power	Great
Supplier power	Great
Threat of new entrants	Great
Threat of substitutes	Little
Competition	Moderate
Market size	EUR 1.2 billion
Expected growth rate	12%
Outsourcing motive	Largely operational efficiency

Table 10: Characteristics of the current market

3 Case organization: Capgemini Outsourcing AMSC Nederland

[Confidential content was removed]

4 Applications management for service-oriented software

4.1 Introduction

This research will analyze how applications management providers can add value to two emerging markets for applications management. In the following, as first of the two, it will be analyzed how the AMSC can add value to the emerging applications management market for service-oriented software by means of a SWOT analysis.

To this end, the opportunities and threats of the applications management market for service-oriented software will be determined. The notion of 'the applications management market for service-oriented software' has three aspects: the software, the according applications management services, and the market. Therefore, each of these aspects will be analyzed. First, service-oriented software will be defined and the benefits and concerns regarding it will be identified, based on a review of relevant academic literature. This analysis will show that although many benefits are attributed to service-oriented software, none is based on empirical findings. Subsequently, it will be analyzed how applications management work for service-oriented software is different from traditional applications management work by analyzing the work itself as well as the value chain of which the work is part. This analysis will show that applications management work for service-oriented software is more complex and requires new and different skills of the applications management providers, such as competency with the technology and a more process-oriented view. Next, the market itself will be analyzed in terms of size, the motives of client organizations to buy the applications management from this market, and the market as a competitive environment. This analysis will show that the market is attractive, among other things because it is predicted to grow rapidly (about 20% yearly). The results of these analyses will again be analyzed to determine any opportunities and threats. Then these will be combined with the strengths and weaknesses of the AMSC—determined before—in a SWOT analysis to find out how the AMSC can add value to this market. [Confidential content was removed] Finally, recommendations will be given.

4.2 Service-oriented software

Currently, service-oriented computing (SOC) arouses academia and the IT industry. While the underlying technology still has to mature expectations are already high. This makes SOC, and its implementations in software, a much-discussed topic. But besides all the buzz and three-letter acronyms, what is service-oriented software really about? To form a solid basis for further discussion of the topic service-oriented software will be detailed. To this end, first, service-oriented software is defined and its characteristics are briefly introduced. Then, the attributed benefits are identified, followed by a discussion of the concerns regarding service-oriented software.

4.2.1 Definition

SOC is the latest generation in the evolution of component-based computing that utilizes services as fundamental elements for developing software applications. A service-oriented architecture (SOA) forms the basis that facilitates SOC. Within an SOA service providers can publish their services to the outside world, so that they may later be discovered and accessed by service consumers (see: Figure 4).

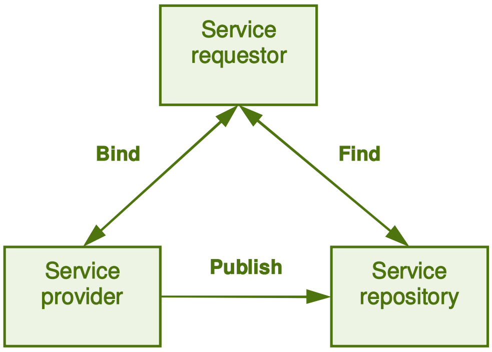


Figure 4: Service-oriented architecture

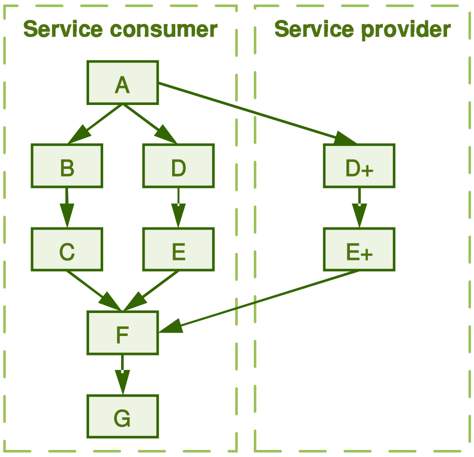


Figure 5: Service-oriented computing (Bichler and Lin 2006)

Service orientation allows the easy composition of individual services into composite services, even to the point that they become full-scale applications capable of supporting entire business processes. The individual services may reside within the company itself, or may be acquired from external service providers (see: Figure 5). Because of their modular nature services are often referred to as 'Lego-blocks' (Iyer et al. 2003; Bloomberg 2006). More formally, services are defined as: "self-describing, open components that support rapid, low-cost composition of distributed applications (Papazoglou and Georgakopoulos 2003)." Nowadays, the most popular implementation of SOA uses Web services (Papazoglou and Georgakopoulos 2003; Zhao and Cheng 2005; Margaria and Steffen 2006; Nezhad et al. 2006).

In literature, many definitions exist that attempt to capture the essence of Web services; some emphasize its capabilities, others focus on the underlying technology. A few examples:

"Web services comprises a set of platform-neutral technologies designed to ease the delivery of network services over intranets and the Internet (Vaughan-Nichols 2002)"

"Web services are self-contained, Web-enabled applications capable not only of performing business activities on their own, but also possessing the ability to engage other Web services in order to complete higher-order business transactions (Yang 2003)"

"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards (W3C 2004)"

"Web services (…) are a framework of software technologies designed to support interoperable machine-to-machine interaction over a network (Leavitt 2004)"

"(…) a service is any nugget of functionality that is directly executable and can be published for use in complex applications (Margaria and Steffen 2006)"

In this research the third definition—from the World Wide Web Consortium[4]—will be applied. From this definition, some key aspects of the Web services technology can be identified. These are described below.

[4] http://www.w3c.org

Decoupling of functionality from implementation. Web services are designed to be interoperable. This is achieved by masking the implementation to the outside world. This means that, although the interface is visible and service consumers know how to interact with the service, the actual operations of the service itself are opaque to the consumer. As a result, interacting with Web services becomes platform-neutral.

Standardized interface. This key aspect is closely related to the former: while the former ensures that the interior workings of a service do not impede collaboration, the standardized interface guarantees interoperability from the outside because all services 'speak the same language'. In Web services the interface and its functionality are described according to the Web Services Description Language (WSDL) standard.

Open standards. Web services are not the first attempt to create interoperability among software systems over a network. Some of its predecessors are Microsoft's Distributed Component Object Model (DCOM), OMG's common object request broker architecture (Corba), and Sun's remote method invocation (RMI) over the Internet inter-ORB protocol (IIOP) (Shirky 2002; Zhao and Cheng 2005). With Web services, however, its standards have become 'open' standards that are supported by all major software vendors (Vaughan-Nichols 2002; Langdon 2003). Being non-proprietary, the standards form no ground for competition among vendors; something that inhibited interoperability among technologies in the past (Vaughan-Nichols 2002).

Modular. A Web service is a useful software system in itself, containing a full, yet single functionality. This makes Web services reusable, but also allows the interaction with other services in larger arrangements (Fremantle et al. 2002). In combination with the standardized interfaces based on open standards, this fosters so called 'loose coupling' of systems, meaning that they can easily be integrated and disintegrated.

Machine-to-machine. Web services are designed for machine-to-machine interoperability (W3C 2004). This way, the foundations are laid for automatic service selection and invocation, paving the way for truly dynamic applications.

Asynchronous. When a service invokes another service it does not have to suspend its operations until it receives the returning result. Though not mentioned in any of the above definitions, this is a notable aspect because it means a departure from the synchronous remote procedure call (RPC) underlying distributed object systems (Jammes and Smit 2005).

The relationships among the SOC, SOA, and Web services concepts are depicted below (see: Figure 6).

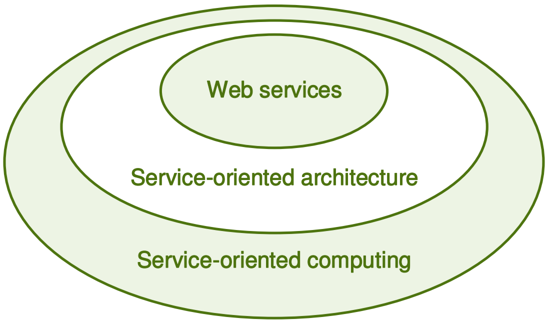


Figure 6: Relationships among service orientation concepts (adapted from Zhao et al. 2007)

In Web Services, two types of software can be discerned. On the one hand there is software that forms the basis for an SOA by taking care of things like message routing among services, keeping a services repository, providing integration with legacy systems by 'wrapping' them with a Web services layer, and by providing the tooling needed for arranging individual services into greater orchestrations with business logic incorporated. In the following, this type of software will be referred to as SOA platforms. Examples of such platforms are IBM's Websphere products, BEA's WebLogic platform, and Software AG's Crossvision Suite. On the other hand there are the applications built on top of these platforms: the orchestrations of services into greater orchestrations that follow certain business logic. These orchestrations are considered service-oriented software and form the subject of the following analysis. The below picture depicts how these two types of software relate to other software and underlying technology (see: Figure 7).

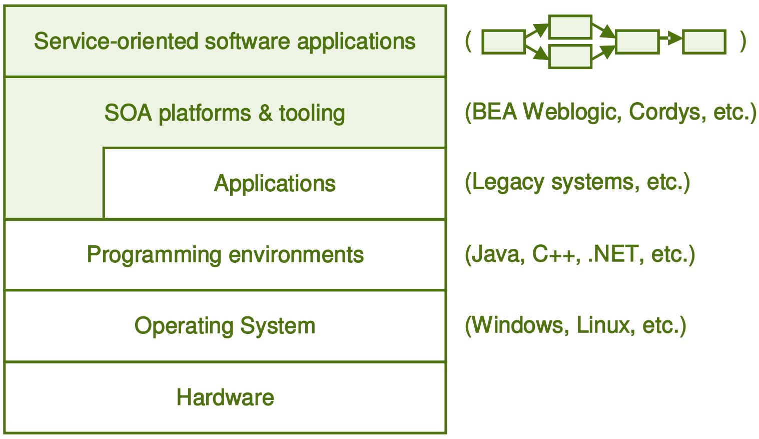


Figure 7: Technology stack

4.2.2 Benefits

As with most new IT technologies, the industry attributes great benefits to service orientation in software. More value for less money, the "Holy Grail", or the "silver bullet" for software engineering. To find out how much of these "mountains of gold" is really to be expected the top-25 information systems (IS) journals as ranked by Mylonopoulos and Theoharakis (2001) were subjected to a structured and systematic analysis to identify any benefits ascribed to service-oriented software. The search process and the results are detailed below. First, the criteria for selecting publications for review are presented, followed by the search method that was applied. Then, the results are presented and discussed.

Selection criteria

The following criteria were used to select publications to be included for review:

- Published in one of the top-25 IS journals;
- The main topic of the article is service orientation in software or service-oriented software, or a specific aspect of it—for example, interoperability of service-oriented software;
- Published in the past ten years (1997 – present); older articles are considered to be outdated.

Letters and opinions were excluded from further review.

Search method

To search the top-25 IS journals for relevant articles the online search engine of Web of Science[5] was used. This search engine covers 23 of the top-25 IS journals. The missing journals

[5] http://www.isiknowledge.com (subscription required)

('Communications of the AIS' and 'Accounting, Management, and IT') were searched separately. 'Communications of the AIS' was not indexed in any search engine, so it was searched by browsing the tables of content. 'Accounting, Management, and IT' (and its successor 'Information and Organization') was searched with the search engine of ScienceDirect[6].

On May 23, 2007 the search string '"service orient*" OR "service-orient*" OR serviceorient* OR SOA OR "web service*" OR "web-service*" OR webservice*' was entered in both search engines. Web of Science returned 178 items that fitted the first and last selection criteria, Emerald zero[7]. Subsequently, the titles and abstracts of the returned items were scrutinized to identify the articles that also fitted the second selection criterion. It was found that 33 articles met all criteria, from which three opinions were excluded. By browsing the tables of content of the 'Communications of the AIS' another two articles were found that met all criteria and could be added to the list. The resulting number of articles selected for further review thus was 32.

Results

The review of the selected articles shows that quite a number of benefits are attributed to service-oriented software. However, it also became apparent that none of the articles provided empirical evidence to back these attributed benefits. In the table below the cited benefits are listed, as well as the articles that cite them. Furthermore, the underpinning each article gives for citing a benefit is given (see: Table 11).

	Better resulting system	Lower development costs	Easier outsourcing	Easier systems integration	Easier enterprise application integration	Easier cross-enterprise collaboration	More flexible	Easier (re-)assembly	Easier maintenance	Reduction of up-front investments	Faster time-to-market	Better scalable	Better performance	More reliable and available	Lower total cost of ownership	Lower maintenance costs
(Hagel III and Brown 2001)	Claim	Claim	Claim	Claim	Claim	Claim	Claim									
(Vaughan-Nichols 2002)				Claim												
(Ferris and Farrel 2003)				Claim												
(Langdon 2003)		Claim		Claim			Claim									
(Curbera et al. 2003)					Claim	Claim										
(Iyer et al. 2003)				Claim			Claim	Claim	Claim							
(Elfatatry and Layzell 2004)									Claim	Claim	Claim					
(Jammes and Smit 2005)	Claim	Claim		Claim			Claim		Claim			Claim				
(Zhao and Cheng 2005)							Claim									
(Moitra and Ganesh 2005)				Claim				Claim			Claim					
(Currie and Parikh 2006)	Claim	Claim										Claim	Claim	Claim	Claim	
(Bichler and Lin 2006)			Claim	Claim												

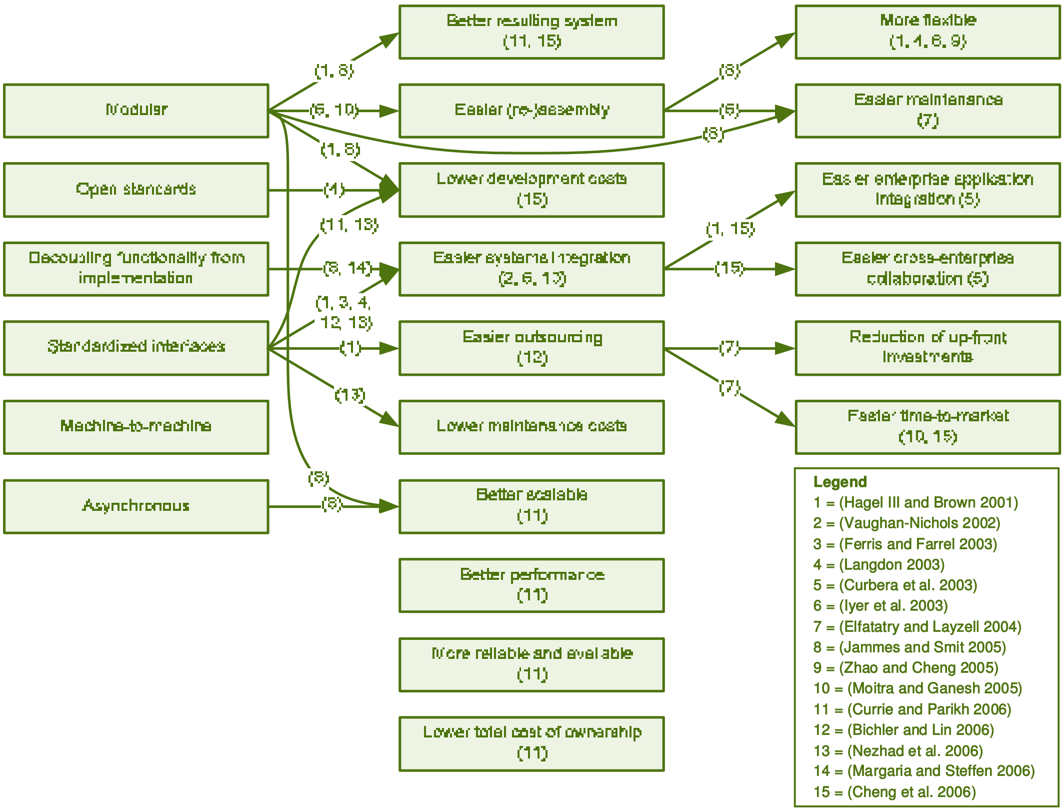
[6] http://www.sciencedirect.com/science/journal/09598022 and ../14717727

[7] This finding was verified by browsing the tables of content of both titles, which confirmed the absence of relevant articles. It was found that the journals treat IS on a systems level, rather than a technological level.

	Better resulting system	Lower development costs	Easier outsourcing	Easier systems integration	Easier enterprise application integration	Easier cross-enterprise collaboration	More flexible	Easier (re-)assembly	Easier maintenance	Reduction of up-front investments	Faster time-to-market	Better scalable	Better performance	More reliable and available	Lower total cost of ownership	Lower maintenance costs
(Nezhad et al. 2006)		Claim		Claim												Claim
(Margaria and Steffen 2006)				Claim												
(Cheng et al. 2006)	Claim	Claim			Claim	Claim					Claim					

Table 11: Benefits of service-oriented software and their underpinning

Although the articles do not provide any empirical findings that support their claims as to the benefits of service-oriented software, many of them do link the benefits to characteristics of service-oriented software, or even to other benefits. In the below figure the found causal relationships are depicted (see: Figure 8). A number within a box means that the corresponding article states the benefit concerned, a number on an arrow means that the corresponding article not only states the benefit, but also identifies a causal relationship between the starting and ending box. The cited causal relations depicted below are depicted for illustrative reasons only. Because none of these relations are based on any empirical findings they are not supported in this research.



Modular
Open standards
Decoupling functionality from implementation
Standardized interfaces
Machine-to-machine
Asynchronous
Better resulting system (11, 15)
Easier (re-)assembly
Lower development costs (15)
Easier systems integration (2, 6, 13)
Easier outsourcing (12)
Lower maintenance costs
Better scalable (11)
Better performance (11)
More reliable and available (11)
Lower total cost of ownership (11)
More flexible (1, 4, 6, 9)
Easier maintenance (7)
Easier enterprise application integration (5)
Easier cross-enterprise collaboration (5)
Reduction of up-front investments
Faster time-to-market (10, 15)
(1, 8)
(6, 10)
(1, 8)
(4)
(11, 13)
(8, 14)
(1, 3, 4, 12, 13)
(1)
(13)
(8)
(8)
(8)
(5)
(8)
(1, 15)
(15)
(7)
(7)
Legend
1 = (Hagel III and Brown 2001)
2 = (Vaughan-Nichols 2002)
3 = (Ferris and Farrel 2003)
4 = (Langdon 2003)
5 = (Curbera et al. 2003)
6 = (Iyer et al. 2003)
7 = (Elfatatry and Layzell 2004)
8 = (Jammes and Smit 2005)
9 = (Zhao and Cheng 2005)
10 = (Moitra and Ganesh 2005)
11 = (Currie and Parikh 2006)
12 = (Bichler and Lin 2006)
13 = (Nezhad et al. 2006)
14 = (Margaria and Steffen 2006)
15 = (Cheng et al. 2006)

Figure 8: Relations of cited benefits with characteristics of service-oriented software

The claimed benefits can be divided into two sorts: benefits for the resulting system and benefits for the development process (Van Hillegersberg 1997). The result of such a grouping can be found in the below picture (see: Figure 9). Again, the cited causal relationships between the characteristics and the benefits as found in literature are depicted, the dashed lines representing indirect relationships. The numbers between parentheses represent the number of times a certain benefit is cited. Note that “easier enterprise application integration” and “easier cross-enterprise collaboration” are contracted with “easier systems integration”, for they resemble the same increased interoperability among systems.

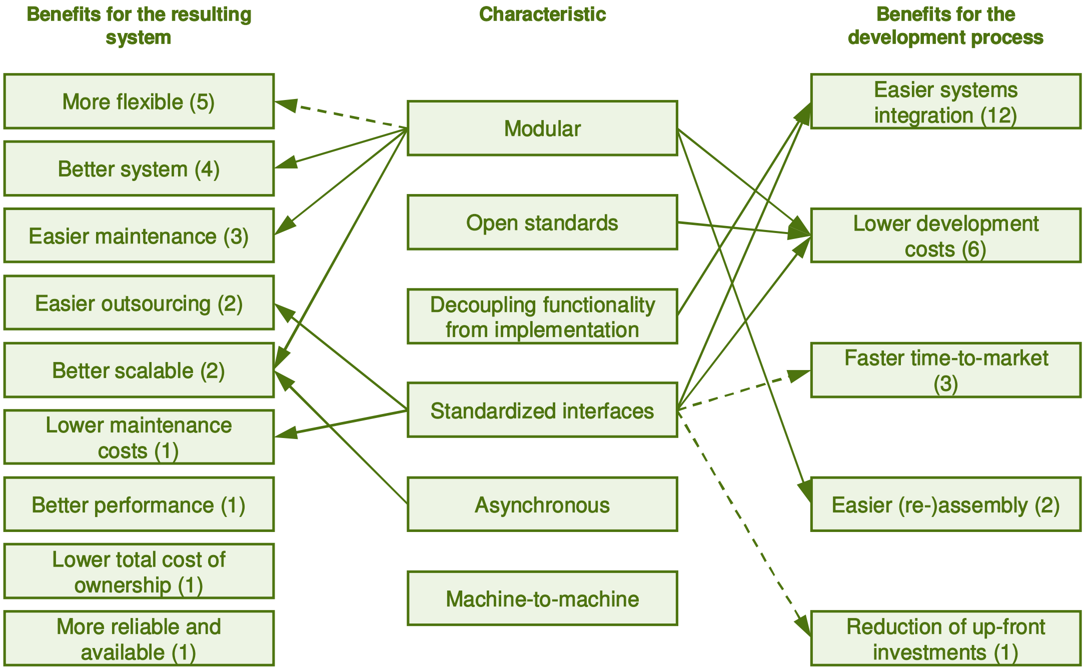

Benefits for the resulting system
Characteristic
Benefits for the development process
More flexible (5)
Better system (4)
Easier maintenance (3)
Easier outsourcing (2)
Better scalable (2)
Lower maintenance costs (1)
Better performance (1)
Lower total cost of ownership (1)
More reliable and available (1)
Modular
Open standards
Decoupling functionality from implementation
Standardized interfaces
Asynchronous
Machine-to-machine
Easier systems integration (12)
Lower development costs (6)
Faster time-to-market (3)
Easier (re-)assembly (2)
Reduction of up-front investments (1)

Figure 9: Relations of cited benefits with characteristics of service-oriented software (grouped)

The above picture shows that the most cited benefits of service-oriented software are the simplification of systems integration, the decrease of development costs, and the increase of systems flexibility and quality. These benefits will be explained below.

Easier systems integration. The greatest advantage Web services offer lies in the interoperability it enables. With Web services functionality is made available through an interface, diminishing implementation concerns. Moreover, the interface is standardized, making it easily accessible. It is the combination of these two characteristics that makes systems integration through Web services easier then ever.

Lower development costs. Service orientation fosters the reuse of services. As existing services are reused in new applications instead of being programmed anew, development costs will decrease (Jammes and Smit 2005). Furthermore, as systems integration only has to deal with one—standardized—type of interface, developing a new application that integrates several systems requires less integration effort, leading to lower development costs.

More flexible. The modular nature of service orientation allows the easy rearrangement of services within an application (Iyer et al. 2003). This makes applications more flexible; for example, if the application supports a business process and the order of activities that form the process changes the order of the supporting services is easily changed accordingly. But also when business processes span multiple organizations, the ease of systems integrations allows business partners to flexibly be added to or removed from the system (Hagel III and Brown 2001).

Better system. Another benefit stemming from the modularity of services is that it allows the selection of the best service for every job. No longer is it necessary to compromise on quality because a single software package may be the best for one part but inferior for another (Hagel III and Brown 2001). With service orientation an application can be built out of single services, each the best in their field, acquired from as many different providers as is necessary. This leads to better resulting systems, if not optimal systems.

4.2.3 Concerns

The previous paragraph has shown that service-oriented software is thought to bring about many good things. Now, any concerns regarding service-oriented software are identified from the articles selected in the previous paragraph by searching them once again. Below, the resulting concerns as well as the articles that cite them are listed (see: Table 12).

	Security, privacy, and trust	Reliability	Service Management and determining Quality of Service	Performance	Semantics	Standardization
(Vaughan-Nichols 2002)	x			x		
(Shirky 2002)	x	x			x	
(Fremantle et al. 2002)		x	x			
(Kreger 2003)	x		x			
(Langdon 2003)					x	
(Peltz 2003)	x		x			
(Casati et al. 2003)			x			
(Elfatatry and Layzell 2004)			x			
(Leavitt 2004)	x	x				x
(Birman 2005)		x				
(Birman 2006)	x	x	x			
(Bichler and Lin 2006)		x	x			
(Nezhad et al. 2006)						x
(Margaria and Steffen 2006)					x	
(Cheng et al. 2006)	x	x		x		
(Bussler 2007)				x		

Table 12: Concerns regarding service-oriented software

The above table shows that a number of issues regarding service-oriented software are cited as a reason for concern by several authors. Below, each of these issues will be detailed, as well as why they are deemed a reason for concern.

Security, privacy, and trust. Making services available over the Internet allows service consumers to access a multitude of services offered by different service providers. However, it also means that the interaction between the services takes place over the Internet. Birman (Birman 2006) cites some of the threats the Internet holds: viruses, spyware, phishing, spoofing, spam, hacking, and denial-of-service attacks. Interacting with services over the Internet means exposing whatever is in your communication to all these threats. Consider, for example, having your credit card data checked through a Web service, or sending a strategic order to a supplier Web service.

Also, Web services can be composites of other services; something the service requestor may not be able to determine beforehand because the internal operations of the invoked services are masked. This means that the service requestor may not be able to determine and judge all parties that will get to see its input message when the 'front-end' service is invoked. As Shirky (2002) states it:

> *"If A calls B and B calls C, A must be able to trust C as well as B, even if A doesn't know C exists."*

This concern in itself is not new. Before, a company's systems could also be integrated with those of some external party. But because of the tight coupling that had to be made between the systems, such connections required an understanding of each other's systems and, probably, of each other as well. However, as standardized services can be bought over the Internet, such connections—now loosely coupled—may well occur more frequently while such an understanding of each other may well be lacking. This makes that more attention may need to be paid to the issue of trust.

Reliability. The interaction messages of Web services are conveyed over the Simple Object Access Protocol, which uses standard internet protocols. These standard internet protocols are unreliable, in that they do not guarantee the ordered delivery of messages, delivering the messages only once, or even delivering them at all (Leavitt 2004). Again, consider a strategic order to a supplier Web service, but now it is never delivered. Or the order message is delivered twice. Web Services Reliable Messaging (WS-ReliableMessaging) has only very recently been approved as a standard by the Organisation for Advanced Structured Information Systems (OASIS)[8], an SOA standards body (OASIS 2007), and just cannot have found wide adoption yet.

Service Management and determining Quality of Service. As services are delivered—or sold—by a service provider to a service consumer, both have to check the service actually delivered. The provider needs service management to ensure it is meeting the requirements; the consumer needs to check delivery according to specifications. However, the number of items to monitor and manage rapidly increases as services are more and more deployed, making the management processes increasingly complex. Yet, the Web Services Distributed Management (WSDM) standard, designed for that purpose, has only recently been adopted by OASIS (Nezhad et al. 2006).

Performance. The common denominator of all Web services standards is that they are based on the eXtensible Markup Language (XML). Also, interaction messages are encoded in XML. But XML is by itself not the most efficient way of data encoding, due to the tags that are used to specify the contents of the message (for an extensive treatment of XML see (Murray 2002)). Furthermore, XML is text-based and thus entails more data for systems to process than binary-based messages, making it slower over HTTP (Vaughan-Nichols 2002).

Next to the inefficiency of the messages themselves, the number of messages that are sent as the result of just simple service invocations can easily explode when logging, management, and data-transformation services are added. Bussler (2007) shows how a single service invocation—service 1 provides an input to service 2 and expects a result message back—can lead to a total 14 messages to be sent. Hence, using Web services may need great network capacity.

Semantics. Web services make possible the easy integration of systems. This, however, does not mean that the integrated systems all the sudden understand each other. Shirky (2002) describes the semantics issue as follows:

> *"Web services' current state is analogous to international snail mail. The ability to send a letter from one country to another does not guarantee the recipient will be able to read it."*

[8] http://www.oasis-open.org

Although several initiatives are undertaken to come to semantic standards—for example ebXML, semantic Web services, and RosettaNet—there still is a long way to go before all ambiguity is removed from Web service interaction.

Standardization. Web services are built on agreed and supported standards. But for Web services to reach their full potential, a lot more standards are needed. Nezhad et al. (2006) provides an overview of the current standardization efforts, a lot of which still have to be decided on. Also, Sholler et al. (2006) show in their Web services hype cycle that it may take these standards some 2 to 10 years before reaching mainstream adoption. Leavitt (2004) cites Evan Data's McKendrick on the subject:

"There are multiple standards competing for each problem space in Web services. (…) [But] More damaging than competing specifications are vendor politics, which create much fear, uncertainty, and doubt in the Web services space a familiar beast to anyone that has been in IT over the past two decades."

Note that none of the above concerns found in literature relate to the development process, but all to the resulting system.

4.2.4 Conclusion

In the above analysis several benefits of service-oriented software are identified. What stands out from this analysis is that none of the attributed benefits are based on empirical findings. This makes their validity doubtful. Moreover, there exist a number of concerns regarding service-oriented software that may have a negative effect on the value that service-oriented software can bring to organizations applying the technology. Therefore, the added value of service-orientated software is considered very arguable.

For example, it can be argued that the claimed easier systems integration as a 'result' of standardization may well be impeded by the sheer number of standards that need to be adhered to. And, arguably, no company will trust its critical business applications to communicate with third-party services over the Internet as long as security issues have not been resolved.

4.2.5 Discussion

Besides the still unproven value of service-oriented software, a further annotation needs be made. Web services technology—like any other information technology—will not supply organizations with a competitive advantage. As Porter (1996) already said, information technology (IT) does not *per se* provide a competitive advantage. After all, the technology is available for all to use and thus loses its ability to provide differentiation. IT can only help in achieving operational efficiency, that is, doing what an organization has chosen to do in the best possible way. It does not diminish the need for organizations to formulate a strategy and to take a position in their competitive environment. It is in this light that Web services technology, and the value it may bring, needs to be seen.

4.3 Resource-based analysis

In the foregoing service-oriented software has been defined and analyzed. In the following it will be analyzed how managing and maintaining service-oriented software is different from traditional applications management, so that any newly required competences can be identified.

To this end the effects of service-oriented software on the actual applications management work and the value chain of which the work is part is analyzed. The effects on the work first are analyzed by reviewing relevant academic literature. This, however, shows that little has been published on the matter. Therefore, the effects subsequently are analyzed by confronting the characteristics, benefits, and concerns associated with service-oriented software with the different parts of the work. This analysis shows that applications management for service-oriented software may cause more applications management work (in terms of the number of problems, enhancement requests, and support requests) and a greater complexity of the work. The analysis of the value chain of which the work is part shows that it may be more complex than the value chain of traditional applications management. From these analyses it is concluded that some new or different competences are required of the applications management provider, such as competency with Web services technology and a more process-oriented view.

4.3.1 Effects on applications management services

Building applications out of orchestrations of individual services represents a new generation in the evolution of component-based software development. As such, it is likely that the maintenance and management of service-oriented applications differs at least to some extent from applications management as it was practiced before. For applications management suppliers it is interesting to know what aspects of their business will change, and how.

To this end, the articles selected in paragraph 4.2.2 were again searched, now for any statements on the management and maintenance of service-oriented applications. Unfortunately, this search returned only three general statements, all claiming that maintenance would be easier (see: Paragraph 4.2.2 and Figure 8). However, nothing concrete was said on how applications management will be different.

So, it was decided to perform a more extensive search on the matter. The search process is described below. First, the altered criteria for selecting an article for inclusion are detailed, followed by the search method. Finally, the results are presented and discussed.

Selection criteria

The following selection criteria were used to select publications to be included for review:

- Published in one of the top-50 IS journals as ranked by Mylonopoulos and Theoharakis (2001) or in a journal dedicated to software maintenance;

- Making any concrete statement as to how applications management of service-oriented software is different from traditional applications management.

Search method

To search the top-50 IS journals for relevant articles the online search engine of Web of Science was again used. From the top-50 journals 39 are covered by this search engine. The missing 11 journals were searched separately. For these journals their respective Websites were consulted. If the Website provided or referred to a specific search engine, that search engine was used to search the journal (search engines: ScienceDirect, ACM Digital Library[9], Emerald[10], and EBSCOhost Business Source Premier[11]). If not, the journal was searched for relevant articles by browsing its tables of content. To identify any journals dedicated to software maintenance the journals listed in Thomson's Journal Citation Report's[12] subject category 'computer science: software engineering'. From this category one dedicated journal was identified: Journal of Software Maintenance and Evolution: Research and Practice. An overview of the search engines used is provided below.

[9] http://portal.acm.org

[10] http://www.emeraldinsight.com

[11] http://www.ebscohost.com

[12] http://www.thomson.com/content/scientific/brand_overviews/jcr

		Web of Science	ScienceDirect	ACM Digital Library	Emerald	EBSCO	Browsing TOCs
1	MIS Quarterly	x					
2	Comm. of the ACM	x					
3	IS Research	x					
4	JMIS	x					
5	Management Science	x					
6	IEEE Transactions (var.)	x					
7	Harvard Business Review	x					
8	Decision Sciences	x					
9	Decision Support Systems	x					
10	Information & Mgmt	x					
11	European J. of IS	x					
12	Sloan Mgmt Review	x					
13	ACM Transactions (var.)	x					
14	Data Base	x					
15	Organization Science	x					
16	Information Systems J.	x					
17	Academy of Management	x					
18	Comm. of the AIS						x
19	IEEE Computer	x					
20	J. of Strategic IS	x					
21	Admin. Science Quarterly	x					
22	Academy of Mgmt Review	x					
23	Int. J. of E-Commerce	x					
24	ACM Computing Surveys	x					
25	Accounting, Mgmt & IT		x				
25	Information and Org.		x				
26	ACM SIG Publications[13]			x			
27	IT and People				x		
28	IBM Systems Journal	x					
29	OMEGA	x					
30	Journal of the AIS						x
31	J. of Org. Comp. & EC	x					
32	Human-Computer Interaction	x					
33	Information Systems Mgmt	x					
34	Int. J. Man-Machine Studies	x					
35	Journal of IS						x
36	The Information Society	x					
37	J. E-U Computing						x
38	Info. Resources Mgmt J.						x
39	Interfaces	x					
40	EM-Electronic Markets					x	
41	JCIS	x					
42	European Journal of OR	x					
43	Operations Research	x					
44	Int'l J. of H-C Studies	x					
45	Journal of the ACM	x					
46	Australian Journal of IS						x
47	Org. Behavior & Human Dec.	x					
48	Behavior and IT	x					
49	Scandinavian J. of IS						x
50	Computer Journal	x					
	J. Softw. Maint. Evol.: Research and Practice	x					

Table 13: Overview of search engines used per journal

On August 20, 2007 all search engines except ACM Digital Library were probed with the following search string: '("service orient*" OR "service-orient*" OR serviceorient* OR SOA OR "web service*" OR "web-service*" OR webservice*) AND (maint* OR "application? management")'. ACM Digital Library required the below sequence of searches to achieve similar results:

+publication:SIGSOFT +maintenance +"web service"*

+publication:SIGSOFT +maintenance -"web service" +webservice*[14]

+publication:SIGSOFT +maintenance -"web service" -webservice +soa

[13] Since only one special interest group (SIG) deals with software maintenance (SIGSOFT, see: https://campus.acm.org/public/gensigqj/SIGLIST/gensigqj_sigdesc.cfm), only publications of that SIG were searched

[14] The ACM Digital Library only allowed one wildcard character per search string, so it was decided to change 'maint*' into maintenance.

```
+publication:SIGSOFT +maintenance -"web service" -webservice -soa +"service orient"*

+publication:SIGSOFT +maintenance -"web service" -webservice -soa -"service orient" +serviceorient*

+publication:SIGSOFT +"application? management"
```

The Journal of Software Maintenance and Evolution: Research and Practice was searched separately, discarding the 'AND maint*' part from the search string, for this is covered by the focus of the journal itself.

Web of Science returned 14 articles, none of which complied with the second criterion. ScienceDirect returned zero articles, which is consistent with the findings in paragraph 4.2.2. The ACM Digital Library searches returned 46, 1, 8, 15, 1, and 9 items respectively, only one of which met the second criterion. Emerald returned five articles, but again zero met the second criterion. EBSCO returned zero items[15]. Browsing the tables of content of the remaining journals did not result in any relevant articles to be added. Searching the Journal of Software Maintenance and Evolution: Research and Practice returned four items, one of which made a concrete statement regarding applications management work. Thus, the final number of relevant publications found was two.

Results

It needs no further explanation that just two publications do not form a solid body of scientific knowledge to build upon. Apparently, academia has not—yet—produced many insights regarding applications management of service-oriented software. When considering academic conferences and symposia as precursors of scientific literature, perhaps the fact that the leading conference on software maintenance (IEEE International Conference of Software Maintenance) dealing with service orientation has only just taken place—but its research papers are yet to be published—explains the meager results of this structured search.

Nonetheless, the two articles found do point out relevant issues. For reasons of uniformity, these few issues are displayed in the below table (see: Table 14).

	New and more software engineering skills required	Versioning is different
(Tilley et al. 2004)	x	
(Stuckenholz 2005)		x

Table 14: Effects on applications management work

These two issues will be explained in greater detail below.

New and more software engineering skills required. Working with Web services demands that one knows a lot about a lot (Tilley et al. 2004). Obviously, working with a new technology requires knowledge about the technology itself. With Web services, this means that someone performing applications management needs to have a thorough understanding of the lengthy list of Web services standards (Tilley et al. 2004)—SOAP, WSDL, UDDI, BPEL, WS-Security, SAML, WS-Reliability, WSDM, and many more. Also, an understanding of the tooling used to deal with the new technology is required. In the case of Web services this means an understanding of SOA platforms and tooling such as Cordys Suite, BEA Weblogic, and IBM's SOA Suite. Furthermore, Web services can be coded in any programming language or on top of existing legacy applications, as long as its interface adheres to the standards. This means that also an understanding of many different programming languages and legacy systems may be required. Then there is the fact that Web services are invoked and delivered over some

[15] This finding was verified by browsing the tables of content of the one journal concerned (Electronic Markets). It was found that the journal deals with (a number of aspects of) e-commerce, but not on a technological level.

network—intranet, the Internet—which also demands insights in all sorts of network issues (Tilley et al. 2004).

Versioning is different. As mentioned before, Web services mask their inner workings. As long as the interface of a service remains unaltered the service is the same to the outside world, even though its inner workings may have been changed. This would make versioning superfluous. Standards like WSDL (which describes the interfaces) and SOAP (used for communication) do not provide for versioning, but a workaround exists (Stuckenholz 2005). Nevertheless, versioning in Web services is different from traditional versioning.

4.3.2 Possible other effects on applications management services

It is deemed hardly likely that software applications developed according to a new way of software development will impact applications management services only in the two ways described in the previous paragraph. Therefore, an initial attempt will be made to identify any other effects of service-oriented software on the nature of applications management services.

The first thing to do in order to identify such effects would be to provide a framework that can identify the nature of applications management work. To this end the widely cited framework of Perrow (1967) for the comparison of work done in different organizations will be used, albeit in a slightly adapted form.

Perrow identified two axes along which the work of an organization can be identified: the number of exceptional cases encountered in the work, and the analyzability of these exceptions. However, as business applications are hardly ever alike—even similar software packages generally are configured differently—acts of maintenance are inherently unique, and therefore exceptional. The same holds for enhancement activities, as the chances of multiple clients requesting exactly the same alteration to their deployments of the same application are negligible. Only in support the same question on the same application may have to be answered more than once. To incorporate this inherent uniqueness of (most of) the applications management work the "number of exceptional cases" is reformulated to the "number of cases". A case is then defined as any set of applications management actions to resolve a single problem, to fulfill a single enhancement request, or to fulfill a single support request.

Also, the "analyzability of the exceptional cases" is slightly reformulated, as analysis is only part of the work. To also incorporate the other activities of applications management work, the term is reformulated to the "complexity of the work".

Now, applications management work can be analyzed along the abovementioned two axes. In order to ensure the analysis is as comprehensive as possible every aspect[16] of service-oriented software identified in paragraph 4.2 is confronted with applications management work. For each of the aspects it is argued whether it will increase, decrease, or have no effect on either the number of cases or the complexity of the work. In order to give these argumentations some empirical value they have been formulated after having a conversation with an applications management practitioner[17] on the topic. The findings are presented below.

[16] All characteristics and concerns are included in the analysis. As the benefits are all claimed, but not empirically proven, their validity is arguable. In this analysis only those benefits cited by at least four authors in four different journals (number arbitrarily chosen) are considered to have sufficient support in the academic world to assume their validity.

[17] Mr. Sjoerd Wittebrood, Service Coordinator at Capgemini Outsourcing Netherlands. Having several years of experience with applications management of traditional applications he is now performing applications management on a service-oriented application.

Effects on the number of cases of applications management work

In the analysis of the number of cases of applications management work the aspects of service-oriented software are confronted with the aspects of applications management work—as opposed to the individual activities. This has been done because although both maintenance and enhancement work comprise the same activities, often their contractual difference is of interest for the applications management provider and the client; maintenance is often sold on a flat-fee basis where enhancement work can be sold as additional work. Recall from paragraph 2.2 that applications management work has three aspects—maintenance, enhancement, and support. In the below table (see: Table 15) the findings are presented. A plus sign represents an argued increase in the number of cases, a minus sign—though absent in this table—represents a decrease. If a cell is empty no reasons were found to assume an effect on the number of cases.

	Characteristics						Benefits				Concerns					
	Modular	Open standards	Decoupling functionality from implementation	Standardized interfaces	Machine to machine	Asynchronous	Easier systems integration	Lower development costs	More flexible	Better system	Security, privacy, and trust	Reliability	Service Management and determining Quality of Service	Performance	Semantics	Standardization
Maintenance							+					+				
Enhancement							+	+	+						+	+
Support							+	+	+			+		+		

Table 15: Possible effects of service orientation in software on the number of applications management cases

The findings are explained below.

Maintenance. Easier systems integration may induce application owners to actually integrate more systems with one another. As a result, more connections among more systems may be made, which would increases the number of dependencies and overall systems complexity. An increase in the number of dependencies as well as an increase in overall systems complexity may well cause more errors to be made, which, in turn, need to be corrected. Therefore, easier systems integration is thought to increase the number of maintenance cases caused by an average service-oriented application, as compared to an average traditional application.

Also, as the reliability of service-oriented software still forms a reason for concern, it may well be lower than the reliability of traditional software. Lower reliability may cause messages between services to be delivered wrongly, which would foster more corrective work. Therefore, reliability concerns are thought to increase the number of maintenance cases caused by an average service-oriented application, as compared to an average traditional application.

Enhancement. The combination of increased flexibility, easy integration, and lower development costs would lower the threshold for application owners to adapt applications to changing demands. As becoming agile is a key concern for IT executives and the pace of today's business is heightened (Luftman and McLean 2004), such a lowered threshold may well lead application owners to exploit these benefits and to have their applications adapted more often, which would increase the number of enhancement cases. Therefore, the combination of increased flexibility, easy integration, and lower development costs is thought to increase the number of enhancement cases caused by an average service-oriented application, as compared to an average traditional application.

Furthermore, there still exist issues of standardization and semantics that need to be resolved. As this resolution process advances, this may cause that some existing applications will need to be adapted to new standards once they come about, which incurs enhancement work. This is in line with the assumption made by a recent Gartner research, which states that applications management work will greatly be impacted by service-oriented applications, mostly because its increased flexibility will foster near continuous change (Cappelli 2006). Therefore, standardization and semantics issues are thought to increase the number of enhancement cases caused by an average service-oriented application, as compared to an average traditional application.

Support. Following the same reasoning as above, the combination of increased flexibility, easy integration, and lower development costs may cause applications to more often be adapted to changing needs. This would also affect the end-users of an application, who are now confronted with a changing system more often. As a result the end-user's knowledge of the functional and the technical matters of an application may become outdated more quickly, leaving them with questions they cannot answer themselves more often. This may well result in a greater need for functional support. Therefore, the combination of increased flexibility, easy integration, and lower development costs is thought to increase the number of support cases caused by an average service-oriented application, as compared to an average traditional application.

In addition, the inferior reliability and performance of Web services may well cause end-users to come across their symptoms. This may lead them to more often call in support on these issues. Therefore, the concerns regarding reliability and performance are thought to increase the number of support cases caused by an average service-oriented application, as compared to an average traditional application.

Effects on applications management complexity

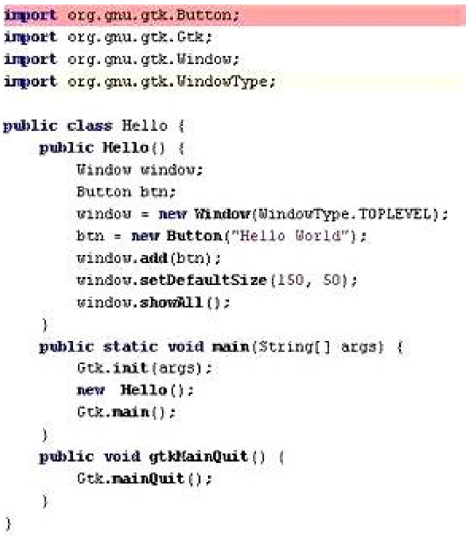
Next to the number of cases generated by an application, now the complexity of the work is analyzed. To this end the effects of the characteristics of service-oriented software on the individual activities of applications management are analyzed. Recall from paragraph 2.2.1 that applications management work consists of six activities—analysis, design, build, test, implement, and support. The result of this analysis is depicted below (see: Table 16). Again a plus sign represents an argued increase in the complexity of an activity; a minus sign represents a decrease. If a cell is empty no reasons were found to assume an effect on the complexity of an activity.

	Characteristics						Benefits				Concerns					
	Modular	Open standards	Decoupling functionality from implementation	Standardized interfaces	Machine to machine	Asynchronous	Easier systems integration	Lower development costs	More flexible	Better system	Security, privacy, and trust	Reliability	Service Management and determining Quality of Service	Performance	Semantics	Standardization
Analysis	+		+		+											
Design	-															
Build	-															
Test	+		+		+		+									
Implement	-		-	-												
Support																

Table 16: Possible effects of service orientation in software on the complexity of applications management work

Analysis. The modularity of service orientation in software makes an application more dispersed. Also, services can be composites of other services, which would create multiple layers of analysis. Furthermore, they may be built on underlying legacy systems, which would also add extra layers of analysis to the application. Instead of one coherent application, as traditionally was the case, service-oriented applications may well exist of a multi-layered orchestration of dispersed services. As a result, finding the cause of an error or determining the impact of a desired change may well require analysis of many more entities on multiple layers. Therefore, the modularity, the decoupling of functionality from implementation, and the machine-to-machine capabilities are thought to increase the complexity of analysis of an average service-oriented application, as compared to an average traditional application.

Design. Although designing a change or bug fix for the inner workings of a service may not be very different from designing a change or bug fix for a traditional application, the modular nature of services may also make designing an alteration a matter of rearranging services. This may mean that a part of the desired change may be designed on a higher level of abstraction (compare the traditional coding depicted in Figure 10 with the tooling of an SOA suite displayed in Figure 11). Also, the modularity of services makes possible the reuse of existing services, which may reduce the need to design certain functionality anew. Therefore, the modularity is thought to decrease the complexity of design activities of an average service-oriented application, as compared to an average traditional application.



```
import org.gnu.gtk.Button;
import org.gnu.gtk.Gtk;
import org.gnu.gtk.Window;
import org.gnu.gtk.WindowType;

public class Hello {
    public Hello() {
        Window window;
        Button btn;
        window = new Window(WindowType.TOPLEVEL);
        btn = new Button("Hello World");
        window.add(btn);
        window.setDefaultSize(150, 50);
        window.showAll();
    }
    public static void main(String[] args) {
        Gtk.init(args);
        new  Hello();
        Gtk.main();
    }
    public void gtkMainQuit() {
        Gtk.mainQuit();
    }
}
```

Figure 10: Traditional coding

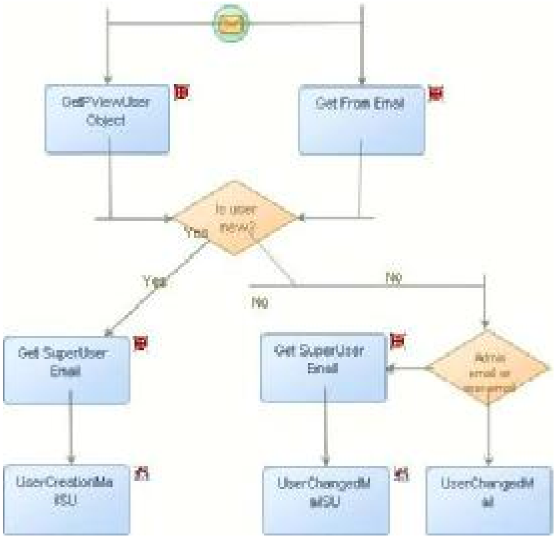


Figure 11: High level SOA tooling

Build. Following the same reasoning as with 'design', a part of all building may proceed at a higher level of abstraction and reusing existing services may reduce to build certain functionality anew. Therefore, the modularity is thought to also decrease the complexity of build activities of an average service-oriented application, as compared to an average traditional application.

Test. Following the same reasoning as with 'analysis', service-oriented applications may well be multi-layered orchestrations of dispersed services. Also, services can be used in multiple applications, which further complicates testing activities (Van Heur 2007). Furthermore, there is no control over third-party services (Lewis et al. 2006). In dynamic service binding—the application automatically selects a service out of a selection of generic services for a specific task, for example selecting the service of a party offering the lowest price in an ordering process—multiple scenarios may exist for a single application. Therefore, the modularity, the decoupling of functionality from implementation, and the machine-to-machine capabilities are thought to increase the complexity of test activities of an average service-oriented application, as compared to an average traditional application.

Implement. The combination of modular design and masking operations behind standardized interfaces are cited as the basis for the abovementioned benefits of Web service-oriented software: easy integration and greater flexibility. Implementing a change to the orchestration of services may well be facilitated by the plug-and-play nature of services. Furthermore, when a change needs to be made to the inner workings of a service, the decoupling of the *what* from the [illegible] would ensure that the rest of the application is not affected, as long as the interfaces are not altered. Therefore, the modularity, the standardized interfaces, and the decoupling of functionality from implementation are thought to decrease the complexity of implement activities of an average service-oriented application, as compared to an average traditional application.

Support. None of the characteristics, benefits, or concerns of service-oriented software is thought to affect the complexity of support activities.

Discussion

The applications management of a service-oriented application is thought to comprise greater amounts of maintenance, enhancement, and support work than is the case with traditional applications. Also, though truly service-oriented applications may lower the complexity of the applications management work, in practice it is to be expected that business applications will be built out of SOA-wrapped back-end systems. This may well make the applications management work even more complex than is the case with traditional applications, because an extra layer is added.

4.3.3 Value chain

Not only may the applications management work itself be affected by service-oriented software, also the value chain of which the value-adding activities of an applications management provider are part may alter. When the results of the analyses in the previous paragraphs are again analyzed, the following effects on the value chain can be identified.

Third-party services. The modularity of service-oriented software allows certain bits of an application's functionality to be acquired from a third-party service provider (Bichler and Lin 2006)(see: Figure 5). However, the applications management provider does not have control over the delivery and performance of the third-party services (Lewis et al. 2006). This is a matter that may require special attention when agreeing on terms of service. Depending on the demarcation of the applications management contract, the applications management provider may take on the responsibility for these services anyhow—in which case the value-adding activities of the external service provider become an input to its own value chain—or decline that responsibility.

Diversity of services technology. The decoupling of functionality from the implementation allows that services are coded in any programming language, on any platform, or built on top of any (legacy) system, as long as the interfaces adhere to the standards (Ferris and Farrel 2003). Inversely, this means that a multitude of different technologies may be used in a service-oriented application. Hence, to be able to maintain such an application the application management provider needs to be proficient at all these technologies. If not, alternatives options such as co-contracting or sub-contracting may bring resolution. Also sub- or co-contractors may come across third-party services.

In the below picture (see: Figure 12) these issues are depicted. It also shows that third-party service provider may require inputs themselves in order to provide their added value. The dotted shapes and lines represent that these are optional.

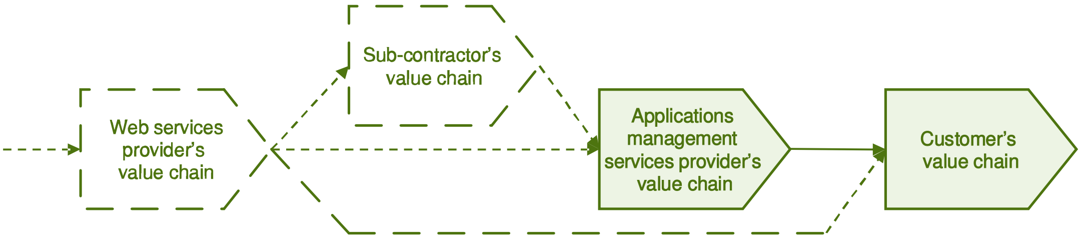

Sub-contractor's value chain
Web services provider's value chain
Applications management services provider's value chain
Customer's value chain

Figure 12: Value chain for the applications management market for service-oriented software

4.3.4 Conclusion: required competences

In the foregoing paragraphs it is detailed how service orientation in software may alter the way applications management is performed. It is thought to bring about relatively more applications management work and to increase the work's complexity. Also the network of value adding activities may become more complex. In order to cope with these alterations, some new or different competences may be required of applications management providers. Below, these alterations and the competences they demand are described.

First of all, Tilley et al. (2004) pointed out the many new technologies and standards that make up Web services and service-oriented software. In order to provide applications management knowledge of the Web services standards needs to be attained, as well as knowledge of SOA platforms and tooling.

Moitra and Ganesh (2005) state that service-oriented applications are more likely to support specific business processes, which is reflected in the business-process model-like nature of SOA tooling (see: Figure 11) based on the BPEL / BPML standards. Such a different approach may require a more process-oriented view and more functional management skills of applications management providers.

The fact that service-oriented applications can be built on all sorts of underlying systems, may require a broad portfolio of programming skills of the applications management provider to be able to manage service-oriented applications. Or, if it is decided to manage an application by means of co-contractors or sub-contractors, specific competences to arrange the associated legal issues and to coordinate efforts spanning multiple contractors may be required.

The possible presence of third-party services within an application may also require new competences. If the responsibility for these services is declined by the application management provider, no new competences are needed. But if the applications management provider accepts the responsibility for these services, it may need to develop new competences to guarantee the functionality of those services according to service level agreements, for example by keeping a list of back-up services providing the same functionality.

4.4 Market-based analysis

In the following the applications management market for service-oriented software will be analyzed, so that the attractiveness of the market can be determined. First, the size of the market is analyzed by consulting the publications of market research agencies. This analysis shows that a market growth of about 20% yearly is predicted. Then, the motives client organizations—current as well as potential—may have to acquire applications management from the market, that is, to outsource their applications management, are analyzed. This shows that clients may have more strategic motives to outsource their applications management than is the case in the traditional applications management market. Subsequently, the competitive environment of the market is analyzed by using the five forces framework. This analysis shows that the applications management market for service-oriented software has a higher threat of new entrants and substitutes, greater supplier power, but also less buyer power and less rivalry among competitors than in the traditional applications management market. Based on these analyses it is concluded that the applications management market for service-oriented software is an attractive one.

4.4.1 Market size

To determine the market size for applications management services for service-oriented software the publications of two major market research agencies (Forrester and Gartner) were analyzed, to find any reports on this matter. First, the criteria for selecting publications are stated, followed by an explanation of the search method. Finally the results are presented.

Selection criteria

The following criteria were used to select publications to be included for review:

- Published by Forrester or Gartner;
- The publication provides a forecast of the market size for applications management services for service-oriented software in Europe, Western Europe, or the Netherlands;
- Published in the past year.

Search method

To search the publications of both market research agencies the search engines offered on their respective websites were used[18]. On September 14, 2007 the advanced search form of Gartner was probed with the search term '"service-orient*" OR "service orient*" OR serviceorient* OR SOA OR "web service*" OR "web-service*" OR webservice*' and the search was restricted to publications of the past year, dealing with region 'Europe', topic 'IT Services', and of content type 'Forecasts'. This resulted in five items found, one of which turned out to be relevant.

The same day the same search term was entered into the advanced search form of Forrester and again the search was restricted to publications of the past 12 months, considering Europe, forecasts only. This search returned nine items, none of which was on topic.

Results

The one forecast found reports the expected market size for application management services for SOA and Web services world-wide for the period 2005-2010. Unfortunately, the report does not present country-specific values of these figures. Nonetheless, it predicts a compound annual growth rate (CAGR) of 22% of the market for application management services for SOA and Web services in the period considered (Cantara et al. 2007).

Besides this finding, another Gartner research states that over 50% of all new, mission-critical applications designed in 2007 will be service-oriented to some extent; in 2010 that figure will be 80% (Natis et al. 2006).

Also, major vendors of packaged software such as SAP, Oracle, Microsoft, and IBM are rebuilding their products to accommodate service orientation (Abrams 2006). For example, SAP already is offering its services-based MySAP ERP-system, Oracle offers it SOA Suite to build one's own SOA or its Fusion middleware to create an SOA among other Oracle products, and Microsoft is already cooperating with SAP to open up its SQL Server to SAP's Enterprise SOA. Furthermore, most major software vendors attribute to SOA standards bodies like OASIS or the Open SOA Collaboration (OSOA)[19].

4.4.2 Motives to outsource applications management

When analyzing the applications management services market for service-oriented software it is not only interesting to determine how much of these services will be asked for, it is also interesting to determine the motives client organizations have to outsource applications management for their service-oriented software, for it may have an effect on the attractiveness of the market.

Client organizations may operate in any market, as long as they apply business applications. In many of today's markets being able to 'sense and respond', i.e. being able to quickly adapt to changing market needs has become imperative. In other markets such qualities can offer an advantage over competitors. But such agility requires the applications that support business to be agile as well. Realizing such alignment between business and software and such agility both

[18] Gartner: http://www.gartner.com/Search?op=31
Forrester: http://www.forrester.com/findresearch/advanced

[19] http://www.osoa.org

are among the key concerns of IT executives (Luftman and McLean 2004). Web services offer great possibilities to flexibly adapt applications to—changing—business demands, thus being able to support agile competitive strategies (Huang and Hu 2004). Their criticality to strategy execution makes service-oriented applications—either in whole or in part—of strategic importance as well. Therefore, outsourcing applications becomes a strategic matter. So clients of applications management services may have more strategic considerations when deciding on outsourcing their service-oriented business applications than they had with their traditional applications. Nonetheless, tactical and operational reasons for outsourcing also remain existent.

4.4.3 Competitive environment

To analyze the market for applications management services for service-oriented software as a competitive environment in which applications management providers operate, the articles returned by the extended search described in paragraph 4.3.1 were scrutinized for any statements on this matter. Yet, no such statements were found.

Nonetheless, it is deemed unlikely that the market for applications management services for service-oriented software will form an identical competitive environment for applications management providers as the market for traditional applications management currently does. Therefore, an initial attempt will be made to identify any differences between both.

To this end each of the five forces that collectively shape the competitive environment of a market will be analyzed for any differences. To improve the empirical validity of this analysis several 'hands-on' experts were consulted on the matter by having them estimate the magnitude of each force on a five-point scale ranging from very little to very great and to comment their estimations (see: Appendix C). Two of the consulted people work for the case organization: Mr. Sjoerd Wittebrood, working as Service Coordinator for the applications management of a service-oriented application, and Mr. Michel Jeschke, Service Manager for the same applications management contract. One of the consulted people—Mr. Anton van Weel—is employed by Capgemini Technology Services as a senior architect. The last person consulted is Mr. Gerard Reeskamp of Cordys, vendor of SOA platform software. The below analyses are the result of combining their aggregated insights and any effects on the five forces that follow from the foregoing analyses. All effects are mapped to the characteristics of service-oriented software and the sources of each force as identified by Porter (1980).

The threat of new entrants. In the below table the results are presented of the analysis of the barriers that prevent companies from entering the market (see: Table 17). A plus sign represents that the statement in the respective column is more true in the market for applications management services for service-oriented software than in the market for traditional applications management services, and thus provides a greater barrier to entry; a minus sign represents the opposite.

		Economies of scale	Product differentiation	Capital requirements	Switching costs	Access to distribution channels	Cost disadvantages independent of scale	Government policy	Expected retaliation
Characteristics	Modular		-						
	Open standards								
	Decoupling functionality from implementation	+							
	Standardized interfaces								
	Machine-to-machine								
	Asynchronous								
Benefits	Easier systems integration								
	Lower development costs								
	More flexible								

		Economies of scale	Product differentiation	Capital requirements	Switching costs	Access to distribution channels	Cost disadvantages independent of scale	Government policy	Expected retaliation
	Better system								
Concerns	Security, privacy, and trust								
	Reliability								
	Service Management and determining Quality of Service								
	Performance								
	Semantics								
	Standardization								
	Growing market								-
	More strategic outsourcing								

Table 17: Possible effects of service orientation in software on the barriers to entry

The findings are explained below. Also, it is noted how many of the 'hands-on' experts mentioned the found differences, if applicable.

The decoupling of functionality from implementation enables services to be built on any underlying platform or legacy system or to be coded in any programming language. In paragraph 4.3.3 it is argued that maintaining an application built out of services may require the applications management provider to be proficient at all the underlying technology. This is different from traditional applications management which allowed providers to focus on a limited number of systems or programming languages. Being proficient at a great number of technologies now creates an economy of scale, which may restrain companies from entering this market.

On the other hand, service-oriented applications are more likely to support specific business processes, as was stated in paragraph 4.3.4. As a result, modeling the functionality of a service-oriented application becomes more like modeling the specific business process. This was traditionally the arena of business consultants, but with service-oriented software the gap between both becomes smaller, making a part of the applications management services less differentiated and reducing this barrier for business consultants to expand into this market. (Mentioned by 1 'hands-on' expert)

Furthermore, the rapidly growing market may well diminish the threat of retaliation to entry by the companies already in the market, because 'there is enough of a pie to share', lowering this barrier to entry as well. (Mentioned by 1 'hands-on' expert)

All in all, the barriers to entry in the market for applications management for service-oriented software are believed to be somewhat lower than they are in the market for traditional applications management services. The 'hands-on' experts estimate the threat of new entrants to be 'very great' twice, 'great' once, and 'little' once. Therefore, the threat of new entrants is believed to be great.

The threat of substitute products. In the below table (axes interchanged for space considerations) the results are presented of the analysis of the threat of substitute products (see: Table 18). A plus sign represents that the threat is greater in the market for applications management services for service-oriented software than it is in the market for traditional applications management services.

	Characteristics						Benefits				Concerns							
	Modular	Open standards	Decoupling functionality from implementation	Standardized interfaces	Machine-to-machine	Asynchronous	Easier systems integration	Lower development costs	More flexible	Better system	Security, privacy, and trust	Reliability	Service Management and determining Quality of Service	Performance	Semantics	Standardization	Growing market	More strategic outsourcing
Threat of substitutes	+						+											

Table 18: Possible effects of service orientation in software on the threat of substitute products

The findings are detailed below.

Service orientation in software makes possible the use of third-party services. These services are maintained and managed by the external provider; hence this bit of functionality no longer needs be managed by an applications management services provider. This clearly is a new substitute.

Also, as said above, high-level enhancement is much easier with service orientation in software. This may lead client organizations to decide to perform these activities in-house again, thus diminishing the need for external applications management services.

Hence, the threat of substitutes is may be somewhat greater in the market for applications management for service-oriented software than they are in the market for traditional applications management services. The 'hands-on' experts rate the threat of substitutes 'little' twice, 'very great' once, and one abstains from rating. Therefore, the threat of substitutes is thought to be moderate.

The bargaining power of buyers. The buyers in the competitive environment of the market for applications management services for service-oriented software are the clients that acquire those applications management services for their software. In the below table it is depicted how their power to bargain the best deal may be different than in the market for traditional applications management services (see: Table 19).

		Concentrated or large purchase volumes	Product represents a significant fraction of buyer's costs / purchases	Product is undifferentiated	Buyer faces few switching costs	Buyer earns low profits	Credible threat of backward integration	Product is unimportant to the buyer's product	Buyer has full information
Characteristics	Modular			-					-
	Open standards								
	Decoupling functionality from implementation								
	Standardized interfaces								
	Machine-to-machine								
	Asynchronous								
Benefits	Easier systems integration								
	Lower development costs								
	More flexible	-							
	Better system								

		Concentrated or large purchase volumes	Product represents a significant fraction of buyer's costs / purchases	Product is undifferentiated	Buyer faces few switching costs	Buyer earns low profits	Credible threat of backward integration	Product is unimportant to the buyer's product	Buyer has full information
Concerns	Security, privacy, and trust								
	Reliability								
	Service Management and determining Quality of Service								
	Performance								
	Semantics								
	Standardization								
	Growing market								
	More strategic outsourcing							-	

Table 19: Possible effects of service orientation in software on the sources of buyer bargaining power

The identified possible effects are described below. Also, it is noted how many of the 'hands-on' experts mentioned the found differences, if applicable.

Service orientation is a new way of software development. As such, service-oriented applications may need specific management services that are not widely available yet. Also, since service orientation is rather new, client organizations may not yet have full insight into the matter, making it hard for them to judge and value the service offerings of different applications management providers. So, at first, the early adopters of service-oriented applications lose their ability to play service providers against each other. Once the technology becomes generally adopted, they may regain this bargaining power. However, according to a recent Gartner research (Sholler et al. 2006) it may take a few years before the technology has become fully mainstream. (Mentioned by 1 'hands-on' expert)

Furthermore, service-oriented business applications are likely to facilitate strategic goals of customers, making the correct functioning of the applications even more important. As the importance rises, the customers are more likely to focus on the quality of the service offering of an applications management provider, rather than price (Porter 1980). This would lower their bargaining power.

Also, the character of continuous change makes it hard to predict the future needed applications management services. Therefore, contract horizons may well be less distant. This would diminish the ability of buyers to artificially increase their purchase volume by bundling the needs of several years into one acquisition.

Therefore, the bargaining power of client organizations in the market for applications management for service-oriented software is thought to be less than it is in the market for traditional applications management services. The 'hands-on' experts rate the buyer bargaining power 'great' twice, 'little' once, and 'neither little nor great' once. Therefore, it is thought to be moderate.

The bargaining power of suppliers. The group of suppliers in the competitive environment of the market for applications management services for service-oriented software still is formed by labor. The below table presents the results of the analysis of the sources of their power to bargain from the applications management providers (see: Table 20).

		The supplier group is dominated by few and is more concentrated than its client industry	The supplier group needs not contend with substitutes	The industry is not an important customer to the supplier group	The supplier's product is an important input to the buyer's business	The supplier group's product is differentiated or has built up switching costs	The supplier group poses a credible threat of forward integration
Characteristics	Modular					+	
	Open standards						
	Decoupling functionality from implementation						
	Standardized interfaces						
	Machine-to-machine						
	Asynchronous						
Benefits	Easier systems integration						
	Lower development costs						
	More flexible						
	Better system						
Concerns	Security, privacy, and trust						
	Reliability						
	Service Management and determining Quality of Service						
	Performance						
	Semantics						
	Standardization						
	Growing market						
	More strategic outsourcing						

Table 20: Possible effects of service orientation in software on the sources of supplier bargaining power

These findings are elaborated on below. Also, it is noted how many of the 'hands-on' experts mentioned the found differences, if applicable.

In chapter 2 it was already stated that the current labor market for IT personnel is tight. The labor market for IT personnel that master the new standards, tooling, and technology for service-oriented software is likely to be even tighter, which makes the labor of those that do have the right skills even more differentiated. (Mentioned by 3 'hands-on' experts)

Accordingly, the bargaining power of labor is considered to be greater in the market for applications management for service-oriented software than it is in the market for traditional applications management services. The 'hands-on' experts rate the supplier bargaining power 'great' twice, 'very great' once, and 'neither little nor great' once. Therefore, it is considered to be great.

Rivalry among existing firms. In the below table the results are presented of the analysis of the intensity of rivalry among competitors (see: Table 21).

		Numerous or equally balanced competitors	Slow industry growth	High fixed or storage costs	Lack of differentiation or switching costs	Capacity augmented in large increments	Diverse competitors	High strategic stakes	High exit barriers
Characteristics	Modular								
	Open standards								
	Decoupling functionality from implementation								
	Standardized interfaces								
	Machine-to-machine								
	Asynchronous								
Benefits	Easier systems integration								
	Lower development costs								
	More flexible								
	Better system								
Concerns	Security, privacy, and trust								
	Reliability								
	Service Management and determining Quality of Service								
	Performance								
	Semantics								
	Standardization								
	Growing market	-	-		-				
	More strategic outsourcing								

Table 21: Possible effects of service orientation in software on the intensity of rivalry among competitors

The findings will be described below. Also, it is noted how many of the 'hands-on' experts mentioned the found differences, if applicable.

Currently, few (major) applications management providers offer applications management services for service-oriented software. Therefore, the competition among those who already are may well not be as intense as in the market for traditional applications management services. (Mentioned by 2 'hands-on' experts)

Also, the market for applications management services for service-oriented software is growing faster than that for traditional applications management services. As a result, competing applications management providers that want to grow can do so less at the expense of each other.

Furthermore, in the new and growing market the competing applications management providers are found to each offer applications management services for a limited variety of service-oriented software, which limits the overlap among their offerings. Therefore, the differentiation among providers is thought to be greater than it is in the traditional applications management services market. (Mentioned by 1 'hands-on' expert)

All in all, the intensity of rivalry among the competitors in the market for applications management services for service-oriented software is considered less than in the traditional applications management market. The 'hands-on' experts rate the rivalry among competitors 'great' once, 'little' once, 'neither little nor great' once, and one abstains from rating. Therefore, it is considered to be moderate.

4.4.4 Conclusion: market attractiveness

In the foregoing paragraphs the market for applications management services for service-oriented software has been analyzed along several aspects. These analyses show that this market is growing rapidly—about 20% a year for the upcoming years—and that the motives for clients to acquire the applications management services offered in this market—that is, to outsource the applications management of their service-oriented software to an external provider—are likely to be of a more strategic nature, which may make clients less price sensitive, which in turn may allow higher margins for applications management providers.

The analysis of the competitive environment of the market for applications management services for service-oriented software shows that entering the market is rather easy. Furthermore, it shows that few substitutes exist for the services offered in this market, which limits their ability to put an upper limit to the prices charged for applications management services for service-oriented software. Also, few providers are already active in this market, and those that are do not have greatly overlapping offerings, and the market is growing, so competition is rather low. The immaturity of the market also causes the bargaining power of client organizations to be only moderate. All these factors add to the attractiveness of the market for applications management services for service-oriented software.

On the other hand, however, the bargaining power of labor with the right skills for this market is great, because their availability is scarce. Also, the easy entry to the market also holds for other parties, which makes the threat of new entrants great.

To conclude, the young and rapidly growing market for applications management services for service-oriented software is considered to be an attractive one.

4.5 Opportunities and threats

[Confidential content was removed]

4.6 Possibilities for the AMSC to add value to this market

[Confidential content was removed]

4.6.1 Success criteria

The above research has shown that there exist a number of possibilities for the AMSC to add value to the applications management market for service-oriented software. To determine whether the competitive strategies found above are sound strategies, they need to be assessed. To this end Johnson et al. (2006) have formulated *success criteria*: suitability, acceptability, and feasibility. Johnson et al. define these as follows:

- "*Suitability* is concerned with whether a strategy addresses the circumstances in which an organization is operating (…)."
- "*Acceptability* is concerned with the expected *performance outcomes* (such as the *return* or *risk*) of a strategy and the extent to which these would be in line with the *expectations* of stakeholders."
- "*Feasibility* is concerned with whether a strategy could be made to work in practice."

In the following, the competitive strategies will be assessed on each of the criteria.

[Confidential content was removed]

4.7 Conclusion

The foregoing analyses have shown that many benefits are ascribed to service-oriented software, but that empirical grounds for those claims are lacking. Also, analysis showed that offering applications management services for service-oriented software requires some new competences, such as mastering Web services technology and related tooling and having a more process-oriented view on software. Furthermore, analysis showed that the young and rapidly growing applications management market for service-oriented software is an attractive one.

[Confidential content was removed]

Value exchange	[Confidential content was removed]
Buyer power	Moderate
Supplier power	Great
Threat of new entrants	Great
Threat of substitutes	Moderate
Competition	Moderate
Market size	Potentially 50% of new mission-critical applications
Expected growth rate	About 20% yearly
Outsourcing motive	Strategic, tactical, and operational

Table 22: Characteristics of the market for applications management for service-oriented software

4.8 Recommendations

[Confidential content was removed]

5 Applications management for open-source software

5.1 Introduction

The second emerging market for applications management services is that for open-source software. In the following it will be analyzed how the case organization can add value to this market by means of a SWOT analysis.

To this end, the opportunities and threats of the applications management market for open-source software is determined. This is done similarly to the analysis of the opportunities and threats of the applications management market for service-oriented software. So, the applications management market for open-source software is analyzed along the three aspects open-source software, applications management, and the market. The analysis of open-source software shows that no evidence exists to irrefutably claim its superiority over proprietary software—although its proponents do so—and that the way open-source software is generally developed—by a community of contributors—is favorable to generic context-independent applications, instead of highly specific business applications, although these are the common subject of applications management. The analysis of how applications management work for open-source software is different from traditional applications management work shows that the only difference is the existence of a supporting community and the possibility to leverage that community. This would require of the applications management provider the skills to interact with the supporting community. The analysis of the market shows that it seems to be unattractive, for its size is considered very limited (due to the nature of the software; see above) and because four of the five forces that form its competitive environment are less favorable than in the traditional one (only rivalry is considered to be equal). These analyses are again analyzed to identify any opportunities and threats. Then these are combined with the strengths and weaknesses of the AMSC—determined before—in a SWOT analysis to find out how the AMSC can add value to this market. [Confidential content was removed] Finally, recommendations are given.

5.2 Open-source software

In the last few years much attention has been paid to open-source software as a feasible alternative for expensive proprietary software. Proponents of open-source software emphasize its benefits, but on the other hand there is a lively group of opponents calling these benefits into question. In the following open-source software is defined and its attributed benefits are presented, as well as the concerns regarding it. This forms the basis for further discussion of the topic.

5.2.1 Definition

To capture the essence of open-source software it is not sufficient to name source code access only. Stewart et al. (2006) identify three defining features to classify open-source software: the license under which the software is distributed, the software development, and the price of the resulting product. Each of these aspects will be detailed below.

License

To understand the importance of the distribution license to open-source software it is useful to know about its origins. The roots of open-source software date as far back as the 1950s: the beginnings of the computing field. In those days, all software was free, and most of it was open. However, this was not out of some ideology, but rather because no-one had yet discovered software as a product with commercial value. It was not before the mid-1970s before the marketplace for software became significant (Glass 2004). But the commercial sale of software was—and still is—accompanied with copyrights and restrictive licenses, limiting the user's freedom to redistribute or modify the software.

In 1985, Richard Stallman founded the Free Software Foundation (FSF)[20], aimed at developing a free operating system—the GNU project. The term 'free' has nothing to do with price, but with the rights the user was given. Free software would let the user, for example, use the program for any purpose and redistribute it, just as in the early days of computing. It would also grant the user the right to modify the program as desired. To be able to effectuate this right, users had to have access to the source code. To safeguard these freedoms, copyright laws were applied, but in a turned around manner: copyleft. This works as follows:

"To copyleft a program, we first state that it is copyrighted; then we add distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program's code or any program derived from it but only if the distribution terms are unchanged. Thus, the code and the freedoms become legally inseparable."(Stallman 1996)

To this end the GNU General Public License (GPL) was set up (Ljungberg 2000).

However, the term 'free software' made certain users—especially business—uneasy, because of its association with ethical issues like freedom. This discomfort was considered to impede the wide adoption of the software (Stallman 1998). In 1998 the term 'open-source' was coined as to overcome the discomfort and to enable the widespread of the software; the same year the Open Source Initiative (OSI)[21] was founded. The OSI and the FSF—both still existent—represent separate movements in the free software field, although both advocate source code access: the FSF does so out of ideology, the OSI just for pragmatic reasons. As Richard Stallman puts it:

"Open source is a development methodology; free software is a social movement. (…) We disagree on the basic principles, but agree more or less on the practical recommendations" (Stallman 1998)

This is illustrated by the below figure (see: Figure 13) which maps the principles of the Open Source Definition (OSI 2006) to those of the Free Software Definition (FSF 2007). As you can see, they match for the greater part.

[20] http://www.fsf.org

[21] http://www.opensource.org

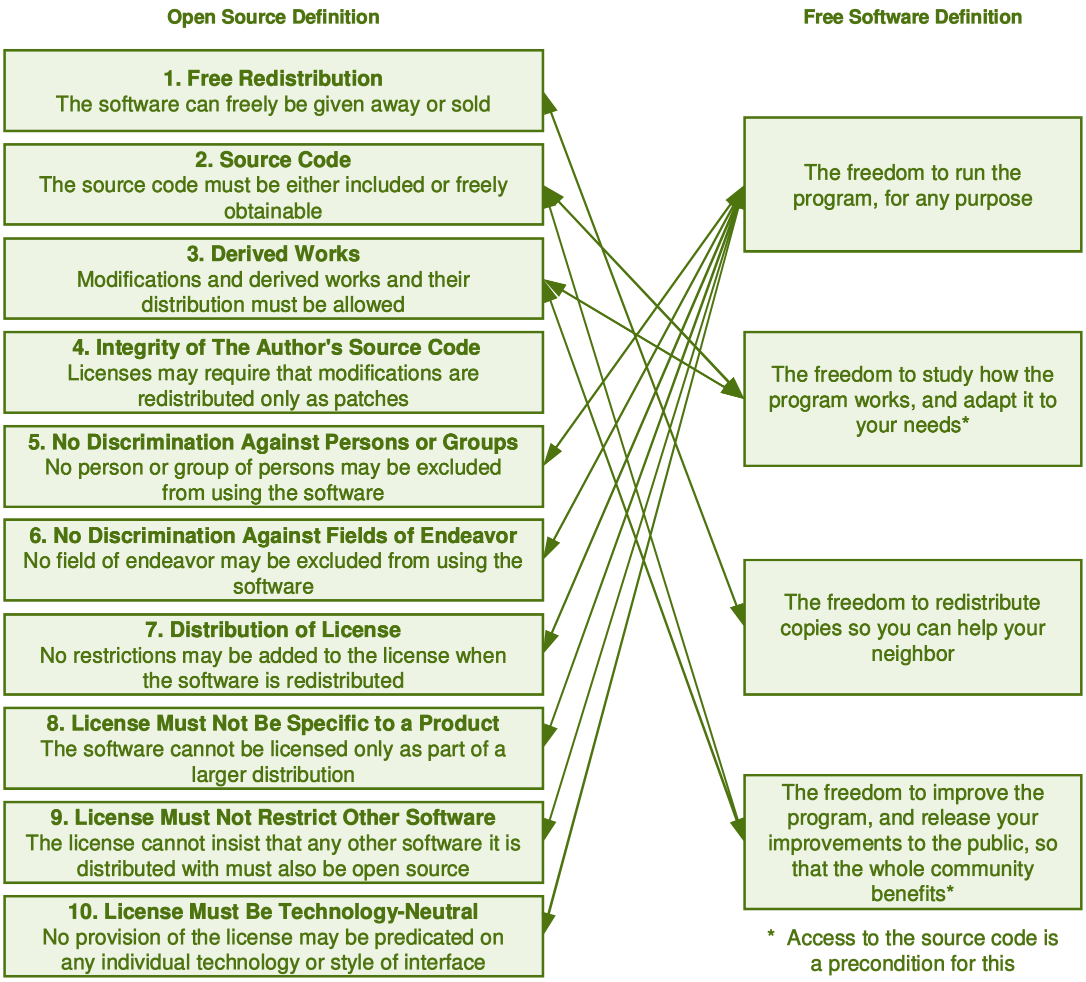

Open Source Definition
Free Software Definition
1. Free Redistribution
The software can freely be given away or sold
2. Source Code
The source code must be either included or freely obtainable
3. Derived Works
Modifications and derived works and their distribution must be allowed
4. Integrity of The Author's Source Code
Licenses may require that modifications are redistributed only as patches
5. No Discrimination Against Persons or Groups
No person or group of persons may be excluded from using the software
6. No Discrimination Against Fields of Endeavor
No field of endeavor may be excluded from using the software
7. Distribution of License
No restrictions may be added to the license when the software is redistributed
8. License Must Not Be Specific to a Product
The software cannot be licensed only as part of a larger distribution
9. License Must Not Restrict Other Software
The license cannot insist that any other software it is distributed with must also be open source
10. License Must Be Technology-Neutral
No provision of the license may be predicated on any individual technology or style of interface
The freedom to run the program, for any purpose
The freedom to study how the program works, and adapt it to your needs*
The freedom to redistribute copies so you can help your neighbor
The freedom to improve the program, and release your improvements to the public, so that the whole community benefits*
* Access to the source code is a precondition for this

Figure 13: Principles of the Open Source Definition and the Free Software Definition

Both sets of principles can also be viewed in the light of modern copyright law. The foundation of modern copyright law is formed by the Berne Convention treaty (Berne Convention 1979), which identifies two types of rights the original creator holds: exclusive rights and moral rights. Unfortunately, the naming of these types of rights may be a cause for confusion. The exclusive rights are to guarantee that the original creator can exploit the work. Therefore these rights are initially granted to the original creator *exclusively*. However, the original creator can decide to transfer the rights to exploit his work to someone else, exclusively as well as non-exclusively. An example of this is often seen in the music industry, where artists hand over the exclusive rights of their songs to a record company in exchange for some remuneration. Moral rights cannot be transferred and comprise "the right to claim authorship of the work and the right to object to any mutilation or deformation or other modification of, or other derogatory action in relation to, the work which would be prejudicial to the author's honor or reputation (Berne Convention 1979)".

Under both sets of definition principles the creators disclaim the greater part of their exclusive rights. Also, both sets of principles allow the original work to be used by anyone in any context, which practically nullifies the author's moral right to object to specific applications of the work. The principles of the Open Source Definition, however, do provide creators the possibility to demand integrity of their work (Principle 4). The principles of the Free Software Definition do not offer such a possibility. And finally, both sets of principles offer creators possibilities to claim authorship. This can be done, for example, by using the license itself, which is inseparable from the code and which may not be changed when redistributed. So by recording their names in the license, creators can make sure their authorship is mentioned with every distribution of the software.

Essentially, software can be defined as open-source when it is distributed under a license that adheres to the principles of the Open Source Definition. So, this definition will be used in this research.

Development

Besides the specific license under which the software is distributed, another distinctive feature of open-source software is that it is developed and maintained by a community of voluntary contributors (Stewart et al. 2006). Such a community, typically, is onion-shaped (see: Figure 14). At the center are the core developers of an open-source software project. Although this generally is only a small group of people (10-15 persons), they are responsible for the majority of code development (about 80%) (Koch and Schneider 2002; Mockus et al. 2002). The founder of the specific open-source project commonly is among the core developers and leads the project. The core developers decide on what to include in the project.

Surrounding the core are the co-developers. These also write code, but their contributions first need core developer approval before it is included in the software. Then there is a layer of active users, that test releases, post bug reports, and answer questions from the passive users, who use the software but do not contribute at all (Crowston and Howison 2006).

Although the developers contribute to the project voluntarily, many of them receive some kind of financial compensation for their work on the project—for example, they are paid by their employer to work on the project (Hars and Qu 2002).

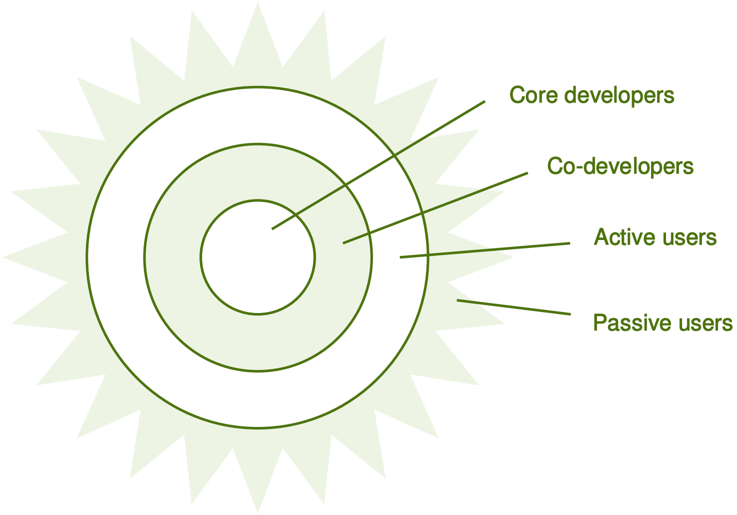


Figure 14: Onion shape of an open-source community (adapted from Crowston and Howison 2006)

However, development by a community is not a requirement. The definition of open-source software leaves room for examples like Mozilla's Firefox and StarOffice's OpenOffice which where originally developed as proprietary software, but which had their source code made public at some point in time to continue as an open-source project.

Product price

Contrary to common belief, open-source software is not equal to cost-free software (AlMarzouq et al. 2005). This misconception is fueled by the term 'free software'. Yet, 'free' refers to the freedoms the user is given, rather than the software being free of charge (Stallman 1998). Nonetheless, the prices and license fees of open-source software are significantly lower than those of proprietary products (Lawton 2002).

From the above description of open-source software some key aspects that affect the maintenance and management of the software can be identified. These are described below.

Source code access. Having access to the source code of an application allows users to study its workings. This is contrary to commercial—or closed-source—software, which tries to protect and exploit its intellectual property by not providing such insight into the source code.

Modification rights. Not only does open-source software provide access to its source-code, it also allows its users to modify the application to their needs—for example to accommodate the software to a local or minority language (Paulson 2004a). Again this is contrary to commercial software that generally denies you the right to alter or extend the software, even if you could do so without knowing the source code.

Development and maintenance by a community. Instead of being developed and maintained by a single software vendor, open-source software often is the product of the continued collective effort of a community of developers.

5.2.2 Benefits and Concerns

Proponents of open-source software claim that it is superior to proprietary software. They say that the community of motivated developers constantly testing and enhancing the code results in higher quality and more secure software, with shorter development cycles (Raymond 2000). To find out whether these claims are supported by research and to identify any concerns associated with open-source software the top-25 Information Systems (IS) journals (Mylonopoulos and Theoharakis 2001) were subjected to a structured analysis. First the criteria for selecting publications are described, then the search method is presented, subsequently the results are stated, and finally, the findings are discussed.

Selection criteria

The following criteria were used to select publications that are included for review:

- Published in one of the top-25 IS journals;
- The main topic of the article is open-source software, a specific aspect of it—for example, code quality of open-source software—or a comparison of open-source software and proprietary software on some aspect;
- Published in the last ten years (1997 – present), because the notion of open-source software did not exist before then.

Letters and opinions are excluded from further review.

Search method

To search the top-25 IS journals for relevant articles the online search engine of Web of Science was used. This search engine covers 23 of the top-25 IS journals. The missing journals ('Communications of the AIS' and 'Accounting, Management, and IT') were searched separately. 'Communications of the AIS' was not indexed in any search engine, so it was searched by browsing the tables of content; 'Accounting, Management, and IT' (and its successor 'Information and Organization') was searched with the search engine of Emerald.

On October 7, 2007 the search engines of Web of Science and Emerald were probed with the following search term: "open-source*" OR "open source*" OR "opensource*". Web of Science returned 154 items that fitted the first and last selection criteria, Emerald zero[22]. Based on article

[22] Again, this finding was verified by browsing the tables of content of both titles, which confirmed the absence of relevant articles.

titles and abstracts, the articles that also satisfied the second criterion were identified; 42 in total. From these 16 letters and opinions were excluded, leaving 26 articles for consideration. Browsing the tables of content of the 'Communications of the AIS' resulted in another three articles that met the criteria. Thus, the final number of articles that were selected for review was 29.

Results

The review shows no convincing support of the benefits claimed by the proponents of open-source software. In the table below (see: Table 23) the possible benefits of open-source software that are cited in the selected articles are given, as well as the articles that discuss the matter. Furthermore, it is stated whether the articles just claim a benefit without further underpinning or that experimental findings on the matter support or contradict the benefit, or that they turn out to be indecisive.

	Higher quality	More secure	Lower costs	Shorter develop-ment cycles	Higher project growth rate	More functions added over time	Less complex	More modular
(Ljungberg 2000)	Claimed							
(Wu and Lin 2001)	Claimed							
(Wall 2001)			Claimed					
(Payne 2002)		Claimed						
(Stamelos et al. 2002)	Contrary findings							
(Lawton 2002)			Claimed					
(Mockus et al. 2002)	Indecisive findings			Supporting findings				
(Huntley 2004)							Supporting findings	
(Paulson et al. 2004)	Supporting findings				Contrary findings	Supporting findings	Contrary findings	Contrary findings
(Paulson 2004b)			Claimed					
(Samoladas et al. 2004)	Supporting findings							
(Raghunathan et al. 2005)	Indecisive findings		Indecisive findings	Claimed				
(MacCormack et al. 2006)								Supporting findings
(Hoepman and Jacobs 2007)		Claimed						
(Riehle 2007)			Claimed					

Table 23: Benefits of open-source software and their underpinning

Besides the fact that the claimed benefits of open-source software are not convincingly supported by research findings, the selected articles also reveal several concerns regarding open-source software. In the below table these concerns and the articles that cite them are depicted.

	Not established in every area	Size of supporting community	Selective code reading	Higher TCO	Not deadline driven / no guarantees
(Lewis 1999)	x	x			
(Glass 2003)			x		
(Fitzgerald 2004)		x			
(AlMarzouq et al. 2005)	x			x	x
(Raghunathan et al. 2005)	x				

Table 24: Concerns surrounding open-source software and their citations

Each of these concerns will be detailed below, in order of the number of citations.

Not established in every area. In 1999, Lewis noted that open-source software had hardly ever "crossed the chasm into mainstream without first becoming a commercial product sold by a commercial enterprise" (Lewis 1999). AlMarzouq et al. (2005) note that, in some areas, open-source software still is not as well established as proprietary software. A possible explanation is provided by Raghunathan et al. (2005). The authors argue that the open-source way of software development is most viable for generic context-independent applications such as operating systems, network software, word processors, and spreadsheets, because the development of those applications require no specific user input. This, however, makes open-source development not applicable for highly specialized business applications.

Size of community. In his often-cited article, O'Reilly (1999) notes that the claimed benefits of open-source software are based on the idea that the Internet enables a larger community of developers than can be applied by even the largest companies. But Lewis (1999) illustrates by comparing the development of open-source Linux to Microsoft's Windows NT operating system that this assumption is not always true. Furthermore, the number of contributors diminishes still further when the hype wears off an open-source application (Lewis 1999). Also, Fitzgerald (2004) notes that studies of two popular open-source software development Web sites have shown that most projects have only one or two developers. Having a small or inactive community may seriously jeopardize the continued maintenance and support of an open-source application.

Selective code reading. Another assumption underlying the claimed benefits of open-source software proponents is that users can read the source code, find errors, and fix them. Glass (2003) points out that this works just fine when the users are programmers, but if they are not this is simply a technical impossibility. This means that only code where programmers are the users is likely to reap the benefits of open source code reading.

Higher total cost of ownership (TCO). Although the initial acquisition costs of open-source software may be lower than proprietary software, AlMarzouq et al. (2005) note that the TCO of open-source software may actually be higher when staff training, customization, and implementation are accounted for.

Not deadline-driven / no guarantees. As open-source software projects are based on voluntary contributions deadlines do not exist. This may become a problem if users depend on an anticipated future event—for example, a bug-fix. So, users may acquire their support from the community, but such support is not guaranteed (AlMarzouq et al. 2005).

It must be said, however, that many of the most popular open-source software are also distributed by commercial parties. These parties, like RedHat and SuSE for Linux, have formed their business model by selecting open-source software from all available, testing their interoperation, distributing this tested collection of software, and by providing updates and support for their distributions. In the case of Linux, SuSE is cited to have a market share of 30 percent in Europe, making it market leader (Wu and Lin 2001). Such a commercial party can provide guarantees.

5.2.3 Conclusion

What stands out from the above analysis is that the claimed benefits of open-source software are not convincingly supported by empirical evidence. Moreover, some of the concerns actually counter the claimed benefits or their supporting grounds. For example, the supposed higher quality of open-source software as a result of numerous contributors reading the code and fixing bugs is undermined by the concerns about the size of the community and the selective code reading of its members. The lower costs that open-source software is claimed to bring about is countered by the concerns about TCO. It is therefore concluded that the superiority of open-source software over proprietary software is very arguable, even though open-source software in itself still may be of real value to its user.

Although the superiority of open-source software as claimed by its proponents is not irrefutably proven by empirical evidence, organizations may have other arguments to prefer open-source software over proprietary software. Government organizations, for example, may choose open-source software because it allows them to verify that discretion is guaranteed. Other reasons may be avoidance of vendor lock-in, the ability to use it as a component of a larger custom system, and usage out of some conviction.

Finally, open-source software is found to be especially viable for generic context-independent applications such as operating systems, network software, word processors, and spreadsheets. These kinds of software, however, are commonly counted among infrastructure technology. Open-source software is found not to be applicable for—highly—specific business applications, while these form the general subject of applications management services. Nonetheless, the right to modify and extend a piece of open-source software may lead client organizations to use open-source software as a building block for some specific business application. In such a case however, the final application becomes very much a bespoke software application, and bespoke applications are already dealt with in the traditional market for applications management. Therefore, the relevance of open-source software as a market for applications management services is considered to be very limited.

5.3 Resource-based analysis

Above, open-source software has been defined and its benefits and concerns have been identified. Now, it will be analyzed how applications management of open-source software is different from traditional applications management.

Therefore, first, the effects on the actual applications management work are identified, followed by an analysis of the value chain of which this work is part. The effects of open-source software on the applications management work are analyzed by reviewing relevant academic literature. This analysis shows that the work is only different in that with open-source software the community supporting the software can be leveraged as an extra aid in performing applications management. The analysis of the value chain of which the work is part shows that it is different from the value chain of traditional applications management, because now the supporting community also is included. From these analyses it is concluded that the applications management provider may require a new competency, namely knowing how to interact with the community if it wishes to leverage its support.

5.3.1 Effects on applications management services

As Richard Stallman is cited above, open source is a development method. Although this may mean a departure from the general—proprietary—way of software development, it does not mean that the resulting software is fundamentally different than other—traditional—software. The analyses of its claimed benefits and the concerns regarding open-source software in the previous paragraphs also did not provide hard evidence for the existence of such a difference.

Nonetheless, the articles selected for review in Paragraph 5.2.2 were searched again, this time looking for any statements on the management and maintenance of open-source software. This second review only yielded the general notion that with open-source software maintenance is performed by a community. Unfortunately, no statements were found that pointed to any concrete differences between the applications management of traditional and open-source software.

Therefore the search has been extended. The search process is described below. First, the altered criteria for selecting an article for inclusion are detailed, followed by the search method. Finally, the results are presented and a conclusion is given.

Selection criteria

The following selection criteria were used to select publications to be included for review:

- Published in one of the top-50 IS journals as ranked by Mylonopoulos and Theoharakis (2001) or in a journal dedicated to software maintenance;

- Making any concrete statement as to how applications management of open-source software is different from traditional applications management.

Search method

The journals searched and the way these were searched were identical to the search described in Paragraph 4.3.1, except for the search strings of course.

On August 20, 2007 all search engines except ACM Digital Library were probed with the following search string: '("open-source*" OR "open source*" OR opensource*) AND (maint* OR "application? management")'. ACM Digital Library required the below sequence of searches to achieve similar results:

```
+publication:SIGSOFT +maintenance +"open source"*

+publication:SIGSOFT +maintenance -"open source" +"open-source"*

+publication:SIGSOFT +maintenance -"open source" +"open-source" +opensource*

+publication:SIGSOFT +"application? management"
```

The Journal of Software Maintenance and Evolution: Research and Practice was searched separately, discarding the 'AND maint*' part from the search string, for this is covered by the focus of the journal itself.

Web of Science returned 18 articles, but none turned out to be relevant. ScienceDirect returned zero items, which is in line with the findings in Paragraph 4.2.2. The ACM Digital Library searches returned 132, 0, 5, and 9 items respectively, only one of which was relevant. Emerald returned—again—five articles, but zero met the second criterion. EBSCO returned zero items, which is consistent with the findings in Paragraph 4.3.1. Browsing the tables of content of the remaining journals did not result in any relevant articles to be added. Searching the Journal of Software Maintenance and Evolution: Research and Practice returned three items, but again none turned out to be relevant. Thus, the final number of relevant publications found was one.

Results

The one article found, (Koponen and Hotti 2005), finds that the maintenance process of two major open-source projects is very similar to traditional software maintenance. This underwrites the suspicion of non-distinctiveness uttered in the beginning of this paragraph.

However, this finding is based on only a single article. So to increase the validity of this claim a practitioner[23] was also consulted on the matter. In this conversation he too indicated that there are no real differences between managing traditional and open-source software: it very much resembles managing a piece of bespoke software. Both provide source code access and both allow modifications to be made to the software. However, with open-source software the community can be leveraged as an extra aid. In the case of open-source software as a packaged distribution from some specific—commercial—party the situation is similar to traditional packaged software, but now with the extra possibility to make modifications yourself and / or to leverage the community.

Conclusion

All the analyses in the previous paragraphs indicate that with open-source software the actual software itself is not essentially different from traditional software. Also, it was found that managing and maintaining open-source software is only different in that the applications management provider has more options to choose from when performing its tasks: it can await

[23] Mr. Ernest Neijenhuis, conversation of August 27, 2007. As a senior Unix administrator he has several years of experience with maintaining Linux.

any inputs from the distributor (if any), it can search or leverage the community, or it can do things all by itself.

5.3.2 Value chain

Just as any market, the market for applications management services for open-source software is built up out of a chain of value adding activities that eventually form the final product. The analyses in the previous paragraphs show that the value chain for this market is a bit different from the market for traditional applications management services. The following differences can be identified.

Interaction with a community. In the previous paragraph it is concluded that an applications management provider may provide its services with or without further inputs from the community. In the case that the applications management services provider does want to use inputs from the community that supports the open-source software application at hand, it needs to interact with that specific community. The applications management provider may attract updates or bug-fixes from the community or it may search the community for help or documentation. But the interaction is not limited to leveraging the community in such a way. The services provider also may contribute its own added value back to the community, for example by posting its self-made bug-fixes.

Distributor instead of a software vendor. If a piece of open-source software is distributed by a specific—commercial—party an applications management provider may also turn to that distributor for inputs or help. Often, these distributors offer support contracts to accompany their distribution of the software that offer helpdesk support, updates, etcetera—much the same as software vendors do. Practically, this boils down to a naming issue.

In the below picture (see: Figure 15) these issues are depicted. It also shows that distributors may also contribute back to the community. The dotted shapes and lines represent that these are optional.

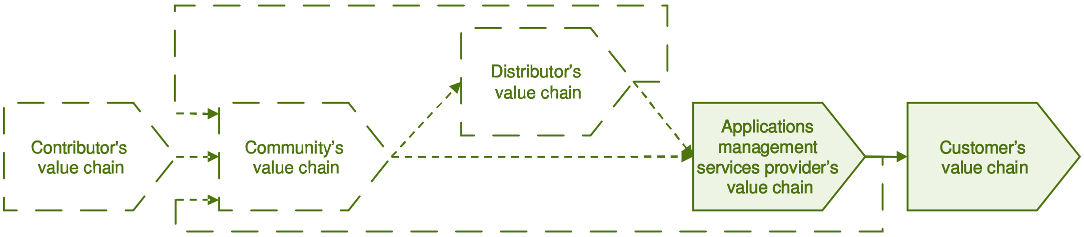

Distributor's value chain
Contributor's value chain
Community's value chain
Applications management services provider's value chain
Customer's value chain

Figure 15: Value chain for the applications management market for open-source software

5.3.3 Conclusion: required competences

In the previous paragraphs it is shown how applications management for open-source software is very little different from that of traditional software. The difference that exists is found in the possibility of interacting with a community that supports a certain piece of software, by using it as an extra resource in the applications management process and / or by contributing back to it.

In order to successfully interact with such a community, an applications management provider needs to know how to do so. This requires knowledge of the community itself, for example knowing what websites, forums, or discussion groups to turn to. It also requires knowledge of how to communicate with such a community to get the best result out of it.

5.4 Market-based analysis

After the analysis of how open-source software affects the applications management work, now the applications management market for open-source software will be analyzed, so that the attractiveness of the market can be determined. First, the market is analyzed in terms of size by consulting the publications of market research agencies. This analysis shows that a market growth of about 5 to 10% yearly is predicted. This is followed by an analysis of the motives client organizations—current as well as potential—may have to have their open-source software managed and maintained by an applications management provider, that is, why they have their applications management outsourced. This shows that clients may have mostly motives of operational efficiency to outsource their applications management. Then, the competitive environment of the market is analyzed by using the five forces framework. This analysis shows that the applications management market for open-source software forms a more hostile competitive environment than the traditional applications management market does, because four of the five forces that form the competitive environment are less favorable (only rivalry is considered to be equal). Based on these analyses it is concluded that the applications management market for open-source software is an unattractive one.

5.4.1 Market size

To determine the market size for applications management services for open-source software the publications of two major market research agencies (Forrester and Gartner) were analyzed, to find any reports on this matter. First, the criteria for selecting publications are stated, followed by an explanation of the search method. Finally the results are presented.

Selection criteria

The following criteria were used to select publications that are included for review:

- Published by Forrester or Gartner;
- The publication provides a forecast of the market size for applications management services for open-source software in Europe, Western Europe, or the Netherlands;
- Published in the past year.

Search method

To search the publications of both market research agencies the search engines offered on their respective websites were used. On August 22, 2007 the advanced search form of Gartner was probed with the search term '"open-source" OR "open source" OR "opensource"' and the search restricted to publications of the past year, dealing with region 'Europe', topic 'IT Services', and of content type 'Forecasts'. This resulted in 2 items found, one of which dealt with North America only and which was therefore discarded.

The same day the same search term was entered into the advanced search form of Forrester and again the search was restricted to publications of the past 12 months, considering Europe, forecasts only. This search returned three items, none of which turned out to be relevant.

Results

The one Gartner forecast found predicts a compound annual growth rate (CAGR) of the Western European market for IT management services for open-source software of 5.6% in the period 2005-2010. Moreover, the CAGR of the market for support services for open-source software is predicted to be 11.2% in the same period (Mac Neela et al. 2007). Unfortunately, no notions on the current size of the market were found.

However, following the argumentation of Paragraph 5.2.3, the market for applications management of open-source software is considered to be very limited.

5.4.2 Motives to outsource applications management

Open-source software is most viable for generic applications and not for highly specialized business applications (Raghunathan et al. 2005). A recent Gartner user survey shows that open-source software is most commonly used as server operating system and database management system (Mac Neela 2006). These generic applications do not provide its users with a sustainable competitive advantage over their competitors. After all, to do so the applications have to be valuable, rare, imperfectly imitable, and non-substitutable (Barney 1991). Though open-source software may be valuable, it fails on the other conditions. This makes that open-source software in unlikely to be outsourced for strategic reasons. Rather, the generic open-source software is outsourced for reasons of operational efficiency, and possibly for tactical reasons like freeing up company IT staff for other purposes.

5.4.3 Competitive environment

The analyses in the foregoing paragraphs already have shown that managing open-source software is only different in that a community exists that supports a certain piece of open-source software and that can be used as an extra resource. This extra resource—which is available to all—may cause the competitive environment of the market for applications management of open-source software to be different from that for traditional applications management. To identify any differences in the competitive environment the one article selected in Paragraph 5.3.1 was searched again, now for any statements on this matter. However, it did not mention any such statements.

Nonetheless, it is deemed unlikely that such a valuable resource does not cause any difference in the competitive environment at all. Therefore, an initial attempt will be made to determine the differences between the traditional market for applications management and that for open-source software.

To this end, the effects of the existence of a community on the source of each of the five forces that shape the competitive environment are analyzed. To increase the empirical validity of this analysis a 'hands-on' expert was consulted on the matter. This 'hands-on' expert, Mr. Ernest Neijenhuis, works as senior Unix administrator at Capgemini Outsourcing Infrastructure Management, and he has about 8 years of 'hands-on' experience with Linux and its community. His insights combined with any effects that follow from the analyses in the foregoing paragraphs are mapped to the sources of each of the five forces as Porter (1980) has identified.

The threat of new entrants. The below table presents the results of the analysis of the effects of the existence of a community that supports a certain piece of open-source software on the barriers that prevent new parties from entering the market for applications management for open-source software (see: Table 25).

	Economies of scale	Product differentiation	Capital requirements	Switching costs	Access to distribution channels	Cost disadvantages independent of scale	Government policy	Expected retaliation
Maintenance by a community			-					

Table 25: Effects of the existence of a supporting community on the barriers to entry

The community that supports a certain piece of open-source software provides anyone with practically all of the knowledge needed to manage it, for free. This reduces the need to invest upfront in personnel and training and, thus, lowers the barrier to entry.

With a lower barrier to entry the threat of new entrants is thought to be greater than it is in the market for traditional applications management services. It is thought to be great.

The threat of substitute products. The effects of a supporting community on the threat of products that can substitute the services of applications management providers are shown in the below table (see: Table 26).

	Threat of substitute products
Maintenance by a community	+

Table 26: Effects of the existence of a supporting community on the threat of substitutes

Obviously, a community of people that supports and maintains a certain piece of open-source software increases the threat of substitutes, because it facilitates clients in their alternative of performing applications management themselves. Therefore, the threat of substitute products is considered to be greater than it is in the market for traditional applications management services. Nonetheless, it still is thought to be little.

The bargaining power of buyers. The buyers in the market for applications management for open-source software are those organizations that acquire those services for their open-source software. The below table shows the effects the existence of supporting community has on the bargaining power of these buyers (see: Table 27).

	Concentrated or large purchase volumes	Product represents a significant fraction of buyer's costs / purchases	Product is undifferentiated	Buyer faces few switching costs	Buyer earns low profits	Credible threat of backward integration	Product is unimportant to the buyer's product	Buyer has full information
Maintenance by a community						+		

Table 27: Possible effects of the existence of a community on the sources of buyer bargaining power

Following the same reasoning as above, the existence of a supporting community facilitates clients in performing applications management themselves. It thus increases the threat of backward integration they can pose. Therefore, buyer bargaining power is thought to be greater than it is in the market for traditional applications management services. It is thought to be great.

The bargaining power of suppliers. The suppliers to the market for applications management for open-source software still are found in labor. The below table presents the results of the analysis of their ability to bargain from the applications management providers (see: Table 28).

	The supplier group is dominated by few and is more concentrated than its client industry	The supplier group needs not contend with substitutes	The industry is not an important customer to the supplier group	The supplier's product is an important input to the buyer's business	The supplier group's product is differentiated or has built up switching costs	The supplier group poses a credible threat of forward integration
Maintenance by a community					+	

Table 28: Possible effects of the existence of a community on the sources of supplier bargaining power

In an already tight labor market finding personnel that also has the knowledge and skills needed to interact with a community may well be even harder to find. This makes the services of those that do possess such knowledge and skills even more differentiated. Therefore, the supplier bargaining power is considered to be greater than it is in the market for traditional applications management services. It is thought to be great.

Rivalry among existing firms. The existence of a supporting community is not thought to have any altering effect on the intensity of rivalry among the existing competitors. This is represented by the empty table below (see: Table 29). It is considered to be moderate.

	Numerous or equally balanced competitors	Slow industry growth	High fixed or storage costs	Lack of differentiation or switching costs	Capacity augmented in large increments	Diverse competitors	High strategic stakes	High exit barriers
Maintenance by a community								

Table 29: Possible effects of a supporting community on the intensity of rivalry among competitors

5.4.4 Conclusion: market attractiveness

The foregoing paragraphs have presented a multi-faceted analysis of the market for applications management services for open-source software. These analyses have shown that the market size is very limited, although it grows about 5-10% yearly. They have also shown that client organizations will outsource the applications management for open-source software mostly for reasons of operational efficiency, striving for lower costs. This puts pressure on the margins that can be achieved. Also, the analysis of the competitive environment of the market shows that four out of the five forces turn out to be less favorable than in the traditional applications management market (only rivalry is considered equal), which already was found to be tough for applications management providers.

In conclusion, the market for applications management services for open-source software is considered to be unattractive.

5.5 Opportunities and threats

[Confidential content was removed]

5.6 Possibilities for the AMSC to add value to this market

[Confidential content was removed]

5.6.1 Success criteria

Now competitive strategies have been identified it needs to be determined whether these are also sound strategies. To assess competitive strategies Johnson et al. (2006) have formulated three success criteria: suitability, acceptability, and feasibility (for a more elaborate discussion of these concepts see: Paragraph 0). In the following, the competitive strategies will be assessed on each of the criteria.

[Confidential content was removed]

5.7 Conclusion

A lively discussion exists about the quality of open-source software. In this research it is shown that there is no irrefutable empirical evidence exists to assume superiority of open-source software over proprietary software. It is also found that open-source software is especially viable for generic context-independent applications such as operating systems and network software, but not for—highly—specific business applications, while these form the general subject of applications management services. Hence, the relevance of open-source software as a market for applications management services is considered to be very limited.

This research has also found that managing and maintaining open-source software requires very few new competences. If the applications management provider desires to leverage the community that supports a certain piece of open-source software it needs to develop an understanding of the community itself, as well as the skills required for interaction with that community.

Analysis of the applications management market for open-source software has shown that it is a very unattractive one. It is very limited in size, not growing fast, and it forms an even tougher competitive environment than the traditional application management market.

[Confidential content was removed]

Value exchange	[Confidential content was removed]
Buyer power	Great
Supplier power	Great
Threat of new entrants	Great
Threat of substitutes	Little
Competition	Moderate
Market size	Very limited
Expected growth rate	About 5-10% yearly
Outsourcing motive	Operational efficiency, possibly tactical

Table 30: Characteristics of the market for applications management for open-source software

5.8 Recommendations

[Confidential content was removed]

6 Conclusion and recommendations

6.1 Research design and method

The dynamic environment in which applications management providers operate forces them to regularly adapt their competitive strategies to the latest developments in their industry. In this research it has been analyzed how the case organization, Capgemini Outsourcing's Dutch Applications Management Service Center (AMSC), can continue to add value to the applications management market as it changes. This research has sought an answer to the following research question:

> **What value can the AMSC add to the changing applications management market?**

Based on the program of the 2007 edition of the International Conference on Software Maintenance the two most critical changes in the applications management market have been identified as the rise of service-oriented software and the rise of open-source software. Therefore, to find an answer to the research question the following approach has been taken. (Note that first the approach will be presented on an overview level. Afterwards, each of the steps will be presented in greater detail.) First, the current applications management market has been analyzed, as well as the case organization and its current value offer. Subsequently, it has been analyzed what possibilities exist for the AMSC to add value to the applications management markets that emerge due to the rise of the two types of software.

The current applications management market has been analyzed in terms of the typical services offered, the value chain of which the activities of applications management providers' activities are part, the size of the market, the motives client organizations have to acquire the applications management services from the market, and the market as a competitive environment.

The case organization has been analyzed in terms of its current competitive strategy, its value offer to the market, its competences, and its strengths and weaknesses.

The analyses of the possibilities for the AMSC to add value to the two applications management markets that emerge due to the rise of service-oriented software and open-source have been performed in a similar manner. First, the both types of software have been analyzed in terms of their benefits and the concerns regarding them. Next, it has been analyzed how the two types of software affect the way applications management is performed and how they affect the value chain of which the applications management providers' activities are part. Then, the markets for applications management these two types of software bring about have been analyzed in terms of size, the motives client organizations have to acquire applications management services from the markets, and the markets as competitive environments. These analyses have again been analyzed to identify any opportunities and threats. The resulting opportunities and threats have been combined with the strengths and weaknesses of the AMSC in a SWOT analysis to identify possible competitive strategies by which the AMSC can add value to both markets.

This research approach has proven to be successful, for it not only has provided insight in the current added value of the AMSC, but it also has produced several suitable competitive strategies by which the AMSC can add value to the applications management markets that emerge as a result of the rise of service-oriented and open-source software. Collectively, these provide an answer to the research question.

6.2 Research findings

In the following, the overall conclusions will be formulated. To this end, the research findings concerning the current market and the AMSC, those concerning the applications management market for service-oriented software, and the findings concerning the applications management market for open-source software are presented. From this the overall conclusions are derived.

Current applications management market and the AMSC

It was found that in the current applications management market demand is non-distinctive, demanded by clients seeking mostly operational efficiency. Furthermore, it was found that there are many equivalent providers, that there exists a considerable threat of entry by new players, and that there exists a potential threat that clients will have applications management performed in-house again. As this market offers little room for differentiation, the applications management providers compete on prices.

[Confidential content was removed]

Applications management market for service-oriented software

The possibilities for the AMSC to add value to the applications management market for service-oriented software have been determined by means of a SWOT analysis. To this end, the opportunities and threats of the applications management market for service-oriented software have been determined by analyzing it on three aspects: the software, the applications management work, and the market.

The first aspect has been analyzed by identifying the benefits and concerns regarding service-oriented software. This analysis has shown that although many benefits are attributed to service-oriented software, none is based on empirical findings.

The second aspect, the applications management work, has been analyzed in terms of the work itself as well as the value chain of which the work is part, and how these are different from traditional applications management. This analysis has shown that applications management work for service-oriented software is more complex and requires new and different skills of the applications management providers, such as competency with the technology and a more process-oriented view.

The last aspect, the market, has been analyzed in terms of size, the motives of client organizations to buy the applications management from this market, and the market as a competitive environment. This analysis has shown that the market is attractive, among other things because it is predicted to grow rapidly (about 20% yearly / from 50% of all newly developed business applications now to 80% in 2010).

These analyses have again been analyzed for opportunities and threats, which have been combined with the strengths and weaknesses of the AMSC in a SWOT analysis. [Confidential content was removed]

Applications management market for open-source software

The possibilities for the AMSC to add value to the applications management market open-source software also have been determined by means of a SWOT analysis. The opportunities and threats in this market have been determined similarly to the way they have been determined in the applications management market for service-oriented software. That is, the applications management market for open-source software has been analyzed along the three aspects open-source software, applications management, and the market.

The analysis of open-source software has shown that no evidence exists to irrefutably claim its superiority over proprietary software—although its proponents do so—and that the way open-source software is generally developed—by a community of contributors—is favorable to generic context-independent applications, instead of highly specific business applications, although these are the common subject of applications management.

The analysis of how applications management work for open-source software is different from traditional applications management work has shown that the only difference is the existence of a supporting community and the possibility to leverage that community. This would require the applications management provider to have the skills to interact with the supporting community.

The analysis of the market has shown that it seems to be unattractive, for its size is considered very limited due to the nature of the software and because four of the five forces that form the market's competitive environment are less favorable than in the traditional market (only rivalry is considered equal).

These analyses have again been analyzed for opportunities and threats, which subsequently have been combined with the strengths and weaknesses of the AMSC in a SWOT analysis. [Confidential content was removed]

6.3 Overall conclusions

[Confidential content was removed]

6.4 Recommendations

In the following some overall recommendations for the AMSC are given, followed by some recommendations for the AMSC of a more general nature. Finally, directions for further academic research are given.

Overall recommendations

[Confidential content was removed]

General recommendations for the AMSC

[Confidential content was removed]

Recommended future research

On some aspects of this research the academic literature has proven a rich source of information. However, on a number of other aspects it provided little information or even none at all. Therefore, a number of recommendations of further research are listed below.

- Empirical research to the actual benefits of service-oriented software
- Empirical research to the effects of service orientation on applications management
- Research to the size of the Dutch market of applications management for service-oriented software
- Empirical research to the competitive environment of the market of applications management for service-oriented software
- Empirical research to the superiority of open-source software
- Empirical research to the effects of open-source software on applications management
- Research to the size of the Dutch market of applications management for open-source software
- Empirical research to the competitive environment of the market of applications management for open-source software
- Research to applications management in general

References

Abrams, C. "Key Issues for SOBA Strategy and Governance," Gartner, November 17, 2006.

AlMarzouq, M., Zheng, L., Rong, G., and Grover, V. "Open Source: Concepts, Benefits, and Challenges," *Communications of the AIS* (16), November 2005, pp. 756-784.

Barney, J. "Firm Resources and Sustained Competitive Advantage," *Journal of Management* (17:1) 1991, pp. 99-120.

Bennet, K.H., and Rajlich, V.T. "Software Maintenance: a Roadmap," in: *The Future of Software Engineering*, A. Finkelstein (ed.), ACM Press, New York, NY, 2000.

Berne Convention "Berne Convention for the Protection of Literary and Artistic Works," WIPO, September 28, 1979 (available online at http://ww.wipo.int/treaties/en/ip/berne/trtdocs_wo001.html).

Bichler, M., and Lin, K.-J. "Service-Oriented Computing," *Computer* (39:3), March 2006, pp. 99-101.

Birman, K. "Can Web Services Scale Up?," *Computer* (38:10), October 2005, pp. 107-110.

Birman, K. "The Untrustworthy Web Services Revolution," *Computer* (39:2), February 2006, pp. 98-100.

Bloomberg, J. "ZapThink :: Research - The Lego® Model of SOA," December 11, 2006 (available online at http://www.zapthink.com/report.html?id=ZAPFLASH-20061212).

Bussler, C. "The Fractal Nature of Web Services," *Computer* (40:3), March 2007, pp. 93-95.

Cantara, M., Adams, C., Clark, L.M., Young, A., Potter, K., Goodness, E., Brown, R.H., and Anderson, D.S. "Dataquest Insight: Worldwide IT Professional Services Market Size and Forecast for SOA, Web Services and Web 2.0, 2005-2010," Gartner, May 7, 2007.

Cappelli, W. "The Top Four Challenges of Effective SOA Management," Gartner, December 7, 2006.

Casati, F., Shan, E., Dayal, U., and Shan, M.-C. "Business-Oriented Management of Web Services," *Communications of the ACM* (46:10), October 2003, pp. 55-60.

Centraal Bureau voor de Statistiek "CBS - Organisatiegraad van werknemers 1995-2004 – Maatwerk," 2006 (available online at http://www.cbs.nl/nl-NL/menu/themas/arbeid-sociale-zekerheid/cijfers/incidenteel/maatwerk/2006-organisatiegraad-mw.htm).

Chapin, N., Hale, J.E., Khan, K.M., Ramil, J.F., and Tan, W. "Types of Software Evolution and Software Maintenance," *Journal of Software Maintenance and Evolution: Research and Practice* (13:1), January/February 2001, pp. 3-30.

Cheng, H.K., Tang, Q.C., and Zhao, J.L. "Web Services and Service-Oriented Application Provisioning: An Analytical Study of Application Service Strategies," *IEEE Transactions on Engineering Management* (53:4), November 2006, pp. 520-533.

Crowston, K., and Howison, J. "Assessing the Health of Open Source Communities," *Computer* (39:2), May 2006, pp. 89-91.

Curbera, F., Khalaf, R., and Mukhi, N. "The Next Step in Web Services," *Communications of the ACM* (46:10), October 2003, pp. 29-34.

Currie, W.L., and Parikh, M.A. "Value creation in web services: An integrative model," *Journal of Strategic Information Systems* (15:2), June 2006, pp. 153-174.

Driver, M. "Managing Open-Source Service and Support," Gartner, December 12, 2006.

Duggan, J., Solon Jr., R.F., and Hottle, M. "How to Start Estimating Software Life Cycle Costs," Gartner, July 1, 2005.

Elfatatry, A., and Layzell, P. "Negotiating in Service-Oriented Environments," *Communications of the ACM* (47:8), August 2004, pp. 103-108.

Ferris, C., and Farrel, J. "What are Web Services?," *Communications of the ACM* (46:6), June 2003, p. 31.

Fitzgerald, B. "A Critical Look at Open Source," *Computer* (37:7), July 2004, pp. 92-94.

Fremantle, P., Weerawarana, S., and Khalaf, R. "Enterprise Services," *Communications of the ACM* (45:10), October 2002, pp. 77-82.

FSF "The Free Software Definition," 2007 (available online at http://www.fsf.org/licensing/essays/free-sw.html).

Giera, J. "Open Source's Impact on IT Services," Forrester, April 5, 2006.

Glass, R.L. "A Sociopolitical Look at Open Source," *Communications of the ACM* (46:11), November 2003, pp. 21-23.

Glass, R.L. "A Look at the Economics of Open Source," *Communications of the ACM* (47:2), February 2004, pp. 25-27.

Hagel III, J., and Brown, J.S. "Your Next IT Strategy," *Harvard Business Review* (79:9), October 2001, pp. 105-113.

Hars, A., and Qu, S. "Working for Free? Motivations for Participating in Open-Source Projects," *International Journal of Electronic Commerce* (6:3), Spring 2002, pp. 25-39.

Hoepman, J.-H., and Jacobs, B. "Increased Security Through Open Source," *Communications of the ACM* (50:1), January 2007, pp. 79-83.

Huang, C.D., and Hu, Q. "Integrating Web services with Competitive Strategies: The Balanced Scorecard Approach," *Communications of the AIS* (13), January 2004, pp. 57-80.

Huntley, C.L. "Organizational Learning in Open-Source Software Projects: An Analysis of Debugging Data," *IEEE Transactions on Engineering Management* (50:4), November 2004, pp. 485-493.

ICT~Office "ICT Marktmonitor 2007," ICT~Office 2007.

Iyer, B., Freedman, J., Gaynor, M., and Wyner, G. "Web Services: Enabling Dynamic Business Networks," *Communications of the AIS* (11), April 2003, pp. 525-554.

Jammes, F., and Smit, H. "Service-Oriented Paradigms in Industrial Automation," *IEEE Transactions on Industrial Informatics* (1:1), February 2005, pp. 62-70.

Johnson, G., Scholes, K., and Whittington, R. *Exploring Corporate Strategy*, (7 ed.) Prentice Hall, Essex, United Kingdom, 2006.

Koch, S., and Schneider, G. "Effort, Co-operation and Co-ordination in an Open Source Software Project: GNOME," *Information Systems Journal* (12:1), January 2002, pp. 27-42.

Koponen, T., and Hotti, V. "Open Source Software Maintenance Process Framework," *ACM SIGSOFT Software Engineering Notes* (30:4), July 2005, pp. 1-5.

Kreger, H. "Fullfilling the Web Services Promise," *Communications of the ACM* (46:6), June 2003, pp. 29-34.

Langdon, C.S. "The State of Web Services," *Computer* (36:7), July 2003, pp. 93-94.

Lawton, G. "Open Source Security: Opportunity or Oxymoron?," *Computer* (35:3), March 2002, pp. 18-21.

Leavitt, N. "Are Web Services Finally Ready to Deliver?," *Computer* (37:11), November 2004, pp. 14-18.

Levina, N., and Ross, J.W. "From the Vendor's Perspective: Exploring the Value Proposition in Information Technology Outsourcing," *MIS Quarterly* (27:3), September 2003, pp. 331-364.

Lewis, G., Morris, E., Simanta, S., Smith, D., and Wrage, L. "Migration of Legacy Assets to SOA Environments," 22nd IEEE International Conference on Software Maintenance, 2006.

Lewis, T. "The Open Source Acid Test," *Computer* (32:2), February 1999, pp. 128;125-127.

Lientz, P.B., and Swanson, E.B. *Software Maintenance Management* Addison Wesley, Reading, MA, 1980.

Ljungberg, J. "Open Source Movements as a Model for Organising," *European Journal of Information Systems* (9:4), December 2000, pp. 208-216.

Luftman, J., and McLean, E.R. "Key Issues for IT Executives," *MIS Quarterly Executive* (3:2), June 2004, pp. 89-104.

Mac Neela, A. "User Survey Analysis: Open-Source Software Services, Europe, 2006," Gartner, November 3, 2006.

Mac Neela, A., Clark, L.M., Igou, B., Lo, T., Cantara, M., and Morikawa, C. "Dataquest Insight: Open-Source Software Services Forecast Scenarios, Worldwide, 2005-2010," Gartner, February 26, 2007.

MacCormack, A., Rusnak, J., and Baldwin, C.Y. "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code," *Management Science* (52:7), July 2006, pp. 1015-1030.

Margaria, T., and Steffen, B. "Service Engineering: Linking Business and IT," *Computer* (39:10), October 2006, pp. 45-55.

Mintzberg, H., Alhstrand, B.W., and Lampel, J. *Strategy Safari* Free Press, New York, NY, 1998.

Mintzberg, H., and Waters, J.A. "Of Strategies, Deliberate and Emergent," *Strategic Management Journal* (6:3) 1985, pp. 257-272.

Mockus, A., Fielding, R.T., and Herbsleb, J.D. "Two Case Studies of Open Source Software Development: Apache and Mozilla," *ACM Transactions on Software Engineering and Methodology* (11:3), July 2002, pp. 309-346.

Moitra, D., and Ganesh, J. "Web Services and Flexible Business Processes: Towards the Adaptive Enterprise," *Information & Management* (42:7), October 2005, pp. 921-933.

Moore, S., and Martorelli, W. "The Apps Outsourcing Showdown Is Accelerating," Forrester Research, January 16, 2007.

Murray, M. "Using Extensible Markup Language (XML) as a Medium for Data Exchange," *Communications of the AIS* (9), September 2002, pp. 105-119.

Mylonopoulos, N.A., and Theoharakis, V. "Global perceptions of IS journals - Where is the best IS research published?," *Communications of the ACM* (44:9), September 2001, pp. 29-33.

Natis, Y.V., Pezzini, M., Schulte, R.W., and Iijima, K. "Predicts 2007: SOA Advances," Gartner, November 17, 2006.

Nezhad, H.R.M., Benatallah, B., Casati, F., and Toumani, F. "Web Services Interoperability Specifications," *Computer* (39:5), May 2006, pp. 24-32.

O'Reilly, T. "Lessons from Open-Source Software Development," *Communications of the ACM* (42:4), April 1999, pp. 32-37.

OASIS "Web Services Reliable Messaging (WS-ReliableMessaging) Version 1.1," June 14, 2007 (available online at http://docs.oasis-open.org/ws-rx/wsrm/200702/wsrm-1.1-spec-os-01.pdf).

OSI "The Open Source Definition," 2006 (available online at http://www.opensource.org/docs/osd).

Overby, S. "Bringing IT Back Home," *CIO* (16:10), March 1, 2003, pp. 54-56.

Papazoglou, M.P., and Georgakopoulos, D. "Service-Oriented Computing," *Communications of the ACM* (46:10), October 2003, pp. 25-28.

Parker, A. "Two-Speed Europe: Why 1 Million Jobs Will Move Offshore," Forrester Research, August 18, 2004.

Paulson, J.W., Succi, G., and Eberlein, A. "An Empirical Study of Open-Source and Closed-Source Software Products," *IEEE Transactions on Software Engineering* (30:4), April 2004, pp. 246-256.

Paulson, L.D. "Localizing Open Source Software," *Computer* (37:4), April 2004a, p. 17.

Paulson, L.D. "Open Source Databases Move into the Marketplace," *Computer* (37:7), July 2004b, pp. 13-15.

Payne, C. "On the Security of Open Source Software," *Information Systems Journal* (12:1), January 2002, pp. 61-78.

Peltz, C. "Web Services Orchestration and Choreography," *Computer* (36:10), October 2003, pp. 46-52.

Perrow, C. "A Framework for the Comparative Analysis of Organizations," *American Sociological Review* (32:2), April 1967, pp. 194-208.

Pierre Audoin Consultants "SITSI Market Figures – Netherlands," 2006.

Porter, M.E. *Competitive Strategy: Techniques for Analyzing Industries and Competitors* Free Press, New York, NY, 1980.

Porter, M.E. *Competitive Advantage: Creating and Sustaining Superior Performance* Free Press, New York, NY, 1985.

Porter, M.E. "What Is Strategy?," *Harvard Business Review* (74:6), November-December 1996, pp. 61-78.

Raghunathan, S., Prasad, A., Mishra, B.K., and Chang, H. "Open Source Versus Closed Source: Software Quality in Monopoly and Competitive Markets," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* (35:6), November 2005, pp. 903-918.

Raymond, E.S. "The Cathedral and the Bazaar," 2000 (available online at http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/).

Riehle, D. "The Economic Motivation of Open Source Software: Stakeholder Perspectives," *Computer* (40:4), April 2007, pp. 25-32.

Samoladas, I., Stamelos, I., Angelis, L., and Oikonomou, A. "Open Source Software Development Should Strive for Even Greater Code Maintainability," *Communications of the ACM* (47:10), October 2004, pp. 83-87.

Shirky, C. "Web Services and Context Horizons," *Computer* (35:9), September 2002, pp. 98-100.

Sholler, D., Abrams, C., Thompson, J., Gootzit, D., Phifer, G., Valdes, R., Wagner, R., Haight, C., Govekar, M., Gall, N., Knox, R.E., Friedman, T., Feinberg, D., Hill, J.E., Sinur, J., Plummer, D.C., Cearley, D.W., Witty, R.J., and Allan, A. "Hype Cycle for Web Services and Related Standards and Specifications, 2006," Gartner, June 11, 2006.

Singendonk, F., and De Graaf, P. "Jaarplan AMSC 2007," Capgemini Outsourcing 2007.

Smith, H.A., and McKeen, J.D. "Developments in Practice XIV: IT Sourcing – How Far Can You Go?," *Communications of the AIS* (13), June 2004, pp. 508-520.

Stallman, R.M. "What Is Copyleft?," 1996 (available online at http://www.gnu.org/copyleft/copyleft.html).

Stallman, R.M. "Why "Free Software" is better than "Open Source"," 1998 (available online at http://www.gnu.org/philosophy/free-software-for-freedom.html).

Stamelos, I., Angelis, L., Oikonomou, A., and Bleris, G.L. "Code Quality Analysis in Open Source Software Development," *Information Systems Journal* (12:1), January 2002, pp. 43-60.

Stewart, K.J., Ammeter, A.P., and Maruping, L.M. "Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects," *Information Systems Research* (17:2), June 2006, pp. 126-144.

Stuckenholz, A. "Component Evolution and Versioning State of the Art," *ACM SIGSOFT Software Engineering Notes* (30:1), January 2005, pp. 1-13.

Takahashi, S. "European Application Outsourcing Spending Forecast: 2006 To 2011," Forrester Research, June 20, 2006.

Tilley, S., Gerdes, J., Hamiltion, T., Huang, S., Müller, H., Smith, D., and Wong, K. "On the Business Value and Technical Challenges of Adopting Web Services," *Journal of Software Maintenance and Evolution: Research and Practice* (16:1-2), January-April 2004, pp. 31-50.

Tramacere, G. "Market Trends: IT Outsourcing, Europe, 2007," Gartner, April 16, 2007.

Van Heur, R. "SOA nog te weinig getest," Computable, June 19, 2007 (available online at http://www.computable.nl/nieuws.jsp?id=2017402).

Van Hillegersberg, J. *Metamodelling-Based Integration of Object-Oriented Systems Development* Thesis Publishers, Amsterdam, The Netherlands, 1997.

Van Hillegersberg, J. "Information Systems for a Networked World," May 18, 2006.

Vaughan-Nichols, S.J. "Web Services: Beyond the Hype," *Computer* (35:2), February 2002, pp. 18-21.

W3C "Web Services Architecture," February 11, 2004 (available online at http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/).

Wall, D.A.E. "Using Open Source for a Profitable Startup," *Computer* (34:12), December 2001, pp. 158-160.

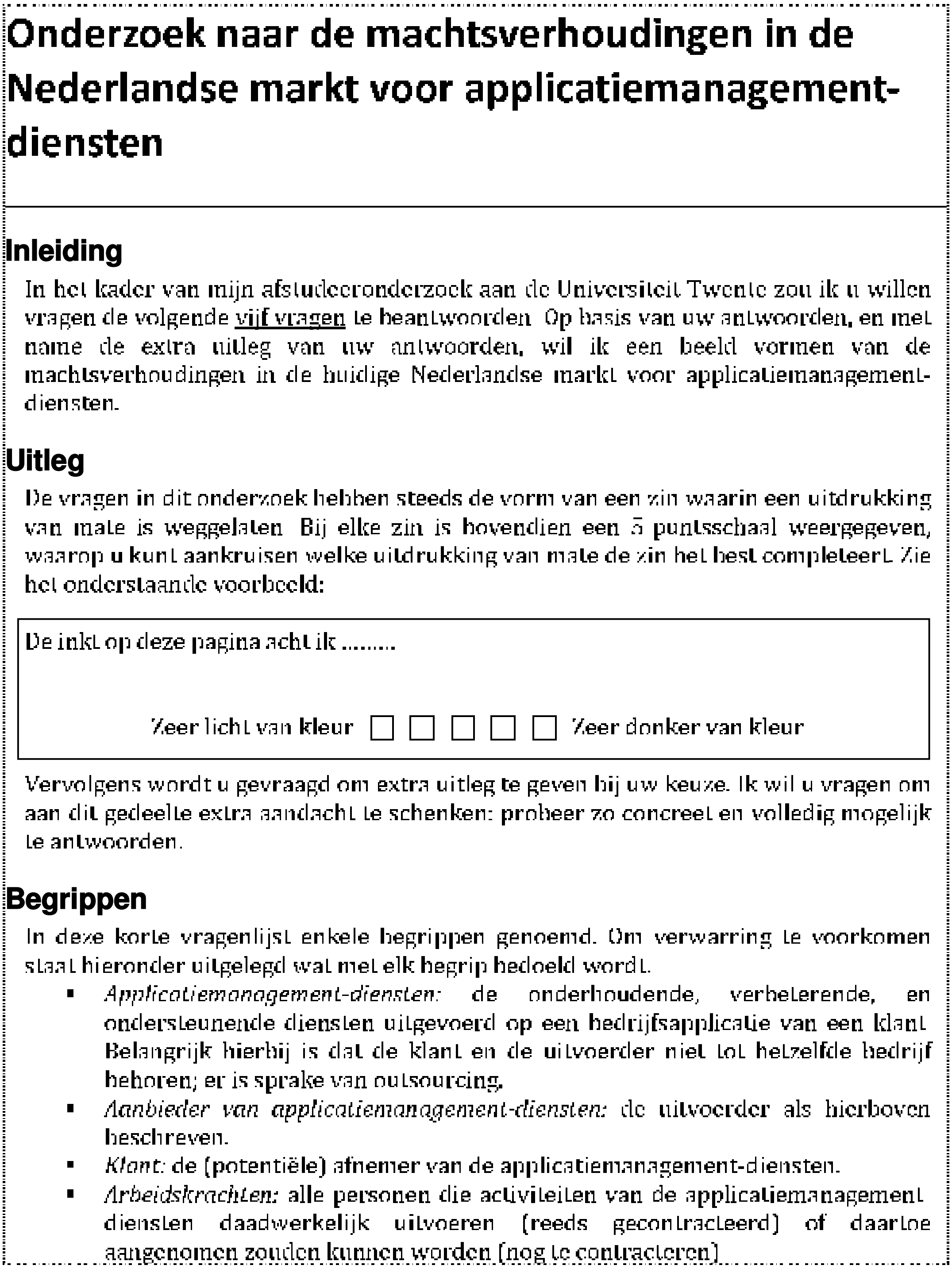
Wu, M.-W., and Lin, Y.-D. "Open Source Software Development: An Overview," *Computer* (34:6), June 2001, pp. 33-38.

Yang, J. "Web Service Componentization," *Communications of the ACM* (46:10), October 2003, pp. 35-40.

Zhao, J.L., and Cheng, H.K. "Web Services and Process Management: A Union of Convenience or a New Area of Research?," *Decision Support Systems* (40:1), July 2005, pp. 1-8.

Zhao, J.L., Tanniru, M., and Zhang, L.-J. "Services Computing as the Foundation of Enterprise Agility: Overview of Recent Advances and Introduction to the Special Issue," *Information Systems Frontier* (9:1) 2007, pp. 1-8.

Appendix A: Questionnaire current AM market



Onderzoek naar de machtsverhoudingen in de Nederlandse markt voor applicatiemanagement-diensten

Inleiding

In het kader van mijn afstudeeronderzoek aan de Universiteit Twente zou ik u willen vragen de volgende vijf vragen te beantwoorden. Op basis van uw antwoorden, en met name de extra uitleg van uw antwoorden, wil ik een beeld vormen van de machtsverhoudingen in de huidige Nederlandse markt voor applicatiemanagement-diensten.

Uitleg

De vragen in dit onderzoek hebben steeds de vorm van een zin waarin een uitdrukking van mate is weggelaten. Bij elke zin is bovendien een 5 puntsschaal weergegeven, waarop u kunt aankruisen welke uitdrukking van mate de zin het best completeert. Zie het onderstaande voorbeeld:

De inkt op deze pagina acht ik ....

Zeer licht van kleur □ □ □ □ □ Zeer donker van kleur

Vervolgens wordt u gevraagd om extra uitleg te geven bij uw keuze. Ik wil u vragen om aan dit gedeelte extra aandacht te schenken: probeer zo concreet en volledig mogelijk te antwoorden.

Begrippen

In deze korte vragenlijst enkele begrippen genoemd. Om verwarring te voorkomen staat hieronder uitgelegd wat met elk begrip bedoeld wordt.

- *Applicatiemanagement-diensten:* de onderhoudende, verbeterende, en ondersteunende diensten uitgevoerd op een bedrijfsapplicatie van een klant. Belangrijk hierbij is dat de klant en de uitvoerder niet tot hetzelfde bedrijf behoren; er is sprake van outsourcing.
- *Aanbieder van applicatiemanagement-diensten:* de uitvoerder als hierboven beschreven.
- *Klant:* de (potentiële) afnemer van de applicatiemanagement-diensten.
- *Arbeidskrachten:* alle personen die activiteiten van de applicatiemanagement-diensten daadwerkelijk uitvoeren (reeds gecontracteerd) of daartoe aangenomen zouden kunnen worden (nog te contracteren)

Vraag 1 van 5

Inleiding

In de markt voor personenauto's zijn al jaren gevestigde aanbieders actief Sinds kort zijn echter nieuwe aanbieders tot die markt toegetreden; denk bijvoorbeeld aan het Chinese merk Landwind.

Kruis het antwoord aan dat, vo gens u, de zin het best afmaakt.

De kans dat eventuele nieuwe aanbieders van applicatiemanagement-diensten tot de huidige Nederlandse markt zullen toetreden, acht ik op dit moment ....

Zeer klein □ □ □ □ □ Zeer groot.

Verk aar waarom eventue e aanbieders de markt (moge ijk) we of (moge ijk) niet zu en betreden:

Vraag 2 van 5

Inleiding

U als persoon heeft een behoefte mobiliteit. Om aan deze behoefte invulling te geven heeft u verschillende alternatieven tot uw beschikking. Zo kunt u hiertoe bijvoorbeeld een auto aanschaffen, of een fiets, of gebruik maken van de diensten van openbaarvervoerbedrijven, taxicentrales, of misschien wel carpoolers. Er zijn echter ook behoeftes waarvoor de mogelijkheden tot invulling zeer beperkt zijn. Denk aan uw behoefte om teksten te verwerken: u kunt dit met de hand doen, of een computer gebruiken, maar veel meer alternatieven zijn er niet

Kruis het antwoord aan dat, volgens u, de zin het best afmaakt.

Klanten hebben op dit moment, naast het afnemen van applicatiemanagement diensten van een externe aanbieder, ... alternatieven om aan hun behoefte invulling te geven.

Zeer weinig □ □ □ □ □ Zeer veel

Indien u alternatieven ziet, noem die dan hier:

Vraag 3 van 5

Inleiding

In het traject voorafgaand aan de tekening van een contract proberen zowel de klant als de aanbieder een voor zichzelf zo goed mogelijke deal te sluiten. De klant wil zo veel en zo goed mogelijk service tegen een zo laag mogelijke prijs. De aanbieder daarentegen wil juist het omgekeerde: zo min mogelijk leveren tegen een zo hoog mogelijke prijs. Onderhandelingen tussen beide partijen bepalen de uiteindelijke service en prijsniveaus

Kruis het antwoord aan dat, volgens u, de zin het best afmaakt.

De onderhandelingspositie van klanten ten opzichte van aanbieders van applicatiemanagement diensten acht ik op dit moment ……… .

Zeer zwak □ □ □ □ □ Zeer sterk

Noem de factoren die aan deze (zeer) zwakke / (zeer) sterke onderhande ingspositie ten gronds ag liggen:

Vraag 4 van 5

Inleiding

Ook arbeidskrachten hebben dergelijke onderhandelingen met de aanbieder van applicatiemanagement-diensten. Een duidelijk voorbeeld hiervan vormen de salarisonderhandelingen bij het sluiten van een arbeidsovereenkomst.

Kruis het antwoord aan dat, vo gens u, de zin het best afmaakt.

De onderhandelingspositie van arbeidskrachten (zowel nieuw aan te nemen als reeds gecontracteerd) ten opzichte van aanbieders van applicatiemanagement-diensten acht ik op dit moment ....

Zeer zwak □ □ □ □ □ Zeer sterk

Noem de factoren die aan deze (zeer) zwakke / (zeer) sterke onderhandelingspositie ten grondslag liggen:

Vraag 5 van 5

Inleiding

In de huidige Nederlandse markt voor applicatiemanagement diensten zijn meerdere aanbieders actief. Hierdoor zullen de verschillende aanbieders met elkaar moeten concurreren om opdrachten binnen te halen.

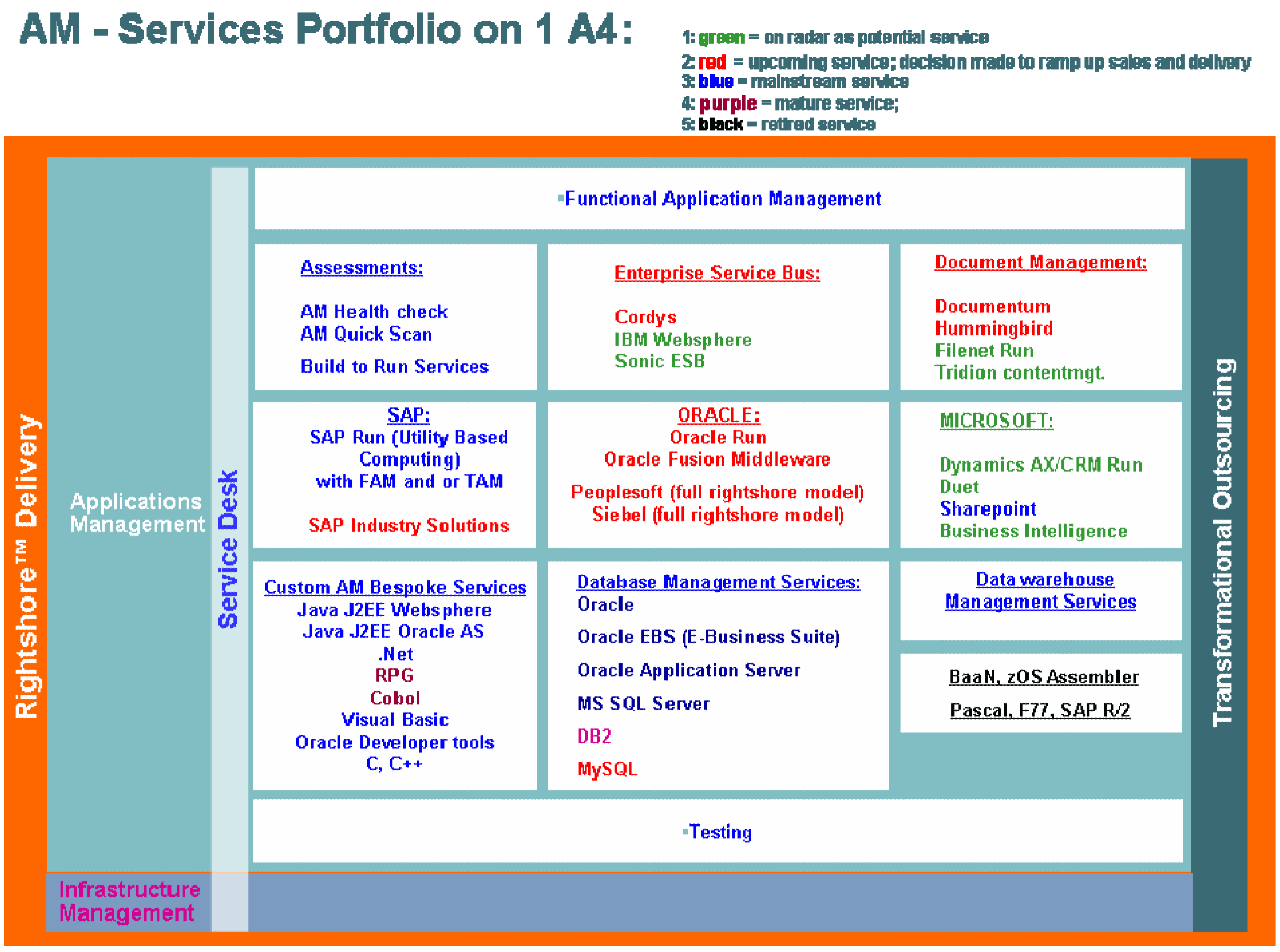
Kruis het antwoord aan dat, volgens u, de zin het best afmaakt.

De concurrentie tussen bestaande aanbieders van applicatiemanagement diensten in de huidige Nederlandse markt acht ik op dit moment ....

Zeer gering □ □ □ □ □ Zeer hevig

Noem de factoren die aan deze mate van concurrentie ten grondslag liggen:

Appendix B: Service Portfolio of the AMSC



AM - Services Portfolio on 1 A4:

1: green = on radar as potential service
2: red = upcoming service; decision made to ramp up sales and delivery
3: blue = mainstream service
4: purple = mature service;
5: black = retired service

Rightshore™ Delivery

Applications Management

Service Desk

Functional Application Management

Assessments:
AM Health check
AM Quick Scan
Build to Run Services

Enterprise Service Bus:
Cordys
IBM Websphere
Sonic ESB

Document Management:
Documentum
Hummingbird
Filenet Run
Tridion contentmgt.

SAP:
SAP Run (Utility Based Computing) with FAM and or TAM
SAP Industry Solutions

ORACLE:
Oracle Run
Oracle Fusion Middleware
Peoplesoft (full rightshore model)
Siebel (full rightshore model)

MICROSOFT:
Dynamics AX/CRM Run
Duet
Sharepoint
Business Intelligence

Custom AM Bespoke Services
Java J2EE Websphere
Java J2EE Oracle AS
.Net
RPG
Cobol
Visual Basic
Oracle Developer tools
C, C++

Database Management Services:
Oracle
Oracle EBS (E-Business Suite)
Oracle Application Server
MS SQL Server
DB2
MySQL

Data warehouse Management Services

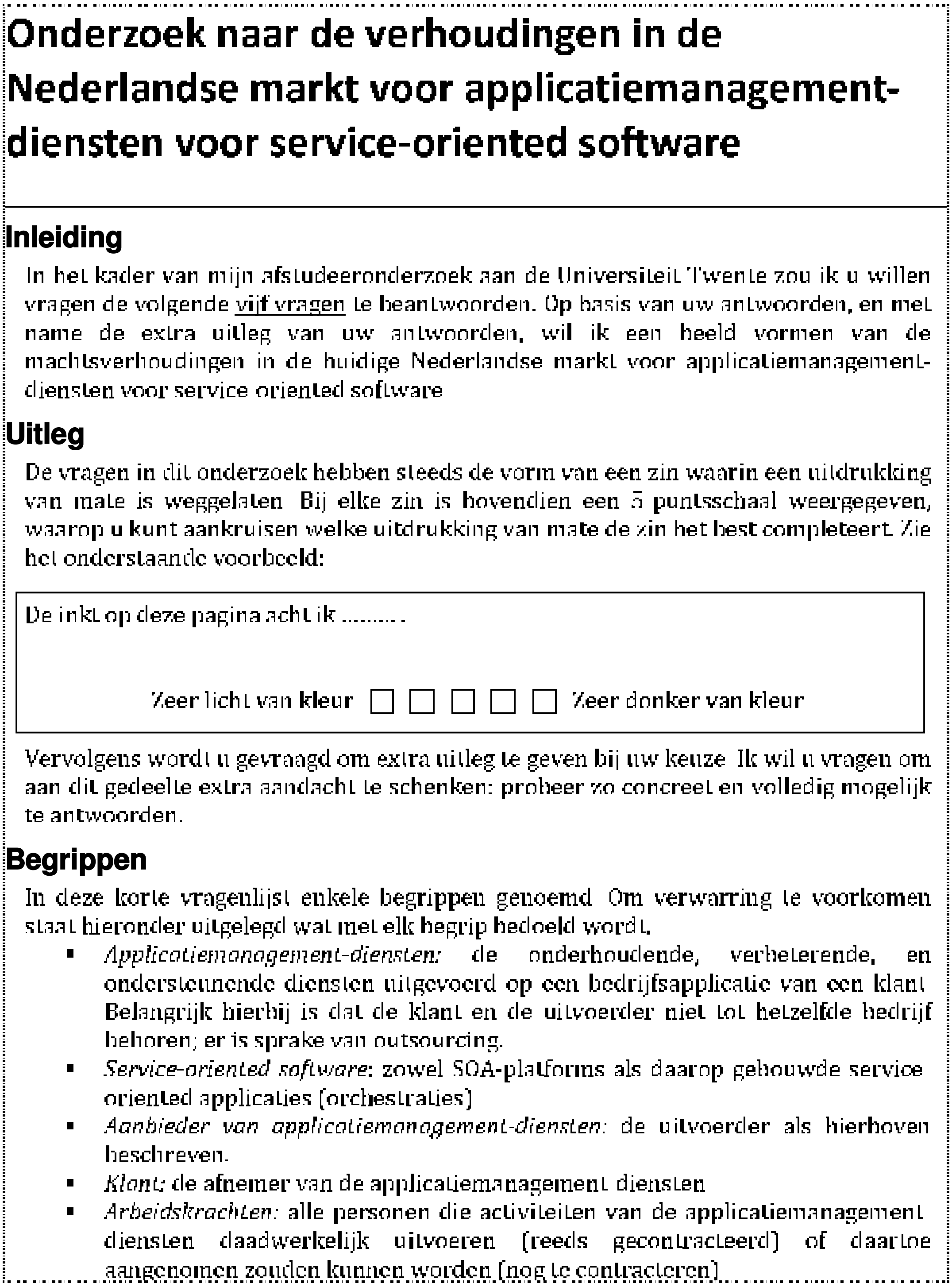
BaaN, zOS Assembler
Pascal, F77, SAP R/2

Testing

Transformational Outsourcing

Infrastructure Management

Appendix C: Questionnaire AM market for service-oriented software



Onderzoek naar de verhoudingen in de Nederlandse markt voor applicatiemanagement-diensten voor service-oriented software

Inleiding

In het kader van mijn afstudeeronderzoek aan de Universiteit Twente zou ik u willen vragen de volgende vijf vragen te beantwoorden. Op basis van uw antwoorden, en met name de extra uitleg van uw antwoorden, wil ik een beeld vormen van de machtsverhoudingen in de huidige Nederlandse markt voor applicatiemanagement-diensten voor service oriented software

Uitleg

De vragen in dit onderzoek hebben steeds de vorm van een zin waarin een uitdrukking van mate is weggelaten. Bij elke zin is bovendien een 5 puntsschaal weergegeven, waarop u kunt aankruisen welke uitdrukking van mate de zin het best completeert. Zie het onderstaande voorbeeld:

De inkt op deze pagina acht ik ……… .

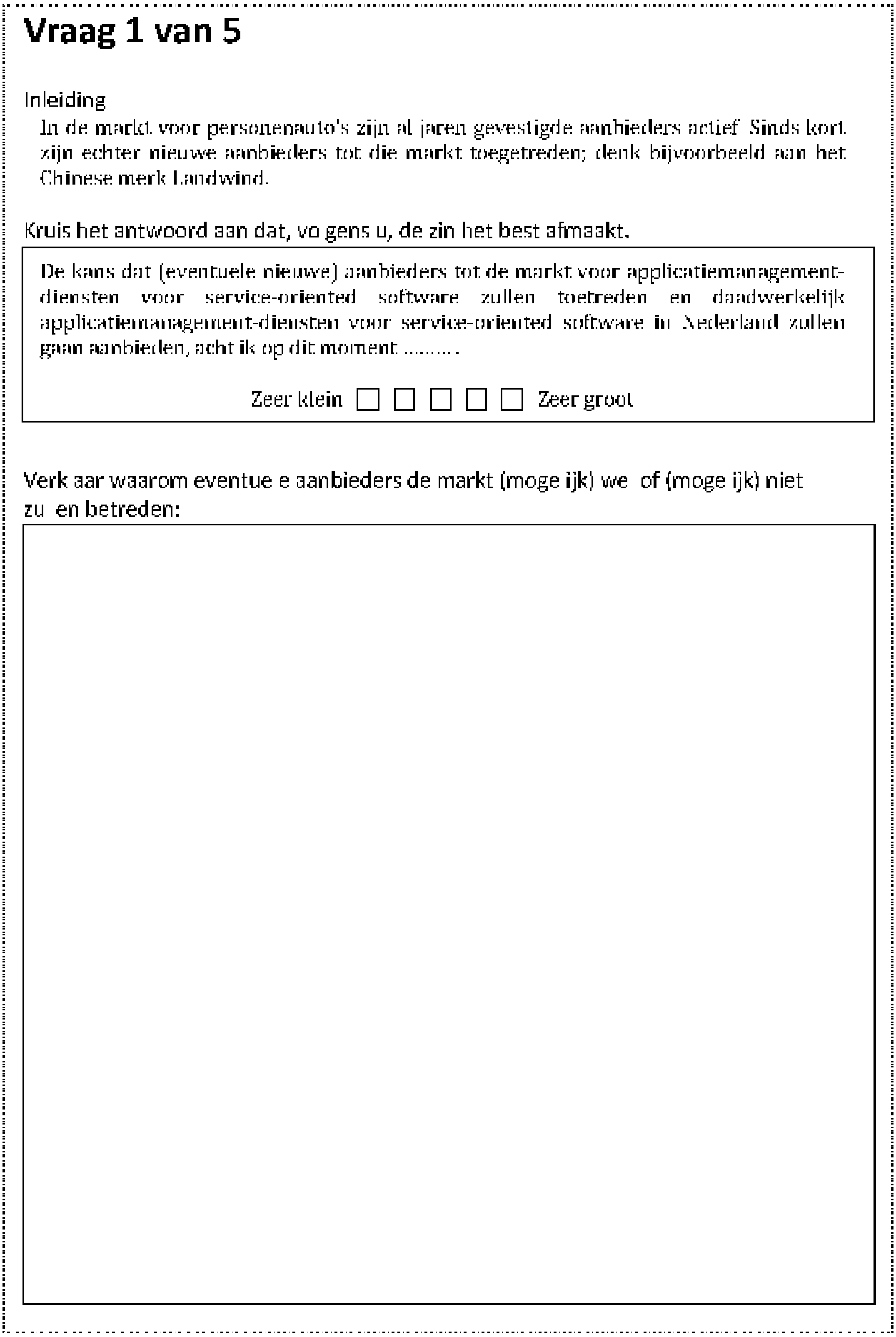
Zeer licht van kleur □ □ □ □ □ Zeer donker van kleur

Vervolgens wordt u gevraagd om extra uitleg te geven bij uw keuze. Ik wil u vragen om aan dit gedeelte extra aandacht te schenken: probeer zo concreet en volledig mogelijk te antwoorden.

Begrippen

In deze korte vragenlijst enkele begrippen genoemd. Om verwarring te voorkomen staat hieronder uitgelegd wat met elk begrip bedoeld wordt.

- *Applicatiemanagement-diensten:* de onderhoudende, verbeterende, en ondersteunende diensten uitgevoerd op een bedrijfsapplicatie van een klant. Belangrijk hierbij is dat de klant en de uitvoerder niet tot hetzelfde bedrijf behoren; er is sprake van outsourcing.
- *Service-oriented software*: zowel SOA-platforms als daarop gebouwde service oriented applicaties (orchestraties)
- *Aanbieder van applicatiemanagement-diensten:* de uitvoerder als hierboven beschreven.
- *Klant:* de afnemer van de applicatiemanagement diensten
- *Arbeidskrachten:* alle personen die activiteiten van de applicatiemanagement diensten daadwerkelijk uitvoeren (reeds gecontracteerd) of daartoe aangenomen zouden kunnen worden (nog te contracteren)



Vraag 1 van 5

Inleiding

In de markt voor personenauto's zijn al jaren gevestigde aanbieders actief Sinds kort zijn echter nieuwe aanbieders tot die markt toegetreden; denk bijvoorbeeld aan het Chinese merk Landwind.

Kruis het antwoord aan dat, vo gens u, de zin het best afmaakt.

De kans dat (eventuele nieuwe) aanbieders tot de markt voor applicatiemanagement-diensten voor service-oriented software zullen toetreden en daadwerkelijk applicatiemanagement-diensten voor service-oriented software in Nederland zullen gaan aanbieden, acht ik op dit moment ……… .

Zeer klein □ □ □ □ □ Zeer groot

Verk aar waarom eventue e aanbieders de markt (moge ijk) we of (moge ijk) niet zu en betreden:

Vraag 2 van 5

Inleiding

U als persoon heeft een behoefte mobiliteit. Om aan deze behoefte invulling te geven heeft u verschillende alternatieven tot uw beschikking. Zo kunt u hiertoe bijvoorbeeld een auto aanschaffen, of een fiets, of gebruik maken van de diensten van openbaarvervoerbedrijven, taxicentrales, of misschien wel carpoolers. Er zijn echter ook behoeftes waarvoor de mogelijkheden tot invulling zeer beperkt zijn. Denk aan uw behoefte om teksten te verwerken: u kunt dit met de hand doen, of een computer gebruiken, maar veel meer alternatieven zijn er niet

Kruis het antwoord aan dat, volgens u, de zin het best afmaakt.

Klanten die service-oriented software gebruiken, hebben op dit moment, naast het afnemen van applicatiemanagement-diensten voor service oriented software van een externe aanbieder, ... alternatieven om aan hun behoefte invulling te geven.

Zeer weinig □ □ □ □ □ Zeer veel

Indien u alternatieven ziet, noem die dan hier:

Vraag 3 van 5

Inleiding

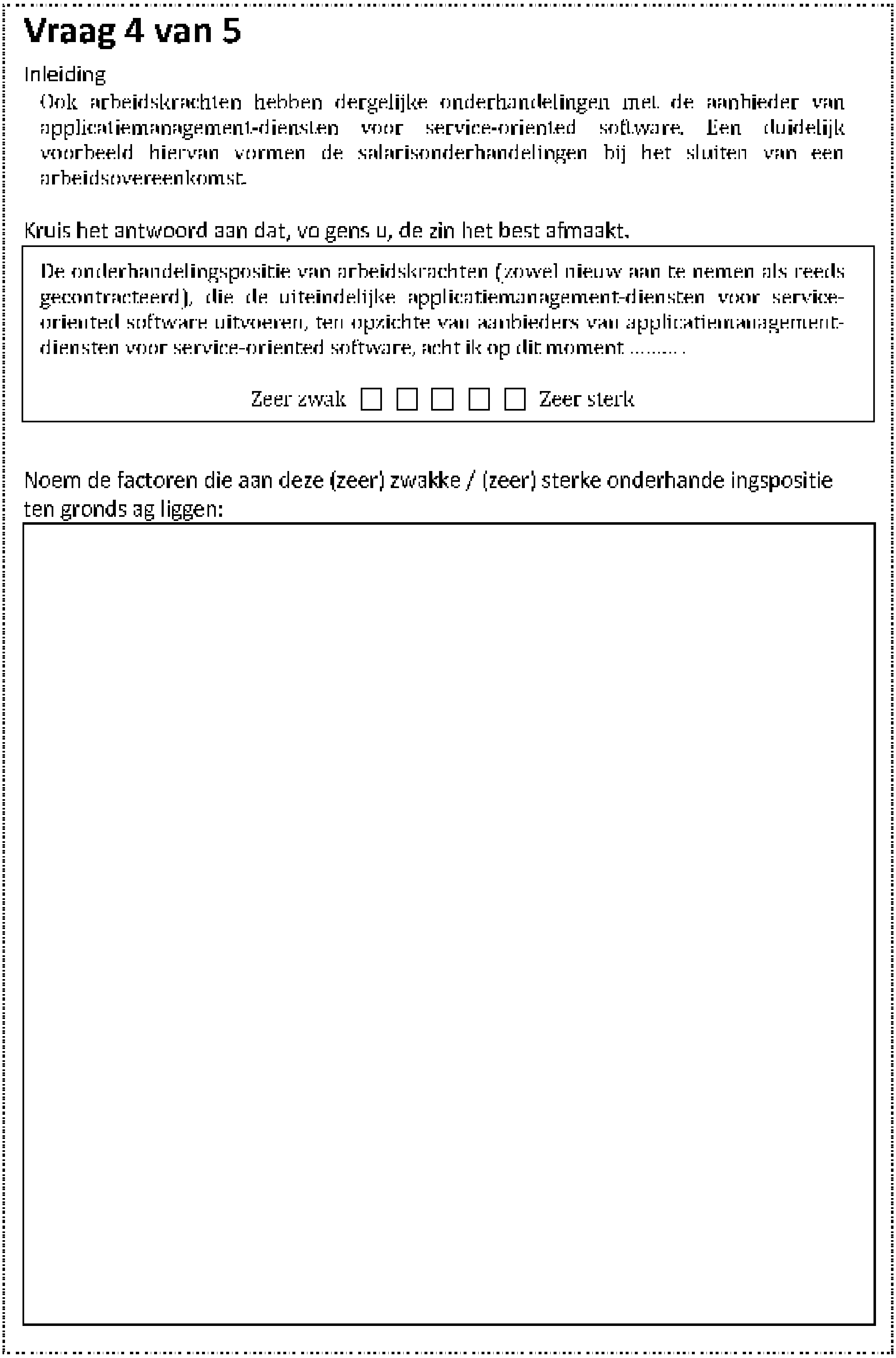
In het traject voorafgaand aan de tekening van een contract voor het beheer van service-oriented software proberen zowel de klant als de aanbieder een voor zichzelf zo goed mogelijke deal te sluiten. De klant wil zo veel en zo goed mogelijk dienstverlening tegen een zo laag mogelijke prijs. De aanbieder daarentegen wil juist het omgekeerde: zo min mogelijk leveren tegen een zo hoog mogelijke prijs. Onderhandelingen tussen beide partijen bepalen de uiteindelijke service en prijsniveaus.

Kruis het antwoord aan dat, vo gens u, de zin het best afmaakt.

De onderhandelingspositie van klanten (met service oriented software) ten opzichte van aanbieders van applicatiemanagement diensten voor service oriented software acht ik op dit moment ....

Zeer zwak □ □ □ □ □ Zeer sterk

Noem de factoren die aan deze (zeer) zwakke / (zeer) sterke onderhandelingspositie ten grondslag liggen:



Vraag 4 van 5

Inleiding

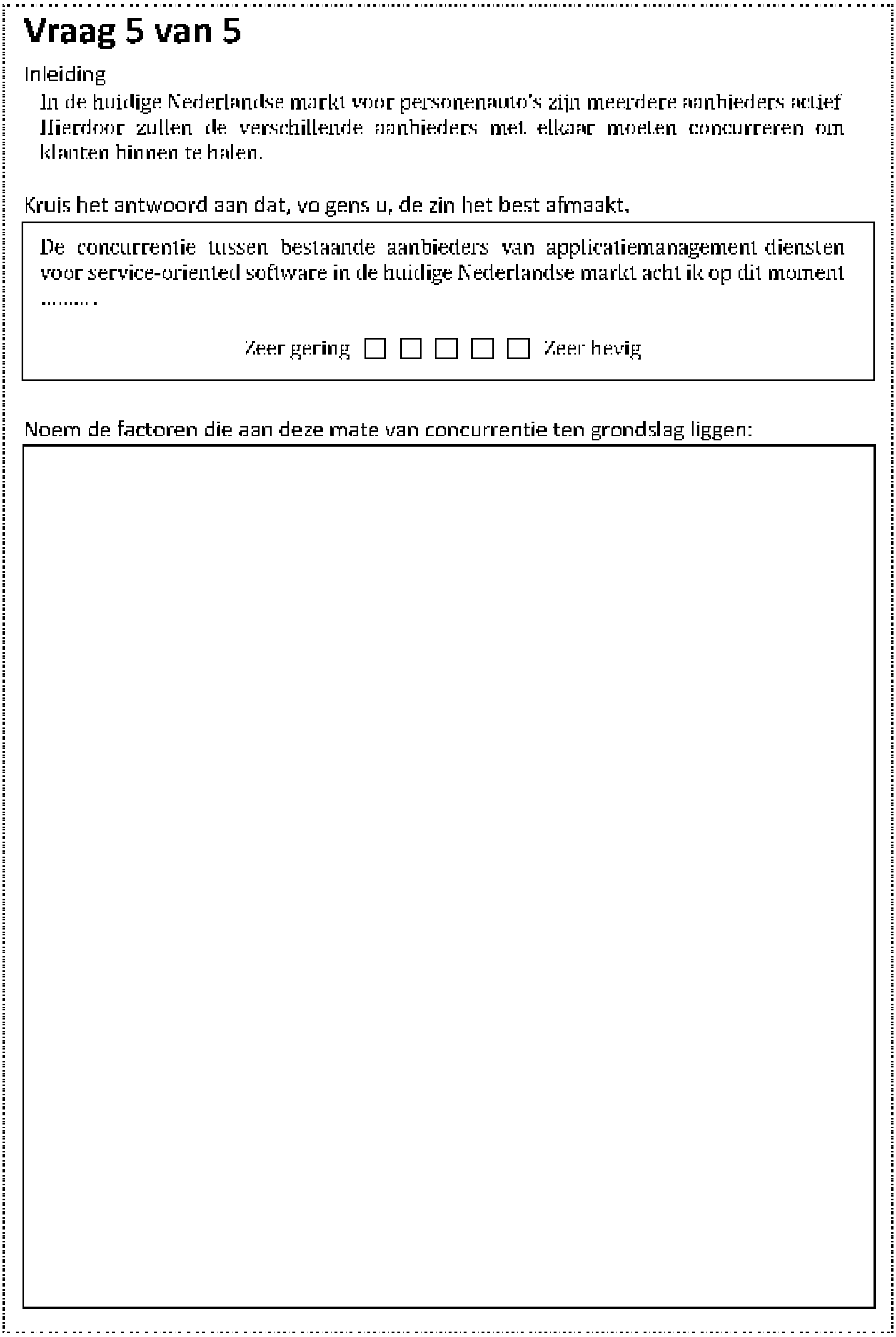
Ook arbeidskrachten hebben dergelijke onderhandelingen met de aanbieder van applicatiemanagement-diensten voor service-oriented software. Een duidelijk voorbeeld hiervan vormen de salarisonderhandelingen bij het sluiten van een arbeidsovereenkomst.

Kruis het antwoord aan dat, vo gens u, de zin het best afmaakt.

De onderhandelingspositie van arbeidskrachten (zowel nieuw aan te nemen als reeds gecontracteerd), die de uiteindelijke applicatiemanagement-diensten voor service-oriented software uitvoeren, ten opzichte van aanbieders van applicatiemanagement-diensten voor service-oriented software, acht ik op dit moment ……… .

Zeer zwak □ □ □ □ □ Zeer sterk

Noem de factoren die aan deze (zeer) zwakke / (zeer) sterke onderhande ingspositie ten gronds ag liggen:



Vraag 5 van 5

Inleiding

In de huidige Nederlandse markt voor personenauto's zijn meerdere aanbieders actief Hierdoor zullen de verschillende aanbieders met elkaar moeten concurreren om klanten binnen te halen.

Kruis het antwoord aan dat, vo gens u, de zin het best afmaakt,

De concurrentie tussen bestaande aanbieders van applicatiemanagement diensten voor service-oriented software in de huidige Nederlandse markt acht ik op dit moment ……… .

Zeer gering □ □ □ □ □ Zeer hevig

Noem de factoren die aan deze mate van concurrentie ten grondslag liggen:

About Capgemini and the Collaborative Business Experience

Capgemini, one of the world's foremost providers of Consulting, Technology and Outsourcing services, has a unique way of working with its clients, called the Collaborative Business Experience.

Backed by over three decades of industry and service experience, the Collaborative Business Experience is designed to help our clients achieve better, faster, more sustainable results through seamless access to our network of world-leading technology partners and collaboration-focused methods and tools. Through commitment to mutual success and the achievement of tangible value, we help businesses implement growth strategies, leverage technology, and thrive through the power of collaboration.

Capgemini employs approximately 68,000 people worldwide and reported 2006 global revenues of 7,7 billion euros.

More information about our services, offices and research is available at **www.capgemini.com**