Pasi Tyrväinen
Slinger Jansen
Michael A. Cusumano (Eds.)

LNBIP 51

Software Business

First International Conference, ICSOB 2010
Jyväskylä, Finland, June 2010
Proceedings

Springer

Lecture Notes
in Business Information Processing 51

Pasi Tyrväinen Slinger Jansen
Michael A. Cusumano (Eds.)

Software Business

First International Conference, ICSOB 2010
Jyväskylä, Finland, June 21-23, 2010
Proceedings

Springer

Volume Editors

Pasi Tyrväinen
University of Jyväskylä
P.O. Box 35, 40014 University of Jyväskylä, Finland
E-mail: pasi.tyrvainen@jyu.fi

Slinger Jansen
IUtrecht University
PO Box 80.089, 3508TB Utrecht, The Netherlands
E-mail: slinger@slingerjansen.nl

Michael A. Cusumano
MIT Sloan School of Management
50 Memorial Drive, Cambridge, MA 02142-1347, USA
E-mail: cusumano@mit.edu

Library of Congress Control Number: 2010928049

ACM Computing Classification (1998): K.6, D.2

ISSN 1865-1348
ISBN-10 3-642-13632-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-13632-0 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180 5 4 3 2 1 0

Preface

The advancement of the software industry has had a substantial impact not only on productivity and on GDP growth globally, but also on our daily work and life. Software business refers to commercial activity of the software industry, aimed at generating income from delivery of software products and software services. Although software business shares common features with other international knowledge-intensive businesses, it carries many inherent features making it an intriguing and challenging domain for research. Until now, however, software business has received little attention from the academic community.

The First International Conference on Software Business (ICSOB 2010) was organized in Jyväskylä during June 21–23, 2010. This inaugural conference brought together a strong Program Committee of 52 members with research disciplines from various fields of business management and technology management as well as international flavor with members coming from 17 countries from South and North America to Europe, India and Australia.

We received 35 research paper submissions. The papers went through a double-blind review process producing at least three reviews for each accepted paper. The Program Committee accepted 13 submissions to be presented as full papers in the conference, equaling 37% of the submissions. In addition, ten papers were accepted as short papers. The accepted papers represent the wide variety of research activity on software business. For the purposes of the conference program, the papers were organized under eight themes: business models, business management, ecosystems, education and research, internationalization, open source software and social media, product management, and software as a service.

In addition to the paper sessions, the conference program included three keynote presentations and a Business Innovation Track containing best-practice presentations from the software industry. The conference program also included two workshops, three tutorials and an adjunct meeting of the Cloud Software Consortia.

As Program Committee Chairs, we would like to thank the members of the Program Committee for their very valuable effort in evaluating the received papers to ensure a high quality of the conference. The experience and efforts of the Steering Committee and all the Chairs were valuable in building up this inaugural conference. Finally, it is our honor to mention the entities that supported the conference: University of Jyväskylä, the Cloud Software Program organized by Tivit Oy and funded by the Finnish Funding Agency for Technology and Innovation, SAP AG, as well as all the collaborating institutions, including Utrecht University, the Software Business Laboratory at Aalto University, VTT Research Center, University of Helsinki, and Springer Science+Business Media.

June 2010

Pasi Tyrväinen
Slinger Jansen

Invited Keynotes

"Staying Power—Six Enduring Principles for Managing Strategy and Innovation in an Uncertain World"
Michael A. Cusumano, MIT Sloan School of Management and MIT School of Engineering

"Software Industry Transformation from Products into Services"
Pirkka Palomäki, Chief Technology Officer at F-Secure Corporation

"Business Models in the Software Industry—Past and Future"
Karl-Michael Popp, Director Mergers and Acquisitions at SAP AG

Workshops

1. Workshop on Global Outsourcing of Software Development
2. Workshop on Competencies for the Globalization of Information Systems in Knowledge-Intensive Settings

Industrial Session

Software Business Innovation Track

Tutorials

1. SaaS Business – Theory and Practice
2. Applying Statistical Research Methods in Software Business
3. Creating Productive Global Virtual Teams – Developing Effective Collaboration across Cultures and Time Zones

Collaborating Organizations

Organization

General Chair

Michael A. Cusumano	MIT Sloan School of Management, USA

Program Chairs

Pasi Tyrväinen	University of Jyväskylä, Finland
Slinger Jansen	Utrecht University, The Netherlands

Review and Publication Chair

Oleksiy Mazhelis	University of Jyväskylä, Finland

Workshops and Tutorials Chair

Nazmun Nahar	University of Jyväskylä, Finland

Industry Track (SBIT) Chair

Jyrki Kontio	R & D-Ware

Steering Committee

Kalle Lyytinen	Case Western Reserve University, USA
Sjaak Brinkkemper	Utrecht University, The Netherlands
Pekka Abrahamsson	University of Helsinki, Finland

Financial Chair

Seppo Puurinen	University of Jyväskylä, Finland

Publicity Chair

Nilay Oza	VTT Research Center, Finland

Local Organizing Chairs

Lauri Frank	University of Jyväskylä, Finland
Eetu Luoma	University of Jyväskylä, Finland

Program Committee

Petri Ahokangas, University of Oulu, Finland
Aybüke Aurum, University of New South Wales, Australia
Jussi Autere, Aalto University, Finland
Jan Bosch, Intuit, USA
Peter Buxmann, Technische Universität Darmstadt, Germany
Erran Carmel, American University, USA
Ernesto Damiani, University of Milan, Italy
Christof Ebert, Vector Consulting, Germany
Anthony Finkelstein, University College London, UK
Leah Goldin, Shenkar College of Engineering and Design, Israel
Tony Gorschek, Blekinge Institute of Technology, Sweden
Jukka Heikkilä, University of Jyväskylä, Finland
Armin Heinzl, Mannheim University, Germany
Thomas Hess, LMU München, Germany
Juhani Iivari, University of Oulu, Finland
Epaminondas Kapetanios, University of Westminster, UK
Timo Koivumäki, University of Oulu and VTT Research Center, Finland
Olli Kuivalainen, Lappeenranta University of Technology, Finland
Olli Martikainen, Research Institute of Finnish Economy, Finland
Lars Mathiassen, Georgia State University, USA
Rod McNaughton, University of Waterloo, Canada
Arto Ojala, University of Jyväskylä, Finland
Balaji Parthasarathy, IIIT Bangalore, India
Oscar Pastor, Valencia University of Technology, Spain
Jan Pawlowski, University of Jyväskylä, Finland
Karl-Michael Popp, SAP, Germany
Björn Regnell, Lund University, Sweden
Matti Rossi, Helsinki School of Economics, Finland
Sami Saarenketo, Lappeenranta University of Technology, Finland
Sowmyanarayanan Sadagopan, IIIT Bangalore, India
Camile Salinesi, University of Paris, France
Steve Sawyer, Syracuse University, USA
Veikko Seppänen, Elektrobit Corporation and University of Oulu, Finland
Kari Smolander, Lappeenranta University of Technology, Finland
Sergiu-Dan Stan, Technical University of Cluj-Napoca, Romania
Clemens Szyperski, Microsoft, USA
Virpi Tuunainen, Helsinki School of Economics, Finland
Inge van de Weerd, Utrecht University, The Netherlands
Juhani Warsta, University of Oulu, Finland
Tony Wasserman, Carnegie Mellon University, USA
Claudia Werner, Federal University of Rio de Janeiro, Brazil
Stanisław Wrycza, University of Gdańsk, Poland

Table of Contents

Full Papers

Business Models and Business Management

Internationalization and Ecosystems

Product Management and Open Source Software

Software as a Service and Green Software

Short Papers

Business Management

Internationalization and Ecosystems

Product Management, Open Source Software and Social Media

Education and Research

Workshops, Tutorials, and Industrial Session

Diversity of Business Models in Software Industry

Aku Valtakoski and Mikko Rönkkö

Aalto University
{aku.valtakoski,mikko.ronkko}@tkk.fi

Abstract. In this paper, we measure the business models of Finnish software firms using survey data. To do this, we first conceptualize the business model based on extant management research, and allows for effective operationalization of the concept. Using our conceptual framework of business model, and data from the Finnish software industry, we find a taxonomy of eight business models. Analyzing the characteristics and performance of the firms using these models, we find that there are significant differences between firms using different business models.

Keywords: Business models, taxonomy, software industry, Finland.

1 Introduction

The software industry is undergoing a significant change in its structure [1]. Caused by the introduction and maturation of Internet technologies, the traditional boundaries of software industry sectors are becoming increasingly blurry [2] In general, the industry seems to be moving from the sales of standalone software products towards provision of various types of services [3,4]. This change is making traditional categorizations of software firms increasingly invalid, as many new types of software businesses, such as Software as a Service and open source models do not fit into the traditional archetypes of packaged software, enterprise solutions, and professional services [5]. This change also creates a problem for empirical research, as it is no longer easy to identify and categorize firms that belong into a certain sector in the software industry. Controlling for the type of firm is thus increasingly difficult.

To overcome these challenges caused by the evolution of the software industry, we use the concept of business model to provide a more detailed view of how software firms are actually conducting business. A business model is often conceptualized as a description of business logic of a firm, i.e. how the firm creates and appropriates value within its business network. Software business models have been studied actively in the information systems research community [6]. However, the current theoretical development of the business model concept is still somewhat dissatisfactory: As noted by multiple authors, there is still a lack of a coherent definition for the concept [7-9]. Furthermore, the theoretical grounding of the concept and its relation to other management research have only been discussed in a handful of papers [10-12,6]. In addition, and most importantly to the current study, very little empirical research has been conducted using the concept. In particular, only few

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 1–12, 2010.

authors have used quantitative methods in measuring business models. Consequently, only very few measurement instruments have been proposed in the literature [11,12].

Therefore, due to the changes in the software industry and the relative immaturity on the scholarly discourse on business models, we chose to perform a quantitative analysis of business models using data from the Finnish software industry. The present paper is an extension to our previous business model papers [e.g., 13,14] using a new set of data and new measurement approach.

2 Literature Review

2.1 History of the Business Model Concept

The origins of the business model concept can be linked to Normann [15], who described the concept of a 'business idea', distinguishing between the external environment (i.e., customer needs and value appropriation), the company's products and services, and internal factors (e.g., values, organization structure, resources and capabilities). However, the concept of business model significantly increased in popularity during the 1990s when it was widely used to describe the business logic of start-up firms during the dot-com boom [16,17]. Since then the business model concept has increasingly been used in mainstream management research on innovation [18], entrepreneurship [19], information systems [6],and strategic management [20,11,12].

However, so far, business model is rarely mentioned in the top-tier management journals. The concept of business model has also been avoided in this literature. For example, Porter has criticized the concept, calling it a part of "internet's destructive lexicon" [21]. This critique is understandable given the lack of theoretical development behind business models, as well as the general vagueness of the concept.

There is indeed considerable variation in the definitions of the business model concept in the extant literature in terms of content and level of detail. While some authors give a full and explicit definition [10,19], others define business model through defining its constituent elements [22,7,17,18]. However, despite this confusion, there are some common themes: First, many definitions of the business model include the idea of a business logic, i.e. an abstract description of how the firm works and how it plans to make money. Second, most definitions include some notion of value - the concept is used to describe how the firm creates and appropriates value.

For example, in two of the rare papers about business models in top-tier management journals, Zott and Amit define business model as design of organization's boundary spanning transactions [11], or "the content, structure, and governance of transactions designed to create value through exploitation of business opportunities" [10] and that "the design elements of a business model are the content, structure, and governance of activities". The focus is clearly on value creation and the activities of the firm, yet the relation of business model to other concepts in this area, such as strategy and resources, remains unclear.

Many authors consider the business model to belong to the domain of strategic management [20,10-12]. This is particularly apparent when we consider the number of studies attempting to explain firm performance by its business model [12]. Yet, as indicated by Zott and Amit [12] and others [20,19], the two concepts strategy and business model can and perhaps should be seen as distinct concepts.

Within the strategic management field, we may differentiate between two complementary key problems: the cross-sectional, static problem of logic of advantage against competition and the longitudinal, dynamic problem of creating and maintaining a competitive advantage [23]. While the purpose of strategy is to describe what the competitive advantage is and how it is pursued, most definitions of business models only describe how the firm creates and captures value. In other words, business model does not address external competition or the attainment of long-term goals of a firm, which are important issues particularly when considering the longitudinal problem of strategy. A business model addresses the configuration of a business at a specific point in time, and thus a part of the static problem of strategy.

The static problem of strategy is related to how a firm positions itself relative to its customers, competition, and partners. On the firm level, a firm can pursue competitive advantage through three generic strategies: cost leadership, differentiation, or focus [24]. However, these generic strategies only describe the high-level business approach of the firm, and do not describe how the firm actually configures its resources to attain competitive advantage. The analysis of this problem thus requires the analysis of firm's activities. In Porter's words, "A firm is a collection of discrete, but interrelated economic activities [...] A firm's strategy defines its configuration of activities and how they interrelate" [23]. When linked, these discrete activities form a value chain within the focal firm, and a value system when connected to other firms' value chains [23,24]. To attain competitive advantage, a firm must configure its value chain in a way that exploits its strengths in capabilities and competitive position.

Another view of strategy considers analyzes the effect of resources on the performance of a firm. This resource-based view (RBV) sees that the source of competitive advantage is the valuable resources a firm possesses [25]. Resources can be seen as complementary to the activities of a firm, as resources are only valuable if they enable the firm to perform activities that create advantages [23]. Correspondingly, activities are only valuable if required resources exist to carry them out. Resources and activities, i.e. business model can be seen as complementary to each other.

Combining these two aspects of strategic management of a firm to the competitive strategy of the firm, as well as the competition environment, including competition, customers, and partners, we have four complementary concepts that describe the strategy of a firm: competitive strategy, competitive environment, resources, and business model. The overarching idea is that these four elements, taken together, affect the firm's success. All elements are under firm's management's control to some degree: it can change its strategy, or market, or acquire new resources. Furthermore, these elements interact; for example, pursuing a certain competitive strategy requires a specific business model and specific complementary resources, and is only effective in certain competitive environment. Porter [23] describes this: "The required mix and configuration of activities, in turn, is altered by competitive scope". Yet there is specific direction of causality between the elements, i.e. one element affecting directly another element.

This conception of business strategy is similar to the research on configurations [26-29]. This perspective suggests that business is seen as a complex configuration of highly interdependent organizational factors, and a holistic, *gestalt* type of fit must exist between these factors to ensure organizational performance [26,30] .

After narrowing down the scope of the concept, we still need to delineate the elements of business model. Nearly all papers on business models list constituent elements of a business model; Common elements that are often found include strategic factors (competition, strategy), customers and target market (including distribution), offering (including value proposition, products and services provided), resources, key activities, revenue logic (including pricing and costs), and value chain/network (relationship and alliances) [7,9,6].

From the discussion on the relationship between strategy, business model, and other concepts and assuming conceptual parsimony as desirable, it follows that some elements of all the potential business model elements found in the papers that we reviewed must clearly be excluded. In our conceptualization, competitive strategy, resources, and market are complementary concepts to business model and cannot therefore be included as components of the business model. What is then left of the potential components are offering, activities, revenue logic, and value network. In summary, based on the literature review of both business model and general management literature, we may now define business model as *a configuration of the firm's offering, activities, value network, and revenue logic, given its competitive strategy, resources, and competitive environment.*

3 Empirical Study

A common assumption about the configuration approach to management research is that despite the large number of variables, only a limited number of configurations are likely to be present in practice. This is due to factors that reinforce both internal and external coherence of a configuration [27]. Hence, we should see only a limited number of feasible configurations. Clearly, business models, when considered on a high level, should be measured as classifications.

Constructing classifications is the domain of the family of statistical methods known as cluster analysis. These methods divide the data into distinct groups based on a measure of similarity of dissimilarity between observations; observations that are only slightly dissimilar belong to the same cluster [31]. However, the cluster analysis methods have their weaknesses. As reported by Ketchen and Shook [32], the method relies heavily on subjective researcher judgment. Furthermore, there is no statistic for assessing the fit of clustering results. However, they also conclude that with proper attention to details when specifying the analysis and post-analysis checks of validity, cluster analysis can yield valuable results.

To collect the data required for cluster analysis, we performed a survey of the Finnish software industry.

3.1 Measures

To operationalize our business model concept, we again relied on the coherence assumption of the configuration approach. Instead of measuring all four elements of a business model, we suggest that measuring the revenue shares of various revenue sources would give enough information to conduct a cluster analysis. First of all, where a firm finds its revenue is related to the firm's offering, activities it performs,

as well as its revenue model. Revenue shares thus cover three of these business model components. Second, the importance of revenue sources is likely to improve the validity of the measurement, as firms are likely to give more accurate information. In addition, revenue shares measures what a firm has actually done.

To create a list of revenue sources, we used our own expertise, as well as several practitioners within the industry. Included revenue sources covered sources related to both product and service income, as well as less familiar sources, such as content and ads, and non-software sources. In the survey form we instructed the informant to divide one hundred percentage points between these alternatives.

3.2 Sample and Data Collection

Our data are a subset of a larger data set that was collected as a part of a research project called the Finnish Software Industry Survey [1]. No lists that would cover the entire industry are available, and so we gathered our data for the sampling frame from multiple sources. However, a large majority of the firms were obtained from the trade register with NACE codes 72.21. ("Publication of software") and 7222 ("Other software consultancy and supply"). Altogether, this resulted in a sampling frame of 4544 firms. We estimated that a third of these firms were not active or not in the software industry.

The survey was implemented following a modified version of the tailored survey design method [33]. The mailing begun with a pre-notice letter, followed by the main survey package using postal mail. A printed questionnaire and a web form were offered as alternative options for the informants. Data collection lasted from June to August 2009, and produced 612 usable responses presenting roughly a response rate of 20%. However, since most of the non-responding firms were very small (one man companies and part-time entrepreneurs) the representativeness of the data of the software industry proper is higher, although an exact figure is virtually impossible to estimate.

3.3 Data Analysis

Our main data analysis method is cluster analysis using with Stata 10.1. Prior to running cluster analyses, we inspected the data and found that the variables were right-skewed. Due to this feature, we chose that distance based similarity measures were inappropriate for the data and instead used the angular similarity measure.

We started the cluster analysis by running a hierarchical average linkage cluster analysis. After this, we analyzed the dendrogram and chose the best solution as a starting point for confirmatory cluster analysis with K-medians clustering. This algorithm was again chosen because of the skewness of the data.

As a standard procedure, this two-step clustering was repeated several times with different similarity measures and agglomeration algorithms to ensure that the final cluster solution was indeed the most appropriate.

The robustness of the final results was checked by repeating the analysis over 50 bootstrap samples. We tested the robustness of the solution by comparing the cluster memberships in each boot strap sample to those calculated based on the full data. The mean of Cramer's V over 50 bootstrap samples was 0.899 for the hierarchical

clustering and 0.965 for the K-median clustering indicating that the final cluster solution was robust.

4 Results

Figure 1 displays the categorization found in the cluster analysis, and what the revenue structure of each business models is. Our analysis indicates that Finnish software firms can be divided into eight categories:

Software product firms. This cluster consists of firms who offer their own software products using the traditional licensing model. The firms in this cluster can be roughly divided into two subgroups: firms who generate most of their revenue from license sales, and firms who offer also an extensive array of services, including maintenance, customization and deployment services.

ASP and SaaS firms. Firms in this cluster are in many ways similar to the software product firm cluster. However, instead of offering their software products as licenses, firms in this cluster instead offer their software over the internet using either an Application Service Provision or Software-as-a-Service arrangement. Income based on these software service fees forms the major share of their revenue. In addition, they draw secondary revenue from software customization and deployment projects.

Development service firms. This cluster consists of firms that provide software development services. This can take two forms: either these firms provide subcontracting services (i.e. developer rental), or they do custom development projects for customers. In addition to development services, these firms also offer additional services, such as system maintenance and deployment services.

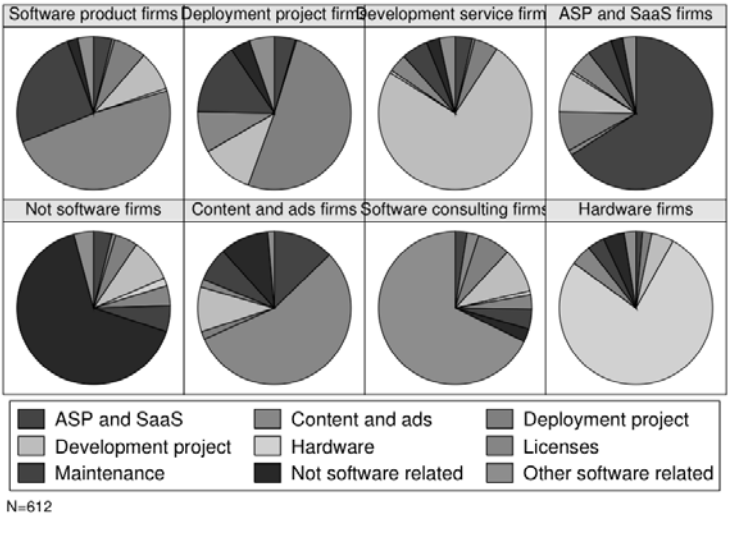

Fig. 1. Revenue sources of identified business models

Deployment project firms. These firms specialize in deployment projects. In other words, the group includes firms who provide installation, customization, and parametrization services mainly for ready software products. Some of these firms may also develop a limited amount of custom-made software for customers, and maintain the systems they have deployed. They also receive some revenue from the sales of software licenses.

Software consulting firms. Firms in this cluster earn most of their revenue from various software-related services, such as consulting and testing. In addition, they also often offer some development and deployment services.

Hardware firms. This cluster generates majority of its revenue from the sales of hardware products in which software is embedded. In other words, they are more manufacturing than software firms.

Content and ads firms. This cluster differs from most other clusters: instead of generating revenue directly from software IPR or from software-related services, this cluster earns majority of its revenue from the development or provisioning of content or advertising. In addition, these firms also generate revenues from non-software sources, software customization and from offering software over the internet.

Non-software firms. This cluster consists of firms receive most of their revenue from non- software sources. These sources include general organizational consulting and provision of IT support services.

Table 1 shows a cross tabulation of the business model classification with the self-reported firm type by the respondent, elicited by asking the firms to select a best fitting description for their main activity on the first page of the survey. Firms considering themselves software product firms naturally fall into the software product business model. As can be expected, most firms in the SaaS and ASP business model category also consider themselves to be software product firms. However, what is surprising is the large share of firms who consider to be product firms but actually have a business model that belongs to the development service category. This can be explained by the fact that young product firms often earn most of their revenue from a small number of customers, and usually do a lot of custom development for these customers while trying to increase the degree of productization. Majority of the firms

Table 1. Comparison between identified business models and self-reported firm types

	Firm type					
Business model	Software product firm	device manu-facturer	Software project contractor	Consulting firm	Reseller	**Total**
Software product firms	45.5%	7.4%	5.4%	6.8%	55.6%	22.2%
Deployment project firms	6.4%	0.0%	9.1%	13.6%	0.0%	8.2%
Development service firms	17.0%	7.4%	61.0%	30.7%	0.0%	36.0%
ASP and SaaS firms	20.0%	7.4%	7.5%	1.1%	11.1%	11.5%
Not software firms	3.4%	18.5%	10.0%	27.3%	22.2%	10.5%
Content and ads firms	2.1%	0.0%	0.8%	2.3%	0.0%	1.5%
Software consulting firms	4.3%	3.7%	5.4%	18.2%	0.0%	6.7%
Hardware firms	1.3%	55.6%	0.8%	0.0%	11.1%	3.5%
Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

that considered themselves device manufacturers seem to use the hardware firm business model. Furthermore, a significant share of these companies seem to use the non-software business model. In summary, we may infer that the hardware firm business model applies to firms for which software products and software development are peripheral activities.

Software project contractor firms are naturally most often associated with the development service business model. The second most relevant business model for these firms is the non-software business model. This indicates that these firms often also engage in business consulting and this can sometimes dominate as a revenue source for the project contractor.

Firms who consider themselves consulting firms are not associated very strongly with any specific business model. On the contrary, these firms can actually employ a wide range of business models, of which development project firm, deployment project firm, non-software, and software consulting business models are the most common. Finally, the few resellers firms in the sample were classified using the software product business model, since most of their revenue comes from selling software licenses.

In summary, the comparison between self-reported firm type and business model categorization provides a fairly consistent picture of the classification of software firms. This consistency is most apparent for firms who reported to be device manufacturers and software project contractors. The discrepancies between the two categorizations can be attributed to the often large differences between the self-projected image of the firm and what the firm actually does. In addition, they demonstrate the difficulties of measuring exactly how a firm actually conducts its business, i.e. what is its business model.

Next we analyze in detail how the firms in each business model differ from each other with some other key variables collected in the survey. Table 2 compares the means of these variables. Exact definitions of variables and comparisons of medians are excluded from this paper due to space constraints, but are available elsewhere [1]. Analysis of medians was required due to skewness of the data. Particularly the negative change in revenue of -32.8% for the year 2008 for hardware firms seems a too large. The medians did support the interpretation that for the average hardware firm in the sample, a third of the revenue did disappear during 2008. However, considering that the data was collected during the bottom of the recession and the fact that hardware firms were hit the hardest during the recession, this drop is actually credible.

Comparing the mean and median age of software firms by business model, we see that the differences in firm age are quite small. Software product firms stand out as the most mature firm with the average age of 11 years, followed by deployment project and not software firms. While there are no great differences between the other business models, it seems that development project, content and ads, and software consulting firms seem to be the youngest.

Concerning the total revenue of the firms by business model, we may immediately conclude that for most business models, there is a significant difference between the mean and median revenue. This is an example of a skewed distribution of a variable, and implies that there are large firms in each business model that skew the distribution towards larger values. In this case the median value gives a more reliable approximation of the typical revenue of a firm in each category.

Table 2. Mean values of business model characteristics and performance

	Business model							
	Software product firms	Deployment project firms	Development service firms	ASP and SaaS firms	Not software firms	Content and ads firms	Software consulting firms	Hardware firms
	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean
Age	11***	10	8**	9	10	6	7	8
Revenue (M€)	4.82**	1.49	2.00**	3.10	13.18	0.46	0.75	5.53
Total personnel	43*	15	215**	23	80	8	7*	52
Growth of revenue	11.9**	43.6*	31.8*	12.9	17.3	19.1	19.1	-32.8*
Growth, 3 year CAGR	13.8*	16.0	29.5	17.1	19.5	36.6	21.9	25.2
Willingness to grow	118.7*	105.1	86.7**	140.8***	70.2**	111.9	81.7	160.3**
Profitability	8.9	12.0	12.4*	2.8*	7.4	9.3	10.1	-1.2**
Productivity (k€)	102.1*	95.9	79.3*	72.0**	113.7**	54.0	86.6	83.2
Has revenue from abroad	53.8***	38.8	30.4***	36.4	40.0	44.4	37.5	68.4**
Plans to internationalize	20.0	20.4	30.4*	34.8	15.0	22.2	25.0	15.8
Revenue share from abroad	36.5	15.6	25.0	23.8	19.3	43.0	30.6	60.6***
R&D / revenue	26.8***	14.4	11.5***	27.9***	11.7	26.4	13.5	34.5
N	139	49	216	70	67	9	40	22

Mann-Whitney U tests between one group and the rest. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Using the median values of revenue to compare firms in different business model categories, we may conclude that software product, ASP and SaaS, hardware, and deployment project firms seem to be larger than the median for all firms. Comparing these median values to means, we see that there are large differences between these two statistics in software product, non-software, and hardware firm categories. This implies that there are several large firms in these business models. In comparison, the difference between mean and median is very small in content and ads and software consulting firms. This indicates that the firms using these two business models are small and that there are no large exceptions in either category.

Since the most important resource for all types of software firms is the human capital, the mean and median personnel counts provide a similar story as the comparison of revenue. Judging by the mean number of personnel, the largest Finnish software firms seem to use one of the development service, non-software, hardware, or software product business models.

The revenue growth in 2008 and over the period 2006–2008 seem to present a consistent story since there is less discrepancy between the mean and median values. This is due to the fact that revenue growth is a relative figure and does not depend on the size distribution of the firms. Using the revenue growth in 2008 figure, we may conclude that, with the exception of hardware and software product firms, software firms in each business model maintained their growth through the economic downturn of 2008. As suggested by Cusumano [3], service firms seem to have withstood the pressures of the recession best. Product-oriented firms seem to have suffered clear slowing down in their revenue growth. This is most dramatic for hardware firms, whose revenue actually dropped by nearly a third in 2008, in comparison to the approximately average growth of 20% in the last three years.

Comparing revenue growth figures with willingness to grow provides interesting insights. A larger figure in the willingness to grow indicates that the firm's

management most important financial target is revenue growth over profitability. Comparing these two figures indicates that the business models hit by the recession - product-oriented firms - are actually the ones who are more willing to grow. By comparison, firms less willing to grow, such as development service, deployment project, and content and ads firms were actually the ones who grew the most in 2008. Hence, despite a high willingness to grow, product-oriented firms were not able to increase their revenues in 2008. This suggests that the recession in 2008 had a significant impact on the business of these product-oriented firms.

The problems with revenue growth are also reflected in the profitability figures of product-oriented firms. As these growth-oriented firms have made investments in new human resources, their cost levels might have increased. Since they were unable to reach the targeted revenue level, this meant a significant drop in profitability as well. By comparison, service-oriented firms who get their revenues from selling the working time of their employees did not experience such drop. Deployment project, Software consulting and development service firms were thus the most profitable business models in 2008. In contrast, software product, ASP and SaaS, and hardware firms struggled to remain profitable. Comparing productivity figures between business models provides another indication of the impact of slow revenue growth. Even though a product-oriented business model should enable the easy replication of business and hence improve the productivity of business per employee, the productivity figures indicate that there are only small differences in productivity between business models. This implies that product-oriented firms were not able to reach the level of revenue per employee as expected from their business model. The results also show that low productivity does not necessarily imply low profitability. Inspecting the figures of revenue share from abroad, plans to internationalize and indicator of having revenue from abroad, we may conclude that hardware firms are the most internationalized software firms, followed by software product and content and ads firms. Deployment project, development service, software consulting and non-software firms are least interested in internationalization and have the least revenue from abroad. These findings support the common view that product-oriented firms are more interested in international markets. By comparison, service-oriented firms provide services that are always local to some degree, implying the necessity of remaining close to the customer.

Somewhat surprisingly, ASP and SaaS firms, who are the most willing to grow, do not stand out in the internationalization characteristics. It may be that the firms are still too young or small to consider internationalization. This is a bit alarming since these business models are by definition global; in other words, there are no local markets for SaaS offerings. Quick internationalization would thus seem to be the most obvious growth strategy for these firms.

5 Discussion and Conclusions

In this paper, we have sought to measure the business models employed by Finnish software firms. Such action is necessary since the evolution of the software industry caused by Internet technologies is quickly dissolving the traditional sectors of the industry. Basing our conceptualization of business model on the configuration

approach to management, and ideas stemming from Porterian value chains, we argue that a business model can be understood as a configuration of a firm's offering, activities, revenue logic, and value network, given the competitive strategy, resources, and competitive environment of the firm.

With cluster analysis of revenue sources, we found that there are eight different main business models in the Finnish software industry. Our results indicate that clear majority of the firms employ a development project model, followed by the traditional software product model. Furthermore, we found that there are differences between firms employing different business models in terms of their characteristics and performance. More specifically, we found statistically significant differences in firm size, willingness to grow, and both revenue growth and profitability. These differences indicate that business models matter – firms using different business models are likely to experience differences in their performance.

Our results contribute to the evolving literature on business models. In particular, we have explicitly related the concept to existing management theory, providing a much stronger theoretical grounding than previous studies. Furthermore, we have provided a framework for operationalizing the business model concept, and demonstrated its usefulness in the context of Finnish software industry.

References

1. Rönkkö, M., Ylitalo, J., Peltonen, J., Koivisto, N., Mutanen, O., Autere, J., Valtakoski, A., Pentikäinen, P.: National Software Industry Survey 2009. Helsinki University of Technology, Espoo (2009)
2. Campbell-Kelly, M., Garcia-Swartz, D.D.: From Products to Services: The Software Industry in the Internet Era. Business History Review 81, 735–764 (2007)
3. Cusumano, M.A.: The Business of Software: What Every Manager, Programmer and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad. Free Press, New York (2004)
4. Cusumano, M.A.: The Changing Software Business: Moving from Products to Services. IEEE Computer 41, 20–27 (2008)
5. Hoch, D.J., Roeding, C.R., Lindner, S.K., Purkert, G.: Secrets of Software Success: management insights from 100 software firms around the world. Harvard Business School Press, Boston (2000)
6. Osterwalder, A., Pigneur, Y., Tucci, C.L.: Clarifying Business Models: Origins, Present and Future of the Concept. Communications of the Association for Information Systems 16, 1–25 (2005)
7. Pateli, A.G., Giaglis, G.M.: A research framework for analysing eBusiness models. European Journal of Information Systems 13, 302–314 (2004)
8. Tikkanen, H., Lamberg, J.A., Parvinen, P., Kallunki, J.P.: Managerial cognition, action and the business model of the firm. Management Decision 43, 789–809 (2005)
9. Shafer, S.M., Smith, H.J., Linder, J.C.: The power of business models. Business Horizons 48, 199–207 (2005)
10. Amit, R., Zott, C.: Value creation in e-business. Strategic Management Journal 22, 493–520 (2001)
11. Zott, C., Amit, R.: Business Model Design and the Performance of Entrepreneurial Firms. Organization Science 18, 181 (2007)

12. Zott, C., Amit, R.: Exploring the Fit Between Business Strategy and Business Model: Implications for Firm Performance. Strategic Management Journal 29, 1–26 (2008)
13. Rönkkö, M., Valtakoski, A.: Business Models of Software Firms. In: Proceedings of the Forty-Second Annual Hawaii International Conference on System Sciences (CD-ROM), 10 p. Computer Society Press, Waikoloa (2009)
14. Valtakoski, A., Rönkkö, M.: The Concept of Business Model in Management Practice and Research: Bridging the Relevance Gap., Chicago, USA, p. 39 (2009)
15. Normann, R.: Management for Growth. Wiley, New York (1977)
16. Mahadevan, B.: Business models for Internet-based e-commerce: An anatomy. California Management Review 42, 55–69 (2000)
17. Timmers, P.: Business Models for Electronic Markets. Electronic Markets 8, 3–8 (1998)
18. Chesbrough, H.W., Rosenbloom, R.S.: The Role of the Business Model in Capturing Value from Innovation: Evidence from Xerox Corporation's Technology Spin-Off Companies. Industrial and Corporate Change 11, 529–555 (2002)
19. Morris, M., Schindehutte, M., Allen, J.: The entrepreneur's business model: toward a unified perspective. Journal of Business Research 58, 726–735 (2005)
20. Magretta, J.: Why business models matter. Harvard business review 80, 86–93 (2002)
21. Porter, M.E.: Strategy and the Internet. Harvard Business Review 79, 62–76 (2001)
22. Hedman, J., Kalling, T.: The business model concept: theoretical underpinnings and empirical illustrations. European Journal of Information Systems 12, 49–59 (2003)
23. Porter, M.E.: Towards a Dynamic Theory of Strategy. Strategic Management Journal 12, 95–117 (1991)
24. Porter, M.E.: Competitive advantage: creating and sustaining superior performance. Free Press, New York (1985)
25. Barney, J.B.: Firm resources and sustained competitive advantage. Journal of Management 17, 99–120 (1991)
26. Miller, D.: Toward a New Contingency Approach: The Search for Organizational Gestalts. Journal of Management Studies 18, 1–26 (1981)
27. Miller, D.: Configurations of Strategy and Structure: Towards a Synthesis. Strategic Management Journal 7, 233–249 (1986)
28. Ketchen, D.J., Thomas, J.B., Snow, C.C.: Organizational configurations and performance: a comparison of theoretical approaches. Academy of Management Journal 36, 1278–1313 (1993)
29. Dess, G.G., Newport, S., Rasheed, A.: Configuration Research in Strategic Management: Key Issues and Suggestions. Journal of Management 19, 775 (1993)
30. Venkatraman, N.: The Concept of Fit in Strategy Research: Toward Verbal and Statistical Correspondence. Academy of Management Review 14, 423–444 (1989)
31. Hair, J., Anderson, R., Tatham, R., Black, W.: Multivariate data analysis. Prentice-Hall, Englewood Cliffs (2006)
32. Ketchen, D.J., Shook, C.L.: The application of cluster analysis in strategic management research: an analysis and critique. Strategic Management Journal 17, 441–458 (1996)
33. Dillman, D.: Mail and internet surveys: The tailored design method (2007)

A Licensing and Business Model for Sharing Source Code with Clients—Leveraging Open Client Innovation in the Proprietary World

Mikko Riepula

Aalto University, Helsinki School of Economics
P.O. Box 1210, FIN-00101 Helsinki, Finland
mikko.riepula@hse.fi

Abstract. While the pure-form open source model has been in the focus of much information systems research recently, research is lacking on truly hybrid software licensing models that combine limited openness of source code with traditional value appropriation logic. Yet such boundedly open models are implied by open innovation literature and by the recent emphasis on client involvement therein. The further amalgamation of traditional proprietary software licensing with open source licensing can be rationalised through commodification of software, shift from products to services, the viability of the pure-form open source model, and the continuing need for better operational efficiency. After reviewing other hybrid-OSS models, a very practical hybrid licensing model is presented that responds to the need of software vendors in various vertical domains. Its benefits and limitations are discussed, and it is positioned relative to the value co-production (co-creation) literature. A set of research questions are raised to the information systems research agenda.

Keywords: client-shared source, shared source, gated source, open source, hybrid OSS, OSS 2.0, software licensing, client innovation, client co-production, distributed development, inner source, corporate source, software business model, software commodification, software commoditization.

1 Introduction

Traditionally, software vendors have guarded their source code rather jealously. Recently, the open source alternative has evolved into something that has also business credibility and a big impact on the software market. However, success stories of companies who had managed to turn around their business from traditional closed source to an open source business remain scarce. At the same time, crowd-sourcing or user contribution systems are becoming viewed as obvious, yet underexploited ways of benefiting from user innovation [1].

We can expect that in commodified software markets access to source code becomes more and more common, but without the software necessarily being released as OSS. Companies are becoming more and more comfortable working with open-source software (OSS) and adopting OSS-like processes. These factors would seem to

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 13–25, 2010.

prepare the ground for *client-shared source* as a hybrid in between the traditional and OSS business logic.

This article takes the viewpoint of a business-to-business (B2B) software vendor, whose clients have—at least to some degree—the willingness and ability to modify source code. The client-shared source model is particularly applicable in cases where the software product needs to be customised or extended for many clients e.g. in different regions or industry sectors, which means the customisations or extensions are many and varied, yet potentially reusable by other clients, but it is by no means a solution to every vendor's needs.

This article presents the client-shared source model as a forward-looking licensing and business model that deductively follows from the results of earlier work, based on a rather extensive two-level backward literature survey. I aim to demonstrate how it and its adoption by certain kinds of software vendors lies on a solid theoretical foundation and how it is therefore to be expected to be seen much more in the future. Secondly, I aim to give very practical guidelines for vendors wishing to implement the model. For practitioners, the impact is potentially very high: the client-shared source model could be implemented not only by vendors of less-differentiating products, but also by consultancies and integrators who find themselves doing similar projects to different clients, the results of which have been traditionally maintained as separate quasi-products.

Section 2 presents the environmental and other factors that then logically lead to client-shared source, which is presented in Section 3 together with its promises as well as challenges to overcome. Section 4 is devoted to discussion, Section 5 outlines certain relevant research questions and Section 6 concludes.

2 The Changing Software Market Environment

Value co-production (or co-creation) refers to two or more economic actors working in co-operation for producing a service or a product for either party, or for a third party, across organisational boundaries. The formal study of value co-production started already in the early 18th century [2]. Here the central thoughts are applied to information systems science. Cook has recently popularised many of the same thoughts using the term "user contribution systems" [1], while the Nambisans urge us to profit from "virtual customer environments" (VCEs) in product idea generation, development, testing and support [3][4]. Both aim at engaging one's clients in the product or service development and delivery process.

In innovation literature, services co-production has been researched especially in knowledge-intensive business services (KIBS) [5][6][7].

2.1 Methodology

The literature study was conducted in 2007-2009 as a combined keyword search, two-level backward search on references and forward references search [8] mostly in the following databases: Ebsco Business Source Complete, ACM, ProQuest Direct, and the ISI Web of Knowledge Social Sciences Citation Index.

Below I examine the changing software marketplace from a software vendor's perspective. Subsections 2.2–2.4 assume that the vendor keeps its market offering unchanged, i.e. as a traditional closed source offering. Subsection 2.5 opens the door to the new offering that is then introduced in Section 3.

2.2 From Commodified Products to Services

The holy grail of software business has been to come up with an innovative software product that can be sold to the masses at very low marginal cost. However, it is a very elusive goal. In reality, most software vendors find themselves customizing their offering to different client groups or even individual clients, which significantly increases the marginal sales and delivery costs. These software vendors thus find their businesses becoming more and more service businesses as opposed to product businesses without them necessarily being prepared for it [9][10].

The same shift from products to services can in part be explained by commodification and Teece's appropriability regimes [11].

In the early stage of a high-tech industry, market success depends on product innovation based on proprietary technology. Once a dominant design emerges, commodification begins and companies need to innovate with complementary assets such as marketing, efficient operations, after-sales support etc. in order to be competitive. Focus moves from product innovation to *business innovation*. [12]

Rather than entering the otherwise topical software products vs. services debate, we can adopt the notion of "offering" comprising both products and services. In co-production it is co-produced offerings, not the 'business unit' actors, which become the central unit of (competitive) analysis [2].

2.3 Continued Competitive Threat from Open Source as an External Factor

There is a consensus in the academic literature about the long-term viability of OSS.

Motivational factors that explain the emergence of OSS as a non-revenue generating activity have been widely researched (see e.g. the references in [13] and [14]). Nambisan and Baron explained why and how people participate in VCEs more generally [4].

Technology maturity models can also be used to explain how the exploration phase preceding the era of a dominant design is conducive to open sharing of information [15]. Even a special "private-collective" innovation model [16], as opposed to a generic collective model [17], has been proposed to better explain OSS as a phenomenon.

Various different business models have emerged and have also been researched in and around OSS [18][19][20][21][22][23][24]. One is that based on services (e.g. support and maintenance, or customisations). Another is *dual licensing,* in which the same software is made available both under a viral license such as the GNU Public License (GPL), for free, and at the same time under a non-viral license sold for a fee.

Apart from the obvious considerations of cost, control and accessibility, *end-user innovation* is a distinct benefit of OSS and is also a widely addressed topic in academic literature [25][26][27][28]. Users are often the best experts in determining how a product should be improved. Knowledgeable and motivated users can even be the best people to determine how to change the source code. They are also often "lead

users", i.e. on the leading edge of technology development [27]. They are, however, far from being in the best position to market the software, especially when their own contribution remains relatively small. The transaction cost for user innovators to license their innovations is thus prohibitively high [28]. An individual user innovator is willing to give away his incremental developments often for free, hoping that others would also incorporate his changes before developing the product further and would thus keep the future improved product versions compatible with his own needs.

A software vendor participating in an OSS project thus often *wants* to contribute their incremental developments to be maintained by the project community, *driven by commercial self-interest* as opposed to purely intrinsic motives [29][30][31].

2.4 Ever-Increasing Pressure for Internal Efficiency: Inner Source as an Emergent Internal Factor

Inner source refers to a company promoting reuse and OSS-like development processes internally, supported by tools and practices familiar from the OSS setting [32][33]. The term "corporate source" has been used in very much the same meaning [34]. "Hybrid-OSS" by Sharma et al. [35] is first and foremost about inner source. Elements of inner source can also be seen in [19][29][36][37].

Fig. 1 demonstrates how the typical technology life cycle overweighs innovation with value appropriation and results in commoditised technology. The development of commodity software in-house as well as releasing differentiating software too openly are both to be avoided (the lower left-hand and upper right-hand corners of Fig. 1).

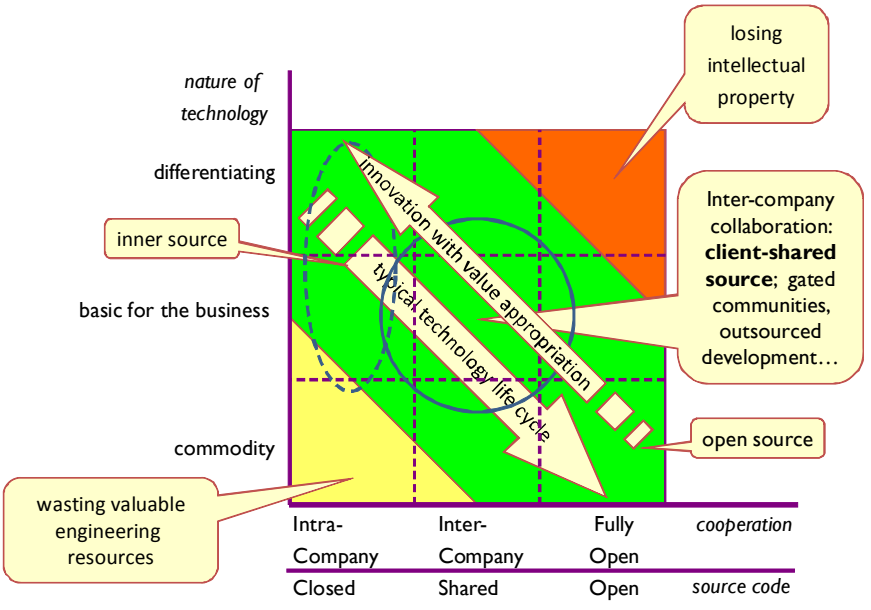

Fig. 1. Shift from differentiating software to commodity over time, with the two corners to avoid (adapted from [33])

For our purposes, we can regard the horizontal axis as a question of code ownership and availability of source code more than as a question of technology sourcing. In Fig. 1, inner source falls in the dashed-line oval. However, internal sharing is still very different from involving one's clients (the solid-line circle in the middle of Fig. 1), which is what will be discussed in Section 3.

2.5 Hybrid Models in the OSS Context

In the OSS context, the term "hybrid" is used in various different ways to refer to related, yet different, constructs.

Bonaccorsi et al. [18] used the term "hybrid business model" to describe a model, where a firm's revenues only partly come from a pure OSS offering. Today this is taken for granted. Their unit of analysis is thus the firm, not the offering.

Fitzgerald foresees more "pragmatic hybrids" of older OSS business models under the term "OSS 2.0", a more purposive and business-oriented mode of OSS development where strategic planning becomes key [19]. Yet the base models are the same old widely known OSS business models, with new OSS (even if not OSI-approved) license variants.

Regarding Fitzgerald's "corporate" and "non-approved" licenses [19], Matusow [38] has elaborated on the rationale behind Microsoft's shared source initiative (not to be confused with the term client-shared source in this article).

"Hybrid-OSS" by Sharma et al. [35] is first and foremost about inner source—it does not particularly focus on the client interface or on the ensuing licensing issues.

Fox et al. [39] talk about "hybrid (technology) ecosystems" and present a model where value creation processes occur on the "periphery". Creation and control incentives as well as infrastructure are all needed for such a hybrid ecosystem to work. While their explicit unit of analysis is such an ecosystem, implicitly they also address the offering made by the firm in the core (as opposed to the partners in the periphery). The focus of their model is on partners, not clients.

Schultze et al. [40] distinguish between "hybrid co-production" and "hybrid peer production" within OSS: the former would be the case with dual licensing or when stripped-down products are offered as OSS; the latter is the case in the traditional, straightforward services sale as an OSS business model. Their unit of analysis is the offering.

O'Mahony [41] seeks to provide a clearer distinction between "true" community-managed OSS projects and those that merely adhere to an OSI-approved license. By "hybrid (governance) models", she refers mainly to corporate-sponsored OSS projects, e.g. a firm giving its earlier proprietary code to the open domain and/or financing ongoing development of an OSS project. Yet these are OSS in the OSI sense. Her unit of analysis is the project or its development community. Participation in corporate-sponsored OSS communities has further been elaborated in [42].

Shah [14] studied certain "hybrid strategies" and "hybrid forms in OSS development" called *gated communities*. Her unit of analysis was the individual software developer. In gated communities the code is available at no cost to anyone willing to agree to the terms of use, but the sponsor is entitled to royalties for any commercial use. Her findings included:

- Participants in the gated source community were clearly driven by need as opposed to fun and enjoyment, which were the prime drivers behind participation in the pure OSS project Shah studied in parallel.
- The primary reasons cited by need-based participants for contributing code included reciprocity (obligation or desire to conform to the norms of the community), future improvements (to get feedback from others, benefit from discussions), and source code commits (future compatibility, see [30][31]).

All the above models fit under some definition of OSS. The next section describes a hybrid model that is neither traditional closed source, *nor* OSS.

3 Client-Shared Source as the Next Step

From involving partners in an inner source arrangement, it is a seemingly short step to including those clients who are capable of collaborative software development into the development community. This however has immediate implications on licensing structures and the vendor's role in the value network. Even if business models in and around OSS have been widely researched (see e.g. [18][19][20][21][22][23][24]), yet none of them caters for the kind of model that client-shared source represents.

3.1 Definition of Client-Shared Source

Here the term *client-shared source* refers to the following kind of arrangement.

- Access to source code becomes part of the object of trade at a price that is typically higher than the price of a run-time license. The client hence pays to participate in a restricted client/developer community. The original vendor most probably continues to also sell regular run-time licenses to its product.
- Suppliers (employees and outsourcing partners) provide their work input as non-client participants in the community using OSS-like processes and tools, as in inner source.
- The software vendor not only retains all rights to its own original IPR but also requires, or at least expects, that participants assign the copyright to all incremental developments back to the vendor. (A similar situation would exist in an OSS dual licensing scheme and in gated source.)
- The participants are not allowed to further license the source code to third parties.
- The software vendor itself acts as a contributing participant to the extent it sees necessary or advantageous.

No other criteria on the license agreement as to appropriation of value are implied by this definition. The license agreement may be limited to client's own internal use or to providing services (SaaS) to third parties, or it can even allow sublicensing or transfer of run-time licenses to third parties, whether under the original vendor's or the reseller's brand. Thus resellers and integrators can similarly act as participants.

A client-shared source arrangement can be seen as a special case of a "user contribution system" [1], or of a VCE [3][4].

It may be difficult to distinguish whether a participant is a client or a supplier in the traditional sense: a motivated client who is able to substantially contribute to the product development may even be paid by the vendor. This is not uncommon in the value co-production view [2]. However, the client's access to the platform should be charged a positive fee, whether fixed, recurrent or in proportion to its own usage of the product—the price can no longer be tied to a specific software version. The contributions by the same client should be then rewarded on a per-task basis, if at all.

The software vendor will thus be supported in its development activities by its clients while being able to extract more license revenue thanks to the higher value associated with a source code offering. The higher value to the client follows the same logic as presented in Subsection 2.3 above: cf. end-user innovation and participants' self-interest to contribute; lower maintenance and support cost etc. It can also be seen as a tool to instill client's trust on the vendor [38], and it naturally does away with the need to negotiate separate, potentially costly, source code escrow arrangements.

The basic model can be extended to cover e.g. joint requirements analysis and testing, which would be good examples of customer-supplier relational processes [47] (see Discussion in Section 4). This is in line with Nambisan and Baron's concept of an overarching VCE [4]. They also found out empirically that *clients' interactions* in value co-creation *can by themselves be an important source of value* and can shape the clients' future participation in such value co-creation [4].

3.2 Similarities with and Differences to Other Models

The recent focus of academic interest has shifted from value appropriation via proprietary models to pure forms of OSS. (By "pure forms" I mean the already quite heterogeneous group of OSI-approved licenses.) While OSS is often cited as a perfect example of open innovation, innovation research and information systems research have joined surprisingly little to study open client innovation, or client coproduction, in a proprietary software setting. What is new about client-shared source as defined above is that it melds proprietary and open approaches in a very pragmatic way at a time when open collaboration practices are beginning to be well known and sufficiently supported by technical collaboration infrastructure.

Similar tendencies have already been brought up under the term "OSS 2.0" [19]. Yet OSS 2.0 mainly envisions source code that is available to all under different kinds of licenses. Platform licensing [43] deals with the same themes but at a very different level and in a context where a platform can be separated from applications running on it. Fink [29] writes about "gated communities" but mainly sees them as a technical issue of access control and network infrastructure. Shah [14] acknowledges the possibility of making licensees who use the code for commercial purposes to pay for the license, but describes essentially a dual licensing scheme.

Neither should client-shared source be confused with corporate-sponsored open source projects, which fit the established OSS definition but happen to originate, and perhaps receive some support, from a commercial vendor [41][42].

Many commercial vendors also offer source code even for free in the form of software development kits (SDKs) or, particularly operating system vendors, driver development kits (DDKs). Like in the above cases, the extent of source code provided is without exception very limited and the modification and redistribution rights allowed

to third parties are granted by the original vendor solely in order to boost its own sales of the base product, or platform, to which these third parties develop modular extensions [38].

The closest equivalents to client-shared source in the current industry practice are those development licenses that software vendors grant their close partners, e.g. integrators. Unfortunately, these license agreements are usually negotiated case by case, are not based on a formalised offering against a price, and most importantly, do not capitalise on concurrent distributed development or user innovation the same way client-shared source does, even if they allowed for some user, or client, innovation.

3.3 Challenges and Limitations

Client-shared source can only be envisaged in a B2B setting where the clients have the willingness to work with the source code, either with in-house or outsourced development resources. They may have vested interests in e.g. critical operational support systems over which they need great control or, say, in resales opportunities, which again may or may not enable selling support and maintenance contracts.

While the benefits of client innovation are obvious, vendors should not expect that all the motivational factors that have led to many successful OSS projects are automatically present in the client-shared source setting [14]. For example, clients may well expect "the mundane but necessary" support tasks [44] to be provided by the original vendor.

For those clients not used to working with OSS projects, it may take a while before they realise that it is in fact in their own best interests to contribute code back to the project. In the beginning, many clients can be expected to jealousy hide their new developments. Therefore I'd suggest the vendor not require further developments to be contributed back, but rather facilitate them, since as a requirement it may come across as arrogant and yet enforcing it is very difficult.

The software vendor probably needs to take on an active role in quality assurance and resolving conflicts between participants' incompatible developments. There is a definite need for finding the right balance in the way the community is handled by the vendor [4], as participants' sense of fairness may even supersede their economic self-interest [45], as was vividly demonstrated in the classic ultimatum bargaining experiment [46].

Many of the same, today somewhat obvious, practical challenges relate to client-shared source as to releasing code as open source [30][38]. However today, the necessary infrastructure for collaborative work over the Internet, such as distributed version and configuration management tools, are no longer a technical issue.

4 Discussion

Tuli et al. [47] have examined what exactly "a solution" means to both vendors and clients. They challenge the predominant view in which a solution is a customised and integrated combination of goods and services for meeting a customer's business needs, and demonstrate how typically suppliers' and customers' views of solutions differ. According to their empirical study, customers tend to view a solution more broadly as *a set of customer-supplier relational processes*.

Interestingly, in Tuli et al.'s research "a supplier had noted importance of flexible source-code software as an enabler of effective solutions" [47]. Perhaps we should not view client-shared source as a potential end solution to the problem of declining profits or resource-consuming maintenance, but rather as an enabler of and means for truly client-serving solutions.

Client-shared source could also be a way for IT consultancies (typical examples of knowledge-intensive business services, or KIBS, firms) to capitalise on their client-specific projects and quasi-products. Whereas above we have taken the viewpoint of a software product vendor, the same change factors (see Section 2 above) are at play for the service houses in the same industry. Thus KIBS firms' clients may no longer require such strict confidentiality and the sole ownership of their bespoke information systems—especially when the systems in question are not differentiating for the clients' core business strategy.

5 Further Research on and around Client-Shared Source

The academic research on OSS so far has been preoccupied with explaining the emergence of OSS and developer motivation in particular, studying its acceptance in firms or drilling down into the OSS communities as loosely governed organisations. Lately efforts have also been made to build comprehensive taxonomies of business models around the commercial use of OSS, but what is of current interest to us is the ongoing fusion of traditional licensing of proprietary technology and pure OSS in a highly commercial setting. Matusow [38] talks about "a move to the middle" and Fitzgerald [19] has coined the term "OSS 2.0" to refer to largely the same shift. The interesting questions in information systems research are no longer *within OSS* but *around* it.

A number of research questions merit further attention:

- To what extent are companies already practicing client-shared source, without necessarily calling it that way or having formalised their strategy in these same terms? Do software companies not practicing source code licensing find the ideas interesting or applicable to their business? Are there common factors preventing companies from benefiting from client-shared source?
- Will price discrimination become more difficult when clients are openly communicating with each other and jointly developing the product?
- How are the development processes and tools affected, when clients are brought into the process and the process and tools themselves—and the whole development community—become part of the offering the clients are buying?
- Is it possible to devise such a cost-effective financial incentive and rewards mechanism that the clients would be better motivated to contribute? Such a mechanism or policy could prove valuable in the context of inner source as well.
- Is the project success related to the extent the vendor itself sets the product roadmap as opposed to letting its clients drive the product requirements? What kinds of approaches do/should vendors take to address feature prioritization and conflict resolution when faced with the question of allocating resources, many of which are not their own, to developing future features or fixing existing bugs?

– How could the vendor better quantify the efforts its clients are performing in testing and maturing the product both in their own test environments and production environments?

Bettencourt et al. [5] give a set of diagnostic questions that could be used to assess co-production readiness in KIBS. An empirical study in the context of software product offerings should also be conducted to find out how well their co-production model applies in more product-oriented client-vendor relationships. In any case, the extant literature on client co-production and value co-creation can be used by information systems researchers to explain and predict new phenomena beyond the current OSS paradigm.

In vertical domains business requirements are more complex and demand more specialized knowledge. A significant challenge is to stimulate open source development in these domains [19]. If in vertical domains OSS—even OSS 2.0—is less likely to succeed, or at least significantly lags behind when compared to the general-purpose, or horizontal, software domain, then this may imply that particularly in vertical domains we are going to see more client-shared source and other hybrid models still leaning strongly towards the traditional, closed value appropriation models.

6 Summary and Conclusions

Despite the recent focus on purely free open models, client innovation and user participation are not *a priori* incompatible with premium licensing schemes. Market pressures may well make the client-shared source model more popular among vendors whose products are suitable for this kind of collaborative development with clients and even among KIBS companies, who could thus better leverage their quasi-products.

After describing software commodification and amalgamation of products and services, the OSS offerings were discussed first as an external threat to traditional software vendors and then also as an opportunity for them to organise their internal processes (inner source), without vendors releasing their code as open source.

Over the past years user contribution, client co-production and value co-creation have been gaining momentum both in academia and manifested in practical offerings by various firms—not only software companies. A brief outlook of the different viewpoints that can also be applied in information systems science was given.

After reviewing the many ways in which "hybrid models" have been understood in the OSS context, I introduced a very concrete and practical client-shared source licensing and business model and spelled out its consequences at a level of detail and specificity that has not yet been done in the information systems literature.

This article has tried to predict the limitations of the client-shared source code model in terms of product type and company and market attributes. The challenges in the software business as outlined above are not insurmountable to many innovative companies. Most successful companies should *not* open up their source code even as widely as suggested by client-shared source. Client-shared source is a mid-tier solution for the "me too" companies, particularly in vertical domains, who do not have or foresee significant differentiators, to survive and develop a new profitable business model by incremental improvements. Client-shared source does not represent such a radical leap of faith as would be required in a transition to a pure OSS model. In the

other extreme, given enough commodification and depending on the type of product, one of the pure OSS business models or the gated source hybrid model might be a better alternative even if there was only a slim chance of success—but that is what commodification in the extreme implies.

This article contributes to academic and industry knowledge and discussion in three ways. First, it sheds light on the hybrid forms, clarifying the concepts and terminology still in flux. Second, a practical client-shared source licensing and business model was outlined. Third, directions were given for future research.

Acknowledgments. I am grateful for the financial support by the Finnish Cultural Foundation and the Foundation for Economic Education and would also like to thank all the reviewers and others who have provided their valuable comments along the way.

References

1. Cook, S.: The Contribution Revolution. Letting Volunteers Build Your Business. Harvard Business Review, 60–69 (October 2008)
2. Ramírez, R.: Value co-production: Intellectual Origins and Implications for Practice and Research. Strategic Management J. 20, 49–65 (1999)
3. Nambisan, S., Nambisan, P.: How to Profit from a Better 'Virtual Customer Environment'. MIT Sloan Management Review 49(3), 53–61 (2008)
4. Nambisan, S., Baron, R.A.: Interactions in Virtual Customer Environments: Implications for Product Support and Customer Relationship Management. J. of Interactive Marketing 21(2), 42–62 (2007)
5. Bettencourt, L.A., Ostrom, A.L., Brown, S.W., Roundtree, R.I.: Client Co-Production in Knowledge-Intensive Business Services. California Management Review 44, 100–128 (2002)
6. den Hertog, P.: Knowledge-Intensive Business Services as Co-Producers of Innovation. Int. J. of Innovation Management 4(4), 491–528 (2000)
7. Möller, K., Rajala, R., Westerlund, M.: Service Innovation Myopia? A New Recipe for Client-Provider Value Creation. California Management Review 50(3), 31–48 (2008)
8. Levy, Y., Ellis, T.J.: A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. Informing Science J. 9, 181–212 (2006)
9. Cusumano, M.: The Business of Software: How to Survive and Thrive in Good Times and Bad. The Free Press, New York (2004)
10. Nambisan, S.: Why Service Businesses are not Product Businesses. MIT Sloan Management Review, ABI/INFORM Global 42(4), 72–80 (2001)
11. Teece, D.J.: Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy. Research Policy 15, 285–305 (1986)
12. Kampas, P.J.: Shifting Cultural Gears in Technology-driven Industries. MIT Sloan Management Review, 41–48 (Winter 2003)
13. Rossi, C., Bonaccorsi, A.: Why Proft-Oriented Companies Enter the OS Field? Intrinsic vs. Extrinsic Incentives. In: Open Source Application Spaces: Fifth Workshop on Open Source Software Engineering (5-WOSSE), St Louis, MO, USA, May 17 (2005)
14. Shah, S.K.: Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. Management Science 52(7), 1000–1014 (2006)

15. Osterloh, M., Rota, S.: Open Source Software Development—Just Another Case of Collective Invention? Research Policy 36, 157–171 (2007)
16. von Hippel, E., von Krogh, G.: Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science. Org. Science 14(2), 209–223 (2003)
17. Allen, R.C.: Collective Invention. J. of Economic Behaviour and Org. 4(1), 1–24 (1983)
18. Bonaccorsi, A., Giannangeli, S., Rossi, C.: Entry strategies under competing standards: Hybrid business models in the open source software industry. Management Science 52(7), 1085–1098 (2006)
19. Fitzgerald, B.: The Transformation of Open Source Software. MIS Quarterly 30(3), 587–598 (2006)
20. Hecker, F.: Setting Up Shop. The Business of Open Source Software. IEEE Software 16(1), 45–51 (1999)
21. Karels, M.J.: Commercializing Open Source Software. In: Many Have Tried, a Few Are Succeeding, But Challenges Abound. ACM Queue, July/August 2003, pp. 46–55. ACM, New York (2003) (1542-7730/03/0700)
22. Krishnamurthy, S.: An Analysis of Open Source Business Models. In: Feller, J., Fitzgerald, B., Hissam, S.A., Lakhani, K.R. (eds.) Perspectives on Free and Open Source Software, pp. 279–296. MIT Press, Cambridge (2005)
23. Välimäki, M.: The Rise of Open Source Licensing. In: A Challenge to the Use of Intellectual Property in the Software Industry. Turre Publishing, Helsinki (2005)
24. West, J.: Value Capture and Value Networks in Open Source Vendor Strategies. In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007), pp. 176–185 (2007)
25. Thomke, S., von Hippel, E.: Customers as Innovators—a New Way to Create Value. Harvard Business Review 80(4), 74–81 (2002)
26. von Hippel, E.: Open Source Projects as Horizontal Innovation Networks—by and for Users. MIT Sloan Working Paper No. 4366–02 (2002)
27. von Hippel, E.: Democratizing Innovation: The Evolving Phenomenon of User Innovation. J. für Betriebswirtschaft 55, 63–78 (2005)
28. Harhoff, D., Henkel, J., von Hippel, E.: Profiting from Voluntary Information Spillovers: How Users Benefit by Freely Revealing Their Innovations. Research Policy 32(10), 1753–1769 (2003)
29. Fink, M.: Business and Economics of Linux and Open Source. Prentice-Hall, New Jersy (2003)
30. Mannaert, H., Ven, K.: The Use of Open Source Software Platforms by Independent Software Vendors: Issues and Opportunities. In: Open Source Application Spaces: Fifth Workshop on Open Source Software Engineering (5-WOSSE), St Louis, MO, USA, May 17. ACM, New York (2005) (1-59593-127-9)
31. Ven, K., Mannaert, H.: Challenges and Strategies in the Use of Open Source Software by Independent Software Vendors. Information and Software Tech. 50, 991–1002 (2008)
32. Lindman, J., Rossi, M., Marttiin, P.: Applying Open Source Development Practices Inside a Company. In: Proceedings of the 4th International Conference on Open Source Systems, Milan, Italy, pp. 381–387 (2008)
33. van der Linden, F., Lundell, B., Marttiin, P.: Commodification of Industrial Software. A Case for Open Source. IEEE Software (2009)
34. Dinkelacker, J., Garg, P.: Corporate Source: Applying Open Source Concepts to a Corporate Environment. In: Proceedings of 1st Workshop on Open Source Software Engineering, Toronto, May 15 (2001)

35. Sharma, S., Sugumaran, V., Rajagopalan, B.: A Framework for Creating Hybrid-Open Source Software Communities. Information Systems J. 12, 7–25 (2002)
36. Gurbani, V.K., Garvert, A., Herbsleb, J.D.: A Case Study of Open Source Tools and Practices in a Commercial Setting. In: Proceedings of the 5th Workshop on Open Source Software Engineering, St Louis, MO, USA, May 17, pp. 24–29 (2005)
37. Gurbani, V.K., Garvert, A., Herbsleb, J.D.: A case study of a corporate open source development model. In: Proceedings of the the 28th international conference on Software engineering, Shanghai, China, pp. 472–481 (2006)
38. Matusow, J.: The Shared Source Initiative: The Microsoft Perspective. In: Feller, J., Fitzgerald, B., Hissam, S.A., Lakhani, K.R. (eds.) Perspectives on Free and Open Source Software, pp. 329–346. MIT Press, Cambridge (2005)
39. Fox, P., Wareham, J., Giner, J.L.C.: Value Appropriation in Hybrid Technology Ecosystems. In: 29th Int. Conference on Information Systems, Paris, France (2008)
40. Schultze, U., Prandelli, E., Salonen, P.I., van Alstyne, M.: Internet-enabled Co-Production: Partnering or Competing with Customers? Comm. of AIS 19, 294–324 (2007)
41. O'Mahony, S.: The Governance of Open Source Initiatives: What Does It Mean To Be Community Managed? J. of Management Governance 11, 139–1150 (2007)
42. West, J., O'Mahony, S.: The Role of Participation Architecture in Growing Sponsored Open Source Communities. Industry & Innovation 15(2), 145–168 (2008)
43. West, J.: How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies. Research Policy 32, 1259–1285 (2003)
44. Lakhani, K.R., von Hippel, E.: How Open Source Software Works: 'Free' User-to-user Assistance. Research Policy 32, 923–943 (2003)
45. Kahneman, D., Knetsch, J.L., Thaler, R.: Fairness as a Constraint on Profit Seeking: Entitlements in the Market. The American Economic Review 76(4), 728–741 (1986)
46. Güth, W., Schmittberger, R., Schwartze, B.: An experimental analysis of ultimatum bargaining. J. of Economic Behavior & Organization 3(4), 367–388 (1982)
47. Tuli, K.R., Kohli, A.K., Bharadwaj, S.G.: Rethinking Customer Solutions: From Product Bundles to Relational Processes. J. of Marketing 71, 1–17 (2007)

Managerial Growth Challenges in Small Software Firms: A Multiple-Case Study of Growth-Oriented Enterprises

Oskari Miettinen, Oleksiy Mazhelis, and Eetu Luoma

Department of CS and IS, University of Jyväskylä, Jyväskylä, Finland
{oskari.miettinen,oleksiy.ju.mazhelis,eetu.luoma}@jyu.fi

Abstract. Notwithstanding the importance of small software firms, research focusing on understanding the growth challenges faced by their management is close to non-existent. By taking a life-cycle approach and focusing on managerial challenges, the aim of this paper is to analyze the growth process of small software firms. After forming a synthesis of possible growth challenges from relevant literature, the results are reflected upon case software firms using thematic interviews and questionnaires. According to the analysis of four software firms, managing human resources represents the greatest challenge for software services firms. Additionally, other challenges stem from competition and sales-related activities. Comparative analysis of the literature and the cases leads toward a theoretical conceptualization of growth challenges from small software firm's perspective.

Keywords: small firms, software business, growth of a firm, life-cycle model, growth challenges.

1 Introduction

In addition to the increasing impact on job and wealth creation in economies around the world, small software firms provide products and services that enable growth of other industries [1]. The majority of firms will never find any significant growth path, though, but instead live and eventually cease to exist as small businesses [2, 3], and software firms are not an exception. The firms that have achieved high growth rates, however, are increasingly more often IT or software companies [4].

Software is developed both in the primary and the secondary software industry [5]. The *primary software industry*, constituting of the actual software firms, can be divided into two business segments: software services and software products [1, 6]. Besides, a significant amount of software is developed in the *secondary software industry,* i.e.*,* in vertical industry enterprises such as telecom operators and banks [5].

As Birley & Westhead [7] suggest, research should focus on specific firm segments, rather than trying to find a general theory to explain the growth of all firms. Although the growth and related managerial challenges of small and medium-sized firms have been researched intensively during the past decades [2, 3, 7-13], similar research focusing on small software firms—especially in the services business segment—has been close to nonexistent in the scholarship. Additionally, small firms are

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 26–37, 2010.

often not differentiated from medium-sized enterprises in studies, even though they do not share all the same characteristics [14]. Furthermore, many studies have focused on growth of young technology-intensive firms [15-20]. Even though the software industry can be seen as a subset of the technology industry, there are specific characteristics that make it unique [21], and hence, should be studied in separation. Specifically, as opposed to the technology industry in general, the core products and services of software industry are intangible. In all, we argue that none of the existing growth models and theories fully describe the management challenges small software services firms are facing during their life cycles, and that more research is required to better understand the specific managerial challenges they are confronted with.

Thus, the aim of this paper is to analyze the growth of small software firms, focusing especially on managerial challenges. The aim is pursued by forming a synthesis of possible growth challenges from relevant literature and by reflecting the results of the literature review upon case software firms. This reflection is based on thematic interviews and questionnaires conducted in four growth-oriented Finnish software firms. According to the analysis of the cases, managing human resources represents the greatest challenge for software services firms. Additionally, other challenges stem from competition and sales-related activities. Comparative analysis of the literature and the cases leads toward a new theoretical conceptualization of growth challenges especially from small software firm's perspective, whereby these challenges are projected on different life-cycle stages of the firms.

The paper proceeds as follows. The next section gives an overview of the literature concerning software business, and firm growth research. Additionally, some previous growth models relating either to small firms or software firms are introduced, which is followed by a synthesis of growth challenges identified in the literature. Next, a short introduction of the case firms and the results of the case analysis are presented. Finally, in the conclusions, the implications of the study are discussed.

2 Growth of a Software Firm and Related Challenges

Growth of a firm is a multidimensional concept [22], combining economical, social and cultural factors [23]. Further, the economic factors include various views to the research of growth, such as entrepreneurship, finance, and management. Davidsson & Wiklund describe entrepreneurial research as a multiple-level analysis, as entrepreneurship occurs and has effect on different societal levels [24]. Some scholars have searched for a universal explanatory model to explain firm growth. This kind of a model has not been found though, and Autio *et al.* [23] argue, one can never be found. One probable reason is that "[t]here is no such thing as a typical growth firm. Rather, there are many different types of growth firms with different growth patterns" [22]. The lack of a universal model causes great diverseness among research methods and settings used by researchers [25]. Thus, researchers have to be able to analyze new research data by combining different kinds of research methods [23]. Altogether, the diverseness and multitudes of research levels cause challenges in forming a coherent view of firm growth by reviewing existing literature and studies on the subject [2].

Anyhow, the growth of a firm can be observed from the *growth process* view, which indicates growth to progress through different stages separated by

organizational crises and changes [2], and from the *determinants of growth* view, which is based on identifying factors affecting the growth including motivation, strategy, resources, firm external opportunities, characteristics, the educational background and business experience of firm founders, social capital, and financing [26]. Furthermore, according to O'Farrell & Hitchens [27], four different paradigms of firm growth research are evident in the scholarship: (1) the stage model or the life-cycle model, (2) the strategic management approach, (3) the stochastic model, and (4) the industrial economics approach. The former two address the factors that affect firm growth internally, while the latter two approach firm growth mainly through external factors such as the market and the industry.

Life-cycle models generally describe a firm's growth as a predictable progress through certain evolutionary stages on which varying crises and management challenges are expected [27]. Life-cycle models serve in understanding and conceptualizing the complexity of a firm growth process. The criticism toward this approach rises from inflexibilities and unidimensionality: all the firms are usually expected to go through the same stages in certain order [28], although there are exceptions. Many life-cycle models from small firm [8, 9], technology-intensive firm [15, 17, 29], and even software firm [21, 30] perspectives have been suggested, but none focuses on small software firms, not to mention the software services segment.

Strategic management approach has focus on business strategies needed by the entrepreneurs and managers in order to maintain continuous growth [21]. Another approach to firm growth is the *resource-based view*. According to Penrose [31], one important reason for growth of a firm is its underutilized resources. Firms have a natural need to eliminate idle workforce by engaging in large operations, and at the same time, "to use the most valuable specialized services of its resources as fully as possible" [31]. Thus, firms need to grow and elevate their operations in order to take full advantage of their highly specialized workforce. The latter is especially true in case of small software firms, wherein highly specialized employees cannot necessarily utilize all their know-how efficiently because the output of the firm is too small. Penrose was also the originator of the *knowledge-based theory of the firm* [31], which states that knowledge is strategically the most important resource for a firm.

Eventually, these paradigms are purely theoretical classifications of types of firm growth studies, and hence many approaches like internationalization are likely to overlap. In the context of this paper, the objective is to focus both on the growth process by taking the life-cycle approach, and on the determining factors that internally affect the growth of small software firms through the strategic management approach. Internationalization is not taken into consideration due to the fact that it is usually relevant only for larger firms or on the later stages of the firm evolutionary cycle.

Delmar *et al.* [22] believe, that conflicting theories on the causes of firm growth are born, when an explanation for *why* firms grow is searched without actual knowledge of *how* firms grow. It is important to understand, that firms can grow in multiple ways: organically, through acquisitions (or mergers), or as in many cases, by a combination of both. At societal level, organic growth creates new jobs, in contrast to acquisitions, where existing jobs are transferred to another organization [22]. Small and young firms in emerging industries are more likely to grow organically due to their lack of resources for acquisitions [22]. Therefore, as opposed to growth of large, established firms, small firm growth creates more new jobs.

Although a multitude of ways to measure firm growth is present in the scholarship, most commonly used are sales, employment, and market share [32]. Sales growth and employment growth are used in the present study as a combination, because, although sales is a good indicator of how customers are increasingly accepting the firm's products or services [32], it does not always lead the growth process [22]: In the beginning of a software firm life cycle there might not exist ready product or service, yet [32].

2.1 Software Business

The importance of knowledge as a resource, as Penrose already suggested in 1959 [31], has become clear in today's world as many industries are becoming increasingly more dependent on and driven by it. Therefore, management challenges faced in other industries are becoming similar to the software industry [1]. Hence, the value of software industry for firm growth research is exceptional since the findings are likely to be applicable to other knowledge-intensive industries as well [21].

Software is developed both in the primary and the secondary software industry [5]. The *primary software industry* can be divided into two main business segments: *software services* and *software products* [1]. However, the distinction between software services and product businesses is not always clear, and there is evidence of more and more software firms, especially on the later stages of their life cycles, straddling both sectors [6, 33], which could also lead to many managerial challenges. According to Hoch *et al.* [1], the dynamics of software product and services businesses differ in many aspects. One of the differentiating factors between the two is the effects of marginal costs: for services firms they are almost constant while in case of product businesses the marginal costs approach zero. Furthermore, the market structure is in general more fragmented in the services segment. Pure software product business is additionally characterized by low variable costs, meaning virtually all the cost of developing software is fixed in the design and implementation of it; many copies need to be sold in order to cover the fixed costs. With regards to the growth strategies and the required management mindset, firms offering mostly services are likely more interested in their capacity utilization rate than market share, which is more important for product firms. Human resources, software development, strategy, and marketing and sales are all important management areas for both, but the level of relevance varies. [1]. According to Cusumano [6], a pure software product business is analogous to book publishing and selling sequels of bestsellers, whereas for services business, like in banking, the importance is in long-term customer relationships and recurring fees. Software products firms seek to take advantage of possible economies of scale, whereas economies of scope are more important for services businesses. [6]. Competing with either software product or software services thus requires managerial skills in marketing and sales to adopt divergent business logics.

One of the major barriers to growth in the labor-intensive software industry is the low number of available professionals, which in turn makes software managers' work more challenging [1]. The effect of this barrier varies, of course, depending on the country, area, industry sector, etc. At least major software countries such as USA and India have been long suffering severely from workforce shortages [1]. According to the authors, the reasons for the lack of talent range from increasing demand, i.e. fast growth of the industry, and competition between firms.

2.2 Growth Challenges in Finnish Software Firms

Software industry can be considered both on a regional and a global scale. Although there were seven Finnish software firms among the top 100 European software vendors in 2008 [34], Finnish software firms are very small on the average; 45 % of them have fewer than 5 employees [35]. Varying approximations of the number of software firms in Finland exists, due to the Statistics Finland's way of categorizing firms inside the IT sector. According to Ali-Yrkkö and Martikainen, software firms represent around two thirds of the IT industry in Finland, which results in around 33,000 employees altogether [36].

As a result of workgroup efforts by Growth Forum 08 several software firm growth challenges were identified and prioritized in the context of Finnish software industry [35]. The challenges are grouped to industrial, national and global challenges. The most important industrial challenges include sales and marketing, small firm size, low knowledge level of the market and customers, and difficulty of forming a growth strategy. The most important national challenges are non-supportive climate towards entrepreneurship, small size of capital market, low level of willingness to take risks, and low ability to take risks.

According to research conducted by Harju [37] there are four challenges for small Finnish software companies that rise above the others: (1) Funding, (2) how to get the right people to the company, (3) competition, and (4) rapidly changing technologies. "How to get the right people to the company" is related to the topic of "low number of available professionals" discussed earlier. High knowledge intensity and labor-intensity of the industry cause individuals to become the most important assets for a software firm, and one of the most important challenges for managers at the same time. Harju's notion of competition being one of the biggest challenges is also supported by the earlier discussions of the characteristics of the industry. Alajoutsijärvi, Mannermaa & Tikkanen attempt to identify the most important marketing challenges

Table 1. Synthesis of possible software firm growth challenges identified in the literature

Area	Growth challenges	References
Environment	Non-supportive climate towards entrepreneurship	[35]
Human	Human resource management	[1]
resources	Workforce shortages / labor supply / recruiting	[1, 27, 37]
Marketing	Knowledge of the market and customers	[35]
and sales	Managing different business logics	[38]
	Sales and marketing skills	[35]
Networking	Small firm size	[35]
Personal	(Growth) motivation	[27, 39]
	Risk taking willingness / ability	[35]
Strategy	Competition	[21, 37]
	Funding / financing	[35, 37]
	Forming a growth strategy	[35]
	Simultaneous management of product and services businesses	[1, 6, 38]
Software	Rapidly changing technologies	[21, 37]
development	Software development	[1]

for small software firms. The authors argue that the most critical challenge for the management is in balancing between entering new business domains, which require differing business logic (e.g. moving from services business towards product business) and maintaining the traditional business operations [38].

As a summary, the most important possible challenges for small Finnish software firms found in the literature are synthesized in Table 1. These challenges are grouped into 7 management areas, partially according to Hoch *et al.* [1] (adding environment, networking, and personal). The table is in alphabetical order (no prioritization).

3 Empirical Research

The challenges in Table 1 have been empirically tested, and a synthesizing model is built based on the literature research and interviews. The objectives of the empirical research were set to list growth challenges small Finnish software firms are facing, to determine which challenges are typical or dominant on certain growth stages, and to learn from entrepreneurs' and executives' attitudes, opinions, and views to growth.

The present study combines both quantitative and qualitative research methods. The qualitative part consists of four thematic interviews and their analyses, as well as of some information collected from the case firm web sites. Thematic interview is conducted as a semi-structured discussion with no detailed questions; the interview is guided only by pre-defined themes [40]. The quantitative data comes from the questionnaire conducted for the interviewees. Its purpose was to collect data prior to interviews that do not necessarily require interview as a method. The quantitative data is used to compare facts and figures of the case firms as can be seen from Table 2. The main business model is based on a classification by Rönkkö & Mutanen [26]. Sales and profit growth refer to relative growth compared to the previous year.

Table 2. Facts and figures of the case firms from year 2008 (β's sales and profit from 2007)

	Alpha (α)	**Beta (β)**	**Gamma (γ)**	**Delta (δ)**
Year of foundation	1995	1997	1987	2005
Employees [growth in 1 year]	13 [+2]	69 [+10]	53 [+5]	26 [+5]
Sales (M€) [growth/decline]	1,4 [+10 %]	4,2 [-20%]	8,7 [+16,8 %]	2,07 [+44 %]
Profit (M€) [growth]	0,4 [+55 %]	1,0 [-]	2,4 [+30 %]	0,15 [+50 %]
Customers [growth in 1 year]	~2000 [+150]	dozens [some]	16 [-]	35 [+10]
Software business segment	Products	Services	Services	Services
Main business model	Standardized product	SW devel. services	SW devel. services	SW devel. services

The case firms were selected by randomly contacting small (or medium) –sized and growing Finnish software business organizations. To determine whether a specific firm had been growing constantly, using publicly available online information sources, the sales growth rates of the firms were studied. The case firms represent rather a heterogeneous sample of small Finnish software firms, as both younger and more experienced firms, smaller and medium-sized firms, and software product and services firms are present. This can been seen mainly as an advantage for the study

because of a better coverage of the industry. The firms that have recently passed the 50-person milestone and become medium-sized (β, γ) are important for the study due to their experience of the small-firm life cycle as a whole. All the case firms are private companies (limited by shares) and have achieved good growth rates either in employment or in sales at least during the past three years. One interview of 42-76 minutes was conducted per case firm. All the interviewees are executives with significant ownership over the firm. The interviewees have been with the respective case firms from the beginning, excluding Interviewee α who joined in 2001. The citations have been translated to English from the Finnish transcript.

3.1 Interview Analysis

The interviews were analyzed by utilizing thematic analysis principles [41]. Table 3 summarizes the interview analysis results. The table includes only those challenges identified and discussed during the interviews. Hence, some of the possible challenges introduced earlier are absent. For each challenge, its importance, as perceived by the interviewee, is indicated. A "-" denotes negative support, "0" means there was no opinion or discussion, or that the challenge does not relate to the case firm, and a "+" is a sign of the interviewee agreeing that the challenge has been essential considering the case firm's growth.

Table 3. Summary of the case firm challenge analysis

Managerial growth challenge	α	β	γ	δ
Competition	+	+	0	+
Education	-	-	-	-
Evolving organization	0	+	+	+
Funding and financing	-	-	-	-
Human resource acquisition and management	+	+	+	+
Motivation	+	+	+	+
Networking	0	0	0	0
Risk taking willingness / capability	+	+	+	+
Sales and marketing	+	+	+	+
Taxation / legislation	-	-	-	-

As could be seen from the table, the most relevant growth challenges for the case firms were related to competition, evolving organization, human resources, motivation, risk taking, and sales and marketing. Neither education, nor taxation and legislation issues were seen as relevant challenges. As opposed to Harju [37], funding and financing received only negative support, meaning the case firms reported not having had any related significant problems. By education, it was discussed whether the national education system produces sufficiently knowledgeable human resources.

Among the interviewees, high motivation is seen as a necessity for growth. Being very small in size is in itself a great motivation to grow bigger, because cooperation, e.g. with potential customers, is expected to become less challenging, and because being bigger reduces the risk of total failure. In addition to money and success, entrepreneur-managers are driven by new challenges and possibilities for personal

development. Thus, the growth becomes a self-feeding process, as Interviewee δ summarizes: "The bigger the firm, tougher the challenges, higher the motivation."

As discussed earlier, software industry operations are very labor-intensive; costs come mainly from labor. Therefore, it is not surprising all the interviewees see personnel as the most important resource for a software firm. According to Hoch *et al.*, the most important managerial challenges for software service firms stem from human resources. The data gathered support this observation, as all the interviewees see both managing and acquiring human resources as major challenges for growth.

Sales and marketing was an interview theme resulting in a wide range of views and opinions. Both Interviewee α and δ mentioned the difficulty of recruiting able salespersons. Additionally, selling something intangible is a challenge for firms offering software developmental services. As a small business, it is difficult to sell ideas when there are no references or successful customer cases to tell about (β, δ). According to Alpha, "sales is just sales after all... no matter whether copy paper or Internet-applications are sold", although it is agreed—also by Interviewee β and γ—that technically oriented personnel usually lack sales and marketing skills.

Although all the case firms have had a history of constant growth, a consensus seems to exist among the interviewees of the fact that it would have been possible to grow faster if more aggressive growth strategy would have been utilized. Some of the interviewees thought they have probably been even too cautious and unwilling to take unnecessarily high risks. Interviewee δ summarizes the most important reason for keeping the risk level as low as possible: "We don't want to cause this highly professional team to lose their jobs by taking too high risks." This is in line with the results from the study by Wiklund *et al.*, wherein well being of employees was listed as a primary reason for small business managers to avoid taking too high risks and even affecting their willingness to grow the firm [39].

3.2 Mapping Growth Challenges on Life-Cycle Stages of Software Firms

Based on the available literature and conducted empirical study, the most important managerial growth challenges for small software firms have been synthesized and mapped to a theoretical life cycle of small software firms (Fig. 1). The conceptualization includes five life cycle stages each corresponding to a firm of different size (number of employees): seed, start-up, growth I, growth II, and growth III. Both managerial challenges specific only to software firms as well as general managerial challenges related to virtually all start-ups are considered.

The *seed stage* starts before the firm is founded. Challenges related to this stage include refining the business concept, finding a suitable team, and gathering capital. Especially service business firms might find it challenging to acquire the first customer without any references or history of previous projects. Indeed, finding the first customer is such a growth boundary that the firm will probably never move to the next stage if one is not found. Software product firms might move to the next stage and start the business without a ready product, and thus without customers.

For software firms, the *start-up stage* will still most likely evolve around acquiring or dealing with the first project or product. Sales for software product firms and customer acquisition for services firms are vital in order to start covering sunk costs and make the business operations profitable. Software product companies might still rely on developing their product(s), and hence might not have acquired any customers, yet.

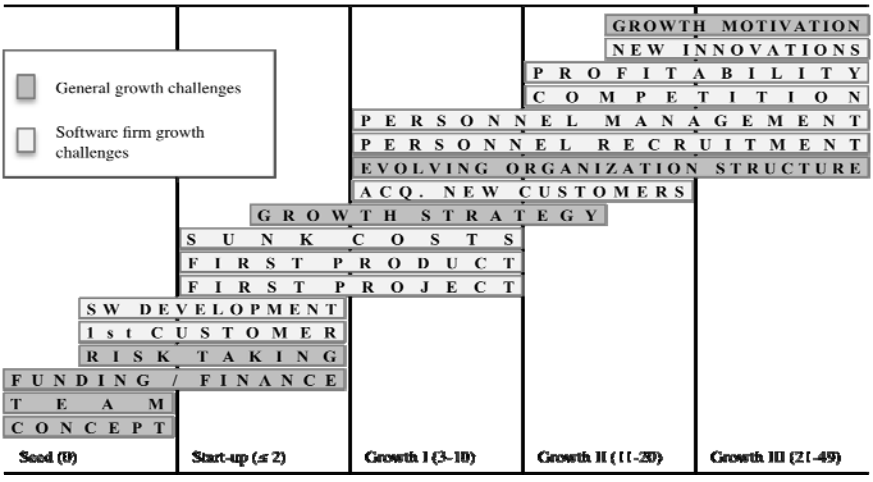

Fig. 1. Managerial growth challenge conceptualization for small software firms

The *growth I stage* for software product firms means ready product(s) and some materialized sales for the most of firms. The biggest challenges for them are likely related to sales, software development, and human resource acquisition and management. Firms offering software services are likely to struggle with acquiring new customers and/or projects after the initial one(s). A lot is depending on the success of the first project; the future of the company lies in closing new deals. Human resources are vital for software services firms, and thus, one of the major challenges for them is acquiring skilled personnel and keeping the existing ones as well. This stage often marks a point where a software firm needs to start making decisions about a growth strategy to be utilized. Whether aggressive or moderate growth is sought depends on the market situation but also on the willingness of the owners to take risks.

The *growth II stage* is a big milestone for a software firm as it has managed to grow beyond 10 employees. The organizational structure starts to take new forms and it might be challenging for the owners to share responsibility. Human resources remain most likely as the major challenges for software companies, and recruiting might become even more challenging because more specialized workforce such as project managers and sales representatives are needed. Competition might become a new big challenge for a software firm, if it is competing on a narrow market segment.

Challenges in the final *growth III stage*, are similar to the previous stage, but they can become more intense. Although it is true that acquiring new customers and employees might be easier than before, because of the references and experience the firm has touted, organizational changes and increasing need for specialized workforce pose new challenges. The growing number of employees makes managing personnel and their skills even more challenging than before. On this stage, competition might become more severe; again, depending on the market segment or the industry sector the firm is operating in. Coming up with new innovations and products might prove challenging, and some setbacks are most likely to occur.

4 Conclusions

One important theoretical finding of this paper is that the theory on managing human resources being the most important managerial challenge for software firms especially in services business receives strong empirical support. This finding is not surprising, however, as some of the major differentiating characteristics of the software industry are its high level of knowledge- and labor-intensity. This finding implicates the current theories on software firm growth challenges seem to be mainly in line with the actual managerial challenges in the Finnish software industry.

Another interesting finding is that financing or acquiring capital has never really been a significant challenge for the case firms, even though they all have grown reasonably fast. All the case firms have been able to sustain their growth through internal financing, and thus, there are no external investors involved. It should be noted, however, this does not necessarily implicate that small software firms in Finland would not have problems with acquiring growth financing. In fact, unchallenging financing might be one of the possible reasons for the firms' success in the first place.

The findings from the interview analysis further implicate a tendency of competition becoming tougher in the Finnish software industry. The current situation of the economy is likely to be one of the causes for this development. It is not clear, however, whether the finding is due to competition actually becoming more radical, or due to the growth the case firms have achieved and thus found themselves fighting for a bigger market share than before. Anyhow, depending on the market positioning of a firm, competition—especially for younger software firms—is a significant challenge.

Sales, especially for younger software firms, cause many challenges. For software services firms, this is mainly due to the tangible nature of the offered services. Selling ideas is extremely difficult without any references to previous success stories. Additionally, software firms are often founded by technically oriented teams, which lack experience in sales and marketing. Hence, if the founding team of a software firm has both technical and business oriented people the future growth of the firm looks more promising, when compared to a software firm managed only by one type of the two.

Risk taking is often a popular topic when discussing firm growth. Although undoubtedly any entrepreneur establishing a new firm has to take personal financial risks to some extent, it seems this fact is too often overemphasized. The findings from the interviews implicate that businesses can be lead to steady growth without taking significant financial risks or outside funding. Further, the theory of low willingness to take risks is also empirically supported by this study to a large extent; entrepreneur-managers are very concerned of the well being of their employees and do not thus want to risk it all. Whether or not this is typical behavior for Finnish software entrepreneurs would require a study of its own. All in all, the analysis implicates that even the most successful Finnish software entrepreneurs are not adept risk-takers.

References

1. Hoch, D.J., Roeding, C.R., Purkert, G., Kindert, S.K., Muller, R.: Secrets of Software Success: Management Insights from 100 Software Firms Around the World. Harvard Business School Press, Boston (2000)
2. Davidsson, P., Achtenhagen, L., Naldi, L.: Research on Small Firm Growth: A Review. Paper Presented at the 35th EISB Conference, Barcelona, Spain (2005)

3. Storey, D.J.: Understanding the Small Business Sector. Routledge, London (1994)
4. Deschryvere, M.: High growth firms and job creation in Finland. The research Institute of the Finnish Economy, Helsinki (2008)
5. Tyrväinen, P., Warsta, J., Seppänen, V.: Evolution of Secondary Software Businesses: Understanding Industry Dynamics. In: León, G., Bernardos, A.M., Casar, J.R., Kautz, K., DeGross, J.I. (eds.) Open IT-Based Innovation: Moving Towards Cooperative IT Transfer and Knowledge Diffusion. International Federation for Information Processing, pp. 381–401. Springer, Boston (2008)
6. Cusumano, M.A.: The Business of Software: What Every Manager, Programmer and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad. Free Press, New York (2004)
7. Birley, S., Westhead, P.: Growth and Performance Contrasts between 'Types' of Small Firms. Strategic Management Journal 11(7), 535–557 (1990)
8. Churchill, N.C., Lewis, V.L.: The five stages of small business growth. Harvard Business Review 61(3), 30–50 (1983)
9. Scott, M., Bruce, R.: Five Stages of Growth in Small Business. Long Range Planning 20(3), 45–52 (1987)
10. Moreno, A.M., Casillas, J.C.: Entrepreneurial Orientation and Growth of SMEs: A Causal Model. Entrepreneurship: Theory & Practice 32(3), 507–528 (2008)
11. Terpstra, D.E., Olson, P.D.: Entrepreneurial Start-up and Growth: A Classification of Problems. Entrepreneurship: Theory & Practice 17(3), 5–19 (1993)
12. Wiklund, J.: Small Firm Growth and Performance: Entrepreneurship and Beyond. Doctoral Dissertation. Jönköping International Business School, Jönköping (1998)
13. Wiklund, J., Shepherd, D.: Knowledge-based resources, entrepreneurial orientation and the performance of small and medium-sized businesses. Strategic Management Journal 24(13), 1307–1314 (2003)
14. Richardson, I., Wangenheim, C.G.V.: Guest Editors' Introduction: Why are Small Software Organizations Different? IEEE Software 24(1), 18–22 (2007)
15. Yli-Renko, H., Autio, E.: The Network Embeddedness of New, Technology-Based Firms: Developing A Systemic Evolution Model. Small Business Economics 11(3), 253–267 (1998)
16. Hanks, S.H., Watson, C.J., Jansen, E., Chandler, G.N.: Tightening the Life-Cycle Construct: A Taxonomic Study of Growth Stage Configurations in High-Technology Organizations. Entrepreneurship: Theory & Practice 18(2), 5–30 (1993)
17. Kazanjian, R.K., Drazin, R.: A stage-contingent model of design and growth for technology based new ventures. Journal of Business Venturing 5(3), 137–150 (1990)
18. Kazanjian, R.K.: Relation of Dominant Problems to Stages of Growth in Technology-Based New Ventures. The Academy of Management Journal 31(2), 257–279 (1988)
19. Laukkanen, S.: Uuden teknologiayrityksen kasvu: toimintatutkimus (The growth of a new technology-based venture: An action-research). TTKK, Tampere (2000)
20. Salonen, A.: International growth of young technology-based Finnish companies. Finnish Academy of Technology, Helsinki (1995)
21. Nambisan, S.: Software firm evolution and innovation-orientation. Journal of Engineering and Technology Management 19(2), 141–165 (2002)
22. Delmar, F., Davidsson, P., Gartner, W.B.: Arriving at the high-growth firm. Journal of Business Venturing 18(2), 189–216 (2003)
23. Autio, E., Miikkulainen, K., Sihvola, I.: Innovatiiviset kasvuyritykset (Innovative growth ventures) (2007), http://www.tekes.fi/julkaisut/innovatiiviset_kasvuyritykset.pdf

24. Davidsson, P., Wiklund, J.: Levels of Analysis in Entrepreneurship Research: Current Research Practice and Suggestions for the Future. Entrepreneurship: Theory & Practice 25(4), 81–100 (2001)
25. Mutanen, O., Rönkkö, M.: Growth Challenges of Small Finnish Software Firms - Comparing Theory and Practice. Paper Presented at the EBRF Conference, Helsinki and Stockholm, Finland and Sweden (2008)
26. Rönkkö, M., Mutanen, O.: National Software Industry Survey 2008: The Finnish Software Industry in 2007. Helsinki University of Technology, Espoo (2008)
27. O'Farrell, P.N., Hitchens, D.M.W.N.: Alternative theories of small-firm growth: a critical review. Environment and Planning A 20(10), 1365–1383 (1988)
28. O'Gorman, C.: The sustainability of growth in small- and medium-sized enterprises. International Journal of Entrepreneurial Behaviour & Research 7(2), 60–75 (2001)
29. Galbraith, J.: The Stages of Growth. Journal of Business Strategy 3(1), 70–79 (1982)
30. McHugh, P.: Making it Big in Software: A guide to success for software vendors with growth ambitions. Rubic, Tiverton (1999)
31. Penrose, E.T.: The Theory of the Growth of the Firm. Oxford University Press, Oxford (1959)
32. Gilbert, B.A., McDougall, P.P., Audretsch, D.B.: New Venture Growth: A Review and Extension. Journal of Management 32(6), 926–950 (2006)
33. Nambisan, S.: Why Service Businesses Are Not Product Businesses. MIT Sloan Management Review 42(4), 72–80 (2001)
34. Truffle Capital: Ranking of the top 100 European software vendors (2008)
35. Kontio, R.M., Mutanen, O., Ahokas, M., Junna, O., Ali-Yrkkö, J., Touru, A.: Kasvufoorumi 08 loppuraportti (Growth Forum 08 Final Report). The Finnish Software Entrepreneurs Association and Microsoft (2008)
36. Ali-Yrkkö, J., Martikainen, O.: The software industry in Finland. The Research Institute of Finnish Economy (2008)
37. Harju, P.: Kilpailukyvyn tekijät pienissä suomalaisissa ohjelmistoyrityksissä (2008) (The determinants of competitiveness in small Finnish software firms), http://www.tekes.fi/julkaisut/KIBS.pdf
38. Alajoutsijärvi, K., Mannermaa, K., Tikkanen, H.: Customer relationships and the small software firm: A framework for understanding challenges faced in marketing. Information & Management 37(3), 153–159 (2000)
39. Wiklund, J., Davidsson, P., Delmar, F.: What Do They Think and Feel about Growth? An Expectancy-Value Approach to Small Business Managers' Attitudes Toward Growth. Entrepreneurship Theory and Practice 27(3), 247–270 (2003)
40. Hirsjärvi, S., Hurme, H.: Tutkimushaastattelu: teemahaastattelun teoria ja käytäntö (The research interview: Theory and practice of thematic interview). Yliopistopaino, Helsinki (2000)
41. Aronson, J.: A Pragmatic View of Thematic Analysis. The Qualitative Report 2(1) (1994)

Internationalization of Software Firms

Mikko Rönkkö and Juhana Peltonen

Aalto University
{mikko.ronkko,juhana.peltonen}@tkk.fi

Abstract. This paper presents the results of a large sample survey of internationalizing firms in the Finnish software industry. We analyze the data descriptively with plots and tabulations and as more analytically with regression analyses. The results support the conclusion that internationalization can be considered as a natural stage in the firm life-cycle, but patterns of internationalization differ across firms. Considering the current theorizing of software firms as prototypical international new ventures, we find it surprising that many firms seem to choose to internationalize only a little and gradually.

Keywords: Software firm, internationalization.

1 Introduction

For a country with small home markets for software, internationalization is often considered to be a natural step in the lifecycle of a software firm. Since internationalization is considered particularly important for software firms operating in small home markets [cf., 1], it has received a fair share of attention in the academic circles [2-9]. Moreover, software firms are often used as a source for empirical data in more general internationalization studies [10-13], and as a source of inspiration for some of the most notable non-empirical theorizing in international business [14,15].

Software industry is in many ways an atypical industry considering international expansion. First, companies using the Internet as the main distribution channel can have an instant access to global markets. A similar channel cannot be easily utilized in many other industries. Second, many software companies get their first international sales before domestic sales. This is particularly the case with providers of specialized systems and applications operating in business to business markets. Hence it is no surprise that software firms are seen as intriguing cases and good sources for theoretical sampling [16] of extreme cases.

Nonetheless, a large share of the current internationalization studies are qualitative and thus limit the maturing of the area of inquiry if we assume that it should increasingly follow the prevalent paradigm in management where larger samples are used to test theories developed with case study research [17]. Due to this we decided to contribute to this stream of research by quantitative analysis of the Finnish software industry focusing on patterns and determinants of internationalization.

One of the key critiques of current mainstream management papers is lack of practical relevance. The reason for this is two-fold. First, as more and more studies are

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 38–48, 2010.

conducted around similar topics, the focus of an individual study necessarily becomes narrower. Second, the advances in statistical methods enable researchers to tackle even more subtle issues in the data. This conflict between rigor and relevance is present in both the information systems [18] and more general management research [19,20]. Due to the multidisciplinary nature of the software business researcher community, we decided to adopt a more general approach on the phenomenon rather than focusing on one of the leading edge niches in internationalization theories.

Due to our broad approach on internationalization, we use somewhat unconventional approach for the structure of this paper: Instead of focusing on theories first and pointing out gaps and filling them with hypotheses that we test, we instead first describe our data and methods. Then, we continue the paper by presenting the results and comparing these to the existing studies.

2 Research Design and Sample

The present paper uses empirical data collected from the Finnish software industry over the years 2001-2009 [21]. The data is analyzed with descriptive graphs and cross tabulations as well as more analytical statistics to uncover how the firms in this market expand internationally.

The details of the sampling frame have evolved over the years to match the changing needs of the survey. NACE code 72.21. ("Publication of software") and 7222 ("Other software consultancy and supply") have typically been included. To cover the entire software industry, also including firms officially registered under other industry codes, we included member lists of several industry organizations. This approach was adopted because the Finnish trade register covers the software industry only partially, as some companies have software as a secondary industry. In addition, the survey project during which the data were collected required that the entire industry is covered [21]. Typically the sampling frame covered all firms with five or more people using the trade register data and smaller enterprises only if they were members of some industry association or had registered on any of the lists covering the software industry. These are e.g. lists of software companies that had applied for product development subsidies from public organizations. However, the coverage of smallest firms varies between years. Moreover, missing revenue data was obtained later from Asiakastieto Ltd., which collects and organizes information from the publicly accessible Finnish trade register.

2.1 Data Collection and Analysis

Data collection for the survey takes place during the late spring and summer using paper and web-based questionnaire loosely following the tailored design approach [22]. For the year 2009 the invitation to participate was sent to 4544 mainly small and medium-sized companies. Since one of the goals of the larger project was to cover the entire industry, this figure represents a significant amount of oversampling to ensure inclusion of all relevant firms. In all, we estimated that the figure contains approximately 30% of firms that are either not active or do not operate in the software industry. We obtained a total of 584 complete responses using this approach representing a

response rate of approximately 20%, which is typical for this survey. Micro-enterprises produced the most non-response, so the effective response rate for the firms that have any meaningful international activities was greater.

The paper and web-based survey forms included questions about internationalization status, revenue, share of international revenue of total revenue, target markets, and business models. The exact variables are presented later when we discuss the study results.

Data analysis was carried out using Intercooled Stata, version 10.1. After data preparation calculating the values for study variables based on the survey responses the data were analyzed using OLS regression analysis.

3 Results

Our results from the internationalization analyses are divided under three sub sections. In the first section we will discuss the prevalence of firm internationalization and the degree of internationalization. The second section focuses on the key markets where the companies go and reflects this to earlier results. Finally, we conclude by presenting and analysis of the determinants of internationalization intentions and the actual internationalization actions.

3.1 Internationalization Patterns

In total, approximately 40% of the firms responding to the latest survey firms reported generating at least part of their revenue from abroad. Also, what is worth noting is that approximately 60% of firms that do not have international operations reported that they were planning on internationalizing underlining the fact that internationalization is seen often as a natural stage in software firm evolution [23,24].

After calculating the simple frequencies from the internationalization statistics, we analyzed how large amounts of international revenues the firms generate. The current theories in internationalization predict increased growth [25] particularly if internationalization takes place at young age [26]. We analyzed this data with kernel density plots. To identify potential time trends, we divided the internationalization data into four classes based on year. The reason for uneven distribution is that the survey, which acts as the source of data, has received substantially more responses during the last three years. The distribution of foreign revenue share in the sample of internationalized firms is shown in Figure 1. Considering that the firms that internationalize should experience growth, the fact that the distribution is right-skewed is a surprising finding. This can mean that there are a large number of firms taking only initial internationalization steps. Reflecting this on prior internationalization research, this can be an indication of sampling bias towards positive cases in the current literature [27]. Or that for some reason this particular sample does not behave similarly to other samples. Clearly more larger samples studies are required to identify the true nature of the phenomenon apparent in the figure.

The smaller peak at the end of the plot can mean that a large share of firms that internationalize successfully gain substantial growth which results in the international revenues surpassing domestic revenues by several orders of magnitude, indicating that

some firms indeed manage to realize substantial growth by internationalizing. The variations across time periods are probably random noise, although we did not do any formal statistical tests to verify this assumption.

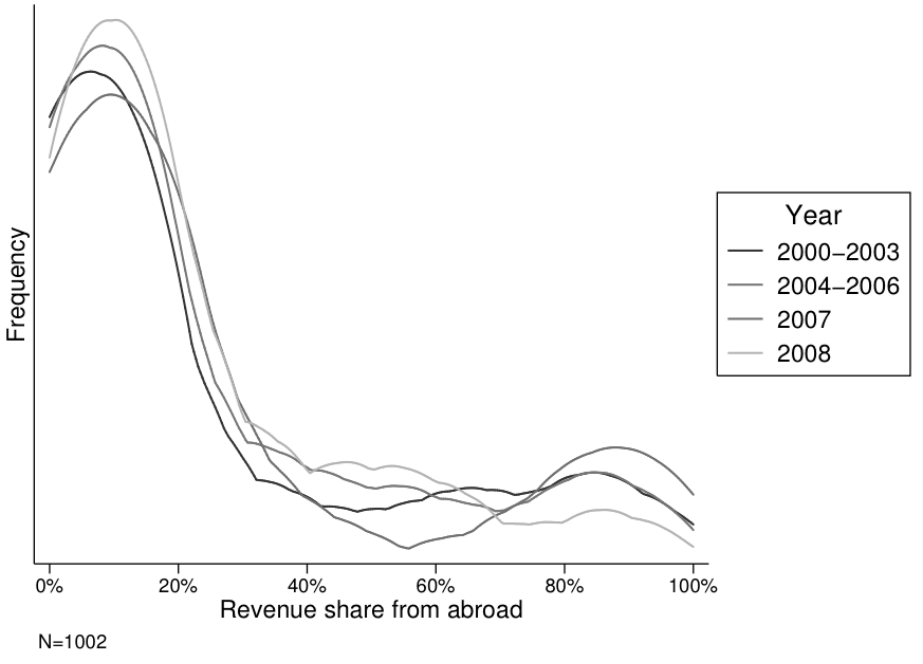

Fig. 1. Revenue shares from abroad by year

After analyzing the distribution of foreign revenue share, we turned our attention on the question of what types of firms internationalize. For this analysis, we grouped the firms into seven classes based on internationalization intensity and plotted the data by business model. Since business model is a problematic construct for research conceptually and difficult to measure [28], we used a simple self-selection. In the survey form the firms were asked to choose the option that best matched their main business from the following alternatives: software product firm, device manufacturer, project contractor, consulting firm, reseller, and not associated with software industry. The last class was excluded due to lack of relevance for this study.

After the total volume of international business, the second most interesting thing is what kinds of firms generate this revenue. Figure 2 shows the distribution of international revenue by these five firm types. Not surprisingly, service businesses are largely local, although some of these firms have generated substantial amounts of international revenue. More interestingly, the most international companies seem to be ones that categorized their main business as device manufacturer having software business. One possible explanation for the international focus of device manufactures is that these companies are probably larger than for example software product firms.

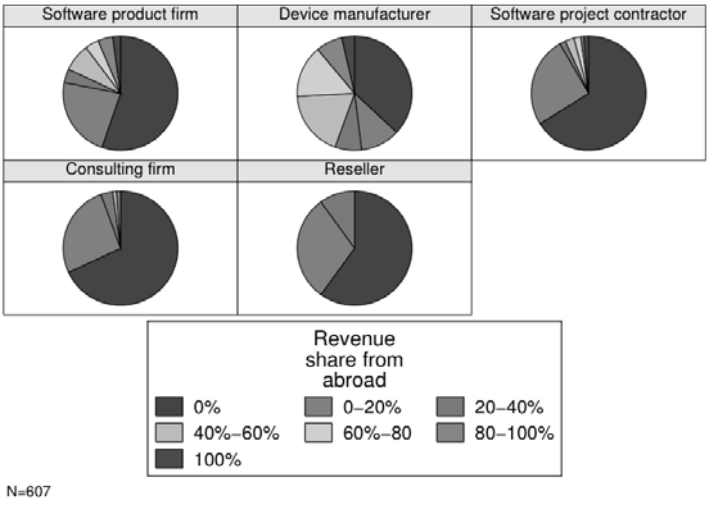

Fig. 2. International revenue shares over firm type

3.2 Target Markets

Figure 3 shows the geographical areas and the frequencies of firms reporting revenue from each of these areas. The figure clearly shows that while Scandinavia indeed is important, as shown in the previous reports in this series, the entire Western Europe is almost equally well present in the graph with more than half of the firms with international operations having revenue from these markets. North America, the largest software market in the world, is the third most targeted geographical market after Western Europe. On the fourth position close behind the North American market come the geographically proximate Eastern Europe including Russia. The different areas of Asia follow North America and Eastern Europe being roughly half as common as the former markets. Finally, South Pacific, Middle East, and Africa conclude the list.

The frequency of firm presence in a particular market, however, does not tell the full story about the importance of each particular market. To understand how important each of these areas are, Figure 4 shows the share of foreign revenue that each of these markets produce. The figure is produced by first calculating how much revenue each responding firm produces from each area and then summing these figures. Since this is figure is based on a sample of 228 firms, the results cannot be completely generalized to the entire population. Regardless, the percentages in this figure are interesting. They follow to a degree the frequencies in Figure 4, which on the surface seems expected. However, due to the very large differences in the market size, we would have expected that firms targeting larger markets would generate more revenue. If the frequency of firms targeting North America is 40% of the firms targeting Nordic countries and the amount of revenue from the American markets is roughly half of what is generated in Scandinavia, this means that for a typical firm, the North American market is only a little more penetrated than the closest

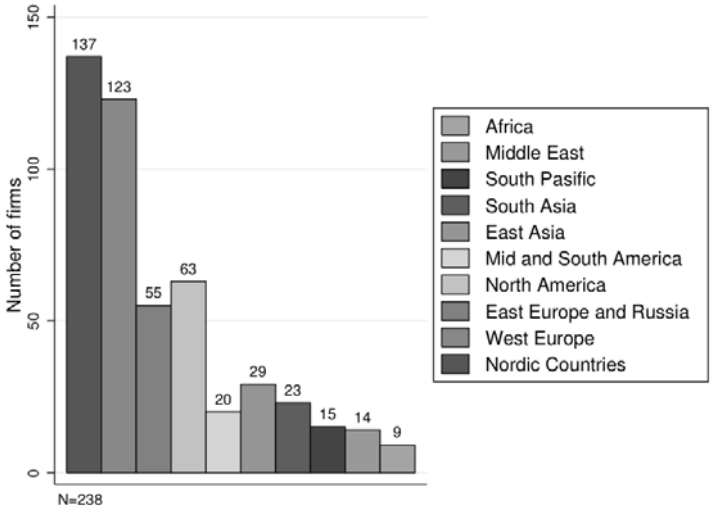

Fig. 3. Number of firms operating in different markets

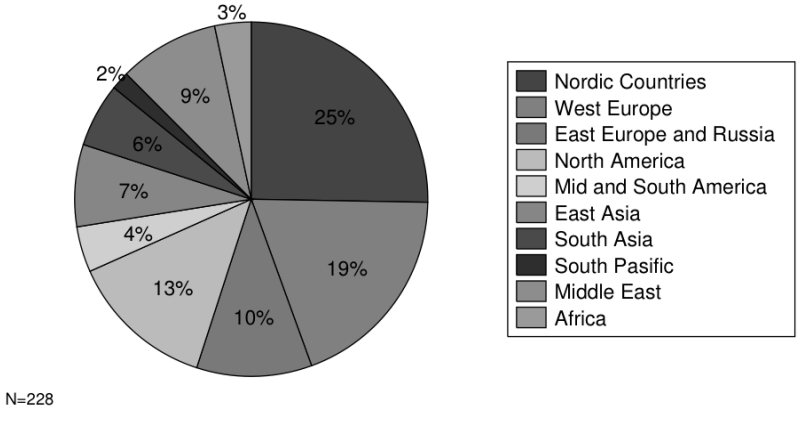

Fig. 4. Share of foreign revenue from different areas

possible international market. While the American market has great potential due to its size, for some reason the Finnish companies in this sample are not able to seize this opportunity.

Finally, Figure 5 shows a histogram of the number of countries targeted. The first thing to notice is that most commonly firms target only one or two areas. Although this was not analyzed, we believe that this is largely explained by service firms operating only regionally in Scandinavia or firms that are taking their initial steps in internationalization path. Overall, the data suggest that software firms internationalize gradually in a stage-based manner instead of following a born-global model [15].

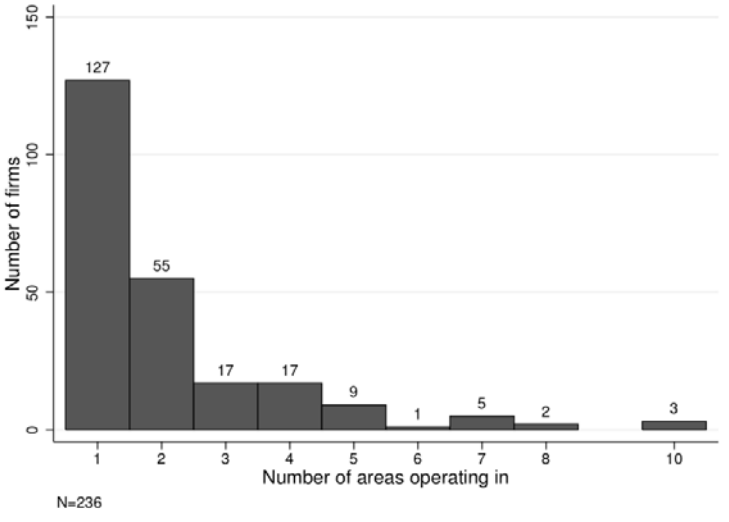

Fig. 5. Number of areas where a firm operates

4 Determinants of Internationalization

The regression analyses in Table 1 show what kind of firms have international revenue or are planning internationalization and if there is a trend over time. The last two models show how these two variables change over time on the firm level. As already covered in the previous section, the first model shows that firm size is strongly related to international revenue. This is due to the fact that in software business, the largest growth options are virtually always outside Finland. Project houses are the least international and according to model 2 also among the least willing to internationalize. This is probably due to the facts that most project services require some kind of local presence while devices or software licenses can be sold through channels. Also the labor costs in Finland do not favor competing internationally, as although the salary levels in IT are not very high compared internationally, the total labor costs are still close to OECD average [21].

The second model also shows that the propensity of those firms who do not have revenue from abroad to plan for international expansion increases with the size of the firm but decreases with the firm age. Although the effect is not strong, we can conclude that if a firm in the software industry does not enter international markets at a young age, it is not likely to do that later either. This is further supported by model three that shows that the likelihood for a firm that on one year reports not having international revenue and the second year as receiving part of the revenue abroad, decreases as the firm size increases.

After comparing which firms are more likely to expand internationally, we will now focus on the differences between firms that are generating revenue from abroad in 2008 and those that are not. Table 2 shows that there is a significant difference in total revenue between internationalized and domestic firms both in terms of mean and median values. Judging by the medians, a typical internationally operating firm had

Table 1. Determinants of international revenue

	Has revenue from abroad	Plans to internationalize	Has revenue from abroad (change)	Plans to internationalize (change)
	(1)	(2)	(3)	(4)
Age (ln)	0.133	-0.373*	-0.173**	0.102
Revenue (ln)	1.635***	0.415*	0.045	0.258†
Year	0.029	0.738	-0.012	
Firm type Software product firm	ref.	ref.	ref.	ref.
Firm type Device manufacturer	1.779†	-0.349	0.204	0.097
Firm type Software project contractor	-1.308**	-0.816*	0.149	-0.171
Firm type Consulting firm	-0.638	-1.118*	0.273	-0.149
Firm type Reseller	-0.521	-0.471	-0.596	-0.487
Intercept	-0.048	-0.299	-0.081	0.412*
Standard deviations of error components				
Firm-level	3.539***	1.549	0.430	1.057
Observation			0.911	0.396
Observations	934	455	307	114
R^2			0.048	0.023
Wald χ^2	47.258	11.367	12.992	9.113
p	0.000	0.123	0.072	0.167
LR test	0.000	0.003	1.000	1.000

† $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$
Mixed-effects logistic regression (Models 1, 2)
Mixed-effects REML regression (Models 3, 4)
Standardized beta coefficients

approximately three to five times the revenue of the average domestic software firm. The year before the average of total revenue for an internationally operating firm was 22.5 M€. Although this was not explicitly tested, there is no reason to believe that the difference in means of revenue would have statistical significance and hence should not be interpreted as a true trend. The same applies to all other comparisons of these descriptive statistics between this year and the previous year.

The comparison of firms that do not yet have international revenues but are planning on internationalizing to those that are neither have international revenue nor plan to expand inter- nationally provides some interesting results. First, we can see that the firms who are planning on internationalizing typically expect to grow twice as fast than other firms including those that are not planning on going international but also those that already do have international revenues. We offer two interpretations for this difference: It is possible that high-growth firms see internationalization as a natural next step or alternatively firms are expecting rapid growth as a consequence of the plans for internationalizing. The lower profitability, higher product development investment, and lower age than in the comparison groups leads us to believe that these firms are typically young software product firms.

Table 2. Descriptive statistics by internationalization

	International operations					
	No international operations		Plans to internationalize		Has international operations	
	Mean	Med	Mean	Med	Mean	Med
Age	9	7	8*	6*	10*	8
Revenue (M€)	2.42***	0.19***	0.83	0.38	14.63***	0.69***
Total personnel	221***	3***	10	6	124***	7***
Profitability	11.25	7.49	6.25	5.71	9.99	7.65
R&D / revenue	12.2***	5.0***	20.5*	12.5	20.4**	11.9**
Expected growth of revenue	86.42*	4.96*	61.02*	16.77**	52.95	9.09
Growth, 3 year CAGR	13.5*	10.1	34.4	21.1	21.4	17.7
N		469		283		899

Mann-Whitney U tests between one group and the rest. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Also when looking at the firms that do not plan to internationalize and comparing these to the others, we can see that these firms are typically smaller, less product-oriented, and more profitable than the other groups. One possible explanation is that these firms are mainly small service firms that seek neither fast growth nor international expansion. For these firms it probably makes sense not to internationalize, since service businesses lack some of the economies of scale present in product business and hence the possible gains from internationalizing the business are probably not worth the risks and cost associated.

5 Conclusions

In this paper, we presented the results of a large sample survey of internationalizing firms in the Finnish software industry. Our descriptive analyses revealed that 41% of the companies have some degree of international operations. Regression analyses further revealed that device manufacturers and software product companies are most likely to expand internationally whereas software service and consulting firms tend to more often remain in the home markets. Firms are also less likely to be international if they are older and more likely to expand internationally if they have higher revenues.

The results support the conclusion that internationalization can be considered as a natural stage in the firm life-cycle, but the patterns of internationalization differ across firms. Considering the current theorizing of software firms as prototypical international new ventures, we find it surprising that many firms seem to choose to internationalize only little and gradually.

References

[1] Kontio, J., Rönkkö, M., Mutanen, O., Ahokas, M., Junna, O., Ali-Yrkkö, J., Touru, A., Ruotsalainen, S., Heikkonen, M., Martikainen, O., Mickelsson, M., Rahkonen, A., Puttonen, V., Niemi, P., Salminen, J., Eloranta, E., Fredrikson, N., Koivunen, A., Tikka, T., Maisala, T.: Kasvufoorumi 08. Ohjelmistoyrittäjät ry., Helsinki (2008)

[2] Bell, J.: The internationalization of small computer software firms. European Journal of Marketing 29, 60–75 (1995)

[3] Bell, J.: A comparative study of the export problems of small computer software exporters in Finland, Ireland and Norway. International Business Review 6, 585–604 (1997)

[4] Moen, Ø., Gavlen, M., Endresen, I.: Internationalization of small, computer software firms. European Journal of Marketing 38, 1236–1251 (2004)

[5] Ojala, A., Tyrvainen, P.: Market entry decisions of US small and medium. Management Decision 46, 187–200 (2008)

[6] Ojala, A., Tyrväinen, P.: Business models and market entry mode choice of small software firms. Journal of International Entrepreneurship 4, 69–81 (2006)

[7] Ojala, A., Tyrväinen, P.: Market Entry and Priority of Small and Medium-Sized Enterprises in the Software Industry: An Empirical Analysis of Cultural Distance, Geographic Distance and Market Size. Journal of International Marketing 15, 123–149 (2007)

[8] Terjesen, S., O'Gorman, C., Acs, Z.J.: Intermediated mode of internationalization: new software ventures in Ireland and India. Entrepreneurship & Regional Development 2008, 89–109 (2008)

[9] Touru, A., Rönkkö, M.: Internationalization Challenges of Small Finnish Software Firms – Comparing Theory and Practice. In: Helander, N., Hannula, M., Iivonen, I., Seppä, M. (eds.) Proceedings of EBRF 2008. Tampere University of Technology, Tampere (2009)

[10] Bell, J., Crick, D., Young, S.: Small Firm Internationalization and Business Strategy: An Exploratory Study of 'Knowledge-Intensive' and 'Traditional' Manufacturing Firms in the UK. International Small Business Journal 22, 23–56 (2004)

[11] Moen, Ø., Endresen, I., Gavlen, M.: Use of the Internet in International Marketing: A Case Study of Small Computer Software Firms. Journal of International Marketing 11, 129–149 (2003)

[12] Nummela, N., Puumalainen, K., Saarenketo, S.: International Growth Orientation of Knowledge-Intensive SMES. Journal of International Entrepreneurship 3, 5–18 (2005)

[13] Zain, M., Ng, S.I.: The impacts of network relationships on SMEs' internationalization process. Thunderbird International Business Review 48, 183–205 (2006)

[14] McDougall, P.P., Oviatt, B.M.: New venture internationalization, strategic change, and performance: A follow-up study. Journal of Business Venturing 11, 23–40 (1996)

[15] Oviatt, B.M., McDougall, P.P.: Toward a Theory of International New Ventures. Journal of International Business Studies 25 (1994)

[16] Eisenhardt, K.M.: Building theories from case study research. Academy of Management Review 14, 532–550 (1989)

[17] Shook, C.L., Ketchen Jr., D.J., Cycyota, C.S., Crockett, D.: Data analytic trends and training in strategic management. Strategic Management Journal, 1231–1237 (2003)

[18] Applegate, L.M., King, J.L.: Rigor and Relevance: Careers on the Line. MIS Quarterly 23, 17–18 (1999)

[19] McGahan, A.M.: Academic research that matters to managers: On zebras, dogs, lemmings, hammers and turnips. Academy of Management Journal 50, 748 (2007)

[20] Van de Ven, A.: Engaged scholarship: a guide for organizational and social research. Oxford University Press, Oxford (2007)

[21] Rönkkö, M., Ylitalo, J., Peltonen, J., Koivisto, N., Mutanen, O., Autere, J., Valtakoski, A., Pentikäinen, P.: National Software Industry Survey 2009. Helsinki University of Technology, Espoo (2009)
[22] Dillman, D.A.: Mail and Internet surveys: the tailored design method. Wiley, New York (2007)
[23] Andersson, S.: Internationalization in different industrial contexts. Journal of Business Venturing 19, 851–875 (2004)
[24] Madsen, T.K., Servais, P.: The internationalization of Born Globals: An evolutionary process? International Business Review 6, 561–583 (1997)
[25] Burgel, O., Fier, A., Licht, G., Murray, G.C.: Internationalisation of High-Tech Start-Ups and Fast Growth - Evidence for UK and Germany. SSRN eLibrary (2000)
[26] Autio, E., Sapienza, H.J., Almeida, J.G.: Effects of Age at Entry, Knowledge Intensity and Imitability on International Growth. The Academy of Management Journal 43, 909–924 (2000)
[27] Ţurcan, R., Mäkelä, M., Sørensen, O., Rönkkö, M.: Mitigating theoretical and coverage biases in the design of theory-building research: an example from international entrepreneurship. International Entrepreneurship and Management Journal (forthcoming)
[28] Rönkkö, M., Valtakoski, A.: Business Models of Software Firms. In: Proceedings of the Forty-Second Annual Hawaii International Conference on System Sciences (CD-ROM), 10 p. Computer Society Press, Waikoloa (2009)

Distance Factors in the Foreign Market Entry of Software SMEs

Arto Ojala and Tanja Kontinen

School of Business and Economics
University of Jyväskylä, Finland

Abstract. Recent studies have indicated that the internationalization process of software SMEs is somewhat independent on the effect of psychic or geographic distance. However, these studies have analyzed the general pattern of entries where software SMEs not commonly follow a step-wise entry route from nearby countries to distant ones. Thus, it remains unknown what the effect of psychic and geographical distance is when these firms enter a distant foreign market. The findings in this case study reveal that psychic and geographic distance inhibited the foreign market entry of software SMEs. However, the distant foreign market entry of these firms was facilitated by distance-bridging and distance-compressing factors enabling foreign business operations despite the significant distance between the home and target country.

Keywords: Psychic distance, geographic distance, SMEs, software firms.

1 Introduction

The impact of geographical and psychic distance on firm internationalization has a long tradition in international business and marketing literature. Already in 1950s, Beckerman [1] proposed that, if transportation costs are equal, entrepreneurs favor psychically close markets. The concept became well known in the 1970s after the studies by Johanson and Wiedersheim-Paul [2] and Johanson and Vahlne [3] known as the Uppsala model. In their study, Johanson and Wiedersheim-Paul [2] define psychic distance as a sum of factors that constrict the flow of information between the firm and the market. Thus, in the model, large psychic distance between countries inhibits the foreign market entry of the firm. In addition to psychic distance, large geographical distance has been indicated having the same effect [4], [5].

In the Uppsala model, firms are expected to enter first into nearby markets which share a similar language, culture, political system, level of education, level of industrial development etc. Thereafter, when a firm's knowledge to operate internationally increases, it gradually starts to develop activities in psychically more distant countries [3]. However, empirical studies related to internationalization of software small and medium-sized enterprises (SMEs) have argued that these SMEs do not follow any particular stages in their internationalization process (see e.g., [6], [7], [8]). For instance, Bell [6, 64-65] announces that "...the data also revealed the some 30-50 per cent of firms had initiated exports with sales to countries which could not

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 49–62, 2010.

be considered either psychologically or geographically proximate" and "…'establishment chain' theories proposed by the Uppsala School authors do not adequately reflect the understanding factors which influence the internationalization patterns of small software firms" [6, 71]. Based on these studies, it seems to be evident that rapidly internationalizing software SMEs do not generally follow the gradual internationalization process from psychically or geographically nearby markets to more distant ones as proposed in the Uppsala model. This conclusion has evoked discussion that psychic and geographical distance have a less important role in the foreign market entry of firms operating in knowledge-intensive sectors [9]. However, the focus of these studies has been on networks [7], entry mode and market selection [8], and internationalization process [6], but the actual effect of psychic or geographic distance on the foreign market entry has been ignored. That is, earlier studies have investigated the general pattern of internationalization but we do not know whether psychic and geographical distance impact on the foreign market entry when firms enter a certain foreign country. In this article, we argue that although these distance factors do not impact on general internationalization process, they still have a remarkable role in the foreign market entry. In addition, we show how these firms are able to tackle the effect of psychic and geographical distance by using proper market entry strategies.

Based on the above discussion, the following three questions are of particular interest in this study: 1) What are the distance-creating factors in the distant foreign market entry encountered by software SMEs? 2) What are the distance-bridging factors that software SMEs use to facilitate their distant foreign market entry? 3) What are the distance-compressing factors facilitating the distant foreign market entry of software SMEs? By investigating to these questions, we use the theoretical framework of Child et al. [10] that divide the components of psychic distance into distance-creating, distance-bridging, and distance-compressing factors.

2 Literature Review

2.1 Distance-Creating Factors

In literature, psychic distance, cultural distance, and geographical distance have been commonly cited as distance-creating factors. In their model, Johanson and Wiedersheim-Paul [2, 308] define psychic distance as "…factors preventing or disturbing the flow of information between firm and market". Thus, the model indicates that psychic distance consists of factors creating distances, such as language, culture, political system, level of education, and level of industrial development. Because of these distance-creating factors, firms are expected to enter first nearby markets where the business environment is similar to the home market. Thereafter, when a firm's knowledge to operate internationally increases, it gradually starts to develop activities in psychically more distant countries. The study of Nordström and Vahlne [11, 42] defines psychic distance as "factors preventing or disturbing firms' learning about and understanding of a foreign environment". This definition refers to the fact that firms have to learn about the environment of the target country and understand the local culture. In their framework, Child et al. [10, 49] argue that "Distance-creating factors

are those responsible for dissimilarity in business environments between the home country and the host countries for investment". They found that culture (including language) was the most important element that created distances and difficulties in the foreign market entry and operations of Hong-Kong family firms.

Cultural distance between countries is also seen as a factor creating difficulties in the foreign market entry. Sousa and Bradley [12, 63] define the difference between cultural and psychic distance as follows: "Cultural distance reflects a difference in cultural values among countries" and "Psychic distance is based on the individual's perception". Cultural differences between countries have commonly been measured by using Hofstede's [13] cultural dimensions and the composite index of Kogut and Singh [14]. These studies (see [15] for further review) have been motivated by the assumption that cultural differences between countries inhibit market entry to culturally distant countries. However, in many cases, the results have been conflicting and there has been a growing amount of criticism toward the usage of Hofstede's [13] cultural dimensions as single determinant for the foreign market entry decision (see e.g., [4], [15]). For instance, the study of Dow and Karunaratna [4] proposes that cultural distance is only one dimension of the larger concept of psychic distance.

The third commonly cited distance-creating factor is geographical distance. Srivastava and Green [5] found that geographical distance has the most significant impact on the trade intensity between countries. This is also line with findings of Dow and Karunaratna [4] indicating that geographic distance is the most influential inhibitor of international trade. In addition, Leamer and Storper [16] indicate that geographical distance is still a valid inhibitor despite improvements of transportation systems and communication technologies. Ojala and Tyrväinen [17], [18] suggest that geographic distance even impacts on the initial market selection of software SMEs.

Altogether, the current literature indicates that psychic and geographical distances are major distance-creating factors whereas the role of cultural distance is more complex and the results have been contradicting. In addition, psychic distance is multidimensional including several factors (see e.g., [4]) and its impact depends on entrepreneur's perceptions about differences between countries [19], [12]. Thus, this study takes a wider perspective to distance-creating factors than the study of Child et al. [10]. In addition, similarly to Sousa and Bradley [12], distance-creating factors are conceptualized here as individual-level perceptions of firm's decision makers. Thus, this study defines distance-creating factors as a sum of factors, based on the perceptions of an entrepreneur or a manager, inhibiting or restricting firm's entry into a new foreign market.

2.2 Distance-Bridging Factors

In their study, Child et al. [10, 50] define distance-bridging factors as "…factors, which are open to the initiatives of firms themselves". These factors consist of strategic and operational activities where strategic decisions are related to locations choice for foreign markets and operational activities are related to the managerial operations in the target country. In line, Nordström and Vahlne [11, 46] indicate the role of distance-bridging factors by arguing that "…distance can be bridged by factors such as knowledge dissemination…or trial and error processes". Thus, distance-bridging factors are largely under the control of an individual firm or an entrepreneur. This is

well present in the international entrepreneurship literature revealing the important role of entrepreneurial activities in the foreign market entry. Scholars [20], [21], [22] have indentified positive relationships between managers' opportunity seeking behavior in the early internationalization. In addition, managers' earlier experiences can facilitate the market entry [20], [21], [23]. Coviello and Martin [24] found that psychic distance can be overcome by recruiting experienced personnel with knowledge of the distant country. Furthermore, the distance-bridging role of network relationships for early internationalization is acknowledged in several studies [25], [24], [26]. For instance, Coviello and Martin [24] argue that the usage of network relationships significantly reduced perceived psychic distance.

Summarizing, there seems to be several distance-bridging factors which facilitate and enable a firm's entry to a distant market despite distance-creating factors. Thus, distance-bridging factors are defined here as any action taken by an entrepreneur or a manager to decrease the impact of distance-creating factors in the foreign market entry.

2.3 Distance-Compressing Factors

According to the definition by Child et al. [10], distance-compressing factors refer to macroeconomic changes, such as social movements, institutional changes, globalization, and technological development. For instance, Oviatt and McDougall [22], [23] indicate the distance-compressing role of the worldwide development of information and communications technologies that facilitate the foreign market entry of new ventures. In addition, some studies reveal the distance-compressing effect of increasing international traveling and mass-media. This has been labeled as global cultures where people share similar values and behavior regardless of their original cultural background or geographic location [27], [28]. This has made foreign markets easier for firms to enter [10] and some products less culturally sensitive. Other examples of distance-compressing factors are the free trade areas and the trade agreements between nations such as the General Agreement on Tariffs and Trade (GATT), European Union (EU), and the North American Free Trade Agreement (NAFTA). In addition, the establishment of organizations like World Trade Organization (WTO) facilitates business between countries. The aim of these agreements and areas is to decrease trade barriers between countries and, consequently, facilitate the foreign market entries of SMEs (see e.g., [29], [30]). All together, these activities or macroeconomic changes are largely out of the individual entrepreneur's or manager's control. Thus, distance-compressing factors are defined here as a sum of factors facilitating firm's foreign market entry that are not under a control of an entrepreneur or a manager.

3 Research Method

This study employs a qualitative case study method [31] including eight Finnish software firms entering the Japanese market. The case study approach was selected because of the need to analyze the firm level and the individual level behavior in detail to come to an understanding about the behavior of the firms in the foreign

market entry process. This enables explaining the significance and cause-and-effect relationships of the phenomena under investigation [31]. In addition, Eisenhardt [32] suggests that the multiple case study method allows studying patterns that are common to the cases and theory under investigation.

Finnish software SMEs operating in the Japanese market were selected as the target group of this study. The selection of the host and target countries is based on several methodological and theoretical reasoning. Firstly, Finland and Japan are culturally and geographically very distant from each others. This helps to find out potential impact of psychic or geographical distance in the market entry that would not be observable if two countries selected are very close to each others. Secondly, both countries have their own languages; Finnish is spoken only in Finland and Japanese in Japan. In addition, both languages differ greatly from other major languages such as English that is commonly used in international business. Thirdly, both countries are culturally very homogenous, and, accordingly, there are no large cultural differences within the countries. This helps us to overcome the criticism of Shenkar [33] related to 'the assumption of spatial homogeneity'.

All eight case firms selected for this study fulfilled the definition by the Finnish government and European Union [34] for SMEs having fewer than 250 employees at the time of their market entry to the Japanese market. The case firms were from software industry, and even if it might be seen as a limitation, several studies analyzing internationalization of knowledge-intensive firms have used software industry as a target group in their studies (see e.g., [6], [25]). Despite the fact that software industry differs somewhat from other industries due to the intangible nature of its products, etc. it still shares common characteristics with other knowledge-intensive industries [35] and the service sector [36].

The interviews for this study were conducted in the headquarters of the firms in Finland and in their units in Japan covering altogether 16 interviews from the eight firms. The main criterion for interviewed persons was that they were actively involved in their firm's entry process to the Japanese market. By selecting the most knowledgeable persons, and by using two informants from each firm, we aimed to get the most relevant knowledge, and to counteract the biases of individual opinions [37]. In addition, having two interviews from each case firm also made it possible to ask more detailed questions of the second interviewee, following on from the first interview. Working in this way improved the validity of the data collected. The interviews were rather conversational and focused mainly on open-ended questions related to the firms' market entry into the Japanese market and its operations there. All these questions were developed according to the guidelines issued by Yin [31], with the aim of making the questions as non-leading as possible. This encouraged the interviewees to give authentic answers to the interview questions. Because the interviews focused on the managers' past experiences, we followed the guidelines for retrospective studies issued by Miller et al. [38] and by Huber and Power [37].

Each interview took approximately 60-90 minutes and was digitally recorded, carefully listened to, and transcribed verbatim with the help of a word processor. A second listening took place to ensure the correspondence between the recorded and the transcribed data. Thereafter, the complete case reports were sent back to the persons interviewed to ensure the validity and the authenticity of the collected data. Whenever the interviewees found some inaccuracies in the text, these were corrected based on

their comments. In addition, e-mail communication was used to collect further information and to clarify any inconsistent issues. To improve the validity of the study we collected and analyzed many types of secondary information (such as web-sites and annual reports). By comparing the interview data with other documents from the case firms, we carried out triangulation on the information [39], [40]. This also provided a more complete picture of the case firms under study [39].

In the data analysis, we arrived at a detailed case history of each firm based on the interviews and written documents in line with Pettigrew [41], who suggests that organizing incoherent aspects in chronological order is an important step in understanding the causal links between events. Thereafter, on the basis of the interviews, we identified the unique patterns of each case and categorized the patterns observed under the sub-topics derived from the three research questions we had set for the study. These sub-topics included distance-creating factors, distance-bridging factors, and distance-compressing factors. In addition, analytical tools were applied within and across the cases as proposed by Miles and Huberman [40].

4 Research Findings

This section presents the empirical findings by categorizing them into distance-creating, distance-bridging, and distance-compressing factors. The average number of employees in the case firms at the time of the interviews was 127. All the case firms were established between 1990 and 2000, except from Firm C that was established already in 1966. The firms had operated in the Japanese market from three to seven

Table 1. Key information on the case firms

	Number of employees	Year of establishment	Foreign direct business operations	Entry modes in Japan
Firm A	30	1998	USA 1998 Hungary 2000 Japan 2002	Representative 2002
Firm B	90	1992	USA 2000 Japan 2002	Representative office 2002
Firm C	300	1966	Sweden 1995 USA 1999 Malaysia 1999 Germany 1999 UK 1999 Japan 2000	Distributors 1999 Representative office 2000 Subsidiary 2001
Firm D	240	1990	USA 1998 Japan 1999	Representative office 1999 Subsidiary 2000
Firm E	100	1995	USA 1998 Japan 2000	Direct sales 1999 Subsidiary 2000
Firm F	210	1991	Sweden 1999 Hong Kong 2000 Japan 2001	Distributor 1997 Joint Venture 2001 Subsidiary 2005
Firm G	12	1998	Japan 1999	Joint venture 1999
Firm H	35	2000	UK 2000 Japan 2003	Corporate 2003

years. Table 1 summarizes the key information of the case firms and demonstrates their foreign direct investments before the market entry to Japan. Entry modes used by each case firm in Japan are presented in a chronological order.

4.1 Distance-Creating Factors Encountered by the Case Firms

All the case firms experienced Japan as a difficult country to enter. The main factors that can be conceptualized as distance-creating factors in the market entry were related to differences in language, business culture, and geographic distance between Finland and Japan.

Language was seen as a distance-creating factor in all firms. Although English was used as an official language in all case firms, the low English proficiency of Japanese customers and partners created problems. The language related problems consisted of networking with potential customers, misunderstandings with customers, and localization of products. For instance, Firm H was searching for a Japanese partner in the Internet. However, language difficulties created remarkable problems. One informant from Firm H explained this as follows:

"Finding a partner in Japan via Internet was difficult, because they have their websites only in Japanese and only few firms have English versions. Of course large multinational firms have their websites in English, but those firms that are of equal size with us, they usually do not."

Language difficulties also increased the need for local presence in Japan because of the needs for local staff that can handle the business negotiations and give after-sales support in Japanese. For instance, one informant from Firm E mentioned that in other countries they have been able to handle their business by using English, but when dealing with Japanese customers, they needed to have employees with very good Japanese skills and cultural knowledge. Firm C also noted that after the market entry and the establishment of their unit in Japan they have not been able to give support for their unit in Japan in the same scale as for other units because of language problems. For instance, they cannot help with market data collection from the Japanese market because no one in the headquarters has Japanese skills needed.

All case firms also confirmed that the way of doing business in Japan differs greatly from other Western and Asian countries. Differences in business culture were related to working times, the hierarchical management style of Japanese customers and partners, slow decision making process, demanding customers, and time needed for building trustful relationships with customers. For instance, Firms A, B, C, E, and G disclosed the slow decision making process of Japanese customers and partners that delayed the sales process and the launching of new products in the market. Firms A, B, and C also highlighted that building trustful relationships with customers was time consuming but needed before the business progressed. Japanese customers were characterized very demanding what comes to products. One informant from Firm E explained the slowness of the decision making and demanding customers in the following way:

"Japanese do not buy anything that is not perfect. When they are 100 percent sure that the product works then they will buy it. For Japanese, it is also important to know who the original developer of the product is and where it is developed."

The hierarchical management style of Japanese customers and partners was also experienced very different compared to other countries. As an example, Firms A, B, and C had difficulties with finding the right contact persons from the customer's side. Their customers were large multinationals and it was hard to get to know who was responsible for technology purchasing. The management style was also closely related to working times that differed greatly from those that the firms were used to. The manager from Firm F explained the differences of the working times as follows:

"When we are busy in Finland, we still commonly go back home when the working time is over, around four p.m. or at least half past four. However, here the customer trusts that if we are busy, we are still working as long as it takes to get everything done. Going back home earlier…they just do not understand it. They do not understand five weeks' vacations -vacation is not a reason for delays."

The geographical distance and the time difference between Finland and Japan were experienced to hinder the business. This was despite the fact that all the firms were able to deliver their products electronically via Internet. Geographical distance was seen as a disadvantage because the Japanese as well as other competitors from geographically nearby countries were able to give support for customers much faster. Appointments and business negations with customers also required lot of traveling and increased costs of doing business. One informant from Firm B expressed this as follows:

"If we have to send employees [to Japan], it easily takes two or three days before they are in our customer's office [in Japan]. Whereas a local competitor can put their whole product development team to a train and they all are there within two hours…it is an obvious advantage for our competitors in Japan."

4.2 Distance-Bridging Factors Used by Case Firms

Despite distance-creating factors discussed above, all the firms were able to use distance-bridging factors to enable their operations in the Japanese market. The distance-bridging factors can be divided into opportunity seeking behavior, recruitment of capable employees, choice of the proper entry mode, networking, and earlier experiences.

All the case firms regarded Japan as a very interesting country for their products already before they started to actively prepare their market entry to Japan. This was mainly related to the large market size and the sophisticated industry structure for the products of the firms. In addition, the domestic markets of the case firms were mentioned to be relatively small and saturated for their niche products. Thus, opportunities motivated managers to enter the Japanese market despite risks and entry barriers. Firms A, C, D, E, and F mentioned that the main reason for the market entry was the large market size and the business opportunities in Japan. In addition, firms G and H mentioned the sophisticated industry structure for their products, such as high capacity of broadband and mobile networks. Firm B got an important customer from Japan

that motivated to enter the market and search for more business potential there. One informant at Firm A explained their reasons for the market entry as follows:

"The target customers of our product are mobile industry, mobile phone manufacturers, mobile operators, and related electronic industry. This is very strong in Japan. Another thing is that Japan is a very difficult market, if we can succeed there, so we can succeed in other markets as well. Thus, the main reason for the market entry was the huge market potential there."

All the managers in the case firms understood that they do not have the required knowledge to handle business activities in the Japanese market because of significant differences in language, culture, and business practices. For this reason, firms C, D, and E acquired the relevant knowledge by recruiting international experienced managers who had sensitivity to psychic distance between Finland and Japan to handle their operations in Japan. It was also important that the selected manager was aware of the business environment, culture, language, etc. in both countries, not only what comes to the target country.

In addition, these firms recruited local employees for marketing and other tasks requiring close cooperation with Japanese customers. Firm F used also this kind of recruitment strategy when they changed their joint venture to a wholly owned subsidiary. Firms A and B were able to handle their customers in Japan by using their current employees as expatriates because of English proficiency of their customers in Japan. Firms G and H did not recruit employees for their units because of their different entry mode strategy discussed below.

The selection of the proper entry mode for the Japanese market was also seen as a very important distance-bridging factor. In all the cases, the reason for the direct entry modes in the market was based on the complexity of the firms' products that required close cooperation with the customers and distributors during the sales process. An own unit in the market enabled the after-sales services nearby the customers, the localization and customization work together with customers, and the recruitment of local employees. In addition, the own unit in the market reduced traveling needs between Finland and Japan and facilitated networking with customers and distributors. Although firms C, E, and F started their operations by using indirect entry modes, such as exporting and foreign distributors, they very soon established direct entry modes in the market. Firms F, G, and H used cooperative entry modes with Japanese firms, although Firm F changed their entry mode into a wholly owned subsidiary later on. This kind of partnering strategy enabled the usage of local knowledge and the decrease of difficulties related to the Japanese language and culture, remarkably in the market entry phase.

Network relationships had also an important distance-bridging role in the market entry phase and later on in networking with customers and distributors. In the market entry, the firms used formal networks with their current business partners (firms D and H), informal networks with friends (firms B and G), and mediated relationships with export promotion organizations (firms A, C, E, and F). The importance of network relationships with the export promotion organizations were highlighted especially among those firms who established wholly owned operations in the market. One of the informants at Firm B highlighted this as the following manner:

"In networking, one good example is Finpro [export promotion organization in Finland], we have used them…they have introduced us to potential customers in Japan. Finpro is good for opening doors to new firms because they have a local authority and a long experience in the field…they know lot of persons."

Firms G and H who used cooperative entry modes were able to benefit from networks of their Japanese employees. The manager at Firm G explained the benefit of the joint venture in networking as follows:

"In Japan, it is a substantial benefit that we have local employees. Taking care of the relationships with customers and distributors is much easier. Networking happens through them…In that way, our unit in Japan has a crucial role because they have very good relationships with the actors in the market."

Earlier experiences from other markets also had an important distance-bridging role for all the firms. This was the case although all the firms had a very limited knowledge from other markets before they established their operations in Japan. These earlier experiences facilitated mainly in operational level activities such as cost estimations, location and entry mode choice, taxation, business models, etc. However, all the firms announced that such experiences are always very personalized. This was the reason why earlier experiences helped only in operational level activities and not with activities where knowledge of local culture was important.

4.3 Distance-Compressing Factors Facilitating the Market Entry of the Case Firms

Distance-compressing factors indicated by the case firms were the technologically advanced industry structure, low governmental entry barriers, and the good image of Finland in Japan. The advanced industry structure in Japan facilitated firms' marketing activities, the distribution of products via Internet, and the delivery of product updates. Entry barriers set by the Japanese government were also experienced to be fairly low. Although some of the problems encountered by the firms were related to industry regulations and intellectual property protection, they were actually due to normal practices faced in all the markets. One of the informants at Firm B expressed this as follows:

"Although Japan is a very bureaucratic country…we have not experienced any problems related to legal issues. The problems are more related to business culture and how to do business…maybe it takes a bit more time, paper work and translation."

Firms B, C, D, E, and F disclosed that the good image of Finland acted as a distance-compressing factor. Japanese customers saw Finland as a very advanced country in technology, mainly because of the mobile phone manufacturer Nokia and Linux operating system that are both Finnish. Although none of these firms could estimate how much it actually influenced their business, it was mentioned to be like an "extra" benefit. One manager at Firm D explained this as follows:

"We profiled as a Finnish firm very clearly, because I and my colleague who was also establishing this firm [the subsidiary in Japan] were from Finland…In Japan, Finland has a very good reputation. It was like an extra benefit for our customers that we were from the same country as Linux."

5 Discussion of Research Results

As the case findings reveal, the main distance-creating factors in the market entry were language, business culture, and geographic distance. These findings indicate that when software firms enter a distant foreign market, psychic and geographical distance impact on the market entry of these firms. Thus, although these firms do not tend to follow a step-wise internationalization process from nearby to distance countries [6], [42], [9], [22], [23], psychic and geographical distance are still important factors inhibiting the foreign market entry.

The actions taken by the case firms to decrease the impact of distance-creating factors were opportunity seeking behavior, recruitment of capable employees, choice of the proper entry mode, networking, and earlier experiences. The findings here reveal that all the distance-bridging factors are not directly related to the psychic distance [2] or environmental differences between the home and the host countries [10]. For instance, the opportunity seeking behavior has very little to do with psychic distance or environmental differences because it is more related to the entrepreneurial behavior of managers. Thus, distance-bridging factors are not solely those that improve the information flow (as opposite for the definition for psychic distance by Johanson and Wiedersheim-Paul [2]). The findings related to distance-bridging factors found here were mainly inline with earlier studies. These studies have revealed that managers' opportunity seeking behavior, [20], [21], [22], earlier experiences [20], [21], [23], recruitment of knowledgeable employees [24], and network relationships [25], [24] facilitate and accelerate the foreign market entry of software SMEs. However, none of these factors alone helped to overcome distance-creating factors in the foreign market entry. Thus, these factors should be studied as a sum of actions taken by a firm or an entrepreneur to overcome distance-creating factors.

Distance-compressing factors, referring to the macroeconomic conditions which are not under a control of an individual firm or an entrepreneur were the technologically advanced industry structure in Japan, low governmental entry barriers, and the good image of Finland in Japan. Thus, as distinct from distance-bridging factors, there were also factors which were not under the control of a firm or an entrepreneur but those facilitated the market entry and decreased the impact of distance-creating factors. For instance, it is very difficult for an individual firm to change these factors, like image of the home country, although it is very important for a firm's survival in the target country (see [43], for further review). The findings in this category also revealed the fact that macroeconomic changes and trade agreements such as GATT and WTO had an impact on the market entry and reduced or eliminated government based entry barriers in Japan. In earlier studies, the Japanese market has been reported to be difficult to enter (see for e.g. [44]), mainly due to the entry barriers set by the Japanese government.

6 Conclusions

This paper contributes to the literature by recognizing the impact of distance-creating factors in the foreign market entry of software SMEs. In addition, it recognizes how distance bridging and compressing factors moderate and facilitate the distant foreign

market entry. Although earlier studies have focused on some of these factors, such as opportunity seeking behavior or network relationships, this study gives a wider perspective covering all the distance-bridging and distance-compressing factors used by the case firms to facilitate their distant foreign market entry. These all are very important concerns for managerial practice as software firms are increasingly investing and operating on distant foreign markets. For the theory development in the field of international entrepreneurship, the findings here indicate that although psychic and geographical distance have only a minor impact on the general pattern of internationalization, the role of these factors have to be considered when a firm enters a distant foreign market. In addition, this paper further develops the analytical framework of Child et al. [10]. The original framework by Child et al. [10] is developed by using large multinationals from Hong-Kong. Findings here reveal that the framework can be also used to analyze the foreign market entry of software SMEs. In addition, this study gives more detailed definitions of the three components (distance-creating, distance-bridging, and distance-compressing) used in the framework.

References

1. Beckerman, W.: Distance and the Pattern of Intra-European Trade. The Review of Economics and Statistics 38, 31–40 (1956)
2. Johanson, J., Wiedersheim-Paul, F.: The internationalization of the firm: four Swedish cases. Journal of Management Studies 12(3), 305–322 (1975)
3. Johanson, J., Vahlne, J.-E.: The internationalization process of the firm: a model of knowledge development and increasing foreign market commitments. Journal of International Business Studies 8(1), 23–32 (1977)
4. Dow, D., Karunaratna, A.: Developing a multidimensional instrument to measure psychic distance stimuli. Journal of International Business Studies 37(5), 1–25 (2006)
5. Srivastava, R.K., Green, R.T.: Determinants of Bilateral Trade Flows. Journal of Business 59(4), 623–640 (1986)
6. Bell, J.: The Internationalization of Small Computer Software Firms: A Further Challenge to "Stage" Theories. European Journal of Marketing 29(8), 60–75 (1995)
7. Coviello, N., Munro, H.: Network Relationships and the Internationalisation Process of Small Software Firms. International Business Review 6(4), 361–386 (1997)
8. Moen, O., Gavlen, M., Endresen, I.: Internationalization of small, computer software firms: Entry forms and market selection. European Journal of Marketing 38(9-10), 1236–1251 (2004)
9. Madsen, T.K., Servais, P.: The Internationalization of Born Globals: an Evolutionary Process? International Business Review 6(6), 561–583 (1997)
10. Child, J., Ng, S.H., Wong, C.: Psychic distance and internationalization: Evidence from Hong Kong firms. International Studies of Management and Organizations 32(1), 36–56 (2002)
11. Nordström, K.A., Vahlne, J.-E.: Is the Globe Shrinking? Psychic Distance and the Establishment of Swedish Sales Subsidiaries during the Last 100 Years. In: Landeck, M. (ed.) International Trade: Regional and Global Issues, pp. 41–56. St. Martin's Press, New York (1994)
12. Sousa, C.M.P., Bradley, F.: Cultural Distance and Psychic Distance: Two Peas in a Pod? Journal of International Marketing 14(1), 49–70 (2006)
13. Hofstede, G.: Culture's Consequences, 2nd edn. Sage, New York (2001)

14. Kogut, B., Singh, H.: The effect of national culture on the choice of entry mode. Journal of International Business Studies 19(3), 411–432 (1988)
15. Tihanyi, L., Griffith, D.A., Russell, C.J.: The effect of cultural distance on entry mode choice, international diversification and MNE performance: a meta-analysis. Journal of International Business Studies 36(3), 270–283 (2005)
16. Leamer, E.E., Storper, M.: The Economic Geography of the Internet Age. Journal of International Business Studies 32(4), 641–665 (2001)
17. Ojala, A., Tyrväinen, P.: Market Entry and Priority of Small and Medium-Sized Enterprises in the Software Industry: An Empirical Analysis of Cultural Distance, Geographical Distance and Market Size. Journal of International Marketing 15(3), 123–149 (2007)
18. Ojala, A., Tyrväinen, P.: Market entry decisions of US small and medium-sized software firms. Management Decision 46(2), 187–200 (2008)
19. Ellis, P.D.: Does psychic distance moderate the market size–entry sequence relationship? Journal of International Business Studies 39(3), 351–369 (2008)
20. Freeman, S., Cavusgil, S.T.: Toward a Typology of Commitment States Among Managers of Born-Global Firms: A Study of Accelerated Internationalization. Journal of International Marketing 15(4), 1–40 (2007)
21. Nummela, N., Saarenketo, S., Puumalainen, K.: A Global Mindset: A Prerequisite for Successful Internationalization? Canadian Journal of Administrative Sciences 21(1), 51–64 (2004)
22. Oviatt, B.M., McDougall, P.P.: Toward a theory of international new ventures. Journal of International Business Studies 25(1), 45–64 (1994)
23. Oviatt, B.M., McDougall, P.P.: Global start-ups: Entrepreneurs on a worldwide stage. Academy of Management Executive 9(2), 30–43 (1995)
24. Coviello, N., Martin, K.A.-M.: Internationalization of Service SMEs: An Integrated Perspective from the Engineering Consulting Sector. Journal of International Marketing 7(4), 42–66 (1999)
25. Coviello, N.: The network dynamics of international new ventures. Journal of International Business Studies 37(5), 713–731 (2006)
26. Freeman, S., Edwards, R., Schroder, B.: How Smaller Born-Global Firms Use Networks and Alliances to Overcome Constraints to Rapid Internationalization. Journal of International Marketing 14(3), 33–63 (2006)
27. Alden, D.L., Steenkamp, J.-B.E.M., Batra, R.: Brand Positioning Through Advertising in Asia, North America and Europe: The role of Global Consumer Culture. Journal of Marketing 63(1), 75–87 (1999)
28. Hannerz, U.: Cosmopolitans and locals in world culture. In: Featherstone, M. (ed.) Global Culture: Nationalism, Globalization and Modernity, pp. 237–251. Sage, CA (1990)
29. Pett, T.L., Wolff, J.A.: Firm Characteristics and Managerial Perceptions of NAFTA: An Assessment of Export Implications for U.S. SMEs. Journal of Small Business Management 41(2), 117–132 (2003)
30. Yamin, M., Sinkovics, R., Hadjielias, E.: EU Harmonization, Managerial Perceptions and SME Export Behavior. Journal of Euromarketing 17(1), 7–21 (2007)
31. Yin, R.K.: Case Study Research: Design and Methods. Sage Publications, California (1994)
32. Eisenhardt, K.M.: Building Theories from Case Study Research. Academy of Management Review 14(4), 532–550 (1989)
33. Shenkar, O.: Cultural Distance Revisited: Towards a More Rigorous Conceptualization and Measurement of Cultural Differences. Journal of International Business Studies 32(3), 519–535 (2001)

34. OECD: Officially-supported export credits and small exporters, Organization for Economic Co-operation and Development, Paris, France (2003)
35. Spence, M.: International Strategy Formation in Small Canadian High-Technology Companies: A Case Study Approach. Journal of International Entrepreneurship 1(3), 277–296 (2003)
36. O'Farrell, P.N., Wood, P.A., Zheng, J.: Internationalisation by Business Service SMEs: An Inter-Industry Analysis. International Small Business Journal 16(2), 13–33 (1997)
37. Huber, G.P., Power, D.J.: Retrospective Reports of Strategic-level Managers: Guidelines for Increasing their Accuracy. Strategic Management Journal 6, 171–180 (1985)
38. Miller, C.C., Cardinal, L.B., Glick, W.H.: Retrospective reports in organizational research: A reexamination of recent evidence. Academy of Management Journal 40(1), 189–204 (1997)
39. Bonoma, T.V.: Case Research in Marketing: Opportunities, Problems and a Process. Journal of Marketing Research 22(2), 199–208 (1985)
40. Miles, M.B., Huberman, M.: Qualitative Data Analysis: An Expanded Sourcebook. Sage Publications, California (1994)
41. Pettigrew, A.M.: Longitudinal field research on change: Theory and practice. Organization Science 1(3), 267–292 (1990)
42. Crick, D., Jones, M.V.: Small High-Technology Firms and International High-Technology Markets. Journal of International Marketing 8(2), 63–85 (2000)
43. Roth, K.P., Diamantopoulos, A.: Advancing the country image construct. Journal of Business Research 62(3), 277–284 (2009)
44. Mason, M.: United States Direct Investment in Japan: Trends and Prospects. California Management Review 35(1), 98–115 (1992)

Partnering Strategies in Global Software Business – A Contingency Perspective

Sami Saarenketo*, Olli Kuivalainen, and Jari Varis

Lappeenranta University of Technology
P.O. Box 20, FI-53851 LAPPEENRANTA
Fax: +358-5-621-7299
sami.saarenketo@lut.fi, olli.kuivalainen@lut.fi,
jari.varis@lut.fi

Abstract. This paper describes a contingency framework for analysis of partnering strategies in software business, a context where international entrepreneurship is highly manifest. The framework (rooted in the literature on international alliances and partnering) consists of three-by-three matrix with the following dimensions: The product strategy of the firm (product, project or hybrid) and the value chain activities (inward, outward or both) that are coordinated through partners. The paper discusses the crucial elements of partnering in each of the resulting nine windows, and make suggestions on software company strategy.

Keywords: partnering strategy, partnering capabilities, management, global software business, product strategy, internationalization of SMEs.

1 Introduction

Partnering with other organizations is a significant mode of international expansion. In the face of intensified globalization CEOs are putting inter-firm collaboration higher on their agendas and partnering has emerged as a compelling strategic alternative for internationalizing firms. Defined in this study as "…on-going relationships between two firms that involves a commitment over an extended time period, and a mutual sharing of information and the risks and rewards of the relationship" [1], these inter-firm and often cross-national partnerships have grown to be "a powerful force shaping firms' global strategies." [2] However, regardless of the increasing popularity of partnering as a growth and internationalization strategy for SMEs, the effectiveness of this strategy has been under-explored in the literature.

In this study, we develop a contingency framework for analysis of partnering strategies in software business, a context where international entrepreneurship is highly manifest. The well-known definition by McDougall and Oviatt [3] identifies international entrepreneurship as "a combination of innovative, proactive, and risk-seeking behavior that crosses or is compared across national borders and is intended to create value in business organizations". Our focus is to explore how this "behavior" and "crossing of borders" unfolds through inter-firm collaboration and partnering.

* Corresponding author.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 63–75, 2010.

Inter-organizational cooperation, especially when crossing borders is problematic field in which to do research as these relationships encompass several contractual modes and there are wide range of strategic motivations for formation of them. [4] What is more, the inherently complex nature of partnerships over the time calls for process oriented research in this area, but the majority of the studies still adopt cross-sectional, structural research approach and are driven by secondary data. [5]

In this paper we describe a new model of partnering strategies, which is applicable to international entrepreneurs within the software domain and perhaps also other fast changing knowledge-intensive businesses. The main dimensions of the model are the product strategy or the firm (product, project or hybrid) and the value chain activities (inward, outward or both) that are coordinated through partners. In the next section we outline the conceptual foundation for the study. Then, we explain our methodology that we use to illustrate the partnering process in the chosen context. Finally, we report the results of the case analyses, discuss the findings, and summarize their implications.

2 Conceptual Foundation

2.1 Partnering in the Internationalization of SMEs

Partnering and strategic alliances are important means of international expansion and a growing number of SMEs have applied this mode in their expansion. However, as Welch et al. [6] note, there seems to be a high drop-out rate from partnerships, and that, despite the many observable appeals of using partners in foreign operations, it is difficult to make international partnerships work in practice. By definition, SMEs have more constraints in resources and capabilities [7] as compared to large MNCs and are subject to the several liabilities such as newness [8], smallness and foreignness. [9] SMEs are also exposed to high rate of uncertainty regarding e.g. potential clients and their needs. [10] They do not possess enough resources at the start-up phase to tolerate any serious business mistakes. [11] Establishing international partnerships and thereby leveraging others' resources is a potential way to overcome the mentioned liabilities. Despite the increasing popularity of international partnering and strategic alliances, the underlying process of partnering remains under-explored as the literature continues to be dominated by cross-sectional studies. The prior research has tackled the partnering strategies and processes but very often partially, i.e. focusing merely on certain (early) phases such as partner selection. There is a need to understand the strategy and process as a whole however, as a large percentage of international partnerships fail because senior management at domestic firms takes a casual or careless approach to foreign involvement. [12]

Processes have been studied in strategy research and process has been conceptualized in different ways. Also partnering or alliance process has been described in several other publications and various other models have been presented. [13] [14] [15] [16] If we see the process as a sequence of events that describe how things change in time, it is also good ground to use explorative methods and case studies. Salk [5] review descriptions of the alliance processes of earlier studies and described put them into three stage model: 1) Formation (including selection of mode of entry, partner

selection, negotiation, structuring), 2) Operating (day-to-day operations and activities including learning), and 3) a Change or Discontinuation, which include instability and failure. The early stages of partnering process has gained the most of the focus, but we see that the management and the later stages at the process deserve increased attention as it has a major role in making the relationship work. The Figure 1 illustrates the partnering process model. The model is agglomerated view of the total partnering process. Each stage has its own sub processes that have been described in the literature. For example, the formation stage the partner selection process has been examined recently by e.g. Holmberg and Cummings. [17]

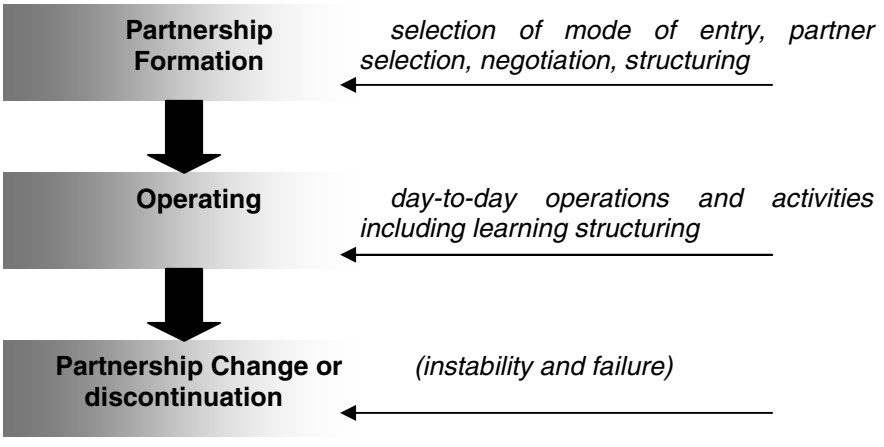

Fig. 1. Partnering process in global software business

2.2 Role of Product Strategy in Partnering

Software companies rely on a variety of product strategies. In principle, however, the companies fall into the categories of either standardized product or tailor-made "software project business", which are the polar opposites in the global software business. [18] [19]

Only few companies fall perfectly into either of the polar opposites because their solutions include both tangible and intangible parts [18] [19] It is also typical for software companies to shift their focus from tailor-made offerings towards more standardized products, and vice versa, especially during their first years of development. However, it is proposed that a product strategy of the firm has a significant impact on the partnering process. Depending on the type of the product (i.e. whether if it is e.g. a software service or packaged software), the need for a partner may increase and the available internationalization options differ. Hoch et al. [20] also look at the differences between the two extremes in the software industry, i.e. differences between professional services (projects) and the product business. Some of the differences in their survey are presented in Table 1. Professional services in the IT or software sector are normally provided by specific IT consultants (either small firms or firms like Accenture and Gap Gemini) or other software firms operating in the field of project business. The industry characteristics in this segment are closely related to that of

management consultants. The business is built around people, relationships and trust with the customers. The other extreme, the software product business is more "tangible business" and its products are more "physical" and exportable.

Table 1. Dynamics of professional service business vs. software product business (Partly based on Hoch et al, p. 46. [20]).

	PROFESSIONAL SERVICES ("PROJECTS")	PRODUCT BUSINESS
Marginal costs	Almost constant (service based on people, small economies of scale effect)	Almost zero (but high first copy costs).
Market structure	Highly fragmented, many actors with similar services.	Drive towards high concentration.
Regional appearance	Mainly regional, with increasing tendency to globalisation (through partnerships and networks)	Highly globalised
Customer relationship	One to one (service based on good customer relationships, trust etc.)	One to few, one to many (tailor-made software products vs. packaged products)
Need for support	Solutions often based on other actors' products: applications, platforms etc.	Mass-market software products stand normally alone. Enterprise solutions need installation, customisation, training etc. (often provided by professional service firms).
Most important number to watch	Capacity utilization rate	Market share (installed base) ("winner-takes-all-economy")
Examples of firms	Accenture, Gap Gemini, many small local players	Microsoft, Lotus, Corel in the mass-market segment. SAP and Baan in the enterprise solution segment.

Similar results were also got in the research conducted by the Telecom Business Research Center (TBRC) [21] in 1999-2000. In a survey consisting of 171 Finnish small and medium sized enterprises (SMEs) operating in the ICT domain packaged software producers proved to be the most international. They also had more customers and more target markets than the firms producing more tailor-made products (see e.g. Kuivalainen). [22] However, there were a lot of similarities in their goals as well: almost all the firms were willing to increase their internationalization and partnering efforts.

Many firms also aim towards rapid internationalization because of the quest for the market leadership in their new developing market segments. This phenomenon is

closely related to the concept of the "law of increasing returns" and to the concept of "killer application". Software business is said to be run by this "returns law"; this is especially important to the packaged software producers who are more able to gain economies of scale and enjoy low marginal costs.

The nature of the software business drives companies towards co-operation. Co-operation may offer one solution to respond to the changing challenges of this turbulent environment. The software market is a high-technology market and it is characterised by high levels of market and technological uncertainty. [23] Partnerships can help the firm to access new markets. [24] Smaller companies may get access to international distribution channels through partnerships with larger companies in the field.

2.3 Value Chain and Partnering Strategy

Oviatt and McDougall [25] state that (in addition to the number of countries entered) international new ventures may be distinguished by the *number of value chain activities* that are coordinated internationally. The literature has mostly emphasized the "outward" or "downstream" activities, i.e. marketing and selling of companies' products in export markets. For example, Varis et al. [15] also focused on "outward" end of value chain in their study on how new ventures select and use partners in their international marketing and distribution.

However, until recently the "inward" activities such as R&D and international sourcing have been left to a lesser attention within the international entrepreneurship literature. In the current "flattening world" it needs to be understood that internationalization of the firm is a comprehensive process that involves the whole value chain. [26] There is some available evidence, however, that partnering is already a reality among entrepreneurial growth-oriented SMEs. For example, according to the TBRC survey mentioned above, [21] 49% of the SMEs in Finnish ICT-sector had partnering relations and 53% of the partnerships were related to sales activities. 47% of the respondents considered that the partnerships were strategically very important for their companies, and also three fourths of the companies saw that they would need partnerships in the future. Moreover, 43% of the companies seek partnerships for internationalization. The companies seeking new partnerships agreed more with the argument that growth is only possible by internationalization, and these companies were generally more growth oriented. These companies also shared the opinion that there is not enough growth potential in the home market, that to succeed in the future it is essential to internationalize, and that they need a partner for internationalization. Consequently, prior research has found evidence that in a global business such as software the international partnering takes place throughout the value chain.

2.4 Proposed Contingency Framework

With the aim to 'unpack' the partnering strategy we utilize contingency approach to study partnering. The contingency approach intends to identify common settings and observe how different structures, strategies and behavioural processes suit each setting. [27] It stresses the importance of situational factors in company management and emphasizes the fact that solutions are situational rather than absolute and may become inappropriate under different environmental conditions. [28] However, in certain

contexts it should be possible to identify particular strategic actions which could be more beneficial than some others. [29] Companies are seen as problem solving entities, where decision makers undertake rational decision processes designed to cope with the complexity and uncertainty of the environment. As an example, Robertson and Chetty [30] saw that according to the contingency theory the export performance of the firm is determined by the extent to which its behaviour fits into its internal and/or external context.

The contingency approach suggests that variations in effectiveness are not random, but depend on the appropriate matching of contingency factors with internal organizational designs. [27] Companies have different features and environmental situations (resources, capabilities, markets) and there is probably no single best way to manage or organize in these different situations. This means also that different strategies should be designed for different environmental contexts. [31]

In this paper, as noted before, we analyze the partnering strategies of the internationalizing software SMEs based on two contingencies, i.e. product strategy of the firm and value chain activities that are coordinated through partners. It can be assumed that there are differences in the partnering activities (e.g. how systematic partnering is) based on the location of the firms in the contingency table presented in Figure 2. Consequently Figure 2 illustrates our contingency framework for analysis of partnering process in global software business.

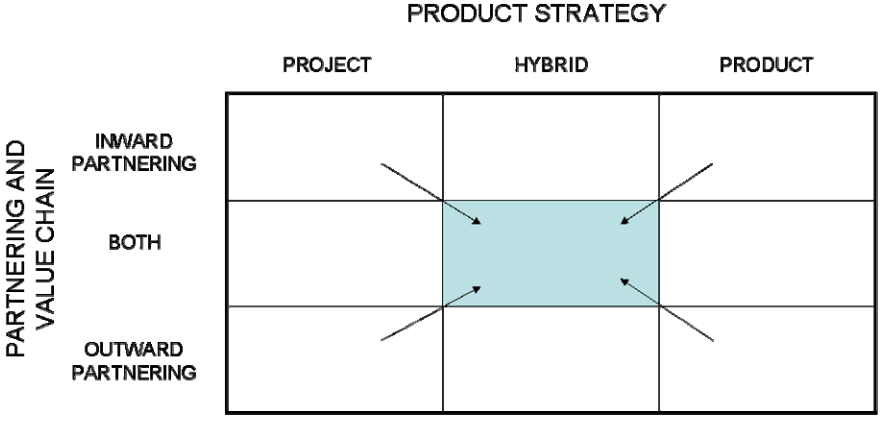

Fig. 2. A proposed contingency framework of the effect of product strategy and value chain activities on partnering process

Our proposition or a working hypothesis is that if a firm has a complex hybrid product strategy and it utilizes partners in all kinds of value chain activities, it should have clear-formulated partnering strategy to be able to be successful in its operations. In practice this would mean e.g. that there are tools for various partnering phases available and in use for partnering managers and partnering would be systematic and part of the organization culture. This is especially an important determinant of success whilst the internationalization process is ongoing, the challenge the focal software SMEs are facing in our empirical sample described in the next section of the paper.

3 Research Design

Research in the area of partnering strategy in internationalizing SMEs is still in its formative stages. Thus, in order to get an insight into this contemporary phenomenon within its real-life context [32] we adopted a multiple case-study approach in our exploration. We followed the principles of data collection set up by Eisenhardt [33] and [32], and used multiple sources of evidence in gathering our data. The primary data collection method used in this study was a series of in-depth interviews with the key informants in the companies (managing directors, marketing directors etc). In-depth interviewing is a time-consuming, costly data-collection technique relative to some other types of data collection such as mail or telephone surveying. However, for the purposes of this study, in-depth interviewing offered an opportunity to gather a rich database of open-ended responses to crucial questions about partnering strategy. The technique is also well grounded as "case studies might be the best means to understand how event unfold over time". [5]

Case selection is key decision in research process and should thus be made after consideration and critical assessment of alternatives. Random selection is neither necessary nor desirable, and theoretical sampling is recommended. Theoretical sampling is carried out with a view to choosing cases that are likely to replicate or extend the emergent theory. [33] The theoretical criteria also have to be considered, e.g. how well they fit the conceptual categories and what their explanatory power is (see e.g. Eisenhardt). [33] We wanted to study companies that would be comparable with the following characteristics:

- Are seen as entrepreneurial SMEs
- Have international activities in inward (R&D, production), outward (marketing, sales) activities, or both.
- Produce software offering either as standardized products or customer tailored services.

The data was collected in Finland between 2004 and 2006 by interviewing at least one key informant in each of the chosen 21 companies. The interviews lasted an hour and a half on average and were tape recorded. After transcribing the interview tapes each researcher became familiar with the data by reading through all the transcripts carefully and independently. The emerging partnering strategies and positioning of the companies in "nine windows" were identified from transcripts, and all the three researchers broadly agreed on them.

4 Findings

Our contingency framework divides different partnering strategies in nine strategic windows (see Table 2). Framework can be utilized in describing how the contingencies of software industry affect the partnering strategy. We present here, based on our empirical research, how the partnering strategies of companies vary depending on their product strategy.

Table 2. Case companies

	Project	Hybrid	Product
Inward partnering	1	1	
Both	1	9	3
Outward partnering		4	2

Window 1 (Project-Inward, 1 company):

In the first window of our framework, companies are concentrating almost merely to project type of selling and they are using mostly inward partnering. Partnering is outsourcing, technology development or other inward partnering activities. This is probably quite natural behavior of companies who are mainly active in project business. We found one case company to this window.

The case company in window 1, named W1a, is a smaller software producer selling complete software solutions for the financial and insurance industry, i.e. their customers are mainly large companies. They have used outsourcing in their software development as a mean to gain cost benefits. This has been done internationally from the quite beginning. This has lead to sales in the same target countries or they have followed their domestic customers to foreign markets.

Basically, case company W1a is a quite representative case of this kind of a product strategy which has lead to partnering strategy with inward partnering as supposed.

Window 2 (Hybrid-Inward, 1 company):

In this window, the companies are having a "hybrid" product strategy, i.e. selling both ready made, "packaged" products and also projects to their customers. Partnering strategy is more inward type relations.

The case company, named W2a, is already quite well established and internationalized company using partner network in product development. Actually, the case company base their products on open source and has a distributed, virtual organisational structure. Thus partnering has a crucial role in their operations, but has a bit different starting point.

Window 3 (Product-Inward, 0 companies):

This window 3 is representing companies who have software products and inward partnering. We did not find any case companies in our sample this window, which is also quite natural as companies who have productized their software do probably not need that much inward partners, but are maybe more interested in finding suitable distribution and marketing channels, and using partnering potentially for these purposes. Inward partnering is probably not an acute issue for these companies.

Window 4 (Project - Both inward and outward, 1 company):

This window is representing project business companies who have both inward and outward partnering relations.

The only case company in this window, W4a, has a software platform which is tailored for each customer. The company has international partners both in technology development and on the distribution side.

Window 5 (Hybrid - Both inward and outward, 9 companies):

Companies in this window are supposed to have a hybrid offering on product side and also both kinds of partnering relations. This window has the biggest group of companies which can be also understood as the companies in the sample are relatively young and they may not have yet reached maturity in their focus and strategy. Also the business area is still developing and it is not so easy to classify companies. Actually, in one interview a company manager was referring to this complexity as follows: "When the product has been designed as glue into this heterogenic mess, so the expertise in clueing. These are very difficult issues and we our selves are trying to figure out this year what we really are. Customers know from their own point of view what it is all about. As we are a solution provider who is making a product. And we are a company with products that produces components. We talk about a mystic niche which does not exist yet."

All in all, this window presents a strategic window which in probably most common for young software firms as they are still in the middle of developing their products and offerings, and also their business is in early stages of development.

Window 6 (product - Both inward and outward, 3 companies):

In the sixth window companies have a product to offer and are using both inward and outward partnerships. There are three companies in the sample representing this kind of position. The first case company, W6a, is a smaller firm, but describes itself as global company (they have activities in 54 countries), and they have used both outsourcing partnerships and marketing partnerships. The other two case companies have already grown and have quite established positions in the market. They also have clear product offerings and are actively using partnering (technology partners, partner programs and value-adding resellers).

Window 7 (Project - Outward, 0 companies):

As well as Window 3, this window is empty, i.e. has no case companies. This can be explained theoretically as companies seemingly have project type of offering and probably need to keep the customer contacts in their own hands. The strategic position that this window is representing is probably not so common in the software business.

Window 8 (Hybrid - Outward, 4 companies):

Window 8 has companies with both project and product offerings but concentrating in outward partnering. In this window case companies vary in size and style, i.e. they are actively selling to quite different fields (e.g. mobile technology, building industry, security applications). Companies clearly have both project type business (tailoring) and product sales.

The first case company in this window (W8a) have partners that act as consultants and technology developers. The second company (W8b) has outward partnering contracts with several large companies. Case company W8c has sales partners in seven countries, and they are waiting for strong growth through one major player in the software field. Last case company W8d has grown very fast during the last years and

it has widely spread international activities. They describe their offering as a mass-tailored product. All the case companies in this window are using partners in international sales.

Window 9 (Product - Outward, 2 companies):

The window 9 was supposed to represent one quite natural strategic window for software companies. Companies in this window have packaged products (as one interviewee referred "a cellophane product") and are organizing their sales with partners. The small number of case companies in this window can probably be explained by the difficulty to develop this kind of standardised software products. The relative smallness and young age of the companies can be one explaining factor.

Cross-case analysis

We can sum up that the heterogeneity of the software business and the complexity of the offerings are making the classification of companies into the different windows quite demanding task. On the other hand, we could find representative case companies to most windows and we can illustrate the changes in partnering strategies according to product strategies. The case companies could be analyzed in more detail to get more thorough analyse, but clearly we can point out certain empirical evidence to our theoretical assumptions and our framework gives a starting point for further studies about partnering strategies, processes and partner management.

5 Discussion and Conclusions

Our aim was to unpack the partnering strategy of software SMEs which are internationalizing and utilizing partners in this endeavour. Although there is an abundance of literature on international partnerships and alliances the alternative strategies in managing the process of partnering is perhaps under-researched and poorly documented compared to other areas of inter-firm collaboration and especially in the context of SMEs. In this paper we have developed a contingency framework for analysis of partnering strategies in software business, a context where international entrepreneurship is highly manifest. Our framework consists of three-by-three matrix with the following dimensions: The *product strategy* of the firm (product, project or hybrid) and the *value chain activities* (inward, outward or both) that are coordinated through partners.

Our empirical analysis was based on a multiple case study of 21 companies. Our preliminary results show that there are differences in partnering strategies based on the contingencies utilized in the study. While some of our results are consistent with earlier research on partnering, many of our findings extend previous research and offer some counterintuitive insights. The first major observation about our results is that most of firms (nine firms) actually were operating in the window number 5 in the middle, i.e. they had both hybrid products and were utilizing both inward and outward partners. This type of position is naturally challenging for a SME as it has to be able to handle 'almost everything'. If we reflect this result to e.g. the classic Oviatt and McDougall [25] typology of international new ventures, it is clear that when

internationalizing this position is the most challenging. Why then are many firms utilizing or aiming for this type of position? One reason for this result can be seen to stem from another contextual factor, i.e. the software industry. The complex and global nature of the software industry leads firms to internationalize and product development towards complete packaged product is not easy. Partners (e.g. in open software development) are often sourced globally by necessity, for example. In addition it can be seen that the offerings of the software companies actually change and develop over the time and in some phases the service element is more important than in others (cf. [19], for example).

Second observation relates to the companies which were most international. The companies which had standardized products seem to be more international than their service/project based counterparts. This result is not surprising and supports earlier findings related to internationalization of services or software services in general (see e.g [34], [22]). However, further studies should dig deeper regarding partnering strategy and process in this development. Both preliminary findings are relevant, or in other words, have useful implications for managers. First, partnering strategy should be planned to fit to the product development strategy and second, internationalization process. Based on the several interviews we came across to the phases in partnering in the companies. In most of the cases the process was more evolving and ad hoc –type of partnering: the challenge lays in the more systematic development of the partnering process. This comes through the experience but for practicing managers and public policy makers/support service providers the key would be to make this process speedier; how to enhance the learning before doing is the challenge in this.

This study needs to taken further in the future. The two key areas for further research are a) the more thorough analysis of the partnering process and b) partnering performance. Cavusgil [12] has emphasized the performance outcomes in various publications but e.g. Contactor (2005) notes that the measures of partnership or alliance performance are still underdeveloped. For us one of the key tasks in the future is to study in in-depth manner how the international partnering performance differs in different contingency windows presented in this study. To conclude, we hope that this study will spur future researchers within the international entrepreneurship domain to explore further the partnering strategies that the entrepreneurial firms use in their efforts to achieve success in international business.

References

[1] Ellram, L.M., Hendrick, T.E.: Partnering characteristics: a dyadic perspective. Journal of Business Logistics 16(1), 41–64 (1995)

[2] Park, S.H., Ungson, G.R.: The effect of national culture, organizational complementarity and economic motivation on joint venture dissolution. Academy of Management Journal 40(2), 279–307 (1997)

[3] McDougall, P.P., Oviatt, B.M.: International entrepreneurship: the intersection of two research paths. Academy of Management Journal 43(5), 902–906 (2000)

[4] Contractor. Alliance Structure and Process: Will the Two Research Streams Ever Meet in Alliance Research? European Management Review 2(2), 123–129 (2005)

[5] Salk, J.E.: Often called for but rarely chosen: alliance research that directly studies process. European Management Review 2(2), 117–122 (2005)

[6] Welch, L.S., Benito, R.G., Petersen, B.: Foreign operation methods: theory, analysis, strategy, UK, Edward Elgar (2007)
[7] Jarillo, J.C.: Entrepreneurship and growth: The strategic use of external resources. Journal of Business Venturing 4(2), 133–147 (1989)
[8] Stinchcombe, A.L.: Social Structure and Organizations. In: March, J.G. (ed.) Handbook of Organizations, pp. 142–193. Rand McNally & Company, Chicago (1965)
[9] Hymer, S.H.: The international operations of national firms: A study of foreign direct investment. MIT Press, Cambridge (1976)
[10] Sharma, D.D., Blomstermo, A.: The internationalization process of born globals: a network view. International Business Review 12(6), 739–753 (2003)
[11] Gabrielsson, M., Kirpalani, V.H.: Born globals: how to reach new business space rapidly. International Business Review 13(5), 555–571 (2004)
[12] Cavusgil, T.S.: Executive Insights: International Partnering – A systematic Framework for Collaboration with Foreign Business Partners. Journal of International Marketing 6(1), 91–107 (1998)
[13] Ellram, L.M.: A Managerial Guideline for the Development and Implementation of Purchasing Partnerships. The International Journal of Purchasing and Materials Management 27(3), 2–7 (1991)
[14] Dyer, J.H., Singh, H.: The relational view: Cooperative strategy and sources of interorganizational competitive advantage. Academy of Management Review 23(4), 660–679 (1998)
[15] Varis, J., Kuivalainen, O., Saarenketo, S.: Partner Selection for International Marketing and Distribution in Corporate New Ventures. Journal of International Entrepreneurship 3(1), 19–36 (2005)
[16] Ruokonen, M., et al.: Global Network Management – Ideas and Tools for ICT firms to thrive in international network environment. Technology Business Research Center. Lappeenranta University of Technology (2008)
[17] Holmberg, S.R., Cummings, J.L.: Building Successful Strategic Alliances -Strategic Process and Analytical Tool for Selecting Partner Industries and Firms. Long Range Planning 42, 164–193 (2009)
[18] Alajoutsijärvi, K., Mannermaa, K., Tikkanen, H.: Customer relationships and the small software firm: a framework for understanding challenges faced in marketing. Information & Management 37(3), 153–159 (2000)
[19] Cusumano, M.A.: The Business of Software – What every Manager, Programmer and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad. Free Press, New York (2004)
[20] Hoch, D., Roeding, C., Purkert, G., Lindner, S.: Secrets of Software Success: Management Insights from 100 Software Firms around the World. Harvard Business School Press, Boston (2000)
[21] Puumalainen, K., et al.: Tietoliikennetoimialan PK-lisäarvopalvelutuottajat Suomessa – Tutkimusraportti (published in Finnish. English name: Small and Medium –Sized Value-adding Service Providers in the ICT-sector in Finland – Research Report), Telecom Business Research Center, Research Report 3, Lappeenranta (2001)
[22] Kuivalainen, O.: Impact of Product Characteristics in the Internationalisation Process of the Born Globals – The Case of the Finnish Telecommunication and Information Technology Software Suppliers and Content Providers. In: During, W.E., Oakey, R., Kauser, S. (eds.) New Technology-based Firms in the New Millennium, pp. 26–41. Pergamon Press, Oxford (2001)

[23] Moriarty, R.T., Kosnik, T.J.: High-Tech Marketing: Concepts, Continuity and Change. Sloan Management Review 30(4), 7–18 (1989)
[24] Mohr, J., Spekman, R.: Characteristics of Partnership Success: Partnership Attributes, Communication Behaviour and Conflict Resolution Techniques. Strategic Management Journal 15, 135–152 (1994)
[25] Oviatt, B.M., McDougall, P.P.: Toward a Theory of International New Ventures. Journal of International Business Studies 25(1), 45–64 (1994)
[26] Servais, P., Zucchella, A., Palamara, G.: International entrepreneurship and sourcing: International value chain of small firms. Journal of Euromarketing 16(1/2), 105–117 (2006)
[27] Zeithaml, V.A., Varadarajan, P., Zeithaml, C.P.: The Contingency Approach: Its Foundations and Relevance to Theory Building and Research in Marketing. European Journal of Marketing 22(7), 37–64 (1988)
[28] Wright, M., Ashill, N.: A Contingency Model of Marketing Information. European Journal of Marketing 32(1/2), 125–144 (1998)
[29] Walters, P.G., Samiee, S.: A Model for Assessing Performance in Small U.S. Exporting Firms. Entrepreneurship: Theory and Practice 15(2), 33–50 (1990)
[30] Robertson, C., Chetty, S.K.: A Contingency-based Approach to Understanding Export Performance. International Business Review 9, 211–235 (2000)
[31] Gardner, D.M., Johnson, F., Lee, M., Wilkinson, I.: A Contingency Approach to Marketing High Technology Products. European Journal of Marketing 34(9/10), 1053–1077 (2000)
[32] Yin, R.K.: Case Study Research, Design and Methods. Sage, Newburu Park (1989)
[33] Eisenhardt, K.M.: Building theories from case study research. The Academy of Management Review 14(4), 532–550 (1989)
[34] Erramilli, M.K.: Entry Mode Choice in Service Industries. International Marketing Review 7(5), 50–62 (1990)
[35] Contractor. Alliance Structure and Process: Will the Two Research Streams Ever Meet in Alliance Research? European Management Review 2(2), 123–129 (2005)

Developing a Maturity Matrix for Software Product Management

Inge van de Weerd, Willem Bekkers, and Sjaak Brinkkemper

Department of Information and Computing Sciences, Utrecht University,
Padualaan 14, 3584CH Utrecht The Netherlands
{i.vandeweerd,bekkers,s.brinkkemper}@cs.uu.nl

Abstract. The quality of processes in Software Product Management (SPM) has a high impact on the success of a software product, as it improves product quality and prevents release delays. To improve the SPM practice, we propose the maturity matrix for SPM, a focus area oriented maturity model concentrating on the SPM functions Requirements Management, Release Planning, Product Roadmapping, and Portfolio Management. In this paper, we describe the development of the SPM maturity matrix, consisting of (a) identification and description of capabilities, (b) positioning the capabilities at the right levels in the maturity matrix and (c) validating the maturity matrix with expert validation and a survey among 45 product managers and product management experts. The result is a validated maturity matrix that will guide further development of methodical support in SPM.

Keywords: software product management, maturity, requirements management.

1 Maturity in Software Product Management

Software product management (SPM) is a crucial area within many software companies. Good product management has a high impact on the success of a software product [1]. This requires a combination of technological, managerial and business skills, such as calculating optimal releases, setting out roadmaps, managing risks, and interacting with many internal and external stakeholders. If these activities do not get enough attention, the quality of a product decreases, release dates are not met, and managing customers' expectations become a large problem.

Although the product manager's function is highly important in the product software industry, little education exists in this area [2]. Almost no education on SPM is being offered, except in the area of marketing and sales. Based on our experience in the market, especially in The Netherlands, Germany and Switzerland, we observe that most software product managers were earlier employed in functions such as development manager, project manager or sales manager. This causes a gap of knowledge that the product manager has to solve by getting experienced in the area. Hence, lifting the quality of the product by improving the SPM processes is often difficult. Most existing software process improvement (SPI) models aim at a broad spectrum of SPI and the area of SPM is usually not the main area of attention.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 76–89, 2010.

In this research, we propose a maturity matrix for SPM that can be used to assess an organization's current SPM capabilities and offer local, incremental improvements to the product manager.

In earlier research, we developed the Reference Framework for Software Product Management [2]. Since its publication, various studies have been done to test the reference framework in product software companies (cf. [3] and [4]). In this research, we use the reference framework as a foundation for our maturity matrix. Therefore, we will provide a brief explanation of the framework.

In Figure 1, the reference framework for software product management is depicted. The framework consists of *internal stakeholders* (product management, company board, sales & marketing, services, support, development and research & innovation) and *external stakeholders* (the market, partners and customers).

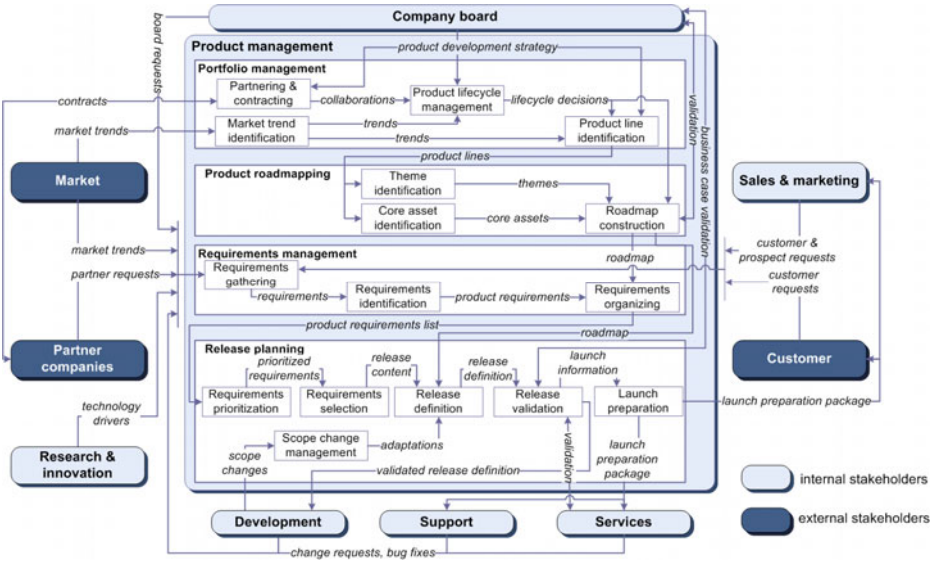

Fig. 1. Reference framework for Software Product Management

The most important internal stakeholder, Product management, consists of four business functions: (1) *Portfolio management* concerns managing the different products that a company owns. Partnering, product lifecycle management and product line identification are part of this function. (2) *Product roadmapping* handles with the development of the product roadmap, in which future releases are planned based on themes and core assets. (3) *Requirements management* contains the activities of requirements gathering, identification and organizing; all ongoing activities within the product management domain. (4) *Release planning* deals with prioritizing and selecting requirements in order to define the new release. Also the activities release validation, launch preparation, and scope change management are part of this function.

2 Research Design

This study follows the design science methodology, in which research is done through the processes of building and evaluating artifacts [5]. The artifact in this research is the maturity matrix for SPM. During our research, we follow the 5 process steps of the design cycle [6]. This design cycle consists of several steps that follow an iterative process; knowledge produced in the process by constructing and evaluating the artifact is used as an input for a better awareness of the problem. The 5 process steps are: (1) *Awareness of the problem***.** In Section 1, we described the problem and its context. (2) *Suggestion.* The suggestion for a solution to the problem identified in step 1 is developed in this step. In Section 2, we describe our approach in tackling the problem and the research methods that we use. (3) *Development.* The development of the artifact, in this case the maturity matrix is described in Section 3. (4) *Evaluation.* This step comprises the evaluation of the method. We used a survey to validate the method, as is described in Section 4. The results of this survey lead to a higher level of problem awareness and suggestions for solutions. (5) *Conclusion.* Finally, in Section 5, conclusions and areas for further research are covered.

In Fig. 2, we depict the structures of the two artifacts in this research. The REFERENCE FRAMEWORK consists of KEY PROCESSES that are grouped into BUSINESS FUNCTIONS. Secondly, the MATURITY MATRIX consists of KEY PROCESSES and SPM CAPABILITIES. Each SPM CAPABILITY contributes to a KEY PROCESSES and it indicates which MATUIRTY LEVEL this process has.

In addition to the artifact structure, the research methods used during the development of both artifacts are provided. At the left, the research methods that were used for developing the Reference framework are listed and at the right the method for developing the Maturity matrix are listed.

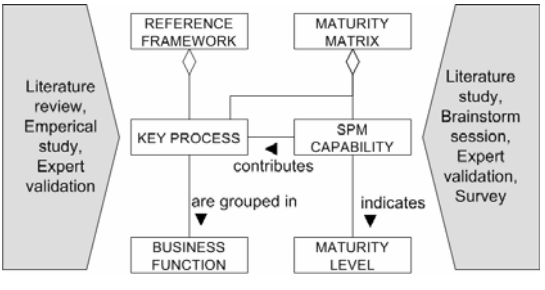

Fig. 2. Research methods and artifacts

As shown in Fig. 2, this research was conducted with the following data collection methods:

Literature study. One of the sources for the capabilities, which are defined for each of the processes in the reference framework for SPM, is a literature study.

Brainstorm session. A brainstorm session was conducted with experts from the scientific community to create the model. The session consists of two parts: 1) the capabilities themselves were determined; 2) the positions of the capabilities in the SPM

maturity matrix were determined. The literature study was used as a basis for the brainstorm session.

Expert validation. An expert from practice validated the results of the brainstorm session: the capabilities themselves and their position within the SPM Maturity Matrix.

Survey. A final validation was conducted based on a survey with SPM experts from practice from all over the world. The goal of this survey was to validate the order of the capabilities relative to each other in the SPM Maturity Matrix.

3 Developing a Maturity Matrix

In this section, we first describe our choice for the type of maturity matrix we use, and then we describe its structure and the development process.

3.1 Variance of Maturity Models

Van Steenbergen et al. [10] recognize three variants of maturity models: (1) *staged 5-level models*, which distinguish five levels of maturity, which in turn have a number of focus areas that are defined specific to that level; (2) *continuous 5-level models,* which contain a number of focus areas, in each area the 5 levels are distinguished; and (3) *focus area oriented models*, in which each focus area has its own number of specific maturity level.

Most well-known maturity models are staged or continuous 5-level models, such as the Capability Maturity Model (CMM) [11], and its follow-up CMMI [12]. Earlier research into the improvement of SPM shows some shortcomings in these methods. CMMI for example, has been found too heavy to use by several organizations [13]. And there are others who say that extensive software process improvement (SPI) frameworks, such as CMMI and ISO/IEC 15504 (SPICE) [14] are too large to implement, or even comprehend [15] [16]. For example, a typical CMM SPI cycle can take between one and a half and two years to complete. It also requires large resources and long term commitment [17], which can be a problem for small and medium companies. Another problem is that small and medium software companies often not only lack the funds required to implement many of the practices from CMM but also have to base their SPI initiatives on practices that do not apply to them [18].

For the reasons above, we choose to develop a focus area oriented model, in order to make local analysis and incremental improvement possible. Similar model have been used for the testing domain [19] and the architecture domain [10].

3.2 Structure of the Maturity Matrix

In Table 1, the SPM maturity matrix is depicted. The matrix consists of columns and rows, which represent the two dimensions of the model. The columns 0 to 12 represent the different maturity levels for the model, where 0 is the lowest level of maturity, and 12 the highest. The SPM processes, or focus areas as they are called in the matrix, are represented by the rows and are divided into four groups (the business functions; ‘Requirements management’, ‘Release planning’, ‘Product roadmapping’,

'Portfolio management'). When a focus area is carried out at a certain maturity level it is called a capability. In Table 1, we can for example identify four capabilities: A, B, C and D. Capability A, located on level 1, is coded as RM1A and described as: "Ad hoc requirements gathering. Requirements are being gathered and registered."

Two more concepts need to be introduced:

Intra-process capability dependency. This is the dependency of one capability within a certain key process to another capability in the same key process. In Table 1 this type of dependency is depicted with an arrow between RM1B and RM1C.

Inter-process capability dependency. Intra-process refers to the dependency of a capability in a certain key process to a capability in another key process. In Table 1 this is depicted with an arrow between RM1D and RM3C.

A more detailed explanation on the structure of the matrix can be found in [7].

Table 1. Maturity Matrix Structure

	0	1	2	3	4	5	6	7	8	9	10	11	12
RM1. Requirements gathering		A		B	←	→	C			D			
RM2. Requirements identification			A			B			C	↘			
RM3. Requirements organizing			A				B				C		
RP1. Requirements prioritization				A		B	C			D			
RP2. Requirements selection				A		B		C			D		
RP3. Release definition			A			B		C					
RP4. Release validation				A			B			C		D	
RP5. Launch preparation		A				B					C		
RP6. Scope change management				A		B				C			
PR1. Theme identification			A		B		C						
PR2. Core asset identification						A			B				C
PR3. Roadmap construction			A			B				C			
PM1. Market trend identification			A		B				C				
PM2. Partnering & contracting			A				B		C				
PM3. Product lifecycle management			A			B			C				
PM4. Product line identification				A			B			C			

3.3 Developing a Maturity Matrix for Software Product Management

We followed three main steps during the development of the maturity matrix:

1. Identification and description of capabilities

First, a literature study was carried out. This literature study was based on a multitude of papers describing specific processes within the field of SPM, e.g. [8] and [9]. Then, a brainstorm session with four SPM experts was held to identify the capabilities. Two of them had extensive professional experience in SPM and the other two were researchers to SPM.

2. Positioning the capabilities in the maturity matrix

The next step concerned positioning the capabilities in the matrix. By analyzing the inter- and intra-process capability dependencies, we decided the order of the

capabilities in the matrix. For example, for the first capability in the process in Requirements identification (RM2A), it makes sense to have gathered and registered requirements, which is capability RM1A. Therefore, RM2A must be placed at least 1 level after RM1A. This activity resulted in a matrix of 13 levels (0-12). During the validation, we will find out whether this is the right size.

3. Validating the maturity matrix

The third and last step is the empirical validation of the matrix. In Section 4, we describe how we use a survey to validate the positions of the capabilities.

4 Survey

Our survey consists of three parts: Introduction questions, general questions and capability questions. It starts with two introduction questions: "Which SPM areas are you familiar with?" and "How are you related to SPM?" The answers to these two questions determine which questions will be presented in the remainder of the survey. First, the respondent can choose which of the four SPM areas will be included in the survey. Only SPM-areas of which the checkboxes are ticked will be included in the survey. The second question is used to find out whether the respondent is a software product manager or another SPM professional. After the introduction question, some general questions are posed concerning company size, experience, etc.

The main structure of our survey is based on the four business functions that are defined in the reference framework for SPM: requirements management, release planning, product roadmapping, and portfolio management. For each function, we asked how our identified capabilities should be implemented in an organization. If we would ask to fill in a whole matrix per area, we would get very large matrices. For example, the release planning area has 21 capabilities and 12 rows. During the first pilots it appeared this would cause a cognitive workload that was too high. Therefore, we decided to use another approach and divided the matrix in three sub matrices. In the first matrix, ranging from 1 to 6, capabilities A and B are covered. In the second matrix, ranging from 4 to 9, capabilities B and C are covered. Finally, if necessary, a third matrix is used to cover capabilities C and D. In Table 2, part of the matrix is depicted, showing the division in three separate matrices.

Table 2. Matrix Division

	0	1	2	3	4	5	6	7	8	9	10	11	12
RM1. Requirements gathering													
RM2. Requirements identification													
RM3. Requirements organizing													

In Figure 3, we depict part of the survey in which all capabilities A and B of the Requirements Management process area are listed. The capabilities are listed on the left and the levels (1-6) in the middle. The last columns can be used to indicate whether the respondent has implemented the capability. The N/A option is for non-product managers that execute the survey.

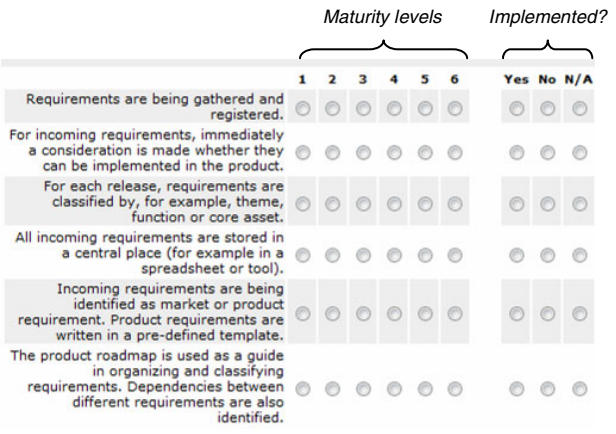

Fig. 3. Survey Questions Requirements Management

4.1 Data Collection

In order to find respondents to the survey, we posted a message to several mailing lists. Below, you can find an overview of the members of these mailing lists:

- Netherlands product software mailing list: 175
- Netherlands SPM community: 68
- International SPM community: 176

In addition, during a national meeting among 20 Dutch product management professionals, the survey was promoted. Finally, the URL of the survey was posted on the SPM community website [20]. Please note that the members of the different mailing lists, visitors to the meeting, and visitors of the website overlap.

The number of respondent to the survey is 84. Of these 84 only 45 were useful, i.e. they did not quit the survey after the first two pages. Not all questions have the same amount of responses. The respondents can make a choice in the beginning of the survey, for which they can indicate on which SPM area they can answer questions. Per SPM area, we have the following amounts of respondents:

- Requirements management: 42
- Release planning: 27
- Product roadmapping: 25
- Portfolio management: 19

The respondents originate mostly from the Netherlands (36). Other individuals originate from Germany, Canada, USA, India, South Africa, Sweden, Switzerland and Spain. 13 of the respondents followed a course or study in SPM and three are certified in SPM. In Table 3, some characteristics concerning the respondents' function, company size (only product managers included) and experience are provided.

Table 3. Respondents' Characteristics

Function		Company size		Experience	
Product manager	30	Less than 50	12	0-2 years	13
Researcher	5	50 - 250	7	3-5 years	12
Consultant	3	More than 250	11	6-10 years	13
Other	7			More than 10 years	7

4.2 Data Analysis

In this section, we will analyze the results from the survey. For each business function, we describe the results of the survey and compare this with our initial maturity matrix. In principle, we use the survey results to update our maturity matrix. However, we make one exception. Dependencies between capabilities in the different business functions were not part of the survey. Therefore, in case a capability is placed on a certain maturity level because of a dependency to a capability in another business function, we keep to the original sequence.

We use box plots to give a graphical overview of the distribution of each capability, as is illustrated in Figure 4.

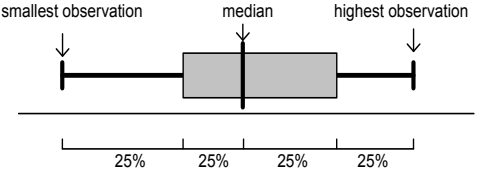

Fig. 4. Box plot

The box plots show the median, smallest and highest observations and the distribution over the quartiles. In addition, the outliers (if present) are shown. We use the medians as an indication of the maturity level of a capability, because we want to know which level was chosen most for a certain capability.

We list all capabilities within one business function in a box plot. At the Y-axis, the different capabilities are listed and at the X-axis, the maturity levels are indicated. Please note that some medians are not integers, but rational numbers. E.g. RM1C has a median of 5.5. The reason for this is that the respondents had to enter these maturity levels twice (cf. Section 4.1). Sometimes this resulted in two different answers, of which we calculated the mean. In such cases, we look at the distribution of the box plot to identify at which level the capability should be placed (in RM1C that means that the capability is placed on maturity level 6).

4.2.1 Requirements Management

In Fig. 5, the results of the Requirements management survey part are depicted in a box plot. In Table 4, the matrix is illustrated with the original results, and the deviations that were found when comparing it with the box plot. We also indicate the status

of a capability. "A" means that the result of the survey was the same as the original position, "A" indicates the result of the survey that was different from the original capability, and "A" shows the old position of the capability.

Several issues are noteworthy. Firstly, the intra-process capability dependencies are the same; all capabilities are positioned in the sequence A, B, C, (D). For the inter-process capability dependencies we see some differences that we will implement in the final matrix.

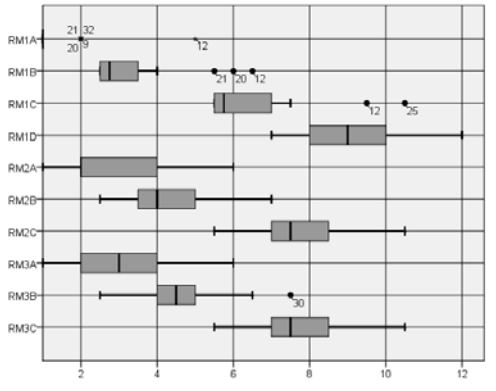

Fig. 5. Box plot Requirements Management

Table 4. Requirements management matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
RM1		A		B			C			D			
RM2			A		B	B			C				
RM3			A	A		B	B		C		C		

4.2.2 Release Planning

In Fig. 6 and Table 5, the results of the Release planning part of the survey are illustrated and compared with the original matrix. Again, the intra-process capability dependencies are the same. However, in the inter-process capability dependencies we see many small deviations. All of which have been incorporated into the matrix. One issue stands out: The prioritizing and selection capabilities (RP1A and RP2A) are

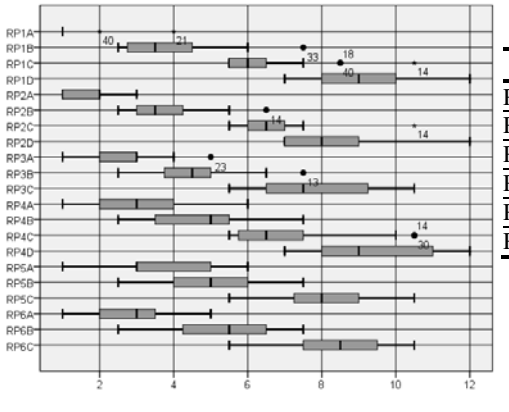

Fig. 6. Box plot release planning

Table 5. Release planning matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
RP1		A		A	B	B	C			D			
RP2			A	A	B	B		C	D		D		
RP3			A	A	B	B		C	C				
RP4				A		B	B	C		D		D	
RP5		A		A		B			C				
RP6				A		B				C			

implemented earlier than all other capabilities. Apparently, the respondents consider these activities as the minimum that a product manager should implement.

4.2.3 Product Roadmapping

The results of the Product roadmapping part of the survey are illustrated in Fig. 7 and compared with the original matrix in Table 6. Also for Product roadmapping, the intra-process capability dependencies are analogous to the original matrix. In the inter-process capability dependencies we see some deviations that will be included in the matrix.

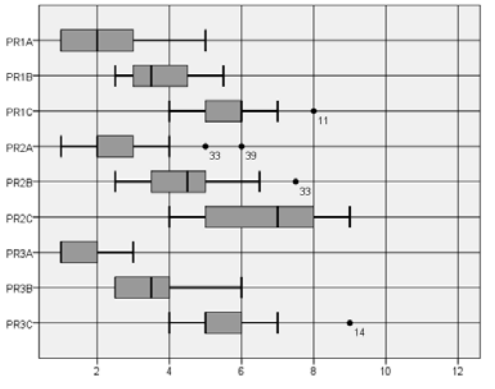

Fig. 7. Box plot product roadmapping

Table 6. Product roadmapping matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
PR1			A		B		C						
PR2			A		B	B		C					C
PR3		A	A	B		C				C			

4.2.4 Portfolio Management

Finally, in Fig. 8 and Table 7, the results of the Portfolio management part of the survey are illustrated and compared with the original matrix. Again, the intra-process capability dependencies are the same as in the original matrix. In the inter-process capability dependencies we see a few small deviations, which we will include in the matrix.

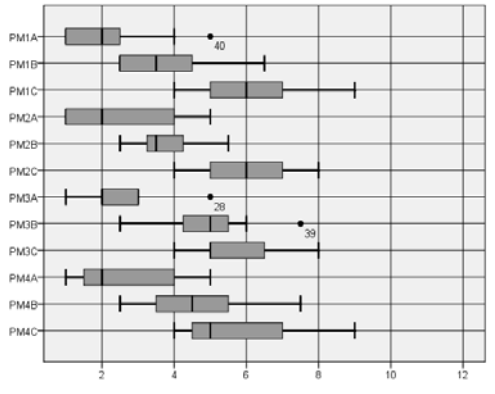

Fig. 8. Box plot portfolio management

Table 7. Portfolio management matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
PM1			A		B		C		C				
PM2			A		B		C		C				
PM3			A			B	C		C				
PM4			A	A		B	C			C			

4.3 Results

In Table 8, the final maturity matrix for SPM is presented. We made some adaptations compared to the survey results. First, all capabilities of the Product roadmapping business function have all been shifted one level to the right. The dependencies between business functions were subject to the survey and we believe that the product roadmapping capabilities should be on a higher level. Finally, several capabilities that are placed on the highest maturity level of a certain area have been shifted to the right, namely RP2D, RP4D, PR2C and PR3C and all the C-capabilities of the Portfolio Management function. Reason for this is that we do not only want to incorporate the fastest way to implement the capabilities in our matrix, but we also would like to provide a balanced way to improvement.

In Table 8, we also show the score for one of the respondents in the benchmark questions. This respondent works for a company with less than 50 employees from the Netherlands. He did not have any education or certification in SPM. The overall SPM maturity level is determined by checking for which maturity level all capabilities (and the preceding capabilities) are implemented. As can be seen in the table, the maturity level in this case is 3. To advance to a higher maturity level, say level 4, work needs to be done in RM2, RP3, PM1 and PM2.

Table 8. Maturity matrix for Software Product Management

	0	1	2	3	4	5	6	7	8	9	10	11	12
RM1. Requirements gathering		A		B			C			D			
RM2. Requirements identification			A		B				C				
RM3. Requirements organizing				A		B			C				
RP1. Requirements prioritization		A			B		C			D			
RP2. Requirements selection			A		B			C			D		
RP3. Release definition				A	B				C				
RP4. Release validation				A		B		C				D	
RP5. Launch preparation				A		B			C				
RP6. Scope change management				A		B				C			
PR1. Theme identification				A		B		C					
PR2. Core asset identification				A		B							C
PR3. Roadmap construction			A		B						C		
PM1. Market trend identification			A		B			C					
PM2. Partnering & contracting			A		B			C					
PM3. Product lifecycle management			A			B		C					
PM4. Product line identification			A			B		C					

4.4 Threats to Validity

Validity refers to the degree to which a response measures what we intend for it to measure [21]. We identified three types of validity threats that are important to discuss, namely threats to conclusion validity, construct validity, and external validity. Internal validity is less important since this study is not about establishing a causal relationship. It is a descriptive study, in which we describe the mostly used implementation sequence of SPM capabilities.

Firstly, conclusion validity is the degree to which conclusions we reach about relationships in the data are reasonable. Important is the composition of participants and the statistical analysis. At the beginning of the survey, respondents were asked to indicate with which of the SPM areas they were familiar. The respondents only got to answer questions about these areas. Therefore, we believe that the respondents were experienced enough to answer the questions they selected. Most of the respondents are not anonymously, since an email address is needed to send them the benchmark report. However, there is no reason to believe that they have not answered according to their best knowledge, since there is no social pressure to answer in a particular way. Therefore, we consider the conclusion validity not to be critical.

Construct validity concerns whether we measure the construct that we believe we measure. On the introduction page of the survey, we have provided all definitions on all important concepts used in the questionnaire. Hence, we minimized the threat that participants interpret concepts differently. However, it is not possible to completely eliminate this threat.

External validity concerns generalization of the results to other groups and contexts than the one studied. The survey was sent to different groups of SPM professionals, as is described in Section 4.2. The respondents are mainly product managers with varying experience, working for companies of varying sizes. Most of the respondents are from the Netherlands. A drawback is that only 46 respondents filled out the survey. However, given the diverse background of the respondents concerning company size and experience, we believe that the validity is high within the product manager population in the Netherlands. Since we believe SPM is not particularly influenced by culture, we believe that we can also generalize the results to other countries. To be sure, more research is needed.

5 Conclusions and Future Research

In this paper, we described the development of a maturity framework for SPM that can be used to assess an organization's current SPM capabilities and offer local, incremental improvements to the product manager. After developing the matrix, we used a survey in which 45 SPM professionals reported by indicating the right order in which SPM capabilities should be implemented in an organization. The results of this survey supported our initial positioning of the SPM capabilities, at least, when looking at the inter-process capability dependencies. Concerning the intra-process capability dependencies, several deviations were found, of which most of them have been used to improve the matrix. Currently, we are carrying out multiple case studies to refine the capabilities.

We believe that the maturity matrix for SPM is a valuable tool for process assessment in SPM. In addition, the matrix can also be used as a means to achieve process improvement, for example by using it in the Product Software Knowledge Infrastructure (PSKI) [22], which creates a process advice for product software companies by taking its situational factors into account [23]. In future research, we will carry out case studies at different product software companies to further validate and refine the matrix in SPM capability improvement.

References

1. Ebert, C.: The Impacts of Software Product Management. Journal of Systems and Software 80(6), 850–861 (2007)
2. van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: Towards a Reference Framework for Software Product Management. In: Proceedings of the 14th Int. Requirements Engineering Conference, Minneapolis/St. Paul, USA, pp. 312–315 (2006)
3. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: The Relevance of Situational Factors in Software Product Management. Technical report, UU-CS-2008-016, Utrecht University (2008)
4. van de Weerd, I., Brinkkemper, S., Versendaal, J.: Concepts for Incremental Method Evolution: Empirical Exploration and Validation in Requirements Management. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 469–484. Springer, Heidelberg (2007)
5. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
6. Vaishnavi, V., Kuechler, B.: Design Research in Information Systems, AISWorld Net: `http://www.isworld.org/Researchdesign/drisISworld.htm` (retrieved December 7, 2007)
7. van Steenbergen, M., Bos, R., Brinkkemper, S., van de Weerd, I., Bekkers, W.: The Design of Focus Area Maturity Models. In: Proceedings of DESRIST 2010: Global Perspectives on Design Science Research, St.Gallen, Switzerland, June 4-5 (2010)
8. Abramovici, M., Soeg, O.C.: Status and Development Trends of Product Lifecycle Management Systems. Ruhr-University Bochum, Chair of ITM, Germany (2002)
9. Clements, P., Northrop, L.: Software Product Lines: Patterns and Practice. Addison Wesley, Reading (2001)
10. Van Steenbergen, M., Brinkkemper, S., van den Berg, M.: An Instrument for the Development of the Enterprise Architecture Practice. In: Proceedings of the 9th Int. Conference on Enterprise Information Systems, pp. 14–22 (2007)
11. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: Capability Maturity Model, Version 1.1. IEEE Software 10(4), 18–27 (1993)
12. CMMI Product Team: Capability Maturity Model Integration (CMMI) Version 1.1 Staged Representation, Tech. Report CMU/SEI-2002-TR-029, Carnegie Mellon University (2002)
13. Cusumano, M.A.: The Business of Software. Free Press, New York (2004)
14. ISO/IEC 15504, Information Technology – Software Process Assessment ISO/IEC (1996)
15. Kuilboer, J.P., Ashrafi, N.: Software Process and Product Improvement: An Empirical Assessment. Information and Software Technology 42(1), 27–34 (2000)
16. Reifer, D.J.: The CMMi: It's Formidable. Journal of Systems and Software 50(2), 97–98 (2000)
17. Zahran, S.: Software Process Improvement: Practical Guidelines for Business Success. Addison-Wesley, Reading (1997)
18. Brodman, J.G., Johnson, D.L.: What Small Businesses and Small Organization Say About the CMM. In: Proceedings of the 16th International Conference on Software Engineering, pp. 247–262. IEEE Computer Society Press, Los Alamitos (1994)
19. Koomen, T., Baarda, R.: TMap Test Topics, Tutein Nolthenius, Den Bosch, The Netherlands (2005)
20. van de Weerd, I.: IT and Software Product Management, `http://www.softwareproductmanagement.org/` (retrieved Februari 1, 2009)

21. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell (1999)
22. van de Weerd, I., Versendaal, J., Brinkkemper, S.: A Product Software Knowledge Infrastructure for Situational Capability Maturation: Vision and Case Studies in Product Management. In: Proceedings of the 12th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2006), Luxembourg, pp. 97–112 (2006)
23. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: The Influence of Situational Factors in Software Product Management: An Empirical Study. In: Proceedings of the 21st International Workshop on Product Management, Barcelona (2008)

Productization: Transforming from Developing Customer-Specific Software to Product Software

Peter Artz[1], Inge van de Weerd[1], Sjaak Brinkkemper[1], and Joost Fieggen[2]

[1] Department of Information and Computing Sciences, Utrecht University,
P.O. Box 80.089, 3508TB Utrecht, The Netherlands
{pgjartz,i.vandeweerd,s.brinkkemper}@cs.uu.nl
[2] MP Objects, Rotterdam, The Netherlands
joost.fieggen@mp-objects.com

Abstract. Developing product software is getting increasing attention from both academics and practitioners. Organizations are recognizing the benefits and importance of developing a product for a market. Also, several software companies that develop customer-specific software have identified a need to change to developing and selling product software. Since few studies are available in this domain, it is difficult for organizations to handle such a transformation. In this research, we introduce the productization process that describes the transformation from developing customer-specific software to product software.

Keywords: productization, software product management, customer-specific software, product software.

1 Introduction

Custom information systems are made-to-order systems and are typically built for specific users [1]. Nowadays, inspired by the success of large software vendors, such as Microsoft and SAP, an increasing number of companies that develop this customer-specific software have a desire to transform to developing product software. *Product software* is defined as "a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market" [2]. Both situations differ in many respects [3], for example where for customer-specific software development usually one release is delivered, the product software development consist of recurrent releases. Consequently, this involves proper requirements management and release planning. Therefore, organizations that make a shift to developing product software need to change their internal processes.

An approach that can be used to structure the internal processes of a product software company is the reference framework for Software Product Management (SPM) [4], which gives product managers and organizations more insight information about the product management processes and involved stakeholders. In the reference framework, key process areas, stakeholders and their relations are modeled. The four main product management functions are Portfolio management, Product roadmapping, Requirements management, and Release planning. Additionally, the interaction

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 90–102, 2010.

of product management with internal stakeholders (company board, research & innovation, development, support, services, and sales & marketing) and external stakeholders (market, partner companies, and customers) are also included.

Currently there is barely any scientific literature available on the explicit transformation from developing customer-specific software to developing product software, which we coin as productization: *the transformation process from customer-specific software development to a standard software product* [5]. As a result, the main research question of this study is:

How can organizations transform from developing customer-specific software to product software?

In this research, we focus on the software product management domain. Evidently, there are more processes that change during a productization process, such as services and sales, but these are not in the scope of this research.

In the following section we first describe the research approach that was followed. Section 3 presents our main result: the productization process. Subsequently, in section 4, this result is evaluated. In section 5, we present the discussion. The final section consists of the conclusions and suggestions for future research.

2 Research Approach

The research described in this paper is a design science research. In order to answer our main research question, we apply the design-science research guidelines proposed by Hevner et al. to describe the problem solving process of understanding, executing, and evaluating research [6]. The design-science guidelines "seek to extend the boundaries of human and organizational capabilities by developing new innovative artifacts". The artifact that we develop in this research is the *Productization process*, which describes the transformation process that companies typically experience when shifting from developing customer-specific software to product software. We combined the five process steps of Vaishnavi and Kuechler [7] with the guidelines from Hevner et al. [6] to give our approach more structure:

1. *Awareness of the problem.* The initial step consists of all activities in order to understand the problem. The main technique we applied in this step is an extensive literature study on several research areas. As a result of that, we defined the problem definition and we determined the contribution of this study. Two guidelines which are covered within this step are 'problem relevance' and 'research contributions'.
2. *Suggestion.* The second step proposes a solution from the existing knowledge and theory base for the problem area. By carrying out a literature study, we identify the differences between developing customer-specific software and developing product software. In addition, we study various methods and approaches that can be used to change the internal processes of an organization. Consequently, this step covers the guideline: 'design as a search process'.
3. *Development.* Based on the suggestions from the previous step, we develop a first version of the artifact, which is, in this case, the productization process.. We use two data collection techniques in order to gain additional knowledge. Firstly, we carry out exploratory interviews among software product managers and software

product management experts, to gain more insight on how organizations evolve over time. Secondly, a literature study is used to obtain more information on the transformation process. The result of this step was the productization process as proposed in Section 3. This third step covers the guidelines 'design as an artifact' and 'research rigor'.

4. *Evaluation.* The fourth step consists of the evaluation of the designed artifacts. We interview several SPM experts, create a survey for software product managers, and carry out a business case. The evaluation is described in Section 4. Accordingly, this step covers the guidelines 'design evaluation' and 'research rigor'.
5. *Conclusion.* Finally, the last step analyzes the results of the study. We use them to deduce the answer towards our research question and provide a discussion of the results. As a result, this step covers the guideline 'communication of the research'. The discussion and conclusion are described in sections 5 and 6.

3 Productization

Two major differences exist between customer-specific software development and product software development: the difference between stakeholders and a major pressure on time to market [1]. The main stakeholder involved for customer-specific software development is the external stakeholder. They specify which requirements need to be implemented and are more involved during the development. Consequently, the main pressures for such organizations are the costs and the customer satisfaction. On the other hand, within a product software company, the developer decides about the requirements and selects the stakeholder representatives. The main pressure for these companies is the time to market, which is covered by regularly releasing new releases. Therefore, it requires a careful release planning and requirements prioritization [3].

In order to evolve customer-specific software into product software, we propose the productization process, consisting of different stages in which the transformation process is depicted (Fig. 1). This process assumes an internal drive to change from developing customer-specific software to a standard software product. The external triggers that can lead to transformation, such as customer and market dynamics, are not taken into account.

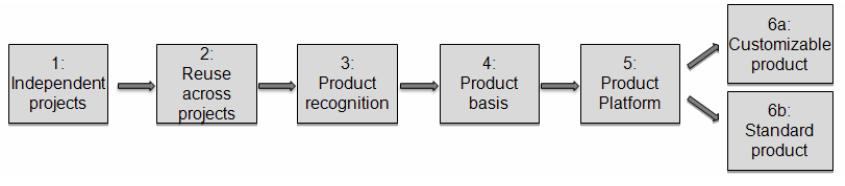

Fig. 1. Productization process

Organizations are able to enter at each stage of the process and continue until one of the two end stages is reached. The two possible end stages of the productization process are: *customizable software product* (stage 6a) and *standard software product* (stage 6b). According to Hietala et al. [8], this "degree of productization", is influenced by product market, concepts, benefits, positioning, requirements, features,

specifications, delivery channel, marketing, selling, and packaging. We applied two end stages for the productization process because often there is a need for some customization in order to integrate software in a customer-specific situation. It can take substantial time and effort to get enterprise product software up and running and therefore pieces need to be customized [9]. Also [10] recognized the intension of selling software products and of providing support services. They identify this approach as a "hybrid approach" and consist of a customer-specific part so that the product is better applicable within current infrastructures.

In the following sections we depict and describe each productization stage. We use a red color for the customer-specific parts and blue for the shared parts.

3.1 Stage 1 – Independent Projects

This first stage of the productization process represents a portfolio of projects that are executed independently. A company is carrying out projects that have barely any standard common functions or features. The software applications that are delivered differ considerably in size, budget, technology, and functionality. These differences are represented in the figure by using different sizes for the software blocks.

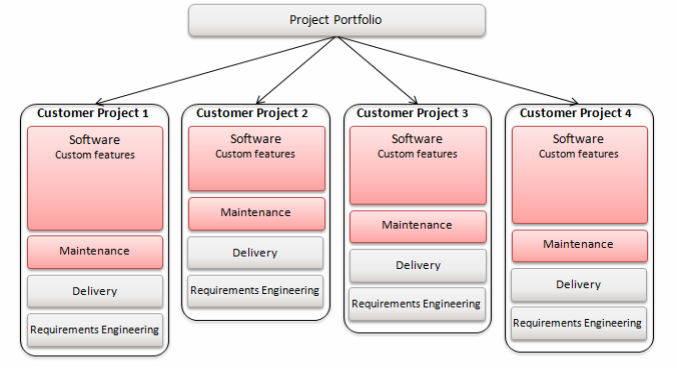

Fig. 2. Independent projects

The projects are especially driven by the customers and therefore they are the main stakeholders within this phase. The success of the projects is measured by customer satisfaction and acceptance. Consequently, there is usually a small physical distance between customers and the developers [11]. Keeping the customers involved is essential in order to satisfy them and therefore all customers' requirements must be obtained. Based on the fixed number of requirements, the delivery date and the required resources can be determined [12]. After the software is developed, it is validated together with the customer and after acceptance it is implemented.

3.2 Stage 2 – Reuse across Projects

The second stage of the productization process is identified as *Project Feature Reuse: Projects are executed differently and features or functionalities are reused across*

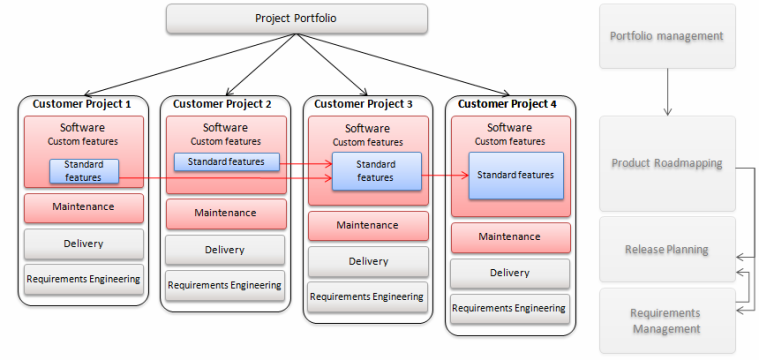

Fig. 3. Reuse across projects

projects. In contrast to stage 1, there is the possibility of reusing specific, already developed, features, functions, components or modules from previous projects. Hence, this stage represents a company that starts to develop projects by reusing elements from other projects. An advantage of reusing functionalities from finished projects is that the overall quality and reliability of the software can increase because specific parts have already been tested within previous projects. This pattern of reusing features from previous projects continues throughout the upcoming projects resulting in a basic set of standardized features or core functionalities. However, at this stage, the amount of customized features developed is still considerably larger than the standardized features.

3.3 Stage 3 – Product Recognition

Compared to stage 2, in stage 3 the standardized part of the projects is larger than the customized parts. We therefore call this stage as *Product Recognition: Large parts of projects are reused and a product scope starts to emerge from the reused functionalities*. Basically, a company is able to identify the similarities of customers' wishes, which results into a more generic basis for / recognition of a product. Important in this stage is that an organization should evaluate the potential to become market-driven. When it is decided to start transforming to product software, stage 3 is the first step of creating a product. During this stage, the main boundaries of the product emerge and based on that a company can define a product for a specific market or purpose.

Within each of the flowing stages, the maturity of the product management functions should increase until at the end all functions are fully in place. We do not provide a specific adoption approach because there is no given hierarchy of implementing the product management functions. In the figure, we indicate this growth in maturity with the gray block on the right-hand side of the figure. The further in the productization process, the higher the product management maturity and the larger the dark segments of the different SPM blocks.

Central in stage 3 is the requirements management function. All new emerging customers' requirements should be stored and managed in one central place. Managing the requirements is necessary because when a company starts to recognize a

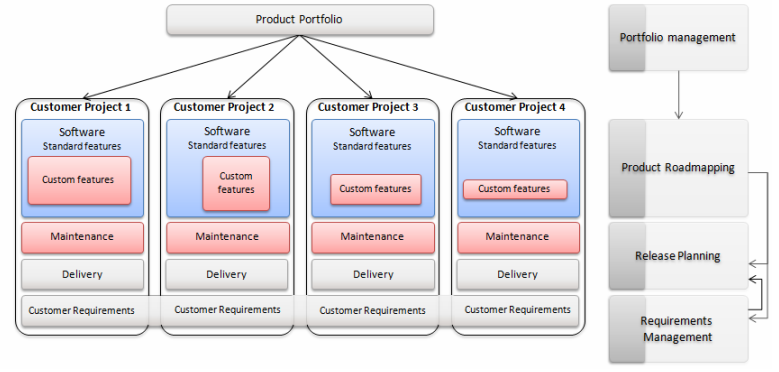

Fig. 4. Product recognition

specific product there is still a need to satisfy the customers. The main focus of this stage is still based on satisfying the customers. Therefore, we cannot speak of a standardized product designed for an entire market. Each project still has its own customized part merged into the main structure of the software in order to satisfy the customer.

3.4 Stage 4 – Product Platform

One of the major differences between developing customer-specific software and product software is the change in the lifecycle of a product. In order to determine the future of a product, a company should start focusing on future releases in order to gain bigger market share. Based on the definition of [13], we define this stage as *Product Platform*: *a set of features that form a common structure, from which a stream of derivative products can be efficiently customized, developed and produced.* As in the previous stage, many features are reused across projects. What is more, the product is recognized and a clearly defined product platform is used as a basis.

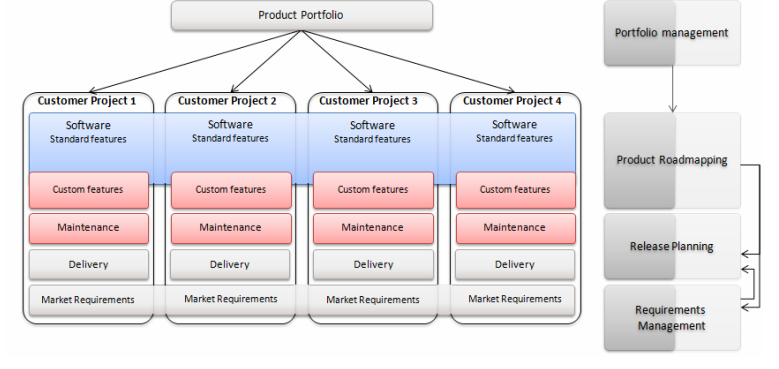

Fig. 5. Product platform

Due to the introduction of the roadmapping function, a company can generate a long-term plan. This means that the main focus on satisfying the customer is decreasing and the focus of gaining more market share should increase. The requirements management functions should also gather market requirements instead of only the customer requirements. These market requirements are needed in order to determine the content of the future software product releases because eventually the customized part should be as small as possible.

At this stage, we still are not able to speak of a first release of a software product; the amount of customized features within the projects is at this point still too large. The standardized part of this stage can be seen as the initial start of creating the product by generating its main core. This core can also be identified as a generic product platform which is extended by a large customer-specific layer.

3.5 Stage 5 – Standardizing Product Platform

The main focus of this stage is still the customer-orientation, but compared to stage 4, it starts to change towards a market orientation and bringing the emerging product to the market. Therefore, the company should create a more standardized product which is applicable for more customers in the market. The definition of this stage originates from the definition of [13]. We define it as *Standardizing Product Platform: increasing the set of features that form a common structure and introduce releases, from which still a stream of derivative products can be efficiently customized, developed and produced.*

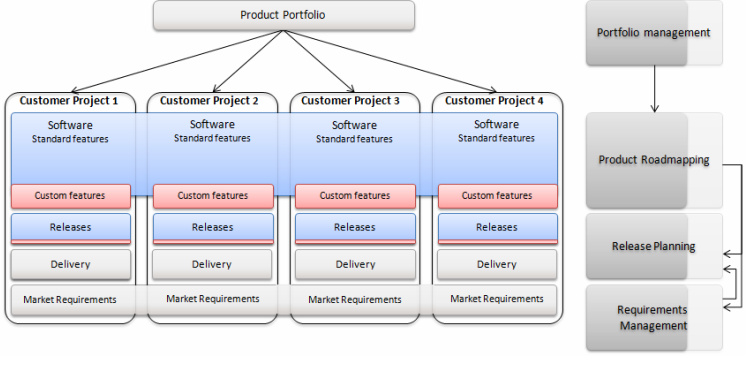

Fig. 6. Standardizing product platform

If necessary, the organization can still create for each customer its own customized part. The result of this approach is a standardized main product with additional event-based (customizable) releases. Due to the introduction of the releases, the lifecycle of the product increases and current customers get acquainted with these releases.

3.6 Stage 6a – Customizable Software Product

Stage 6a is one of the end stages of the productization process. By adapting the definition of [2], we define this stage as *Customizable Software Product*: *a packaged*

configuration of a software-based service, with auxiliary materials, which is released for and traded in a specific market and customized for a specific customer. In contrast to the previous stage, in stage 6, releases are the same for each customer.

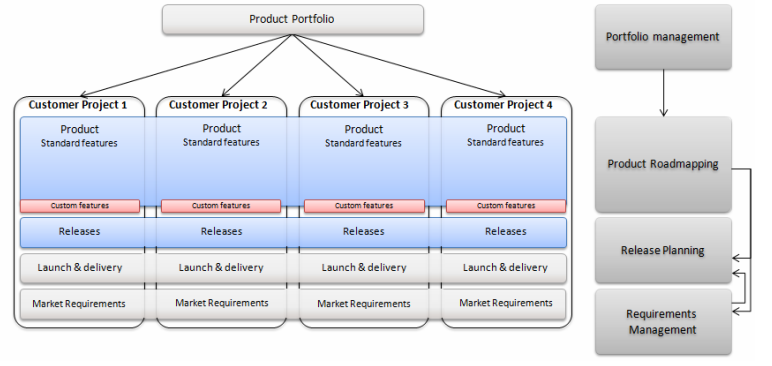

Fig. 7. Customizable software product

For some companies, there is still a need to be able to customize the software product so that it can be integrated within specific situations. This product type is characterized by software that is too complex to be sold 'off-the-shelf' and that requires customization or special integration and installation work [10]. Therefore, there is the need for a customized layer on top of the product so that the required functionalities can be added. The size of the functionality and code of is customization is, in general, small in comparison to the standard product.

Also [14] identified the essence of still having a customizable part in order to be able to apply the product to different situations. Additionally, [9] identified this type of product software as "enterprise solution systems" because usually this type of product software is developed for other enterprises.

3.7 Stage 6b – Standard Software Product

As described in Section 3, the productization process has two end stages. Stage 6b is the other end stage of the productization process. Inspired by [2], we define this stage as *Standard Software Product: a packaged configuration of software components, which is released for and traded in a specific market.*

After several stages still focusing on the customers, this stage focuses on a market in general. By performing active marketing and sales activities, the company should start to sell the product to a mass-market and start looking at the wishes of that market. In order to bring the product to the market, it is required that there are no customized features included within the product and the product is completely configurable. Furthermore, the main measurement in order to determine success is mainly based on getting a bigger market share and having a shorter time-to-market. An advantage of selling product software is that no additional development is required in order to be able to sell the product to new customers. This results in a much bigger potential market and more customers [8].

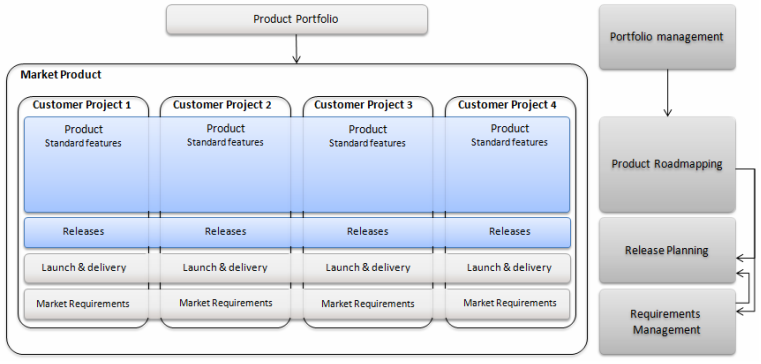

Fig. 8. Standard software product

4 Evaluation

The methods we used for the evaluation of the artifacts are expert validation (consisting of interviews and a survey), and a business case. In this section, we describe both types of validation and the related validity threats.

4.1 Expert Validation

For our expert validation of the productization process, we used an expert panel of five experts, because the interviewed experts provide an informed, objective, and unbiased opinion as well as suggestions about the developed artifact [16]. The focus of the interviews was to evaluate the acceptance and recognition of the productization process in the past of organizations. Key questions were related to how Dutch companies actually transformed in order to become a product software company. Additionally, we evaluated whether the productization process is technically sound in relation with the viewpoints of the SPM experts.

Secondly, we created a survey in order to get the opinions of eight product managers from a wide range of companies which participated on an SPM course. The survey was designed in such a way, that we were able to examine the structure of artifact for static qualities (e.g. complexity or readability) [6]. We wanted to find out whether the participants were able to recognize such process and the possibility to select a stage which is applicable for their product(s). As a result, several suggestions emerged and all participants were able to select an applicable stage. More details on the expert validation can be found in [5].

4.2 Productization in Practice: A Case Study

In addition to the expert validation, we also carried out a case study at MP Objects to validate the applicability of the productization process in a business environment. By performing a case study, the validation process provides an in depth evaluation in a business environment [17]. MP Objects is a small sized software vendor which is a specialist in providing ICT solutions for the logistic sector. The software is a unique

IT solution for Supply Chain Management within and across organizations and they support companies with the integration across the supply chain, the collaboration between people in the order cycle, and the optimization of processes in the supply chain.

In order to apply the productization process, we designed a specific approach that consists of three steps. The first step determines the initial position by carrying out an assessment. The assessment uses the maturity matrix for determining the maturity of the (present) SPM processes [19], situational factors for the determination of the best suitable maturity levels [18], and the process deliverable diagram for the visualization of the organizational structure of the initial situation [20]. Secondly, a gap analysis must identify the distance from the initial position until being fully Software Product Management oriented. We used situational factors [18] to determine the best suitable maturity levels for the product management functions. Consequently, with the initial maturity levels we determine which processes need to be implemented or improved. Finally, based on the gap analysis, recommendations can be identified, which an organization can use when they continue in transforming to become market-driven. An elaborate case study description can be found in [5].

4.3 Validity Threats

In conducting our case study, we considered the three validity threats as described by [21]. Firstly, *construct validity* is covered by using multiple sources of evidence for the data collection of specific information. While gathering the required data for the productization approach, we applied the viewpoints of consultants and developers in order to get a better and representative result. Secondly, *external validity* refers to the approximate truth of conclusions that involve generalizations. As for the exploratory interviews and validation interviews, we combined the gathered results of the scientific field with the gathered results of the business field. Also, for the survey we involved organizations that differ in several areas (by using the situational factors). Finally, we also tried to verify the results of the validation process with references from the literature before we applied them. The last threat, *reliability* concerns the demonstration that the results of the study can be replicated. For the business case, we covered this by creating an approach for the implementation of the productization process. Additionally, all other relevant information and conclusions are documented so that the results can be replicated.

5 Discussion

Based on the evaluation described in section 4, we feel that more validation is required to prove the added value of the productization process and to determine the validity and applicability of the identified stages. For example, there still exists some discussion on the identified stages of the productization process. Some members of the expert panel proposed to merge two stages and others commented on the fact that the end stages of the productization process are limited to customizable software products and standard software products (stage 6a and 6b). Furthermore, some experts mentioned that it could well be possible to use one of the other stages as end stage. In

addition, some suggested that there should be a possibility to move backwards in the productization process, since there is always a possibility that an organization decides to change back to a customer specific focus. Additional case studies in different types of software companies should be carried out to validate the productization process in an organization that is transforming.

Another issue that needs to be discussed is the parallels that can be drawn between the productization approach and the evolution of product lines as described in, for example, [22]. In this article, Bosch describes the evolution of a product line through different maturity levels, ranging from independent products, via a shared product platform, to a configurable product base. Similarities with the productization process, of which the stages range from independent projects, via a shared product platform, to a standard software product, are easy to recognize. However, there are some important differences. The aim of software product line evolution is to transform the way in which products are being built. By using a configurable product base to develop a variety of products, the cost of development and the time to market can be decreased [22], compared to developing products from scratch. In the productization process, the aim is not directly to decrease costs and time to market, but to change the whole business model. Instead of carrying out customer-specific software projects, a shift is made to developing a standard software product for a market, hence serving a much larger array of customers. Nevertheless, the analogy with product line evolution is present.

6 Conclusions and Further Research

The research question, as we stated in Section 1, is:

How can organizations transform from developing customer-specific software to product software?

To answer this question, we proposed the productization process that describes the transformation from developing customer-specific software to product software and that can assist organizations in becoming a product software business. The entire productization process is supported by the implementation of the product management functions of the reference framework for SPM. Additionally, per stage a brief description is presented which explains in more detail the characteristics of that particular stage. The stages represent the different phases that can occur during such a transformation. Eventually, when an organization has reached its desired end stage, all product management functions should be in place.

We believe the productization process is a valuable contribution in the research domain of software business. However, this research also identifies the need for more research on the transformation from customer-specific software development to product software development.

Future research and more expert interviews or case studies are required in order to be able to provide a valid answer. Finally, research carried out by [2] also identified the overlap of Open Source software. Further research should look deeper into the integration of Open Source software in the productization process.

References

1. Sawyer, S.: Packaged software: implications of the differences from custom approaches to software development. European Journal of Information Systems 9, 47–58 (2000)
2. Xu, L., Brinkkemper, S.: Concepts of product software: Paving the road for urgently needed research. In: Castro, J., Teniente, E. (eds.) PHISE 2005, pp. 523–528. FEUP Press (2005)
3. Regnell, B., Höst, M., Natt och Dag, J.: An industrial case study on distributed prioritization in market-driven requirements engineering for packaged software. Requirements Engineering 6, 51–62 (2001)
4. Weerd, I.v., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support. In: International Workshop on Software Product Management (IWSPM 2006 - RE 2006), pp. 312–315. Minneapolis/St. Paul, Minnesota (2006)
5. Artz, P., Weerd, I.v., Brinkkemper, S.: Productization: An exploratory research project Technical report UU-CS-2010-003. Institute of Computing and Information Sciences, Utrecht University, http://www.cs.uu.nl/research/techreps/UU-CS-2010-003.html
6. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly: Management Information Systems 28, 75–105 (2004)
7. Vaishnavi, V., Kuechler, B.: Design Research in Information Systems (2007), AISWorld Net http://home.aisnet.org/displaycommon.cfm?an=1&subarticlenbr=279
8. Hietala, J., Kontio, J., Jokinen, J., Pyysiainen, J.: Challenges of Software Product Companies: Results of a National Survey in Finland. In: 10th International Software Metrics Symposium, pp. 232–243 (2004)
9. Hoch, D., Roeding, C.R., Purkert, G., Lindner, S., Müller, M.: Secrets of Software Success: Management Insights from 100 Software Firms around the World. Harvard Business School Press, Boston (1999)
10. Cusumano, M.A.: The business of software: What every manager, programmer and entrepreneur must know to thrive and survive in good times and bad. Free Press, New York (2004)
11. Keil, M., Carmel, E.: Customer-developer links in software development. Communications of the ACM 38, 33–44 (1995)
12. Carlshamre, P.: Release Planning in Market-Driven Software Product Development Provoking and Understanding. Requirements Engineering 7, 139–151 (2002)
13. Meyer, M.H., Seliger, R.: Product platforms in software development. Sloan Management Review 40, 61–74 (1998)
14. Codenie, W., De Hondt, K., Steyaert, P., Vercammen, A.: From Custom Applications to Domain-Specific Frameworks. Communications of the ACM 40, 71–77 (1997)
15. Bohl, O., Frankfurth, A., Schellhase, J., Winand, U.: Guidelines - A Critical Success Factor in the Development of Web-Based Trainings. In: International Conference on Computers in Education (ICCE 2002), Auckland, New Zealand, pp. 545–546 (2002)
16. Sandelowski, M.: The call to experts in qualitative research. Research in Nursing & Health 21, 467–471 (1999)
17. Zelkowitz, M., Wallace, D.: Experimental Models for Validating Technology. Computer 31, 23–31 (1998)

18. Bekkers, W., Weerd, I.v., Brinkkemper, S., Mahieu, A.: The Influence of Situational Factors in Software Product Management: An Empirical Study. In: Proceedings of the 2008 Second International Workshop on Software Product Management, pp. 41–48 (2008)
19. Weerd, I.: v., Bekkers, W., Brinkkemper, S.: Developing a Maturity Matrix for Software Product Management. UU-CS (Int. rep. 2009-15). UU WINFI Informatica en Informatiekunde
20. van de Weerd, I., Brinkkemper, S.: Meta-modeling for situational analysis and design methods. In: Syed, M.R., Syed, S.N. (eds.) Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, pp. 28–38. Idea Group Publishing, Hershey (2007)
21. Yin, R.: Case study research: Design and methods, vol. 3. Sage Publishing, Beverly Hills (2003)
22. Bosch, J.: Maturity and Evolution in Software Product Lines: Approaches, Artefacts and Organization. In: Chastek, G.J. (ed.) SPLC 2002. LNCS, vol. 2379, pp. 247–262. Springer, Heidelberg (2002)

Implementing Open Source Software Governance in Real Software Assurance Processes

Claudio A. Ardagna[1], Massimo Banzi[2], Ernesto Damiani[1], and Fulvio Frati[1]

[1] Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano
via Bramante, 65 - 26013 Crema (CR) - Italy
[2] TelecomItalia TILab - Vertical Platforms & VAS - Vertical Platforms Innovation
via V. Zambra, 1 38100 Trento - Italy

Abstract. Open Source is giving rise to new methodologies, competences and processes that organizations have to investigate both from the technical and the managerial point of view. Many organizations are studying the possibility to adopt open source products or migrate their systems to open frameworks, even for mission-critical application. In this paper we discuss a roadmap for organizations that want to establish a formalized governance methodology for the management of the open source products, taking into consideration issues such as software quality and community reliability. The governance framework is designed to be included in a more complete software assurance system for open source software.

Keywords: Software Assurance, Open Source, Governance.

1 Introduction

The concept of *Software Assurance* (SwA) [4] involves a number of different phases of software process and varies depending on the specific context where it is applied. For instance, the implementation of SwA actions can be targeted to the reaching of specific software quality, security or dependability requirements.

Generally speaking, the term *assurance* has been associated to the task of reducing general information risks while improving, at the same time, the overall quality of data reported to management and decision makers. In fact, the assurance concept is associated to different tasks, such as risk assessment, information systems security, internal audit, and customer satisfaction surveys, focused on specific aspects of the environment they are applied to [1].

As far as software engineering is concerned, the assurance process includes all the activities that an organization deems necessary to provide a correct level of confidence that a software effectively will satisfy the functional and non-functional requirements requested by its users. As depicted in Fig. 1, assurance activities, in a traditional waterfall-based development process, are process-oriented. Each phase of software lifecycle is covered by specific assurance activities, starting from the *Requirements* and *Use Cases* definition, where specific tests analyze the fulfillment of security requirements and the presence of possible abuse cases [13], and examining, for instance, the *Code* phase, where specific tools are exploited to calculate

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 103–114, 2010.

the sets of product metrics that are able to assess code quality, dependability, performance, and the like.

Of course, in the case of cooperative development processes, like the ones used for Open Source (OS) development, is not possible to identify a complete and rigorous development process and, consequently, associate all assurance tasks showed in Fig. 1. In this scenario, the definition of assurance mechanisms for OS solutions is a difficult task.

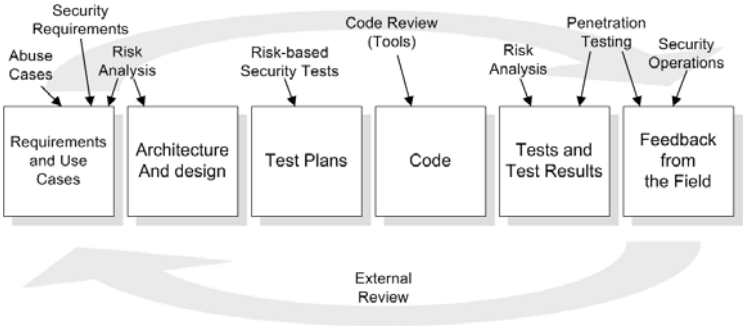

Fig. 1. Assurance tasks in a traditional, lifecycle-based development process

The problem of defining an assurance methodology for OS projects has been studied by Damiani *et al.* in [3], where the problem of establishing assurance policies and methodologies for OS software has been addressed, focusing on the specific security requirements of critical telco environments. In this paper we try to extend the concepts described in [3] focusing on issues related to OS software governance, providing a guide and a roadmap for managers and decision makers for driving them through the decision of adopting OS frameworks in their critical applications.

The paper is organized as follows. Section 2 gives an introduction of open source governance. Then, Section 3 describes OS governance methodology, introducing a new vision for the OS management, and Section 4 an outlook of the future work to implement the model. Finally, Section 5 gives our conclusions.

2 Open Source Governance

In the literature, the concept of "governance" has been mostly associated to the creation and the management of the community of users and developers that characterizes an OS product [12]. As for instance O'Mahony and Ferraro [14] define a model for OS community governance, while in [11] the author gives insights how the lessons learned from OS communities could be applied to other productive environments.

The concept of governance has to be extended in order to consider OS environments not only from the point of view of an inside-analyzer, but taking into

consideration the needs of an organization or a group which wants to integrate OS products following precise and reliable adoption policies.

The common policies used in the case of adoption of commercial products could not be applied, since they are mostly focused on the purchase cost variable that could not be directly applied to OS products. Otherwise, such OS governance should take into account aspects that are typical of OS development, like community management, the number of developers, the quality of the available source code, the type of license, and so forth. If not managed, the "free" world of OS software can rapidly lead to inconsistent situations from both technical and licensing perspectives. Therefore, governance policies are becoming increasingly important as a means of ensuring the long-term viability and acceptability of OS projects inside and across enterprises.

The problem has been debated from many point of view, trying to give high-level recommendations on how to implement an existing governance framework to drive specific education and enforcement actions [7], or how to define a governance structure suited to the symbiotic relationship of actors involved in the OS process [10].

An enforceable program of OS governance is of paramount importance for gaining more value from OS products, protecting, at the same time, the interest and assets of organizations. In this paper we describe the governance model that an important telco player (Telecom Italia) studied and developed by exploiting researches on OS environments undertaken by the academic research group at Università degli Studi di Milano. The model is aimed at understanding the context in which the software will be adopted and having a full knowledge of the impact that the adoption will have in organization common activities. In fact, in the telco context the exploiting of OS can range from simple deployment of selected applications, to the embedding of OS packages in the development of critical systems. The model provides a set of metrics, recommendations, and procedures to analyze the OS software to be embedded in legacy solutions.

The implementation of a reliable governance framework will be integrated in the overall SwA strategy that could be implemented by organizations to raise their awareness and control of the adopted software packages.

3 Open Source Governance Model

The OS governance methodology implemented by TelecomItalia is composed of two distinct phases: *i)* the selection of the suitable OS tool and *ii)* the management of the OS environment.

3.1 Open Source Tools Selection Methodology

The need for TelecomItalia of having a model for OS governance comes from the results of an analysis of installed OS packages across all the company's business areas. The results was reported in [3] and shows that, while OSS adoption was pervasive, it was carried out without any kind of adoption strategy: a number

of different applications were installed to perform the same activity, resulting in multiple outcomes when performing the same action.

The typical business process for telco companies is based on eTOM model (enhanced Telecom Operations Map) [19], released by the TeleManagement Forum. The model includes a hierarchy of process definitions, offering a structure to represent and develop areas of the process. Furthermore, it is important in the management of licenses and Intellectual Properties Rights (IPRs). In fact, different process areas imply different levels of visibility and customization of the adopted OS package. This issue leads to a careful analysis of the license types and the area where they are going to be integrated, ensuring that the license of the adopted package is consistent with its specific license and business area. According to the specific service in which the OSS component is embedded, special care has to be taken in verifying the suitable licenses.

The methodology answers three specific questions: *i)* **Identify reliable communities**, identifying a set of parameters to be collected and evaluated periodically to rate the communities themselves, *ii)* **Verify the type of licenses used**, comparing them with the business area they are going to be included, and *iii)* **Verify the quality of the adopted solution**, evaluating it in terms of adherence to coding and security standards.

Community Reliability Metrics. The evaluation of the reliability of an OS community requires the definition and evaluation of a set of metrics that have to be applied to public data released by the communities themselves, or extracted analyzing forums, download logs, and releases frequency. Currently, there is any standard framework of metrics, but a number of independent projects (e.g., as described in [16]) provide their own metrics, each one focused on specific aspects.

Even though well-known portals like SourceForge, Freshmeat, and the OW2 consortium, release information about the projects they host and distribute, those data are semantically heterogeneous and do not allow cross-portal evaluations. For instance, the management of issue tracking system is handled very differently at different portals, and the definition of "closed" for an issue is subject to different interpretations. Also, the importance given to the number of downloads varies when the community considers more valuable the quality of a product rather than its diffusion.

Currently, the most-established assessment frameworks are the BRR Model [15], which provides a complete set of metrics to evaluate if a project is enough mature to be adopted but does not define rules on collecting data, and FLOSSmole [9], that collects and distributes data from several communities. However, these two sets of metrics are not homogeneous and are difficult to compare.

In order to fulfill the task of evaluating more the community aspect than the software itself, evaluating at the same time the reliability and the reactivity of the community to newly discovered bugs and vulnerabilities, we consider FOCSE (Framework for OS Critical Systems Evaluation) [5], a framework, developed by the Università degli Studi di Milano, based on a set of general purpose security-related metrics that perfectly fit the needs of the analysis. In fact, FOCSE includes some specific metrics expressing the capability of the

Table 1. First level metrics

Metric	Description
Date of Search	Date when current data were collected
First Release Date	Date of the first documented stable release of the product
Last Stable Release Date	Date of the last product stable release
#ofBugsDetected	No. bugs detected from bug reports
#ofBugsFixed	No. bugs fixed from bug reports
#ofForumReplies	No. community answers to forum messages
#ofForumThreads	No. groups of specific user questions
#ofReleases	No. releases since project start up
#ofPatches	No. patches since project start up
#ofDownloads	No. downloads since project start up
Project Core Group existence	Evaluate the existence of a group of core developers (1 if exist, 0 if do not exist)
Number of core Developers	No. core developers contributing the project
Forum and Mailing List Support	Check forum and mailing list availability (0 if no mail no forum, 1 if mail but no forum, 2 both mail and forum)
RSS Support Check	RSS availability (0 no, 1 yes)
Documentation fulfillment	1 to 5 check for the presence of the following type of documents: admin, user, config, release delta notes,white paper guides

community to adapt to continuous changes in security threats and vulnerabilities. It provides a set of metrics focused on multiple areas: *i)* Generic Application, *ii)* Developers Community, *iii)* Users Community, *iv)* Software Quality, *v)* Documentation and Interaction support, and *vi)* Integration and Adaptability with new and existing technologies.

The original formalization of FOCSE provides two different sets of metrics: the first one includes all metrics that can be readily computed from available information on the OSS projects, while the second one requires parameters whose measures involve privileged access to the developers' community. Such metrics has been improved and adapted for the particular needs of the Telco context where they are going to be applied to; Tab. 1 and Tab. 2 summarize the metrics, giving a short description of them.

To generate a single estimate, it is necessary to aggregate the value of metrics in Tab. 2. This way, two or more projects, each one described by its set of attributes, can be ranked. To this aim, an aggregation algorithm has to be

Table 2. Second level metrics

Metric	Description
Age	Age of the project calculated from first stable release to *Date of Search* **Date of Search - First Release Date**
Last Update Age	Age of the last stable project update to *Date of Search* **Date of Search - Last Stable Release Date**
Project Core Group	Evaluate the existence of a group of core developers **[1 exist 0 do not exist]**
Number of core developers	Number of core developers contributing the project
Replies/Threads (Community Vitality)	Vitality of the community in terms of no. answers in response to specific user questions **(#ofForumReplies/#ofForumThreads)**
Number of Releases	Number of releases since project start up
Update Average Time (time/release)	Vitality of developers group,i.e. mean number of days to wait for a new update **Age/(#ofPatches+#ofReleases)**
Average of downloads per release	**#ofDownloads/#ofRelease rate**
Bug Fixing Rate	Rate of bugs fixed, weighted over the total number of bugs detected **(#ofBugsFixed+1/#ofBugsDetected+1)* (1-exp(-#ofBugsDetected))**
Problems per release	**(#ofBugsDetected - #ofBugsFixed) /#ofReleases**
Mean Time Between Failures (MTBF)	**(#ofBugsDetected - #ofBugsFixed)/Age**
Forum and Mailing List Support	Check forum and mailing list availability (0 no mail no forum, 1 mail no forum, 2 mail e forum)
RSS Support Check	RSS availability **[1 exists, 0 do not exist]**
Documentation Fulfillment	Check for the presence of the following type of documents: admin, user, config, release delta notes, white paper guides **[1..5]**

provided to compute a weighted sum of all metrics value that will return an indicator to establish if a product is acceptable or not.

A solution of this problem has been found in [5] using the *Ordered Weighted Average* (OWA) operator. Introduced by Ronald Yager in [20], the OWA operator provides a parameterized aggregation method, characterized by a reordering step that makes it a nonlinear operator. The OWA operator is different from the classical weighted average in that coefficients are not associated directly with a particular attribute but rather to an ordered position. The operator is defined as follows:[1]

Definition 1. *Let* $w = [\omega_1, \omega_2, \ldots, \omega_n]$ *a weight vector of dimension* n*, such that* $\omega_1 \in [0, 1]$ *and* $\sum_i \omega_i = 1$*. A mapping* $f_{OWA} : R^n \rightarrow R$ *is an OWA operator of dimension* n *if*

$$f_{OWA}(a_1, a_2, \ldots, a_n) = \sum_i \omega_i a_{\sigma}(i)$$

where $\{\sigma(1), \sigma(2), \ldots, \sigma(n)\}$ *is a permutation of* $\{1, \ldots, n\}$ *such that* $a_{\sigma}(i) \leq a_{\sigma}(i-1)$ *for* $i = 2, \ldots, n$

All decision processes that involve multiple selection criteria (i.e. metrics), like OS selection methodology described in the article, require some compensatory trade-offs. They occurs in the presence of conflicting goals, when compensation between the corresponding compatibilities is allowed. The OWA operator can realize trade-offs between objectives, by allowing a positive compensation between ratings, i.e. a higher degree of satisfaction of one of the listed metric will compensate for a lower degree of satisfaction of other criteria to a certain extent.

The metrics framework has been tested on a number of open source products to show its potential and effectiveness. Tab. 3 compare three open source SSH clients using FOSLET metrics: PuTTy, WinSCP, and ClusterSSH. Finally, Tab. 4 shows the aggregated values for the three applications, calculated using the OWA operator. A precise and suitable calibration of OWA weights, and the application of the aggregation operator, will lead to the creation of ranking of the examined OS products, allowing to choose for the best one with respect to organization policy.

License Type Evaluation. The intersection between the requirements of each business area defined in the eTOM model (i.e. visibility of the services to common users, customizations of existing code), and the specific characteristics of the different types of available OS licenses, make their analysis an important issue of the overall governance framework.

The definition of a standard procedure for license analysis starts from the review of the literature in terms of OS licensing. The great success of OS paradigm has lead to the proliferation of a great number of OS license types. Every license shares the principle that everyone should be able to use, copy, modify, and distribute the code, but each one has particular ways to manage such actions.

[1] A more complete formalization of the OWA operator is out of the scope of this paper. A more detailed explanation could be found in [5][20].

Table 3. Comparison of proposed SSH implementations [5]

Metrics	Putty	WinSCP	ClusterSSH
Age	2911 days	1470 days	1582 days
Last Update Age	636 days	238 days	159 days
Project Core Group	Yes	Yes	Yes
Number of Core Developers	4	2	2
Number of Releases	15	32	15
Bug Fixing Rate	0.67	N/A	0.85
Update Average Time	194 days	46 days	105 days
Forum and Mailing List Support	N/A	Forum Only	Yes
RSS Support Check	No	Yes	Yes
Reply/Threads(Community Vitality)	N/A	3.73	5.72

Table 4. OWA-based SSH applications comparison [5]

	Putty	WinSCP	ClusterSSH
f_{OWA}	0.23	0.51	0.47

Several works categorize OS license in term of Intellectual Property Rights [17] and try to track the transformation of the OS licensing with respect to the continuous changes of OS environment [8]. A complete listing of available license has been published by the Open Source Initiative (OSI, www.opensource.org) and by the Free Software Foundation (FSF, www.fsf.org), reporting more than one hundred license types that differs in some parts and that could affect the adoption and use in production environments.

Discussing how to manage all these license types is outside the scope of our work. Instead, we categorize licenses in three overall categories. An understanding of these licensing categories lays the groundwork over which the approach of making decisions about OS licensing is based. In our three-category model, licenses are categorized basing on the extent to which it requires a derivative work to share same license in the produced code [18].

Category A: Unrestricted Licenses. The licenses in that category are wholly unrestricted with regard to the scope of license use in derivative works. A developer operating under a Cat. A licenses may therefore use the commons to create virtually any work, in any way, and then use any kind of licensing he or she desires for the derivative work. The developer may, for example, use a free or OS license for the derivative work, or even use a commercial closed license for the work. Examples are the BSD-style licenses.

Category B: File-based Community-fostering Licenses. Like Cat. A, includes licenses that allow developers to use the open code to create virtually any work, in any way. They assume, however, that these derivative works will be made up of source files that will ultimately become a single binary

file, and that any source file that contains such code must be licensed with the same license. They do not, however, impose this requirement on files that do not contain code from the original code base; those files can be licensed in any way the developer wishes. Examples are Mozilla-style licenses and the Sun's Common Development and Distribution License.

Category C: Strong Copyleft Licenses. Like Cat. A and B, it contains licenses that allow developers to use the open code in any way. Like Cat. B licenses, they require any file that contains derived code derived to be licensed under the original license. Furthermore, they also require that any file, regardless of code origin, which is combined under certain circumstances with the common files must be licensed under the original license. Examples are the GNU General Public License.

This classification has been used to create a check list table to help decision makers on deciding how to behave in front of the different tools according to their usage within an organization. Together with such categorization, that will help to restrict the number of open tools we can include in our products, a deeper check could be executed exploiting free products, like the Koders tool (www.koders.com), that will examine the product code base ensuring that it is developed free of concerns raised by OS licenses, known security vulnerabilities or corporate policies regarding OS code usage.

Software Quality. The quality of OS software is expected to be good enough. Anyway, some quality metrics, taken by literature, are applied to deeply evaluate it checking its intrinsic quality, the accordance to coding standards, and the solution of known security issues. In particular, software quality is evaluated using the well-established metrics for Coupling, Cohesion, Cyclomatic complexity, and the Change Risk Analysis and Prediction (CRAP) metrics. At the same time, the accordance to specific coding standards is assessed using free verifier tools specific of the programming language used for the implementation, such as the J2EE Verifier Tool (java.sun.com/j2ee/avk/), the High Integrity C++ Conformity (www.codingstandard.com), or the MISRA-C and MISRA-C++ tools (www.misra-cpp.org).

Finally, security is taken into account. Security aspects are usually handled by OS developers by exploiting community feedbacks and issues tracking environments. The need of some form of security certification based on a rigorous and in-depth analysis has been raised, and the formalization of a certification framework that will allow, on the one side, suppliers to certify the security properties of their software and, on the other side, users to evaluate the level of suitability of different OS security solutions, is strongly required [1].

3.2 Open Source Environment Management

The final, and may be the most critical, aspect of the OS governance is the definition of the correct way the adoption has to be managed. The investment in OS is also an investment in the IT department, since the adoption of established, good-quality products will increase the internal skill level and know-how.

Starting from this point, it is possible to operate in five distinct ways, differentiated by the approach of the organization versus the OS environment.

Train people on the product. Developers are trained about the product, giving them the knowledge for critical bug-fixing or possible customizations.

Train people to community work. Developers are trained to work in community, with a win-win approach, taking into consideration also the needs and requirements of other members.

Ignite the community with developers. In case the OS product is considered important enough to be able to operate on the product, it can be wise to exploit internal developers to contribute to the community either for small enhancements or for bug-fixing. This allows to acquire competence on the code, but understanding the behaviour and feelings of the community, on how the community operates and on the timing it can react to requests.

Join the community board. If the organization wants to have the highest Return of Investment, once ensured that problems with the licenses and internal skills are overcome, the choice will be of operating directly on the OS community board in implementing the necessary functionality, driving it, and defining future tasks and functionalities. The result is approaching the OS as a resource, where developers are either internal and external, with savings not just on the maintenance of the component, but also in its development.

Create a community. If a suitable community does not exist, it can result profitable to share and open an internally developed solution to extend the test and code base. Such an approach has been followed by TelecomItalia for WADE, where a community has been created and driven to extend the OS platform JADE [6]. A Support Group, that clearly owns the major skills on the solution and will represent the core developers group, and a Technical Manager, responsible for deciding on the roadmap of the product within the arising community, have to be identified.

To protect the investment, where possible, the organization has to negotiate privileges with the community, such as implementing certain features in a protected development environment, where the commit control is delegated to the organization itself. Furthermore, a major involvement of organization in the communities core group, will ensure a major stability of project requirements and achievements.

Such a vision, applied to the last two scenarios, paves the way for a new perspective within the OS paradigm. The strict collaboration with OS communities gives to commercial organization the opportunity of consistent savings since they can enter and work in a working environment complete with all the communication, bug tracking, and versioning tools typical of OS projects. In addition to that, the possibility to negotiate privileges with the board of the community, gives the opportunity to work in a protected context.

For instance, organizations can ask to the board to open a new development branch exploring solutions that are of interest for them; at the end of the implementation, such code will be released, as usual, under the adopted OS license,

but, at the same time, organizations will see their investment in OS protected and profitable.

4 Future Work

At present, a major problem of OS adoption is the complete lack of standards that force to normalize available data on the quality/confidence of the communities. This often results in a huge analysis effort that, due to the continuous evolution of OS communities, has to be periodically repeated. The adoption rate of OS components by large companies suffers from the lack of standards. Furthermore, the FOCSE framework, which has been used as the basis for the metrics framework, has to be improved to include more sophisticated and community-based concepts, like for instance the alignment between project strategy and development objectives.

TelecomItalia, is trying to create consensus on some sort of standards - the Open Source Governance Model as described in Section 3 - by operating within TeleManagement Forum: the world's leading industry association focused on improving business effectiveness for service providers and their suppliers. The idea is to export the work performed in collaboration with Università degli Studi di Milano and integrate it with the experiences of the largest telco service providers in the world to define *de-facto* standards to be proposed to interested OS service supplier companies and indirectly to communities.

5 Conclusions

In that paper we analyzed the state of the art in OS governance, highlighting known issues and proposing a new model for OS governance that could be included in a overall OS software assurance strategy. Our model takes into consideration the evaluation of communities reliability, the type of used license, and the quality of the released software. Furthermore, we give a schema for OS environment management, introducing a new vision for the OS paradigm that considers commercial organizations as active elements of the OS communities, reserving to them some sort of privileges that can protect their investment.

Acknowledgment. This work was partly founded by the European Commission under the project SecureSCM (contract n. FP7-213531) and by the Italian Ministry of Research under FIRB TEKNE (contract n. RBNE05FKZ2_004).

References

1. Ardagna, C.A., Damiani, E., El Ioini, N.: Open Source Systems Security Certification. Springer, Berlin (2008)
2. Ardagna, C.A., Damiani, E., Frati, F., Madravio, M.: Open Source solution to secure e-government services. In: Encyclopedia of Digital Government. Idea Group Inc., USA (2006)

3. Ardagna, C.A., Banzi, M., Damiani, E., Frati, F., El Ioini, N.: An Assurance Model for OSS Adoption in Next-Generation Telco Environments. In: Proc. of 3rd IEEE Int. Conf. on Digital Ecosystems and Technologies (DEST 2009), pp. 619–624. IEEE Press, New York (2009)
4. Evans, M.W., Marciniak, J.J.: Software quality assurance & management. Wiley-Interscience, New York (1987)
5. Ardagna, C.A., Damiani, E., Frati, F.: FOCSE: An OWA-based Evaluation Framework for OS Adoption in Critical Environments. In: Open Source Development, Adoption and Innovation, pp. 3–16. Springer, Boston (2007)
6. Banzi, M., Bruno, G., Caire, G.: To What Extent Does It Pay to Approach Open Source Software for a Big Telco Player? Open Source Development. In: Communities and Quality, pp. 307–315. Springer, Boston (2008)
7. Best Practices in Open source Governance (2010), `https://fossbazaar.org/content/best-practices-open-source-governance`
8. Fitzgerald, B.: The transformation of Open Source Code. Management Information Systems Quarterly 30(3) (2006)
9. FLOSSmole: Collaborative collection and analysis of Free/Libre/open Source Project Data (2010), `http://flossmole.org`
10. Franck, E., Jungwirth, C.: Reconciling Investors and Donators - The Governance Structure of Open Source. Lehrstuhl für Unternehmensführung und politik Universität Zürich, Working Paper No. 8 (2002)
11. De Laat, P.B.: Governance of Open Source Software: State of the Art. J. Manage Governance 11, 165–177 (2007)
12. Lattemann, S., Stieglitz, C.: Framework for Governance in Open Source Communities. In: Proc. of the 38th Hawaii IEEE International Conference on System Sciences. IEEE Press, New York (2005)
13. McDermott, J., Fox, C.: Using Abuse Case Models for Security Requirements Analysis. In: Proc. of the 15th Annual Computer Security Applications Conference (ACSAC 1999), Washington, DC, USA (1999)
14. O'Mahony, S., Ferraro, F.: The Emergence of Governance in an Open Source Community. Academy of Management Journal 50(5), 1079–1106 (2007)
15. OpenBRR: A Framework for Evaluating Open Source Software (2010), `http://www.openbrr.org`
16. Qualipso Project: Trust and Quality in Open Source systems (2010), `http://www.qualipso.org`
17. Rosen, L.: Open Source Licensing. Prentice-Hall, Englewood Cliffs (2005)
18. Sun Whitepapers: Free and Open Source Licensing (2010), `http://mediacast.sun.com/users/sunmink/media/sunlicensingwhitepaper042006.pdf`
19. TeleManagement Forum: Business Process Framework (eTOM) (2010), `http://www.tmforum.org/browse.aspx?catID=1647`
20. Yager, R.R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Transaction Systems, Man, Cybernetics 18(1), 183–190 (1988)

How to Define Software-as-a-Service – An Empirical Study of Finnish SaaS Providers

Tuomas Mäkilä[1], Antero Järvi[1], Mikko Rönkkö[2], and Jussi Nissilä[3]

[1] University of Turku, Department of Information Technology, Finland
{tuomas.makila,antero.jarvi}@utu.fi
[2] Aalto University, Software Business Laboratory, Finland
mikko.ronkko@tkk.fi
[3] Turku School of Economics, Business and Innovation Development, Finland
jussi.nissila@tse.fi

Abstract. Software-as-a-Service (SaaS) is a relatively new and much hyped model of software delivery where the software is procured as a service over the Internet. We found that there were several different definitions of SaaS. Based on these definitions, we distilled a list of five characteristics that are required for a firm to be considered a SaaS provider. Using survey data from the Finnish software industry, we tested the proposed criteria. Closer examination of the survey indicated the criteria were necessary but not sufficient. Therefore, we extended the criteria to better grasp the phenomenon and tested the changes through a qualitative validation. Also, we found that while a large number of firms are producing software that is technically SaaS, pure SaaS-based business models are much more rare in Finland.

Keywords: Software as a service, survey, qualitative study.

1 Introduction

Software industry is increasingly moving to services, and turning products into tools for vendors to sell services [1]. Software-as-a-Service (SaaS) is a concept that is often mentioned in this context. In general, SaaS refers to a software deployment model, where the software is provisioned over the Internet as a service. A key difference between SaaS and other new "on demand" models and the more traditional internet-based deployment models (such as Application Service Provider, ASP), is that the service is to some extent standardized, whereas tailored software providers can use ASP model to deliver the software. The new on demand services are not limited to providing only software application as a service, but sometimes extend to as far as business process outsourcing [2]. In this paper, we however focus on the application provisioning side since it clearly belongs to software business, while pure business process outsourcing or hardware maintenance does not.

Globally, the period during which SaaS model became well known and popular was in the mid 2000s. In 2005 IDC [3] predicted that 10 percent of enterprise software markets would move to pure SaaS model by 2009. The forecast has not fully

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 115–124, 2010.

materialized. Even though the SaaS industry is growing at 40-50 percent annually, the global SaaS market this year is estimated to be $6.6B, which is about three percent of total global software and related industry. Moreover, the user acceptance has not been great. In a survey of 333 enterprises in December 2008 in the US and UK, Gartner [4] found low level of approval from customers, describing overall satisfaction levels as "lukewarm". Particularly high cost of service, difficulty with integration, and technical requirements were seen as issues. Still, Gartner concluded that SaaS has become mainstream and growth of SaaS business would continue.

SaaS is a difficult topic for a study since there is no one generally accepted definition of the concept. Instead, there are multiple related phenomena without clear boundaries; traditional application software providers, SaaS providers, and even web portals, online media and media business. It is clear that SaaS is about selling a productized software service and doing this online. However, the exact borderlines between SaaS model and other e-commerce models are unclear. The purpose of this paper is to clarify the definition of SaaS by setting criteria for classifying companies as SaaS and to test this classification approach empirically.

The structure of the article is as follows. In the next section we will review some of the literature on Software-as-a-Service covering a broad range of topics and disciplines from computer science to management. Next we describe our research approach that uses a combination of mail survey and qualitative analysis to test our proposed SaaS classification criteria. In the empirical results chapter, we will go into detail with the results that the quantitative and the qualitative studies produced. Discussion of the results concludes the paper.

2 Definition of Software-as-a-Service

In this chapter we present examples of SaaS definitions and discuss the main characteristics of a SaaS solution, with an aim to understand better the nature of SaaS phenomenon. The definitions of SaaS typically include both business and technical perspectives, the former being the dominating viewpoint.

The term Software-as-a-Service entered the mainstream computing vocabulary a few years into this millennium. Initially, the term was used for various forms of service oriented computing (see e.g. [5]), but is currently used for software that is provisioned over the internet and used usually with a web browser. The same naming convention is currently used also for other parts of the computing stack, e.g. Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) [6].

As a software delivery model SaaS can be considered either as an extension or as a replacement to Application Service Provisioning (ASP), which is a delivery model that contains hosting, maintenance, and support of software [7]. In both models, SaaS and ASP, the software is accessed through the internet or other computer network and the vendor charges service fees. The main difference is that in SaaS the provided software is standardized for all customers, whereas in ASP each customer uses its own instance of the software. The technical solution where several customers use a single instance of the software is called multi-tenancy [8, 10, 9]. It is often argued that multi-tenant architecture is merely a technical solution to a business problem, and thus not essential part of the SaaS business model. From the business perspective, the

key issue is whether or not there is client specific installation and operating work. This impacts the scalability of the solution, its operational and maintenance costs and the deployment speed of new software versions. While multi-tenancy is probably present in virtually all SaaS offerings, it would be possible to achieve the same business benefits by spawning a new instance of the software for each client automatically instead of serving the clients from the same instance of the software.

Table 1. Definitions for Software as a Service

Source	Definition
[12]	In the software as a service model, the application, or service, is deployed from a centralized data center across a network - Internet, Intranet, LAN, or VPN - providing access and use on a recurring fee basis. Users "rent," "subscribe to," "are assigned", or "are granted access to" the applications from a central provider. Business models vary according to the level to which the software is streamlined, to lower price and increase efficiency, or value-added through customization to further improve digitized business processes.
[11]	Software as a Service is time and location independent online access to a remotely managed server application, that permits concurrent utilization of the same application installation by a large number of independent users (customers), offers attractive payment logic compared to the customer value received, and makes a continuous flow of new and innovative software possible
[13]	Software as a service (SaaS, typically pronounced 'sass') is a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own web servers or upload the application to the consumer device, disabling it after use or after the on-demand contract expires. The on-demand function may be handled internally to share licenses within a firm or by a third-party application service provider (ASP) sharing licenses between firms.
[14]	In this form of computing, a customer runs software remotely, via the Internet, using the service provider's programs and computer infrastructure.
[15]	SaaS is different from traditional software licensing, which involves the buyer's purchasing a perpetual use license from the software publisher and then making additional investments for hardware, installation, and maintenance. In contrast, in the SaaS model, users buy a subscription to the software and the software publisher (seller) runs and maintains the software on his own hardware. Users with current subscriptions can obtain access to the software using the Internet.
[8]	Software as a Service (SaaS) is a software delivery model, which provides customers access to business functionality remotely (usually over the internet) as a service. The customer does not specially purchase a software license. The cost of the infrastructure, the right to use the software, and all hosting, maintenance and support services are all bundled into a single monthly or per-use charging.
[16]	SaaS is defined as a model of software deployment via the Internet whereby the SaaS provider licenses an application to customers as a service based on usage or periodic subscription payments. SaaS software vendors typically host the application on their own web servers or enable customers to download the application to consumer devices via the Internet.
[17]	Under SaaS, the software publisher (seller) runs and maintains all necessary hardware and software and buyers obtain access using the Internet.

Table 1 lists various definitions of SaaS in the literature. Five distinct characteristics are typically associated with SaaS in these definitions:

1. Product is used through a web browser.
2. Product is not tailor made for each customer.

3. The product does not include software that needs to be installed at the customer's location.
4. The product does not require special integration and installation work.
5. The pricing of the product is based on actual usage of the software.

The multi-tenancy aspect [9] is regarded as critical in many SaaS definitions, but was not included in the list. We regard it as a technological choice in SaaS implementation, not a critical feature from business perspective. The impact of multitenancy from user's point of view is included in the second and fourth criterion.

3 Empirical Study

To clarify the definition of SaaS and to develop a study method for identifying companies using SaaS-based business models, we developed a set of questions and conducted a survey of Finnish software companies. The sequential explanatory strategy [18] was utilized to further elaborate the result of the quantitative survey by one of the authors. After the analysis of the quantitative data, the criteria of identifying SaaS companies were evaluated independently by the other authors using qualitative techniques.

The purpose of this approach was two-fold: 1) The internal validity of the quantitative survey was evaluated. The main focus was on analyzing whether the existence of the five criteria characterizing SaaS business actually implicates that a firm provides SaaS products. Just as it is possible, but rather imprecise to identify for example a carnivorous creature from the teeth, claws and other easily identifiable "technical" qualities of the creature, the real question is if the creature hunts and eats other animals. Similarly, many firms and their products may have technical SaaS qualities, but the real question is whether the product can provide the values inherent in the SaaS model. 2) Additional information about the identified SaaS companies was collected in order to better understand the quantitative results and summarize the state of SaaS phenomenon in Finnish software product firms.

3.1 Quantitative Survey of Finnish Software Companies

To identify the SaaS providers, the survey started from the premise that a SaaS provider must consider itself as a software firm. This excluded for example a large share of web portals from the study. To identify which software firms are SaaS providers and which are not, the survey used the five criteria presented in the literature review above.

The five items were measured on a five point Likert scale measuring the degree of the agreement on the statement (1=Strongly disagree to 5=Strongly agree). In addition to this scale, the survey form also contained other questions related to for example firm size, growth, and internationalization. In the ideal case, categorized as "pure SaaS", the responses were positive (Agree or Strongly Agree) onto all the five questions. In the "high SaaS characteristics", one response was allowed to deviate. The "Web based solutions" group included two to three positive responses, and largely consisted of hosted non-standard solutions. In addition to the SaaS scale, the

questionnaire included a question of nine different revenue sources (e.g. license sales) and the respondents were instructed to estimate how many percents of their total revenue came from each of these sources. The survey form is available online at *http://www.softwareindustrysurvey.org/*

The data was a subset of a larger data set that was collected as a part of a research project surveying the software firms in Finland. The sample was a list of firms considered to cover the whole Finnish software product industry and majority of the service firms as well. Oversampling was necessitated by the project resulting in the inclusion of 3396 firms in the list, of which 60% was estimated to belong to the population. The sampling frame is further described in [19]. One recognized limitation of the SaaS analysis was that these questions were presented to only a subset of firms. The focus of the SaaS questionnaire was on SMEs and therefore the public listed firms and microenterprises were not included in the study.

The survey was implemented following a modified version of the tailored survey design method [20]. The mailing begun with a pre-notice letter, followed by the main survey package using postal mail. After initial mailing each firm was contacted two to four times using email and telephone. A printed questionnaire and a web form were offered as alternative options for the informants. This phase of the data collection lasted from June through August 2009.

The data was analyzed using Intercooled Stata, version 10.1. After data preparation the data was analyzed with the help of plots and tabulations. In the data screening phase also factor analysis and cluster analysis were used to familiarize with the data. The results from these analyses were not interesting, so we decided to exclude them. Finally, cluster analysis was used to develop a classification based on revenue sources. This analysis is reported in detail in [19] and excluded due to space constraints.

3.2 Qualitative Validation

We performed a qualitative analysis of the SaaS providers identified in the survey using the providers' web-pages as data source. Our goal was to test the technical characteristics, which was used to define SaaS, and analyze whether the criteria were able to capture the essence of SaaS.

Our perception was that the original criteria concentrated on analyzing the SaaS-like aspects of the product itself but did not evaluate whether the actual SaaS business model is in use. To test the possible flaws in the original criteria, two additional characteristics were added: 6) Does the product has multitenant architecture and 7) Can the product be purchased on-line on-demand. These additional characteristics were selected from several possible choices as the most prominent ones. Multitenant architecture is a technical characteristic that is directly connected to many of SaaS business benefits. Therefore it seemed to be a good attribute for analyzing the business potential of a SaaS solution, although the survey omitted this characteristic as too technical detail. On-line on-demand purchasing characterizes many of the well-known SaaS products. This was also an easy attribute to identify through firm web pages and was therefore selected.

In the qualitative study we investigated the web pages of the firms that had met either five ("Pure SaaS") or four ("High SaaS characteristics") characteristics of the

original criteria. The existence of the original characteristics and two added characteristics were determined by examining the SaaS product descriptions in the providers' web pages. Based on the personal experience and the evidence found from the web site, the researchers made a subjective judgment whether the analyzed SaaS product 1) was pure SaaS, 2) had high SaaS characteristics or 3) was not SaaS service at all. In addition, the researchers took notes about the nature of the found SaaS product.

One shortcoming of the qualitative approach is that only the information found from the providers' web sites is used. On the other hand, we believe that one of the key characteristics of a SaaS product is that most of the transactions between service provider and buyer can be done or at least initialized online. Therefore the web sites provide a good view into phenomenon from the buyer's perspective.

4 Empirical Results

The first interesting thing is how prevalent SaaS firms are. Figure 1 illustrates the number of firms in different SaaS classes in our sample. In total, we can see that "pure SaaS" model is very rare with only 4% of the responding firms qualifying with all five criteria. Analysis of revenue (not reported) indicated that these firms were predominantly small and accounted for only one percent of the total revenue of the sample.

If we include also those firms that have "high SaaS characteristics" i.e. have only one SaaS characteristics missing, the size of the subset of the sample increases to 17 percent of the firms and to six percent of the revenues. Based on these values the Finnish software industry is at least at median level in SaaS transformation compared to the global industry that has below five percent of the revenues coming from SaaS. However, this conclusion includes a caveat that the firms in the sample do not include any large firms.

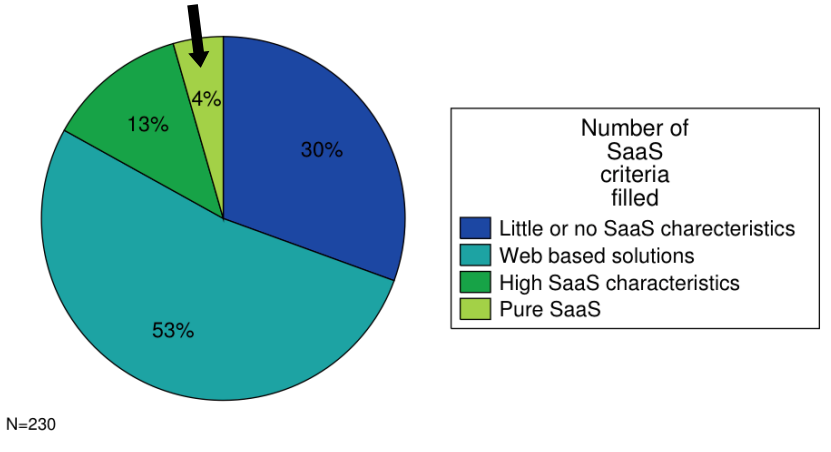

Fig. 1. Prevalence of different types of SaaS firms in the sample ("Pure SaaS" section marked with arrow)

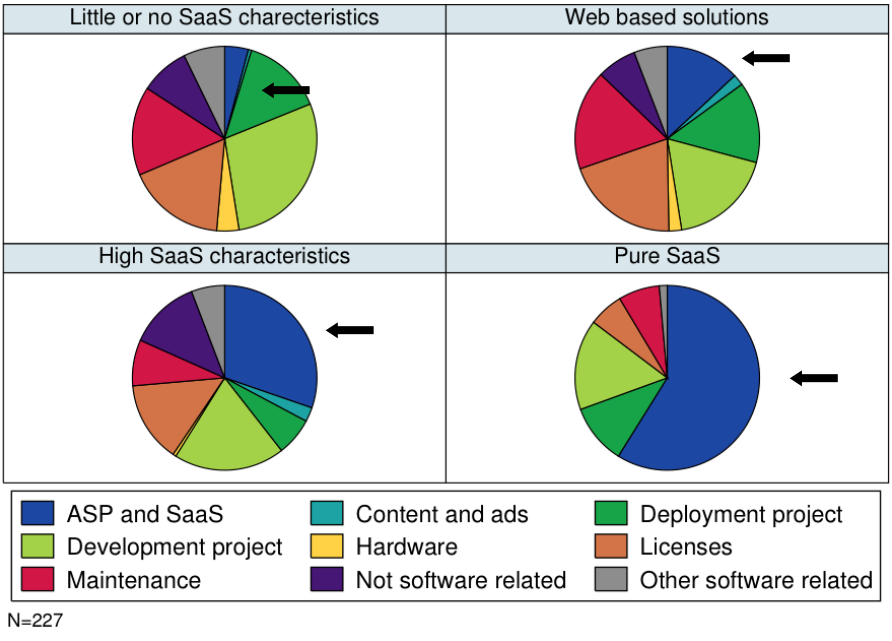

Fig. 2. Revenue shares of firms in different SaaS classes ("ASP and SaaS" section marked with arrow, other categories follow clock-wise in row-by-row order)

After analyzing the prevalence of SaaS firms in the software industry, we will now take a closer look of what these firms are and particularly what the firms that almost meet the criteria are. Based on the self-reported sources of revenue, there is great variance in how these companies make money, as shown in Figure 2. The most distinguishing factor with the "high SaaS" and "pure SaaS" firms is that the "high SaaS" firms deliver products that require customer specific deployment. This resembles the traditional ASP model where a new instance of an application is opened and configured for each customer and offered on the servers of the provider. To make classification of the firms yet more difficult, these firms are sometimes promoting themselves as SaaS firms to show that they are keeping up with the latest trends. If the firm argues itself as being a SaaS firm and the customers agree, but the firm does not meet our five criteria, is the firm as SaaS firm or not? As the definition of SaaS is not yet fully stabilized, it is difficult to give a clear answer.

In Table 2 we cross-tabulate the categories of the firms meeting different numbers of SaaS criteria with the business model classification developed with cluster analysis of the revenue share data. The "pure SaaS" providers are predominantly classified in the ASP/SaaS or product firms based on the business model classification, and even with the firms meeting four of the five criteria ("high SaaS") the standardized offering based business model is in the majority. But already in this category, the development service firms have over 20 percent share, indicating that even if a firm has a product that conforms to all or most of our SaaS criteria, this is not categorically removing the need for customer adaptation and tailoring. As a sign of the existence of continuum

from software to content and producing services based on the software, there are multiple firms having content as their main business model and 17 percent of firms do not have software as their main business.

Table 2. Cross-tabulation of business models and SaaS criteria

	Number of SaaS criteria filled				
Business Model	Little or no SaaS charac-teristics	Web based solutions	High SaaS characteristics	Pure SaaS	**Total**
Software product	26.9%	35.5%	17.2%	10.0%	29.5%
Deployment project	11.9%	14.9%	0.0%	0.0%	11.5%
Development service	34.3%	21.5%	20.7%	20.0%	25.1%
ASP and SaaS	4.5%	11.6%	37.9%	70.0%	15.4%
Not software	11.9%	9.9%	17.2%	0.0%	11.0%
Content and ads	0.0%	2.5%	3.4%	0.0%	2.8%
Software consulting	6.0%	2.5%	3.4%	0.0%	3.5%
Hardware	4.5%	1.7%	0.0%	0.0%	2.2%
Total	100.0%	100.0%	100.0%	100.0%	100.0%

During the qualitative validation the web pages of "Pure SaaS" and "High SaaS characteristics" firms were benchmarked against established SaaS providers (e.g. Salesforce.com, Google Apps, Amazon EC2) and their way to communicate about the SaaS products through their web sites. Analysis of the "pure SaaS" category firms revealed that almost none of the firms came close to the benchmark level; It was not immediately clear what the product actually was, what was the pricing, and how the product could be purchased. On the other hand, all of the firms in the "pure SaaS" category clearly had a web-based product with high productization level.

The more detailed results of the analysis on the "pure SaaS" category firms are presented in Table 3. The table shows that firms were not eager to reveal information on the more business-related the characteristics of the product (i.e. the 4th and 5th criterion) in their web sites. If the original criteria had been extended with on-line on-demand purchase requirement, almost none of the firms would have been classified into "Pure SaaS" category.

Based on this analysis, the "pure SaaS" category of the quantitative survey gives no direct false positives but neither identifies correctly pure SaaS businesses if the

Table 3. Qualitative analysis on the characteristics of the "pure" SaaS firms

Characteristics	**No**	**Uncertain**	**Yes**
1. Product is used through a web browser.	8,7 %	13 %	78,3 %
2. Product is not tailor made for each customer.	8,7 %	30,4 %	60,9 %
3. The product does not include software that needs to be installed at the customer's location.	4,3 %	17,4 %	78,3 %
4. The product does not require special integration and installation work.	13 %	47,8 %	39,1 %
5. The pricing of the product is based on actual usage of the software.	13 %	56,5 %	30,4 %
6. Does the product has multitenant architecture	4,3 %	47,8 %	47,8 %
7. Can the product be purchased on-line on-demand	78,3 %	17,4 %	4,3 %

term SaaS is interpreted strictly. Conclusion might be that the original criteria miss some key characteristics of the SaaS businesses. “High SaaS characteristics” category had false positives but also included pure SaaS firms. This suggests that the criteria might be redundant and weight of different characteristics is not equal.

The more detailed analysis of the firms in “Pure SaaS” and “High SaaS Characteristics” categories showed a high variation in business model and product types. Some firms sold software as their core business whereas for the others the software was clearly an addition to consultancy and training services. The product complexity varied from simple standalone solutions to large systems involving several individual software products. Some firms operated locally in Finland without notable growth potential or willingness to grow, while others had global presence and clear growth potential. It also seemed that the word SaaS itself possessed some kind of hype value. Therefore in some cases a web-based software product was advertised as a SaaS product, although the pure SaaS business model was not yet fully implemented nor full SaaS characteristics were met.

5 Discussion and Conclusions

The definition of SaaS is significant for understanding the essence of the SaaS as a phenomenon. Without a proper definition a scientific, especially quantitative, analysis is impossible. In pragmatic level a clear definition is needed for the SaaS providers to identify unleashed business potential, and for the customer to evaluate SaaS providers’ ability to produce long-term value.

Based on the variation in businesses of Finnish software product firms, it is clear that the firms have not implemented SaaS offering and -business in uniform way. Thus it is difficult to capture the SaaS phenomenon into a single set of simple criteria for identification purposes. Such criteria would tell if the firm has a SaaS offering in "technical" sense, but would not identify all the firms that run a SaaS based business. Instead, the identification criteria should reflect more if the firm is trying to benefit from generally expected business benefits of SaaS; Such as low implementation cost, fast sales cycle and deployment, and flexible pricing that fits the customers perceived value.

As a conclusion, we can say that in the maturing software business, SaaS is a small segment having characteristics of an emerging industry. According to market research companies, SaaS is growing fast and approaching mainstream at the global markets. While the Finnish SaaS industry has not been standing still, its development during recent years has possibly not been as good as generally assumed, especially when compared to the global development.

The main finding in this study is that Finnish SaaS firms typically have a technically mature SaaS product, but have not elaborated their business models to fully take advantage of the potential of SaaS. In other words, the firms in the sample produce software with all the technical characteristics of SaaS, but the sales model follows the traditional ASP model more than SaaS.

Considering the overall development towards SaaS, this can in the near future mean that the majority of the application areas where SaaS is used will be taken over by global competitors and potential domestic advantage of smaller regional players disappear.

References

1. ITEA Roadmap for Software-Intensive Systems and Services, 3rd edition. ITEA 2 Office Association, Eindhoven, NL (2009)
2. Pettey, C., Stevens, H.: Gartner says Worldwide SaaS Revenue to Grow 22 Percent in (2009), http://www.gartner.com/it/page.jsp?id=968412
3. Traudt, E., Konary, A.: Worldwide and US Software as a Service 2005-2009 Forecast and Analyses: Adoption for the Alternative Delivery Model Continues, IDC, Framingham, MA (2009)
4. Plummer, D.: The Real Truth About Cloud, SaaS and Saving Money Now, Webinar (2009)
5. Gold, N., Mohan, A., Knight, C., Munro, M.: Understanding service-oriented software. IEEE Software 21, 71–77 (2004)
6. Schaffer, H.E.: X as a Service, Cloud Computing and the Need for Good Judgement (2009)
7. Susarla, A., Barua, A., Whinston, A.B.: Understanding the 'Service' Component of Application Service Provision: An Empirical Analysis of Satisfaction with ASP Services. In: Information Systems Outsourcing, pp. 481–521 (2006)
8. Sun, W., Zhang, K., Chen, S., Zhang, X., Liang, H.: Software as a Service: An Integration Perspective. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 558–569. Springer, Heidelberg (2007)
9. Chang Jie Guo, W.S.: A Framework for Native Multi-Tenancy Application Development and Management, Tokyo, pp. 551–558 (2007)
10. Mietzner, R., Metzger, A., Leymann, F., Pohl, K.: Variability modeling to support customization and deployment of multi-tenant-aware Software as a Service applications. In: ICSE Workshop on Principles of Engineering Service Oriented Systems, pp. 18–25. IEEE Computer Society, Los Alamitos (2009)
11. Sääksjärvi, M., Lassila, A., Nordström, H.: Evaluating the Software as a Service Business Model: From CPU Time-Sharing to Online Innovation Sharing, pp. 177–186 (2005)
12. Hoch, F., Kerr, M., Griffith, A.: Software as a Service: Strategic Backgrounder. Accessed 21, 4 (2001)
13. Software as a service - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Software_as_a_service
14. Campbell-Kelly, M.: Historical reflections - The rise, fall, and resurrection of software as a service. ACM Commun. 52, 28–30 (2009)
15. Choudhary, V.: Comparison of Software Quality Under Perpetual Licensing and Software as a Service. Journal of Management Information Systems 24, 141–165 (2007)
16. Huang, K.W., Wang, M.: Firm-Level Productivity Analysis for Software as a Service Companies. In: Proceedings of ICIS 2009, p. 21 (2009)
17. Choudhary, V.: Software as a service: Implications for investment in software development. In: Hawaii International Conference on System Sciences, p. 3464 (2007)
18. Creswell, J.: Research design: qualitative, quantitative and mixed method approaches. SAGE, London (2002)
19. Rönkkö, M., Ylitalo, J., Peltonen, J., Koivisto, N., Mutanen, O., Autere, J., Valtakoski, A., Pentikäinen, P.: National Software Industry Survey 2009. Helsinki University of Technology, Espoo (2009)
20. Dillman, D.: Mail and internet surveys: The tailored design method (2007)

The "As-a-Service"-Paradigm and Its Implications for the Software Industry – Insights from a Comparative Case Study in CRM Software Ecosystems

Daniel Hilkert, Christian M. Wolf, Alexander Benlian, and Thomas Hess

LMU Munich, Institute for Information Systems and New Media,
Ludwigstrasse 28, 80539 Munich, Germany
{hilkert,wolf,benlian,thess}@bwl.lmu.de

Abstract. By presenting insights from our comparative case study of two CRM ecosystems, we indicate how "as-a-service"-based ecosystems differ from traditional "on-premise" ecosystems and how the particular roles of the market players might change due to the increasing diffusion of the "as-a-service" paradigm. Within the scope of our case studies, we differentiate two views on the ecosystems: the relationship between the platform provider and providers of complementary extensions (ISVs) as inside perspective and the relationship between customers and the CRM ecosystem as a whole as outside perspective. Based on transaction cost theory and intermediary theory our results let us assume that (1) "as-a-service"-based ecosystems will have a higher level of market coordination than "on-premise" ecosystems and (2) the task profiles of intermediaries and platform providers participating in "as-a-service"-based software-ecosystems will differ from task profiles in "on-premise" ecosystems. Based on these findings, we discuss practical implications for involved market players.

Keywords: software-ecosystems, analysis of value chains and value creation structures, "as-a-service" vs. "on-premise", cloud computing, transaction costs, intermediaries, case-study research.

1 Introduction

Following the latest news from the software industry, buzz words such as cloud-computing or platform- and software-as-a-service have gained ground in the attention span of IS executives. Driven by numerous examples of successful pure software-as-a-service providers like Salesforce.com, traditional software companies, such as SAP with Business ByDesign or Microsoft with the Azure platform, have significantly expanded their "as-a-service" activities in recent years. These new service based products now form the basis of "as-a-service"-based software ecosystems.

The term ecosystem refers to the fact, that the platform provider together with vendors of complementary applications and services can be described as a sort of "ecosystem" in terms of a "biocoenosis" including the surrounding environment. A specific characteristic of software ecosystems' core products is that customers derive

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 125–137, 2010.

added value only if the core product is extended with functions that are outside the core competencies of the particular core platform provider and that can be delivered by independent software vendors (ISVs), instead. Hence, on the one hand participants of ecosystems commonly benefit from their commitment and on the other hand their commitment is also necessary for the long-term survival of the system. Due to these occurring reciprocal dependencies, also called indirect network effects [1], market players who are involved in these ecosystems strongly urge to align their activities and integrate themselves into the system [2-4].

Unlike "on-premise"-software, ecosystems based on the "as-a-service"-paradigm are characterized by the fact that the software is no longer sold to the customer as a *product* but operated on the infrastructure (i.e. servers, …) of the suppliers and therefore provided as a *service*. This technically entailed change is also the starting point of numerous economic and organizational implications for the involved participants of "as-a-service" ecosystems [5]. For all market players in these "as-a-service" ecosystems (including platform providers, ISVs as providers of complementary extensions, system integrators and customers) the question arises, how "as-a-service"-based ecosystems differ from traditional "on-premise" ecosystems and accordingly how the particular roles of the involved market players might change due to the increasing diffusion of the "as-a-service" paradigm.

Hence, the purpose of this paper is to compare a typical "on-premise" ecosystem with a typical "as-a-service" ecosystem within the scope of a comparative case study. In section 2, we therefore review the literature on the analysis of value creation structures and pinpoint the research gap addressed in the paper at hand. In the subsequent sections 3 and 4, we first introduce the design of our case studies and report its results. Our paper closes with a conclusion in which we summarize possible practical implications and suggest starting points for further research.

2 Related Work

The possible paradigm shift in the software industry, which we address in our research question, is to be considered on the level of the involved market players and hence aims at the analysis of value creation structures. Such analyses of value creation structures are already intensively discussed in the literature. In their overview article, Picot et al. (2007) summarize established concepts and distinguish them on a macro-, meso- and microeconomic level [6]. Models on a macroeconomic level examine the contribution of different economic sectors to the gross national product while in case of a consideration on the microeconomic level a single company is in the focus of research activities. The meso-level of analysis resides in between the aforementioned levels and refers to value creation in different branches. Since the intention of our paper is the investigation of software ecosystems, our analysis focuses on the meso-level.

Although the body of knowledge on value chains has also dealt with changes of value creation structures at the meso-level, amongst other things also within the context of digital goods [7-8], there were no studies which analyzed value creation structures of software-ecosystems in particular. Hence, beside the general literature on added value creation structures, the emerging body of knowledge on economic

changes in the software industry is chosen as theoretical basis. Here, the "application service provider" (ASP) concept which is considered as predecessor of the "as-a-service" paradigm, has been elaborately discussed (see e.g. [9-10]). Although we meanwhile know that the ASP-concept has not lead to a sustainable structural change in the software business, Cusumano (2008) could still empirically prove that a change from products to service business is taking place in the software industry [11]. Consequently, in recent literature on the software business, the "as-a-service" paradigm is still intensively discussed. In their empirical study based on transaction cost economics, Susarla et al. (2009), e.g., have identified the form of contracts as a critical success factor for service business models [12]. Regarding the transformation of value creation activities, the reference model for "as-a-service" value chains by Altmann et al. represents an particularly important preliminary work of this study [13].

Summing up, we find that topical contributions agree in the assumption that the shift towards the "as-a-service" paradigm will result in a sustainable structural change of the software industry. Nevertheless, to our knowledge, no contributions exist that specifically address the differences between "on-premise"- and "as-a-service" ecosystems and indicate which changes could arise for the involved market players. This research gap is to be addressed in the paper at hand.

3 Research Concept

The design, procedures and evaluation of our case study is based on the concept and the recommendations of Yin (2008) [14]. In order to achieve an adequate level of representativeness, we consciously selected two ecosystems for our comparative case studies that can be considered as typical in terms of the respective paradigm ("as-a-service" or "on-premise") [14]. The chosen enterprises *CAS Software AG* and *Salesforce.com* as well as their respective ecosystems are therefore introduced in detail in section 3.1. In the following sections 3.2 and 3.3, we develop the applied research framework and present the theoretical backgrounds for the analysis. The detailed and transparent description of the research design ensures the repeatability and therefore guarantees the reliability of our research [14].

3.1 Presentation of the Case Study Objects

CAS Software AG. Founded in 1986, the CAS Software AG with its headquarter in Karlsruhe is the German market leader for CRM solutions for small and medium-sized businesses. In the preceding fiscal year the company gained a turnover of approx. € 35 million with 190 employees and more than 150,000 users (see: CAS website).

Since 1998 CAS produces the CRM-Groupware software "genesisWorld", which we regard in the following. Because of the fact that more than 100 ISVs develop diverse horizontal and vertical complementary extensions on it, genesisWorld is to be understood as a core product of an "on-premise" software ecosystem for the purpose of this paper. The mentioned cooperation partners produce horizontal extensions which are specific for a branch, as well as vertical extensions for additional functionalities which are independent from branches. All extensions are listed and priced in

the official solution catalog of CAS genesisWorld. Beyond that, integration partners exist in the CAS ecosystem, which offer services like the installation and configuration of the solution and can act as general contractors.

Salesforce.com. Salesforce.com was founded in 1999 by the former Oracle manager Marc Benioff with the purpose to provide enterprise applications and in particular CRM solutions over the internet. Salesforce accomplished a remarkable development since its foundation – up to date, more than 63,000 customers and 1.5 million users, e.g. numerous famous enterprises like Dell, Motorola or CNN, use the CRM solution which is now called "Sales Cloud 2". The Salesforce group, which appoints more than 3,500 employees worldwide, reached a turnover of $ 316 million in Q2 2009. Hence, Salesforce is regarded as a SaaS pioneer as well as one of the most prominent success stories in the field of SaaS providers (see: Salesforce.com website).

An important driver of the depicted growth is the open architecture of the Salesforce ecosystem. With the introduction of the "Force.com"-platform in 2007, Salesforce offers a "platform-as-a-service"-solution which enables ISVs to provide extensions of the Salesforce products as well as Salesforce-independent "software-as-a-service"-solutions. In the associated AppExchange marketplace for Salesforce extensions topically more than 1,000 horizontal and vertical extensions are listed by approx. 500 partners (see: Salesforce.com AppExchange portal).

3.2 Case Study Design

Starting point of the case study survey were interviews with experts conducted with responsible persons from the respective ecosystems[1]. To guarantee the comparability of both cases, the interviews were partly standardized, meaning that a questionnaire and an interview guide were used to structure the conversations. To ensure the validity of the research, according to the recommendations of Yin (2008), we carried out a document analysis of publicly available information like corporate websites and press releases of the participating market players in the sense of a data triangulation [14].

As we discussed in chapter 2, the analysis of ecosystems constitutes a consideration at the meso-level. Important for the structures of value chains at this level are on the one hand the relations between the platform povider and the ISVs and, on the other hand, the relations between customers and the ecosystem as a whole [4]. To take into consideration both views, our case study design is divided into the analysis from an inside and an outside perspective. We therefore consider four market players: The (1) *platform provider* is the hub of the ecosystem since he develops and provides the core system. (2) *ISVs* offer complementary extensions, while (3) *integrators* typically help (4) *customers* to install and configure the software solution. The inside perspective examines the relationship between the platform provider and the ISVs while the outside view refers to the connection between the customers and the software ecosystem as a whole, as well as the support of integrators where appropriate. These two perspectives are illustrated in Fig. 1.

[1] Altogether, we conducted 9 circumstantial interviews over a period of six months in 2009. Our interview partners included managerial executives of the respective platform providers (Salesforce.com and CAS) as well as general managers and leading employees of ISVs and system integrators.

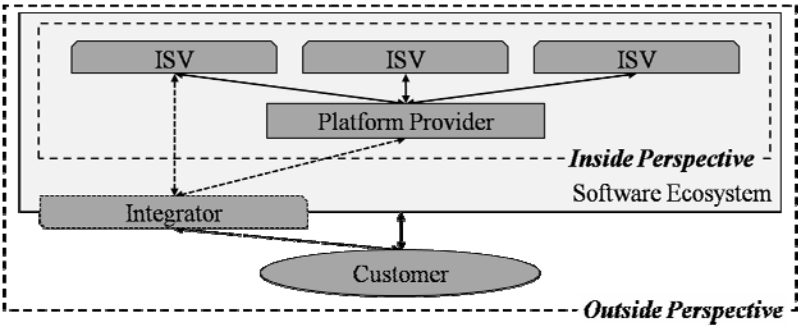

Fig. 1. Case study design: *Inside-* and *Outside Perspective* of a software ecosystem

3.3 Two Research Perspectives on Value Creation

Inside Perspective. A key aspect when considering the relationship between the platform provider and ISVs, which we refer to as inside perspective, is, to find the optimal coordination form of this relationship. The question of the optimal coordination form of an economic relationship, regarding its efficiency is often analyzed by the transaction cost theory [15]. The transaction cost theory is based on the fact that the coordination of distributed value creation activities generates costs. Therefore, transaction costs cover the costs for the initiation, agreement, execution, control, and adaption of the transaction [16]. The purpose of an analysis based on transaction costs is to determine the coordination form of a relationship that minimizes its transaction costs, depending on the value of certain parameters. Based on the assumption of limited rationality and opportunistic behavior of the involved market players, the parameters asset specificity, uncertainty and transaction frequency determine the extent of the transaction costs [17]. The transaction costs theory generally recommends hierarchic coordination for relationships which are characterized by high asset specificity and uncertainty, hybrid coordination for average specificity and uncertainty and market coordination for low specificity and uncertainty [15]. Since the transaction frequency only leads to a strengthening of the recommendation towards hierarchic coordination, if the asset specificity and/or uncertainty are increased, it is commonly considered as supportive only and can therefore be neglected in the following [18].

The purpose concerning the inside perspective therefore is to analyze, whether "as-a-service" ecosystems differ from "on-premise" ecosystems in the parameters asset specificity and uncertainty. If yes, the transaction cost minimizing coordination form in "as-a-service" ecosystems would differ from the coordination form in "on-premise" ecosystems.

Outside Perspective. In terms of the outside perspective, the relevant question particularly is to what extent intermediaries are needed to support the relationship between customers and the overall system [4]. In the given context, market players who act neither as suppliers nor as consumers, but who, for a commission, facilitate the functioning of the market are called intermediaries [15]. According to this definition, integrators, which adapt software solutions to the individual needs of customers and support the integration into their processes, represent typical examples of intermediaries in the

software industry. The central functions of intermediaries in electronic markets are (1) to supply the market participants with information (2) to organize the composition of the individual solution, (3) to build trust between the market participants and (4) to offer additional services like handling of payments or financing [15].

Therefore, based on the intermediary theory, the purpose concerning the outside perspective is to point out whether a change of market structures caused by a shift towards the "as-a-service"-paradigm also leads to a differentiated task profile of intermediaries in software-ecosystems [19].

4 Results

We derived our results by performing a qualitative content analysis of the conducted interviews[20]. A document analysis of corporate websites and press releases on the two selected ecosystems was used to complement our information basis [14]. According to the perspectives introduced above, the presentation of our case study results is subdivided into inside- and outside perspective. In chapters 4.1 and 4.2 the results from the chosen ecosystems are compared and the implications are pointed out.

4.1 Comparative Analysis of the Inside Perspective

The comparative analysis of the inside perspective is based on the transaction cost theory. As discussed in the preceding chapter, the influential factors asset specificity and uncertainty are analyzed in terms of their impact on the transaction cost occurring in the relationship between the platform provider and the associated ISVs.

Asset specificity. From the perspective of ISVs, the asset specificity of the relationship between them and the platform provider is determined in particular by the extent of their need to make specific investments for being able to provide complementary products. These investments can either be to acquire specific knowledge or to buy specific hard- and software, which would only be useful in the context of the respective ecosystem [18]. The higher these specific investments are, the higher is the resulting threat potential of the platform provider which he could use opportunistically e.g. to enforce a higher revenue share for himself. From the view of the platform providers, however, the specificity is measured through the extent to which he has to reveal specific knowledge about his product core to the ISVs. This is also correlated with an imminent behavioral uncertainty, because opportunistic ISVs could use the specific knowledge otherwise, e.g. to become competitors or to pass it on to an existing competitor of the platform provider.

Investments in transaction specific hardware and software are in both worlds equally of little importance, because the development of extensions is possible in each case with standard development tools and with no specific hardware needed. Nevertheless, ISVs associated to CAS as well as to Salesforce.com have to intensively invest in gaining specific knowledge about the respective product core before being able to produce suitable extensions. Because the documentation of how to develop complementary applications for Salesforce.com is publicly available on the Internet, it can be supposed that the obtainment of specific knowledge is easier with Salesforce.com than with CAS and accordingly that it will be easier to recruit employees

with the required skill sets. Furthermore, Salesforce.com offers numerous SOA interfaces. With their help, ISVs can integrate standardized external software. Therefore, the knowledge ISVs have to build up in the Salesforce.com ecosystem can possibly be used in other SOA-based ecosystems as well. Consequently, regarding the perspective of ISVs, asset specificity in the Salesforce.com ecosystem tends to be lower in comparison to the ecosystem of CAS genesisWorld. Because both, CAS and Salesforce.com offer open interfaces to their core systems, both platform providers only have to reveal very little information and insights about their specific product core. Because the conceptual design of Salesforce.com was very open from the beginning on, we however expect that Salesforce.com has to reveal even less specific information about the core system in their interface descriptions than CAS has to for genesisWorld. Hence, the platform providers' perspective allows us to confirm our assumption of a comparatively low specificity of the relationship between Salesforce.com and the associated ISVs. Hence, we can overall assume a higher extent of specificity in the relationship between platform providers and ISVs at CAS genesisWorld.

Uncertainty. The uncertainty as an influencing factor of transaction costs is usually subdivided into environmental uncertainty and behavioral uncertainty [17]. *Behavioral uncertainty* is especially high if one party is able to fleece his transaction partner due to an existing opportunism potential, e.g. because of high transaction-specific investments [16]. The relationship between Salesforce.com and its ISVs was built on a purely electronic processing from the very first; hence, the partner management is entirely handled via the online-platform. At CAS genesisWorld, however, interested ISVs first have to complete questionnaires and send them to the partner management office of CAS before CAS gets in touch with the potential ISV and initiates the contract negotiations. Overall it is to be stated that the relationship between Salesforce.com and the associated ISVs is characterized by a higher behavioral uncertainty because purely electronic connections generally offer bigger opportunities for opportunistic behavior than relationships that are at least to some extent built on a personal basis [7].

Besides the just mentioned behavioral uncertainty, the second important influencing factor of transaction costs is the *environmental uncertainty*, which is characterized by the complexity and dynamic of a relationship [18]. In the case of the here examined relationship between platform providers and ISVs of a software ecosystem, complexity refers in particular to technical developments as well as to the resulting necessary abilities and knowledge of human capital, while dynamic applies to the change of complexity over time [21]. In terms of the analyzed ecosystems, we assume that the complexity and dynamics and therefore also the environmental uncertainty are a little higher in the ecosystem of CAS genesisWorld than at Salesforce.com. This is caused by the fact that Salesforce.com's platform force.com is also widely established as stand-alone programming environment and therefore Salesforce.com must guarantee a certain continuity and compatibility. In contrast, CAS genesisWorld is, like other established on-premise software systems such as SAP's R3 or Oracle's E-Business Suite, a proprietary platform that is only utilizable to develop specific extensions for CAS genesisWorld. Hence, because the impact of modifications is generally less far-reaching, CAS as a manufacturer could make changes easier and more often than Salesforce.com could do it.

Table 1. Recommendations of the transaction cost analysis

influencing factor	relative parameter value in the "as-a-service" ecosystem	"on-premise" ecosystem	result of analysis
asset specificity	lower	higher	indication for a relatively higher degree of market coordination in the selected "as-a-service" ecosystem
environmental uncertainty	lower	higher	indication for a higher degree of market coordination in the selected "as-a-service" ecosystem
behavioral uncertainty	higher	lower	indication for a higher degree of market coordination in the selected "on-premise" ecosystem

Implications. The results of our transaction cost analysis are summarized in table 1. We find that two central influential factors of the transaction cost theory, specificity and environmental uncertainty, are relatively lower in the relationship between Salesforce.com and its ISVs compared to the relationships between CAS and its partners. However, the results of the behavioral uncertainty point at the opposite direction.

Following the general notion that the asset specificity is the most important influential factor of transaction costs [16], our results allow the following careful assumption: Under the condition that the transaction partners manage to control the given opportunism potential caused by the purely electronic relationships (e.g. by implementing accordant contract conditions and adequate possibilities for sanctions), in the "as-a-service" world a higher degree of market coordination is reasonable than in the "on-premise" world because this coordination causes the comparatively lowest transaction costs in the relationship between platform providers and ISVs. The following scenario is imaginable: the platform provider of an "as-a-service" ecosystem establishes a special online marketplace that enables ISVs to offer their extensions to the customers of the ecosystem. In doing so, the core manufacturer keeps control of all terms and especially entry conditions of this marketplace. As we will elaborate more thoroughly in the following chapter, Salesforce.com, for example, has already established an element of market coordination by offering the AppExchange Marketplace.

4.2 Comparative Analysis of the Outside Perspective

The analysis of the relationship between customers and the CRM-system as a whole, which is covered by the outside perspective, aims to reveal whether the presumed paradigm shift results in a modified task profile of integrators. The analysis is carried out along the central functions of intermediaries as introduced in chapter 3.3.

Supply of market players with information. Especially in opaque markets the supply of market players with information is an important function of intermediaries. Salesforce.com offers a considerably larger number of extensions, currently about 1,000, compared to CAS genesisWorld (with about 100 extensions) or other typical "on-premise" software ecosystems. This implies a greater need of a central bundling

of information by an intermediary. Salesforce.com is, however, trying to shape the market for extensions as transparent as possible and has established the AppExchange platform for this purpose. Besides providing a complete list of all available extensions and their prices, this platform allows commenting and evaluating extensions as well as browsing reviews of other users. At CAS genesisWorld all extensions are documented in an official solution directory, too, but a public platform for ratings or comments does, however, not exist. Altogether, the higher number of available extensions in the "as-a-service"-ecosystem entails a stronger need for an intermediary to bundle and provide information about the supply. In this case, however, this task is undertaken by the platform provider itself.

Composition and configuration. The second fundamental function of an intermediary is to support customers in the composition and configuration of the specifically customized software solution. Due to the fact that CAS tries to establish long-term relationships with their partners, the frictionless integration of extensions is largely guaranteed. Because of the high complexity of their software system, it is however intended that the installation and configuration is supported by CAS or a certified partner. In accordance to the SaaS-paradigm, the solution of Salesforce.com is operated on their dedicated infrastructure and accessed by customers solely via the browser. Hence, an installation is not necessary and accordingly the integration of specific extensions can be done with a few clicks by the customers themselves. Although Salesforce.com does offer support services (especially to key accounts), such as the adaptation of special templates, customers should generally be able to accomplish most of the necessary adjustments by themselves. Overall, our analysis indicates a higher need for clients of the CAS genesisWorld ecosystem to task an intermediary with the composition and configuration of the software solution than for clients of Salesforce.com.

Trust Building. Another function of intermediaries arises from the need to build trust between the various market players. Salesforce.com addresses this issue by providing the above mentioned evaluation and annotation features and thus ensures a certain transparency and security within their extension marketplace. As the partner network of CAS genesisWorld consists of significantly fewer partners, who moreover are tightly bounded to CAS by means of detailed partnership agreements, however, it can be assumed that within the CAS genesisWorld ecosystem the need of confidence building between customers and complementors is less important in comparison to Salesforce.com.

Additional services. In addition to the mentioned core functions, intermediaries often offer additional services to support the functioning of the market. One of these additional functions is the handling of payments. Whereas at CAS genesisWorld integrators can appear as general contractors, and so undertake this function, Salesforce.com takes care of the handling of payments by itself. Furthermore, according to the SaaS paradigm customers of Salesforce.com do not have to pay for the acquisition of the software and pay a monthly fee instead. Hence, for customers of Salesforce.com no significant costs of acquisition occur and accordingly generally no need of financing these upfront costs arises.

Table 2. Results of the intermediary functions analysis

function of intermediaries	greater need in
information supply of market participants	"as-a-service" ecosystem
composition and configuration of the software system	"on premise" ecosystem
trust building	"as-a-service" ecosystem
additional services (handling of payments and financing)	"on premise" ecosystem

Implications. The results of the intermediary functions analysis are summarized in Table 2. The analysis of the two considered cases indicates a change of integrators' task profiles as a result of the postulated shift towards the "as-a-service" paradigm. Especially those integrators that have specialized on the composition and configuration of software systems and/or that were acting as general contractors in the sense of handling payments and offering financing will possibly have to take on a different role in "as-a-service" ecosystems which are comparable to the one we considered.

Our analysis reveals that besides the informational function, which is largely undertaken by the platform provider itself, the function of building trust will gain further importance. A possible scenario for the change in the specific role of the integrators would be that in addition to traditional integration services, trust building services such as customized security consulting will become increasingly important in "as-a-service" ecosystems. How necessary these confidence-building services can be, became obvious at Salesforce.com: the SaaS-provider had to react on customer demands for more transparency and now provides permanently updated data on safety and the system status of the infrastructure on their publicly available portal trust.salesforce.com.

5 Limitations, Further Research and Discussion

The goal of this paper was to take a closer look at possible changes in the software industry resulting from the increasing diffusion of the "as-a-service" paradigm. For this purpose, we analyzed the value creation structures of an exemplary "on-premise"-(CAS genesisWorld) and an exemplary "as-a-service" software ecosystem (Salesforce.com). The transaction cost theory was used for the inside perspective and the intermediary theory has built the basis for the analysis of the outside perspective.

The results of both views let us assume that a paradigm shift towards "as-a-service" based ecosystems will result in changes for all involved market players. It should be noted, however, that, although the usual quality criteria for case studies have been sufficiently taken into account, a generalization of the results has only a limited legitimacy [14]. The findings presented in chapter 5.1 are therefore consciously formulated as propositions which should be understood as starting points for further research on the given topic. Moreover, in chapter 5.2 we discuss concrete practical implications resulting from our findings.

5.1 Propositions for Further Research

Proposition 1: "As-a-service"-based software ecosystems will have a higher level of market coordination than "on-premise" ecosystems. The tendency towards a

comparatively lower specificity and environmental uncertainty, which was detected in the context of the inside perspective analysis, could indicate a relatively higher degree of market coordination in "as-a-service" ecosystems, because a coordination implementing a market would result in comparably lower absolute transaction costs in the relationship between the particular platform provider and the associated ISVs. For established ISVs this development raises the question whether they have to be afraid to lose their contractually secured position in the future and therefore encounter the mere price mechanism of market coordination. Platform providers will face the challenge to retain those ISVs that are providing especially important extensions within their ecosystem and therefore control the increased behavioral uncertainty resulting from the trend towards all-electronic extension marketplaces.

Proposition 2: The task profiles of intermediaries and providers participating in "as-a-service"-based software ecosystems will differ from task profiles in "on-premise" ecosystems. Consistent with comparable findings on "as-a-service" implications, our analysis showed that a significant shift of task profiles is to be expected [22-23]. Our findings imply that the importance of conventional integrators that primarily undertake the task of composing and configuring the software solution will comparatively rather decrease in "as-a-service" ecosystems, because providers themselves might undertake this task. At the same time, our analysis however reveals that the task of building trust becomes more relevant. Hence, a possible scenario would be that integrators increasingly focus on addressing the confidence issues occurring in the "as-a-service" context, for example by becoming a "trusted third party" [24]. Conversely, providers whose role in "on-premise" ecosystems is especially the production of the core software will undertake additional tasks in "as-a-service" ecosystems. Together with the higher degree of market coordination (see proposition 1), our findings imply that platform providers will increasingly undertake the task of supplying the market participants with information. So, beyond the role of being pure software manufacturers, platform providers will increasingly also act as orchestrators of their platform ecosystems.

5.2 Discussion of Practical Implications

Though, as we already denoted above, only a limited generalizability of our results is possible, we want to discuss which implications could arise for the individual market players of "as-a-service"-based software ecosystems.

As our results indicate that in "as-a-service" ecosystems the supplies of extensions will rather be organized in markets, we expect an intensification of the competition among the ISVs compared to "on-premise" ecosystems. Hence, *customers* of "as-a-service"-based solutions should find a wider variety of extensions at lower prices. In addition, upfront costs for the integration and configuration of the software solutions will rather decrease, because customers become less reliant on integrators. As a result, it can be assumed that *integrators* undertake more confidence-building functions in "as-a-service" ecosystems, because their function of customizing and integrating the software solution becomes less relevant. For example, established integrators could offer additional services, like legal counseling services, involving aspects of individual forms of contracts and the potential risks, which could arise from the global distribution of the IT-infrastructure. In doing so, integrators could use their reputation to

convince those customers, who are still critical for "as-a-service"-solutions especially because of security concerns. *ISVs* that could rely on their contractually protected position within the ecosystem so far are likely to face greater competition in "as-a-service" ecosystems. Due to the increased importance of markets, it will in reverse also be easier for ISVs to offer their extensions in further ecosystems and there to assert themselves against well-established competitors. We expect the task profile of *platform providers* to change as well. In addition to the changes, which are directly determined by the "as-a-service"-paradigm, such as the development of an own multi-tenant server infrastructure or the implementation of a service-based pricing model, providers will most likely have to fundamentally rethink the character of their relationship with customers, integrators and ISVs. Whereas customers of "on-premise" solutions were tied to providers because of extensive expected switching costs, this lock-in effect is much lower with "as-a-service" solutions. Therefore, it can be assumed that "as-a-service" platform providers will increasingly invest in customer loyalty. Also the shift of partner management, moving away from a small number of partners bound by contracts towards a market-organized ecosystem, will make it inevitable for platform providers to develop new skills in the management and orchestration of ecosystems.

References

1. Church, J., Gandal, N.: Complementary network externalities and technological adoption. International Journal of Industrial Organization 11(2), 239–260 (1993)
2. Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: A research agenda for software ecosystems. In: Proceedings of the 31st International Conference on Software Engineering (ICSE 2009), Vancouver (2009)
3. Boucharas, V., Jansen, S., Brinkkemper, S.: Formalizing software ecosystem modeling. In: Proceedings of the 1st International Workshop on Open Component Ecosystems, pp. 41–50. ACM, Amsterdam (2009)
4. Messerschmitt, D., Szyperski, C.: Software Ecosystem. The MIT Press, Cambridge (2005)
5. Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W., Stößer, J.: Cloud Computing - A Classification, Business Models and Research Directions. Business & Information Systems Engineering 1(5), 391–399 (2009)
6. Picot, A., Schmid, M., Kempf, M.: Die Rekonfiguration der Wertschöpfungssysteme im Medienbereich. In: Hess, T. (ed.) Ubiquität, Interaktivität, Konvergenz und die Medienbranche, pp. 205–257. Univ.-Verl. Göttingen, Göttingen (2007)
7. Bailey, J.P., Bakos, Y.: An exploratory study of the emerging role of electronic intermediaries. International Journal of Electronic Commerce 1(3), 7–20 (1997)
8. Dietl, H., Royer, S.: Intrasystem competition and innovation in the international videogame industry. Innovation: Management, Policy & Practice 5(2-3), 158–169 (2003)
9. Seltsikas, P., Currie, W.: Evaluating the Application Service Provider (ASP) Business Model: The Challenge of Integration. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS), pp. 2801–2809. IEEE Computer Society, Hawaii (2002)
10. Smith, M.A., Kumar, R.L.: A theory of application service provider (ASP) use from a client perspective. Information & Management 41(8), 977–1002 (2004)

11. Cusumano, M.A.: The Changing Software Business: Moving from Products to Services. IEEE Computer 41(1), 20–27 (2008)
12. Susarla, A., Barua, A., Whinston, A.: A Transaction Cost Perspective of the Software as a Service Business Model. Journal of Management Information Systems 26(2), 205–240 (2009)
13. Altmann, J., Mohammed, A.B., Hwang, J.: Grid Value Chains: Understanding Business and Value Creation in the Grid. In: Neumann, D., Baker, M., Altmann, J. (eds.) Economic models and algorithms for distributed systems. Springer, Heidelberg (2009)
14. Yin, R.: Case study research: Design and methods. Sage Publications, Thousand Oaks (2008)
15. Picot, A., Reichwald, R., Wigand, R.: Information, Organization and Management. Springer, Berlin (2008)
16. Williamson, O.: The economic institutions of capitalism: firms, markets, relational contracting. Free Press, New York (1998)
17. Williamson, O.: The economics of governance: framework and implications. Journal of Institutional and Theoretical Economics 140(1), 195–223 (1984)
18. Poppo, L., Zenger, T.: Testing Altermative Theories of the Firm: Transaction Cost, Knowledge-Based and Measurement Explanations for Make-or-Buy Decisions in Information Services. Strategic Management Journal 19(9), 853–877 (1998)
19. Buxmann, P., Diefenbach, H., Hess, T.: Die Softwareindustrie: Ökonomische Prinzipien, Strategien, Perspektiven. Springer, Berlin (2008)
20. Mayring, P.: Qualitative Inhaltsanalyse: Grundlagen und Techniken. Beltz, Weinheim (2008)
21. Duncan, R.B.: Characteristics of Organizational Environments and Perceived Environmental Uncertainty. Administrative Science Quarterly 17(3), 313–327 (1972)
22. Anding, M.: SaaS: A Love-Hate Relationship for Enterprise Software Vendors. In: Benlian, A., Hess, T., Buxmann, P. (eds.) Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen, Gabler, Wiesbaden (2010)
23. Bossert, O., Freking, U., Löffler, M.: Cloud Computing in Practice – Rain Doctor or Line-of-Sight Obstruction. In: Benlian, A., Hess, T., Buxmann, P. (eds.) Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen, Gabler, Wiesbaden (2010)
24. Benlian, A., Hess, T., Buxmann, P.: Drivers of SaaS-Adoption - An Empirical Study of Different Application Types. Business & Information Systems Engineering 1(5), 357–369 (2009)

Software-as-a-Service in the Telecommunication Industry: Problems and Opportunities

Eetu Luoma[1], Oleksiy Mazhelis[1], and Pertti Paakkolanvaara[2]

[1] Dept. of Computer Science and Information Systems, University of Jyväskylä
firstname.lastname@jyu.fi
[2] Faculty of Economics and Business Administration, University of Oulu
firstname.lastname@oulu.fi

Abstract. This paper examines the telecommunication software market and the adoption of topical Software-as-a-Service (SaaS) model in this vertical market. The aim of this paper is to estimate which telecommunication software products are likely to be provided as a Service, and to examine the potential factors disallowing SaaS adoption. The set aim is pursued by studying both the supplying software vendors and the communication service providers (CSP) as customers. A set of thematic interviews, software industry statistics and information disclosed by the software vendors are used in the analysis. The results of the analysis reveal challenges in adopting SaaS for telecommunication software as well as suggest that SaaS offerings are more likely to appear in specific areas of CSP operations, rather than as a comprehensive solution.

Keywords: Software-as-a-Service, Communication Service Provisioning, Operations Support System, Business Support System, Business Networks.

1 Introduction

Telecommunication service providers (CSPs) use a variety of software applications to support their business processes. A crucial role in supporting CSP's daily operations, such as sales, service provisioning, service management, charging and billing, is played by the so-called Operations and Business Support Systems (OSS/BSS) automating the network infrastructure related and customers related processes, respectively.

The market of OSS/BSS software is a relatively young and dynamically evolving vertical software market. According to the model explaining the evolution of vertical software markets [1], software offerings in such markets gradually advance from proprietary software, to competing software products, and later to a dominant design. Towards the end of this evolution cycle, the competitive advantage of using software applications diminishes and they become subject to cost-efficiency considerations. Accordingly, the software vendors adjust their offerings to provide low cost applications.

We assume that under these circumstances, the vertical enterprises likely outsource the development and operations of the software applications, further causing increase

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 138–150, 2010.

in the demand of cost-efficient service offering. The most prominent approach to the cost-efficient software provisioning is Software-as-a-Service (SaaS) which can be defined as providing online access to a software product that is managed and maintained by the provider [2]. OSS/BSS software vendors are expected to eventually switch to the SaaS as their main form of software provisioning [3].

The OSS/BSS systems are large systems, composed of numerous subsystems with complex interdependencies between them. As a result, integration of these subsystems is a highly costly process [4]. To cut transaction costs, and to reduce the integration efforts, the telecommunication service providers would prefer to purchase a OSS/BSS software products covering several operational area, from a single provider [5].

This paper examines the current state of and possible future development scenario for the telecommunication software market, while focusing on the OSS/BSS software applications. The aim of this paper is to estimate which OSS/BSS software products are likely to be provided as a Service and, in addition, to examine the potential factors preventing SaaS adoption in this vertical software market.

The above aim is pursued by studying both the supplying software vendors and the CSPs as customers. Observing the demand viewpoint, we assume that OSS/BSS software applications provided as a service would be of interest to the vertical enterprises due to lower TCO of the SaaS offering [6]. Consequently, the analysis focuses on whether there are technical or business issues disallowing enterprises in the telecommunication industry to utilize OSS/BSS offerings as a service. This analysis is based on a set of thematic interviews conducted to study the CSP perspective.

From the supply viewpoint, the focus is on finding the most probable candidates to implement OSS/BSS as SaaS, i.e. finding a candidate having capabilities to produce and provide the offering. The analysis is based on two factors: i) the commitment to the development of OSS/BSS software applications and to SaaS mode of software provisioning, and ii) the coverage of vendors' current offering against industry standards. For this, both the vertical software industry statistics and information disclosed by the software vendors are used in the analysis. We consider the coverage of the vendor as an important factor in enabling SaaS offering, since the vendor needs to be capable of providing solutions covering several application areas. Otherwise, the cost benefits from SaaS would be diminished as a result of the high costs of integrating software applications and transaction costs when acquiring from several vendors.

The paper proceeds as follows. First, the envisioned OSS/BSS offering as a Service is described based on the OSS/BSS standardization efforts and SaaS-related literature. Findings of the CSP interviews on the potential issues are introduced in section 3. In Section 4, the large software vendors present in the OSS/BSS market and their current offerings are analyzed, the vendors most likely to implement a OSS/BSS solution as SaaS are identified, and the parts of OSS/BSS stack most likely to be provided as a Service are estimated. The concluding section summarizes the findings and discusses their interpretations.

2 OSS/BSS Software-as-a-Service

The OSS/BSS are systems for managing processes of the telecommunication service providers. The OSS/BSS software applications are usually supported by middleware

(such as database software), and it manages data either at customer or at telecommunication network interface [7],[8]. Initially, the software offering targeted the management of the telecommunication networks, that is, the resource management layer. More recently, emphasis has shifted to supporting the processes close to customer interface [9], thereby creating an entry initiative for companies providing software applications for other industries. In addition, the market is affected by new technologies and convergence with digital media technologies.

The essential business processes of CSPs are being standardized by the TeleManagement Forum (TMF) community. TMF is a global non-profit association that develops practical solutions and guidance in a form of benchmarks, roadmaps, best practices, and software standards and interfaces, for service providers and their suppliers. Among other activities, TMF initiatives have produced[1]:

- The enhanced Telecom Operations Map (eTOM) defining and hierarchically structuring business process of CSPs,
- The Shared Information and Data (SID) framework providing information definitions to be used across different applications, and
- The Telecom Applications Map (TAM) describing the functionality of applications and services to support the identified business processes.

The eTOM provides a CSP hierarchical business process model, where at the highest level three process areas are identified: i) Strategy, Infrastructure & Product, ii) Operations, and iii) Enterprise Management. Among these three, the Operations area is of particular interest for this paper, since the OSS/BSS software primarily automates processes in this group. In particular, the Operations area includes the vertical end-to-end processes of service fulfillment, service assurance, and billing, also facilitated with the operations support and readiness processes. The Operations area also describes the horizontal layers of customer management, service management, and resource management. [10].

An overview of the Telecom Application Map is presented in Figure 1. In TAM, the applications automating the eTOM business processes are grouped into functional areas. These functional areas employ the definitions from the SID, and they are in line with the vertical and horizontal process groupings defined in eTOM. The current map specifies a total of 158 applications [11]

The TAM is used in the paper as source of information in studying the OSS/BSS software market. Namely, we are assuming that a comprehensive OSS/BSS solution – *full-stack OSS/BSS* – is a software system providing the functionality to support all the essential operational processes of the telecommunications service provider. Since the eTOM processes are automated with applications present in the TAM, the full-stack OSS/BSS solution is expected to cover all the functional areas of the TAM. Therefore, to assess software vendors' capability of providing OSS/BSS software applications as a service, be they the full-stack or parts of it, the offerings of the vendor are compared against the TAM.

[1] http://www.tmforum.org/SolutionFrameworks/

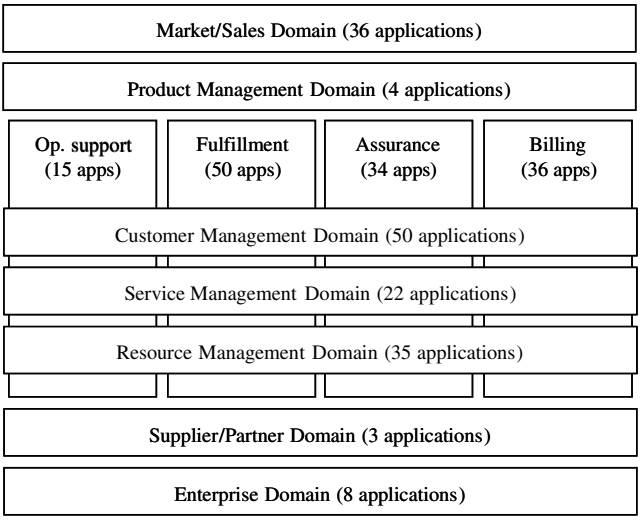

Fig. 1. An overview of the Telecom Application Map (TAM) [10]

2.1 Software-as-a-Service in the Telecommunication Industry

From the CSP's perspective, Software-as-a-Service (SaaS) refers to the means of acquiring and deploying standardized software applications, which are being operated by the software vendor or a third party. The SaaS model differs from bespoke and traditional software product business models in that the ownership of the software remains at the provider [12] and instead of a license agreement, the importance of service level agreement is stressed. There are also financial differences. Comparing SaaS model to the traditional software-related business models, the initial costs to CSPs should be much lower, fixed costs become variable costs and the source of TCO changes [13], [14]. SaaS model is said to be cost-efficient, especially for small and medium size enterprises [15].

Benlian et al. [16] studied the drivers of SaaS adoption using transaction cost theory, resource-based view of firm, and theory of planned behavior. They found higher adoption rate for applications characterized as less specific, less strategic relevant, and associated with a lower level of adoption uncertainty. In contrast, the lowest adoption rates were among applications with higher levels of specificity, strategic significance, and adoption uncertainty. Examples include office and collaboration applications in the former category and ERP systems in the latter category [16]. We suggest that OSS/BSS share the properties of ERP systems in this regard. The OSS/BSS have been developed over time on top of vendor-specific network element interfaces [17] and their integration resulted in systems with low level of modularity. The OSS/BSS also support the single most important process to the CSP, the billing process where usage records are rated and bills to the customers are produced.

From the viewpoint of CSPs, SaaS has several limitations. The high level of standardization in SaaS model implies that all customers are using the same software version (i.e. code base) and system infrastructure, including multi-tenant database. By

using commodity software applications, the CSP loses a potential for competitive advantage, and also outsources the responsibility of developing the process area to a software vendor. In case the integration into existing infrastructure is difficult, the cost benefits from SaaS offering are diminished. Further, as operating the system infrastructure is outsourced, the software applications shall be accessed over the Internet. Such deployment model together with multi-tenant database causes two issues for CSPs, namely data security issues [18] and performance issues in mission-critical software systems. Thus, in addition to covering the functional areas of TAM, the prospective full-stack vendor has to overcome the problems related to the provisioning of OSS/BSS software as a service. These problems include:

Ensuring adequate performance. Performance characteristics specified in a service level agreement need to be ensured, which may be difficult due to executing the applications remotely and using the Internet as communication channel [19].

Significant integration efforts. The full-stack implementation has to be integrated with multiple CSP networks and a multitude of services built on top of them. The network and services differ among CSPs, and the full-stack OSS/BSS solution should be flexible enough to be integrated with the environments in different CSPs.

Heterogeneous network elements. Various types of network elements are deployed by the CSPs, and many of them have proprietary protocols. As a result, the efforts of integrating with these network elements may be significant [17].

Security concerns. As the data storage and management is outsourced to the software vendor, the confidentiality, integrity, and availability of the data need to be managed effectively and transparently to the customer CSP.

3 Problems in Utilizing OSS/BSS as a Service: CSP Interviews

In order to assess, whether overcoming the above issues by software vendors is feasible, a set of thematic interviews with representatives of CSPs were conducted. The set of interviews included five European CSPs selected to represent different sizes measured by number of subscribers and breadth of operations. The breadth of operations refers to geographical classification: A regional CSP operates within a regional area in one country (less than 50 000 subscribers), the national CSPs have activities mainly in one country (with approximately 2 million subscribers), and the international CSPs have notable operations in several countries (more than 20 million subscribers). All the operators provide their consumer and business customers mobile, fixed-line and broadband network connections. However, the regional CSP does not have its own brand for mobile subscribers. The interviewees had varying positions in their organization, however, they were selected to have proper view of how software applications were acquired.

The scheme for the interviews was semi-structured [20]. Topical areas were given and both the interviewers and respondents had the opportunity to direct the conversations and bring into the discussion points they considered important. However, to guide the interview, the discussion was organized around three themes: adoption of SaaS, problems with SaaS in CSP business, and the costs of SaaS.

In the following, the analysis on the two former areas is reported. Specifically, under these themes, two specific questions were asked: i) Are you acquiring software systems with SaaS model? ii) Which factors hinder or would hinder applying SaaS model to support operations? Discussing on the latter question, the interviewer made further probing questions on whether network interfaces, security, performance requirements and integration was seen problematic by the representatives of the CSP.

One of the aims of the interviews was to examine the potential problems with deploying SaaS for OSS/BSS. In this setting, the first question was to verify whether the interviewee had considered the SaaS option to be used in their operations. Out of the five CSPs, only two (a national and an international) CSP is currently utilizing SaaS model for selected software applications. The regional CSP is planning to use SaaS model for OSS/BSS and, therefore, had estimated the benefits and problems of the deployment model. When asking on which software applications were under consideration, the interviewee answered that they would be interested in buying the whole OSS/BSS stack as a Service in case their partner, responsible for developing their OSS/BSS infrastructure, finds a functioning offering from the market. The other international CSP is not using SaaS in OSS/BSS. The interviewee responded that there are no commercial platforms, which would serve their needs. In addition, a significant effort to customize the software vendor's offering would be needed.

In Table 1, the data collected on the second question is presented. The middle column indicates whether the interviewee considers the suggested area problematic in SaaS or not. The rightmost column presents the given comments to each individual area. Although the other international CSP has not considered the SaaS option, the respondent could elaborate some problematic areas.

From the data, the following observations can be made, in accordance to the given answers. First, the interfaces with the network and integration of new software applications (i.e. deployed using SaaS model) to existing systems does not seem to prevent utilizing SaaS, although the integration effort might be laborious and adoption would therefore be a slow and gradual process. While the representatives of the international CSPs mentioned integration problematic, both of their comments relate to general problems in integrating OSS/BSS systems. The other interviewees provided similar answers. Secondly, security issues are seen as likely problems hindering the adoption of both full-stack OSS/BSS and SaaS. The CSPs rather keep confidential data within their premises and in software applications they fully control. On the performance matter, three of the respondents approached the performance requirements matter with different approach. On one hand, the real-time requirements may disallow using software applications accessed through Internet connections, especially in software applications managing charging on pre-paid subscriptions. On the other, in case reliable network connections are established, the performance can be increased merely by acquiring efficient hardware. Also, CSPs should have the tools to address this issue. In all, a cautions conclusion can be made that once the trust in SaaS is established, it may be a viable option to acquire and deploy OSS/BSS, especially on less complex domains of eTOM and TAM. However, the detailed answers suggest that for parts of the full-stack there are major problems hindering the adoption of SaaS for CSP's applications.

Table 1. Interview data on the potential problems in deploying SaaS for OSS/BSS

Problem area	Answer	Comments
Regional CSP		
Network interfaces	Not problematic	Intefaces with the network are not problematic, since these are encapsulated and available through web services.
Security	Not problematic	Security is a difficult question. Network connections are secure, but the software systems and especially browsers may have security problems. The CSP is unable to fix the problems in software systems. However, with OSS/BSS the problems can be avoided with good architectural design.
Performance	Not problematic	Performance issues are managed by the hardware vendors and the CSP is building high-capacity fibre network enabling SaaS.
Integration	Problematic	Integration is not technically challenging, but the high number of interfaces makes integration labourous. The effort to integrate new software systems is also generally challenging because of required domain knowledge. Out of the four potential challenges, integration is to most difficult.
National CSPs		
Network interfaces	Not problematic	In the less complex domains of eTOM, like CRM, SaaS is likely to appear earlier.
Security	Problematic	Security also sets boundary conditions. In the telecom industry, the regulator has predetermined that certain sensitive data cannot be stored in a foreign country. This limits the providers capable of offering OSS/BSS as a Service to CSPs. Securing data in a multitenant database is also questionable - can one trust that the data is not visible for other customers of the SaaS provider.
Performance	Problematic	Performance requirements are an issue in case customers experience disturbing latency. In time-critical software systems, such as in charging software systems taking care of pre-paid billing, this disallows using SaaS. The reliability problems due to network connections disallow using SaaS, e.g. in case VPN connections are lost. Network overhead due to the need to transfer large amount of data to/from sfotware vendor is likely to be a problem in some domains, like testing.
Integration	Not problematic	Technically integration is nowadays no longer a problem. Rather the problems arise in defining interfaces and functionalities behind them. However, a single version serving all customers is an unrealistic ideal so far.
International CSPs		
Network interfaces	Not problematic	The CSPs may not achieve competitive advantage through OSS/BSS. Rather more generic systems will gain popularity.
Security	Problematic	SaaS offering is not considered reliable. The level of trust is not sufficient to acquire software as a service. The CSP wants to manage their customer information internally.
Performance	Not problematic	Operator should have tools to deal with this on the network level. However, some realtime systems will require on-premise operating.
Integration	Problematic	The current systems are so complex and so customized that replacing current systems would not be possible. Deploying SaaS and virtualization techniques would require redefining OSS/BSS architecture. Therefore, the adoption of SaaS will happen gradually over a long time. For some applications, there are no standard interfaces. In such cases, tailoring is required that is difficult to implement in SaaS mode.

4 Potential Supply for OSS/BSS as SaaS

The OSS/BSS market was served by 186 software companies in the year 2005, out of which roughly one third were larger software vendors providing a broad portfolio of OSS/BSS software systems [7]. With the aim of examining the potential supply of OSS/BSS as a service, a study of the vendors' offering and their strategic intent were conducted. The study comprised of three steps.

First, the largest software vendors were identified by using the commercial OSS/BSS knowledge base by Dittberner Associates [21], where the products and market performance of 186 companies in the OSS/BSS industry are surveyed. The following selection criteria were used: The corporate turnover must be above $2000 Million, and the OSS/BSS turnover alone must be more than $500 Million. Thereby, a set of twelve large vendors focusing on OSS/BSS software have been determined. The analysis focused on large vendors as it was assumed that telecom-specific application requires specialized technical knowledge to customize, operate and support, and only larger OSS/BSS vendors would be capable of pursuing a SaaS solution for the market. The selected vendors are listed in Table 2.

Second, the strategies of the selected software vendors have been analyzed based on public sources. The Table 2 presents the details of the analysis, resulting in the identification of the vendors most likely to implement OSS/BSS as a service. From the initial set, the companies having little interest in OSS/BSS software development and provisioning it in the mode of SaaS were eliminated. In addition, only the companies focusing on both OSS and BSS have been preserved. Further, as the SaaS offering is assumed to be built on standardized applications, the companies deploying software product strategy have been preserved, whereas the companies focusing only on software services and system integration services were eliminated.

As a result of the analysis, three vendors are expected to provide the software systems covering several application areas of the TAM. All three - Amdocs, Ericsson and Oracle – have strong commitment to serving the telecommunication industry and broad portfolio of OSS/BSS software products[2]. However, the vendors have different backgrounds, which affect the scope of their offering. Of the three, Amdocs is the only vendor purely focusing on the OSS/BSS market. As a network element manufacturer, Ericsson has traditionally provided OSS at the resource management domain, whereas Oracle had entered the market with acquisitions of CRM companies at the customer management layer of the TAM. From the publicly available sources, the business interest of these vendors towards SaaS is not clear. Amdocs and Oracle claim to have SaaS offering[3], but information on the successful deployments seems to be missing. In addition, Ericsson does not seem to provide its software systems with SaaS model. Conversely, they are the market leader in managed services[4].

[2] http://www.oracle.com/us/industries/communications/index.htm
http://www.amdocs.com/Offerings/CES-Portfolio/Pages/CES-Portfolio.aspx
http://www.ericsson.com/ourportfolio/telecom-operators

[3] http://www.amdocs.com/News/Pages/Unveils.aspx
http://www.oracle.com/us/technologies/saas/index.htm

[4] http://www.ericsson.com/ourportfolio/services/managed-services

Table 2. Analysis on the strategies of OSS/BSS vendors

Software vendor	Analysis on the strategy
Accenture	Not likely to provide SaaS - Focus on software services.
Agilent	Excluded due to narrow focus on test and measurement applications.
Alcatel	Not likely to provide SaaS - Focus on software services.
Amdocs	Likely to provide OSS/BSS as a Service - Committed to OSS/BSS (close to 95% of the turnover). Wide offering covering both OSS and BSS. Is already providing SaaS.
Atos Origin	Not likely to provide SaaS - Focus on software services.
CGI	Not likely to provide SaaS - Focus on software services.
Convergys	Potential unclear - Focuses on BSS applications (mainly billing). Has strong on-demand offering. Partners with SaaS offering. Excluded due to narrow focus.
Ericsson	Likely to provide OSS/BSS as a Service - Strong commitment to telecom industry. Market leader in wireless network equipments. Providing OSS as managed services. Has to trust third parties in delivering BSS. Possibly sees solutions implemented as SaaS as a threat and, therefore, prepares offering to prevail in competition.
IBM	Not likely to provide SaaS - Focus on software services and hardware.
Nokia-Siemens	Potential unclear - Strong commitment to telecom industry. Focuses on billing applications, but certainly has the expertise to manage network interfaces. Excluded due to little interest to SaaS.
Oracle	Likely to provide OSS/BSS as a Service - Strong presence in telecom industry as a system infrastructure provider. Entered the OSS/BSS market through several acquisitions. Strategy targetted for economies of scale. Has existing SaaS offering.
SAP	Potential unclear - Low commitment to telecom industry. Is missing telecom-specific offering, however, has partners on their product map for telecommunications.

To examine the supply, a third step was performed, where the offering of the selected software vendors were given a closer examination. The objective was to solve whether one of the vendors is currently capable of providing a comprehensive OSS/BSS solution covering all the application in the Telecom Application Map. On the other hand, the interest was in the coverage of the selected vendors offering, to observe in which application areas the potential SaaS offering will appear. To achieve these objectives, information regarding the functionalities of the software vendors' products were gathered and mapped against the specifications of the TAM. The offerings of the vendors were evaluated according to the coverage of the application map. This analysis was performed both for the whole TAM and for the central layers and verticals. The results of the analysis are provided in Table 3.

From the result, the following estimates and remarks can be made. First, while the coverage of Amdocs and Oracle on the customer management and service management domain are relatively high, none of the vendors is likely to be capable of providing the envisioned full-stack OSS/BSS. All the three vendors are missing coverage especially at the resource management layer, close to the network interfaces. Presumably, because of its background, Ericsson has a small number of applications for

the layers closer to the customer interface. The low overall coverage of Ericsson, as opposite to high coverage of Oracle, may be explained by the high number of horizontal applications included in the Telecom Application Map.

The second observation relates to the question of which parts of OSS/BSS stack are most likely to be provided as a Service. This likelihood can be assessed based on the coverage of the vendors' offering as elaborated above: SaaS offering will be of interest to CSPs once it provides cost-advantages achieved through pre-integrated software packages. Based on the analysis of coverage, the offerings are more likely to appear in the billing, customer management, and fulfillment areas, as well as parts of the service management layer. This estimate can be based on the high coverage of the leading software vendors, further indicating that standardized software product offering does already exist in the market. Moreover, as also evidenced by the CSP interviews reported above, providing the OSS software, in particular in the resource management layer, in the SaaS mode is more challenging than providing the BSS software as a service.

An indication of existing standardized software product offering for the BSS is of relevance to the possible future developments. In addition to the larger vendors, we may see formation of business networks as a potential way to create SaaS offering for the OSS/BSS market. In such case, we expect the business networks to comprise of leading companies holding the network together, and of niche vendors filling in the technical gaps of the leading company. Examples of the leading companies would be SAP (already serving the market, strong partner program) or Salesforce (a horizontal player entering the market through BSS software).

Table 3. Coverage of the TAM by vendors' present offerings

	Amdocs		Ericsson		Oracle	
	# of apps	coverage	# of apps	coverage	# of apps	coverage
Customer Mgt.Domain	39/50	78 %	17/50	34 %	43/50	86 %
Service Mgt. Domain	19/22	86 %	8/22	36 %	19/22	86 %
Resource Mgt. Domain	19/35	54 %	13/35	37 %	11/35	31 %
Fulfillment vertical	35/50	70 %	8/50	16 %	37/50	74 %
Assurance vertical	20/34	59 %	16/34	47 %	16/34	47 %
Billing vertical	26/36	72 %	20/36	56 %	30/36	83 %
Full TAM	108/158	68 %	41/158	26 %	116/158	73 %

5 Discussion and Conclusion Remarks

The OSS/BSS software applications are widely used by CSPs to support their business processes. Often, these applications are acquired from software vendors, deployed on-premise, and used and managed by the CSPs. In future, some of the major OSS/BSS vendors are expected to start offering their software as a service. Furthermore, to reduce transaction costs, vendors will offer greater sets of OSS/BSS applications, and in the extreme case, the complete set of OSS/BSS applications - the full-stack OSS/BSS software - might be provided by a vendor as a readily integrated solution for the CSP. In this paper, the possibility for OSS/BSS software applications

to be provided as a service has been examined. For this, both the technical feasibility of such a solution has been assessed, and the capabilities of leading OSS/BSS vendors to provide such a solution have been evaluated.

In order to assess the technical feasibility, thematic interviews have been conducted with the representatives of five CSPs. Based on the interviews, the challenges of implementing OSS/BSS software applications as a service have been identified. Some of the important challenges include the security concerns, difficulty of interfacing and integrating with complex infrastructure, and strict performance requirements. As a result, a conclusion has been made that the full-stack implementation is unlikely to emerge in near future, whereas subsets of OSS/BSS application areas may be offered in SaaS mode by the vendors.

In order to evaluate the vendors' capabilities, the current offerings of three leading OSS/BSS software vendors were analyzed by estimating the percentage of the TAM application areas covered by these offerings. As a result, no single vendor capable of well covering all application areas has been found. Rather, the capabilities of the vendors varied depending on the layer and the vertical of TAM: Oracle has the widest coverage in the billing, customer management, and fulfillment areas, whereas Amdocs and Ericsson cover better the assurance and the resource management areas.

Since SaaS is more common mode of software provisioning for horizontal applications applicable across industries, it is expected that SaaS offerings are more likely to appear in the billing, customer management, and fulfillment areas, as is also indicated by the strength of horizontal player in these areas. Meanwhile, in the resource management and assurance areas, which are inherently more telecom-specific, the SaaS may be more difficult to offer, but if and when it is offered, the telecom-specific vendors have a better position to provide them.

Since SaaS is relatively new phenomenon, little is known about its likely adoption in different vertical industries. Therefore, exploratory research aiming at recognizing regularities (such as critical problems and opportunities) is appropriate [22]. Two types of data have been used for the purposes of this research: data from thematic interviews [23] and secondary data [24]. Software business research can be seen as a part of Information Systems (IS) research studying the application of ICT to organizations and management thereof. According to the IS research framework by Hevner et al. [25] combining the behavioral-science and design-science paradigms, information systems research is evaluated in terms of its relevance to the practice and the rigor with which it was conducted.

In the research reported in this paper, the relevance is implied by the fact that it addresses the existing business needs (perceived by software vendors and CSPs) to predict whether the OSS/BSS software will be provided as a service. The rigorousness of the research conduct is achieved by applying accumulated research foundations and following research methodologies. In particular, thematic interviews are conducted to identify potential problems with adopting SaaS by enterprises, and secondary data analysis (of industry statistics and software vendor public reports) is applied to find the likely candidates for implementing OSS/BSS as a service. Still, due to the exploratory nature of the study, its results should be taken with care, and further studies are needed to validate the reported findings.

Whereas the analysis in this paper focuses on the specific market of the telecommunication software, similar analysis can be applied to other vertical software

markets, and also to other forms of software provisioning. For instance, the analysis of SaaS applicability can be performed for the verticals, where software systems are highly complex and heterogeneous, notably automotive and embedded software. In this paper, only the capabilities of individual vendors to provide OSS/BSS software applications as a service have been analyzed. In future work, possible business networks of multiple vendors will be analyzed from the viewpoint of their capability and strategic interest in developing a full-stack OSS/BSS solution and offering it as a service.

References

1. Tyrväinen, P., Warsta, J., Seppänen, V.: Evolution of Secondary Software Businesses: Understanding Industry Dynamics. In: León, G., Bernardos, A., Casar, J., Kautz, K., DeGross (eds.) Open IT-Based Innovation: Moving Towards Cooperative IT Transfer and Knowledge Diffusion. IFIP International Federation for Information Processing, vol. 287, pp. 381–401. Springer, Heidelberg (2008)
2. SIIA: Software as a Service: Strategic Backgrounder, Software & Information Industry Association (2001)
3. McNeice, S.L.: The Emergence of Software-as-a-Service (SaaS) in Telecom, Stratecast Perspectives & Insights for Executives (SPIE) – 2008 #11, Frost & Sullivan (2008)
4. Unterschütz, T.: Recommendation for Actions based on the mapping of topics and activities for NG-OSS, Report of project OSS for NGN – Co-ordination of Telco Activities, Eurescom (2004)
5. Frank, L., Luoma, E., Tyrväinen, P.: Market Scope of the vendors in the OSS software market. In: Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, pp. 2096–2010 (2007)
6. Xin, M., Levina, N.: Software-as-a-Service Model: Elaborating Client-Side Adoption Factors. In: Boland, R., Limayem, M., Pentland, B. (eds.) Proceedings of the 29th International Conference on Information Systems, Paris, France (December 2008)
7. Frank, L., Luoma, E.: Telecommunications Software Industry Evolution. In: Proceedings of 7th Conference on Telecom, Internet & Media Techno-Economics (2008)
8. Terplan, K.: OSS Essentials – Support System Solutions for Service Providers. John Wiley & Sons, New York (2001)
9. Thomsen, O.: The Lean Approach. Wireless Personal Communications 38, 17–25 (2006)
10. Kelly, M.: The TeleManagement Forum's Enhanced Telecom Operations Map (eTOM). Journal of Network and Systems Management 11(1), 109–119 (2003)
11. TMF GB929: Telecom Application Map - The BSS/OSS Systems Landscape. Version 3.11. TeleManagement Forum, Morristown (October 2009)
12. Laplante, P., Zhang, J., Voas, J.: What's in a Name? Distinguishing between SaaS and SOA, IT Professional, pp. 46–50. IEEE Computer Society, Los Alamitos (2008)
13. Greschler, D., Mangan, T.: Networking lessons in delivering, Software as a Service' Part I. International Journal of Network Management 12(5), 317–321 (2002)
14. Jacobs, D.: Enterprise Software As Service. Online Services are Changing the Nature of Software. In: ACM Queue, July/August 2005, pp. 36–42 (2005)
15. Wang, R., Brown, E.G., Erickson, J., McEnroe, W., Metre, E.V.: Comparing The ROI Of SaaS Versus On-Premise Using Forrester's TEITM Approach, Measuring The TEI For Enterprise Applications series, Forrester Research (2006)

16. Benlian, A., Hess, T., Buxmann, P.: Drivers of SaaS-Adoption – An Empirical Study of Different Application Types. Business & Information Systems Engineering 1(5), 357–369 (2009)
17. Mazhelis, O., Tyrväinen, P., Matilainen, J.: Analyzing Impact of Interface Implementation Efforts on the Structure of a Software Market: OSS/BSS Market Polarization Scenario. In: Proceedings of the 3rd International Conference on Software and Data Technologies (ICSOFT 2008), Porto, Portugal (July 2008)
18. Lee, J.-N., Huynh, M.Q., Kwok, R.C.-W., Pi, S.-M.: IT Outsourcing Evolution, Past, Present and Future. Communications of the ACM 46(5), 84–89 (2003)
19. Carraro, G., Chong, F.: Software as a Service (SaaS): An Enterprise Perspective. Microsoft Corporation (2006), http://msdn.microsoft.com/en-us/library/aa905332.aspx
20. Fontana, A., Frey, J.H.: The Interview: from structured questions to negotiated text. In: Denziu, N.K., Lincoln, Y.S. (eds.) Handbook of qualitative research, pp. 645–672. Sage, London (2000)
21. Dittberner. OSS/BSS KnowledgeBase. Vendor financials 2000 and 2006 (2007)
22. Tesch, R.: Qualitative research: Analysis types and software tools. Falmer, New York (1990)
23. Hirsijärvi, S., Hurme, H.: Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö (2001)
24. Järvinen, P.: On Research Methods. Tampere, Juvenes-Print (2004)
25. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)

How Green Is Your Software?

Juha Taina

University of Helsinki, Department of Computer Science

Abstract. Green IT is a mission to reduce carbon emissions of information technology. Although immediate savings come from hardware, software also plays an important role. Since a software has a life cycle, it creates direct and indirect carbon emissions: it has a carbon footprint.

In this paper we present an approach to analyse software carbon footprints. We analyse a typical software life cycle step by step and give estimates of how large carbon footprints each step produces. A software vendor that claims to be green needs to show that his software has a small carbon footprint. From the green software point of view, it matters how to develop and deliver software, and how usable the software is.

1 Introduction

Information technology (IT) is present in practically every field, process, and system in the world. It requires a huge amount of energy and other resources. Currently at least 2% of global carbon emissions are directly due to IT systems (`http://www.gartner.com/it/page.jsp?id=503867`). The amount is expected to rise since new IT solutions are implemented every day. For example, it has been estimated that 2011 the energy consumption of the USA's servers and data centers may be more than 100 billion kWh [1]. Thus, the carbon emissions of IT systems are and will be important to measure.

The need to reduce IT energy requirements and carbon emissions has been noted all over the world. Several research projects for more resource-efficient IT are either implemented or under implementation. As Murugesan states [2]: "We are legally, ethically, and socially required to green our IT products, applications, services, and practices." It includes all computing, inside and outside the data center. It is called *green IT* [1].

Carbon emissions are measured as a *carbon footprint (CF).* A CF defines how much carbon dioxide a product or service produces. The size of a CF can be compared to how many trees are needed to absorb it in a year. Together they give a simple metric to compare greenness of IT products and service including software.

While immediate savings are easiest to achieve from hardware and large data centers, also software can be green. Software has a life cycle, so it also has a carbon footprint. We can estimate and measure it, analyse its sources, and find ways to reduce it.

In this paper we analyse how the CF of a software can be calculated. We first define the life cycle of a software package and then analyse each step in the cycle. Each step has a CF that can be estimated.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 151–162, 2010.

2 Idea

It is not straightforward to define a CF for a software. A software does not directly require resources: the hardware behind it requires them. However, a software has indirect resource requirements. Its development, delivery, and maintenance create a CF. It uses CPU cycles that create a CF. It may request services from peripherals that create a CF. All these can be calculated during the software life cycle.

The life cycle of a software can be divided into the following phases:

1. Development: The software is implemented.
2. Beta testing: Potential customers test the software.
3. Delivery and re-delivery: The software is delivered to customers.
4. Usage: The software is used.
5. Maintenance: The software is updated.

Since the life cycle consists of the previous phases and each phase has a CF, it is safe to define that the carbon footprint of a software is the sum of the CFs of the phases. This gives us the following definition:

Let us have a software s. The CF of s, cf_s, is

$$cf_s = (d_s + b_s + dr_s + m_s)/n_s + u_s \tag{1}$$

where n_s is the number of delivered software packages, d_s is the CF of development, b_s is the CF of beta testing, rd_s is the CF of delivery and re-delivery, m_s is the CF of maintenance, and u_s is the CF of usage.

The CF of each phase consists of two items: 1) material CFs, and 2) energy usage CFs. A material CF depends on the production, delivery, usage, and disposal of a product. An energy usage CF depends on how the required energy is produced.

A material CF can be calculated if its life cycle is known. An exact analysis of various material CFs is beyond the scope of this paper. We use estimates from various carbon footprint calculators when necessary.

An energy CF can be calculated directly but usually it is easier to count the energy required for a task and multiply it with an *energy carbon footprint factor* F. It shows how much carbon dioxide is emitted from generating one kWh energy. The energy carbon footprint factor is calculated from the life cycle of a power plant. Its value is always over zero regardless of the energy source.

A detailed analysis of each life cycle phase is required to estimate the CF. We will give a brief summary of the steps and show how the CFs can be calculated.

In the analysis we use a few predefined constants. Changing these walues will strongly affect the resulting CFs. Due to this the individual CF values are less important than how they relate to each other.

The constants we use are as follows:

Energy carbon footprint: $F = 0.25$kg/kWh. This value is based on a Finnish CF calculator (`http://www.hs.fi/viesti/hiilijalanjalkitesti`, in Finnish) and is a relatively clean energy source.

Paper carbon footprint: $cf_p = 0.001$kg/sheet.
Binder carbon footprint: $cf_b = 2$kg/binder. The paper and binder CFs are based on various carbon footprint calculators on the Internet, for instance `http://www.stopglobalwarming.org/carboncalculator.asp`.
Printer power consumption: $p_p = 400$W. This is a combined estimate of printer standby, ready, and printing power consumption.
Computer power consumption: $p_c = 400$W.
Relay node power consumption: $p_n = 10$W. The computer and relay node power consumptions are based on modern computer technology.
Office power consumption: $p_o = 600$W.
Meeting room power consumption: $p_m = 1000$W. Room power requirements are rough estimates of light and heating power consumptions.

3 Development

Development usually follows a well-defined process. The process has phases, best practices, and people. A typical development process consist of one or more cycles, each of which have the following major tasks: requirements gathering and analysis, design, implementation and unit testing, integration, and verification and validation. The techniques used in each of the tasks vary considerably between process models but in most cases the phases are present. We can calculate a CF for each of the phases:

The CF of development, d_s of software s, is

$$d_s = dg_s + dd_s + diu_s + di_s + dv_s \tag{2}$$

where dg_s is the CF of requirements gathering and analysis, dd_s is the CF of design, diu_s is the CF of implementation and unit testing, di_s is the CF of integration, and dv_s is the CF of verification and validation.

A detailed analysis of the development CF depends on the used process model and is beyond the scope of this paper. For example, a simple requirements gathering process might go as follows:

1. A requirements team requests a brainstorm meeting. Result: a call and an agenda.
2. The meeting is in a conference room. Result: a list of raw requirements.
3. The raw requirements are analysed. Result: analysed new requirements.
4. A negotiation meeting is required for the new requirements. Result: a call and an agenda.
5. The negotiation meeting is held. Result: an updated list of new requirements.
6. The new requirements are stored to the project requirements pool. Result: an updated list of requirements.

Most outputs require physical resources (mostly paper) although sometimes digital output may be enough.

The first step is the brainstorm meeting call. It requires an office. We get

$$CF_{1w} = (p_c + p_o) * t_{1w} * F$$

where p_c is the required power of a computer, p_o is the required power of office lights and heat, t_{1w} is the required working time at the office, and F is the carbon footprint factor.

The result of the first step is a call and an agenda. Let us assume that both the call and the agenda are delivered in one paper in via the office mail system. We get

$$CF_{1r} = n_{1p} * cf_p + (p_p * t_{1pr} + n_{1p} * p_p * t_{1pp}) * F$$

where p_p is the required power of the printer, t_{1pr} is the time to get the printer ready for printing, n_{1p} is the number of printed papers, cf_p is the CF of a paper sheet, t_{1pp} is the time to print one sheet of paper, and F is the carbon footprint factor.

The second step is the brainstorm meeting. It requires a conference room and a computer for each participant. We get

$$CF_{2w} = (n_{2m} * p_c + p_m) * t_{2w} * F$$

where n_{2m} is the number of participants (members), p_c, p_m and F are as before, and t_{2w} is the length of the brainstorm meeting.

The result of the brainstorm meeting is a list of raw requirements. Let's assume that the list is stored during the meeting. Thus, its CF is included in the meeting computer usage. We get

$$CF = 0.$$

The next step is to analyse the collected raw requirements. We assume that analysis is done in offices. The work requires a computer and an office. We get

$$CF_{3w} = (p_c + p_o) * t_{3w} * F$$

where p_c, p_o and F are as before, and t_{3w} is the time required to analyse the gathered raw requirements.

The requirements analysis does not produce anything extra. We can estimate that its carbon footprint is zero. We get.

$$CF_{3r} = 0.$$

The next step is to call a negotiation meeting. It has a similar structure and analysis than the brainstorm meeting call. We get

$$CF_{4w} = (p_c + p_o) * t_{4w} * F$$

with similar symbols than in CF_{1w}.

The result of the negotiation call is similar to the result of the brainstorm call. We get

$$CF_{4r} = cf_p * n_{4p} + (p_p * t_{4pr} + n_{4p} * p_p * t_{4pp}) * F$$

with similar symbols than in CF_{1r}.

The negotiation meeting is similar to the brainstorm meeting. We get

$$CF_{5w} = (n_{5m} * p_c + p_m) * t_{5w} * F$$

with similar symbols than in CF_{2w}.

The result of the negotiation meeting is a paper binder that includes new agreed requirements. We get

$$CF_{5r} = cf_b + n_{5p} * cf_p + p_p * (t_{5pr} + n_{5p} * t_{5pp}) * F$$

where cf_b is the CF of a binder, n_{5p} is the number of printed sheets,t_{5pp} is the time to print one sheet, and cf_p, p_p, and F are as before.

Finally, the new requirements are stored to the project requirements pool. The step requires an office and is similar to the brainstorm request meeting. We get

$$CF_{6w} = (p_c + p_o) * t_{6w} * F.$$

The result of the last step is an updated list of requirements. Let us assume that the requirements are uploaded to a database server. We get

$$CF_{6r} = (p_s + p_r) * t_{6r} * F$$

where p_s is the power consumption of the sending computer, p_r is the power consumption of the receiving computer, t_{6r} is the transfer time, and F is the carbon footprint factor.

Let us assume that a meeting takes two hours with 10 members. The result is a binder with 300 pages of paper. Meeting calls take one hour each. Raw requirements analysis takes 20 hours. Each meeting call session requires 30 sheets of paper et cetera. (We have a total 20 variables to consider, plus the constants in the previous chapter. Most values do not add much to the total picture.) We get

$$\begin{aligned} CF =& CF_1 + CF_2 + CF_3 + CF_4 + CF_5 + CF_6 \\ =& (0.28 + 2.50 + 5.0 + 0.28 + 4.82 + 0.25)\text{kg} \approx 13\text{kg}. \end{aligned}$$

Thus, three rounds of requirements gathering gives $CF \approx 39$kg. The result does not include any travelling. Travelling would dominate the CF if included.

Sometimes a detailed CF estimation is not possible or would take too much time or other resources to execute. A simple way to estimate a CF is to calculate how much each person works in a project and count individual CFs.

> Let us have a project P with n workers. The estimated size of the developed software is S KLOC (thousands lines of code). The workers have an average production efficiency p KLOC/pm (person-month), and an average CF/t cf kg/pm. The CF of the development phase is then
>
> $$d_s = n * (S/np) * cf = S * cf/p. \quad (3)$$

The average CF/t depends on the workers' CFs in the project. An easy estimate for this is to calculate the CF of an average working day and then multiply the result with the average number of working days in a month.

A month-based CF estimation is not as good as a process-based estimation. Days are not similar and it may be difficult to estimate averages over a long period of time. It may also lead to a situation where workers' CFs are compared to their productivity. It is, however, a useful tool when used properly.

For example, a greenish project worker's average work day and its CF could go as follows:

- Car to work: 160g/km*15km = 2.4kg.
- Meetings 2 hours average: 2h*(0.400kW+1kW)*0,25kg/kWh = 0.7kg (meeting room, computer).
- Office 6 hours: 6h*(0.400kW+0.600kW)*0.25kg/kWh = 1.5kg (office, computer).
- 20 papers printed: 20g = 0.020kg (paper, printer)
- Car to home: 2.4kg.

This gives a CF/day = 7.0kg/day. With a regular 22 work-day month we get CF/pm = 154kg/pm. Let us assume that the average working efficiency is 0.3KLOC/pm. A 10KLOC software would then have the following development CF:

$$d_s = 10\text{KLOC} * 154\text{kg/pm}/0.300\text{KLOC/pm} = 5100\text{kg}.$$

4 Beta Testing

In modern beta testing, the development company gives a prototype of the developed software for beta testers. The beta testers will use the software and report found anomalies to the company. Usually beta testers download the software from a server and report found anomalies via a web form.

The beta testing phase consist of the following steps:

1. a development team offers a beta software for downloading,
2. a set of beta testers download the software,
3. the beta testers use the software,
4. the beta testers report found anomalies,
5. the development team analyses listed anomalies, and
6. the development team corrects anomalies that they find severe enough.

The development team may release a new beta version of the software in which case the above list has a cycle from step 6 to step 1.

Again, we can calculate the CF from the previous phases:

The CF of beta testing, b_s of software s, is

$$b_s = n_{bs} * (bo_s + k_{bs} * (bd_s + bu_s + br_s) + ba_s + bc_s) \qquad (4)$$

where n_{bs} is the number of beta testing cycles, bo_s is the CF of loading software to the server, k_{bs} is the number of beta testers, bd_s is the average CF of downloading the software to a beta tester, bu_s is the average CF of beta tester using the software, br_s is the average CF of anomaly reports, ba_s is the CF of anomaly analysis, and bc_s is the CF of correcting chosen anomalies.

The first step is to offer a beta software for downloading. This phase requires at least two computers (one to send and one to receive) and a network connection between them. The CF produced is

$$bo_s = (p_s + p_r) * t_{bo} * F$$

where p_s is the power of the sending computer, p_r is the power of the receiving computer, t_{bo} is the time required to send the software internally, and F is the energy carbon footprint factor.

The second step is the actual downloading phase. A similar analysis as above can be used with two exceptions: 1) the network outside is probably slower than the LAN of the previous example, and 2) if the download is done via Internet, the CFs of all routing computers should be counted as well. We have a modified formula

$$bd_s = (p_s + p_r + n_{bd} * p_n) * t_{bd} * F$$

where p_s and p_r are the power consumptions of sending and receiving computer, p_n is the average power consumption of a relaying node, n_{bd} is the number of relay nodes, t_{bd} is the time to download the software, and F is the carbon footprint factor.

The next step is interesting since it covers the actual use of the software. We will get a deeper analysis of the usage CF in Section 6. Here we use a simplification where we only consider the used electricity of the executing computer.

Let us assume that the average length of the beta testing period is n_{bu} days. For this time, each beta tester uses an average k_{bu} computers for average t_{bu} hours each day for testing and storing found anomaly reports (bug reports). The computers have an average p_c power consumption. With these values we get

$$bu_s = n_{bu} * k_{bu} * p_c * t_{bu} * F.$$

Finally, the final step for beta testers is to report found anomalies. This step is similar to the first and second step. While the anomaly report transfer if fast, only seconds at most, the beta tester may need extra time to prepare the report for uploading. Let p_s be the power consumption of the sending computer, p_r be the power consumption of the receiving computer, p_n be the power requirement of a relay node, n_{br} the number of relay nodes, t_{br1} preparation time, t_{br2} sending time, and F the carbon footprint factor. Then with a similar analysis than in the second step we get

$$br_s = (p_s * t_{br1} + (p_s + p_r + n_{br} * p_r) * t_{br2}) * F.$$

The steps 5 and 6 take place in the development environment and hence do not directly affect beta testing. We can use similar formulas than in the development phase to calculate them.

The analysis above gives us an estimate of the CF of the total beta testing phase excluding fault analysis and correcting.

For example, let us assume that we have a hundred beta testers in a two-cycle beta testing program. Each cycle takes six months (180 days). Internal download takes ten minutes, external download an hour. The average number of relay nodes is 10. Each beta tester has computer that he uses an average two hours a day for testing the software. Time to prepare an anomaly report is one hour. Time to send it is two minutes. We get

$$\begin{aligned} b_s =& 2 * (0.03\text{kg} + 100 * (0.23\text{kg} + 36\text{kg} + 0.11\text{kg})) \\ =& 7300\text{kg}. \end{aligned}$$

As we can see from the analysis, the beta testing phase counts surprisingly lot in the CF calculation. A hundred beta testers is not much with modern software. The number can be several times larger, although in large beta testing usually the time to test is not as long as in our example.

Even a large CF in beta testing is acceptable if the phase reduces the number of faults efficiently from the final software. We can count the CF/fault ratio to show how effective the beta testing is. In order to save carbon dioxide emissions, the CF/fault ratio from beta testing has to be smaller than the CF/fault ratio from development. Otherwise it would be more environment friendly to do better development with less beta testing.

5 Delivery and Re-delivery

In traditional delivery and re-delivery, the final products are packed, shipped to importers, stored, shipped to stores, and sold to customers. In software it includes burning the software to a CD or DVD. However, this is a somewhat old-fashioned way to deliver software. Currently more and more software is directly available for downloading from a server.

The traditional way to deliver software has a CF that consist of the following items: packing materials, stock CF, delivery to importers, importer stock CF, delivery to re-sellers, re-seller stock, and delivery to customers. While currently dominant, the traditional delivery method will eventually give room for a more efficient network-based delivery. Hence we will not further analyse the CF of traditional delivery and re-delivery.

A modern delivery process consists of the following steps:

1. software is uploaded to a server,
2. re-sellers download software from the server to their servers,
3. software is sold in web-stores (local and re-sellers'), and
4. customers download software to their computers.

All the listed steps are similar to the steps in beta testing. Thus, we get the following formula.

The CF of modern delivery and re-delivery, d_s of software s, is

$$dr_s = do_s + n_{dr} * dc_s + k_{dr} * (ds_s + dd_s) \tag{5}$$

where do_s is the CF of downloading the software to our server, n_{dr} is the number of re-selling sites, dc_s is the CF of downloading the software to the re-selling sites, k_{dr} is the number of purchases, ds_s is the CF of selecting the software from a web store, and dd_s is the CF of downloading the software to the client.

For example, let us assume that we deliver our software to 500 re-sellers. We expect the total number of sold software packages be 10,000. The average time in a net store purchasing our software is 10 minutes. Our software requires 100 secs to deliver internally and 1000secs to deliver externally.

With the values above we get the following:

1. Upload to a server: $do_s = (p_s + p_r) * t_{do} * F = 0.01\text{kg}$.
2. Download to a re-seller:$dc_s = (p_s + p_r + n_{dc} * p_n) * t_{dc} * F = 0.06\text{kg}$.
3. Selling software:$ds_s = (p_s + p_r) * t_{ds} * F = 0.03\text{kg}$.
4. Downloading to a customer:$dd_s = (p_s + p_r + n_{dd} * p_n) * t_{dd} * F = 0.23\text{kg}$.

This gives us the following result:

$$\begin{aligned} dr_s =& 0.01\text{kg} + 500 * 0.006\text{kg} + \\ & 10,000 * (0.003\text{kg} + 0.23\text{kg}) \\ =& 2600\text{kg}. \end{aligned}$$

The result is not bad considering that we just sold 10,000 copies of our software.

6 Usage

The longest period in a software life cycle is its usage time. At that time the software is executed in a set of computers. As an abstract entity the software does not directly create a CF. It, however, requires resource which in turn create a CF.

The most intuitive resource for a software is a processor cycle. Each cycle requires some amount of energy in the computer where it is executed. We can calculate its CF and include it in the total CF. Thus we get the first definition

The CF of usage, u_s of software s, is

$$u_s = (\sum_i c_i * E_i) * F \tag{6}$$

where c_i is the CPU cycle i of the software usage, $E0_i$ is an energy factor that tells how much energy the cycle takes, and F is an energy carbon footprint factor.

We simplify the usage analysis by assuming that the power consumption of all CPUs is equal. While this is not entirely true, it is a good approximation. It saves us from deep analysis about CPU scheduling and installing the software to a new computer.

While the equation above is correct it is not very useful. It is practically impossible to estimate the total number of CPU cycles in a software life cycle. An easier approach is to count CPU time for the software and estimate how much energy it requires. Thus

$$u_s = (\sum_j t_j) * p * F \tag{7}$$

where t_i is the ith chunk of CPU time in seconds, p is the power requirement of the CPU, and F is the energy carbon footprint factor.

In many cases counting the used CPU time directly is perhaps too difficult. It can be used with simple applications whose life cycle and CPU usage are easy to estimate. We need easier and more accurate estimates for normal software CFs.

A good way to estimate software resource usage is via software service requests. If we can list typical service requests and count how much CPU the software needs to fulfil them we can estimate the total CF of the software use.

Software performance analysis with a suitable queue model is a good tool for service request estimation . A working performance analysis model will estimate how much CPU time each type of service request will need on average, how much they will wait on queues, and how much disk, network or other external services they will require. Once we have estimates for each request type, we can estimate how often each request type will occur and hence count the total. We get

$$u_s = (\sum_{k=1}^{n} T_k * f_k) * p * F \tag{8}$$

where n is the number of request types, T_k is the estimated CPU time for request type k, f_k is the estimated number of type k requests in the software life cycle, p is the CPU power consumption, and F is the energy carbon footprint factor.

Unfortunately software performance analysis is often a too complicated tool for larger systems. A simpler but less accurate tool is to measure software complexity. The more complex a software is, the more it will require CPU cycles to fulfil requests, and the larger CF it will have.

So far we have analysed mostly CPU usage. A more accurate model includes memory, disk, network, printer, display, and other peripheral equipment usage. With this analysis we get a fairly accurate estimate of the software's resource requirements. However, all these estimates are inputs to the software: something that the software needs in order to function. In a realistic CF estimation also the outputs of a software should be included in the CF.

A regular output of a software is a report. If the report is printed to a display, it can be included in the peripheral analysis. If it is printed to a printer, also its paper usage should be included. If it is sent to several people, the resource requirements to process the output should be included from all receivers. This can expand the required CF considerably. For example, consider the CF of a spam-sending software.

When inputs, outputs, and execution are included in the CF analysis, the CF analysis suddenly becomes a usability issue. A good software does not waste resources: neither from the system nor from its users. A bad software steals time and increases its CF. In the future software greenness will be a software quality factor with usability elements.

Even a large CF can be justified if it is for a good cause. A software can use resources as long as it is for a useful software system and is built as efficiently as possible. In fact, consumed CPU cycles are almost always useful. Idle cycles, on the other hand, are never useful.

Any idle cycle of a CPU is basically a wasted resource. It uses energy and creates a CF but does not give anything back. A typical home desktop or even an IT server can be 90% idle [1]. That implies a lot of wasted cycles and is like flying a jet with 10% of occupied seats. Fortunately the waste cycle problem has already been acknowledged and with proper virtualization it is possible to reduce the number of wasted cycles.

As an example of a software CF we take a compiler. It is a simple example because it uses only two types of resources: CPU cycles and disks. Let us assume that a typical compilation (a service request) requires 300 seconds CPU time and 50 seconds disk access time. The input of the software is a set of source files. The output of the software is a set of compiled files. The CPU requires 60W and the disk 20W. Thus we get

$$CF = (p_{cpu} * t_{cpu} + p_{dsk} * t_{dsk}) * F$$

where p_{cpu} is the power consumption of the CPU, p_{dsk} is the power requirement of the disk, t_{cpu} is the CPU time required for the compilation, t_{dsk} is the time required for disk access and F is the energy carbon footprint factor. If the compiler is used 10 years for 240 days/year and 50 compilations each day, it would have 120,000 compilations in its life cycle.

This gives us

$$\begin{aligned} u_s =& 120,000 * CF = 120,000 * \\ & (0.060\text{kW} * 300/3600\text{h} + \\ & 0.020\text{kW} * 50/3600\text{h}) * 0.25\text{kg/kWh} \\ =& 160\text{kg}. \end{aligned}$$

In that time the computer has the following CF (assuming 10h/day and 240 days): $CF = 0.400kW * 10 * 240 * 8\text{h} * 0.25\text{kg/kWh} = 2400\text{kg}$. About 6,6% of the CF of the computer belongs to the compiler.

7 Conclusion

In this paper we showed how we can estimate the carbon footprint of a software and gave example estimates. Let us assume that the examples are of a compiler software. We get the following values:

Development	5100kg (10 KLOC code)
Beta testing	7300kg (100 testers)
Delivery	2600kg (10,000 copies)
Usage	160kg (per installed copy)

The estimated values are small and the actual CF usage is probably higher especially in development. Yet the small analysis already showed that cutting beta testing will easily cut emissions.

It matters how we develop software, how we deliver it, and how our customers use it. With proper analysis we can ensure our both authorities and our customers that we develop environment-friendly software.

References

1. Lamb, J.: The Greening of IT - How Companies Can Make a Difference for the Environment, 1st edn. IBM Press (April 2009)
2. Murugesan, S.: Harnessing green it: Principles and practices. IT Professional 10(1), 24–33 (2008)

Board Interlocks in High Technology Ventures: The Relation to Growth, Financing, and Internationalization

Juhana Peltonen and Mikko Rönkkö

BIT Business Innovation Technology, Aalto University,
P.O. Box 15500, FI-00076, Aalto, Finland
{juhana.peltonen,mikko.ronkko}@tkk.fi

Abstract. The significance of boards of directors is often considered minor in unlisted companies due to the low degree of agency problems. Research on the topic has also been limited due to data access issues. However, networks are essential to the success of internationalizing new ventures and interlocked boards of directors have the potential to act as important information and resource conduits. To improve understanding on these dynamics, we performed an exploratory cluster analysis on board interlocks of Finnish software companies using board data we obtained from the Finnish Trade Registry and a custom survey. Our results indicate that companies that have international revenue are often interlocked with each other. In addition, companies without international revenue plan to internationalize if they have an interlock to a company that has international revenue. Our results are mirrored against management theory and future research is outlined.

Keywords: boards of directors, board interlocks, internationalization, venture capital, small and medium companies.

1 Introduction

Research on boards of directors is mostly tied to corporate governance issues in large corporations. Analyzing the link between board composition and firm performance has been one of the most popular topics of management research. This stream of research typically applies agency theory [1]. For smaller firms, agency problems are less significant, as ownership and management are often tied closely. However, a small group of individuals has a greater impact on managerial issues, suggesting that governance issues may be more pronounced in smaller firms cf., [2]. This has motivated interest in studying the role of outside directors in the SME context cf., [3-5], which are generally considered essential to monitoring executive behavior in larger enterprises and to good corporate governance.

A similar vein, interest in smaller firms is however not observable in board interlock studies. In a review of research on SME boards, [6] listed only one paper concerning boards and emphasizing networks. [7] found that board members play an important role in the management of networking activities. The board members were

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 163–168, 2010.

motivated by reputation, professional standards, legal responsibilities and ownership. A later review of board roles in SMEs by [8] lists one other paper [9]. This study measured board interlocks as a part validating new a board's networking strategy construct, which they later positively link to ROA.

Given the limited but consistent evidence on boards of directors being able to aid SMEs in networking activities, further hypotheses can be drawn on their implications on internationalization. According to the stage theory of internationalization [10], firms' internationalization is limited by knowledge of foreign cultures and markets, which increases gradually as a company gains experience operating in a particular country setting. This enables the company to gradually increase their presence in that country. Facilitating the transfer of critical cultural and market knowledge, for example from another company operating in given foreign market, would therefore help companies to expand internationally. Thus, it may be hypothesized, that an interlocked directorate may in some cases be beneficial to internationalization. Similar arguments can also be derived from other internationalization theories, such as international new ventures [11].

In addition to the more active information transfer mechanisms, interlocked directorates may cause organizations to resemble each other through a more passive institutionalization process [12]. Interlocked boards can spread organizational practices [13] and strategies [14], which may promote internationalization in two ways. First, building international operations may require companies to reorganize themselves to gain internationally competitive marketing and product development capabilities. Second, internationalization is clearly a strategic choice along with any other significant expansion decisions. Therefore, especially in an environment with lots of growth-oriented startups, internationalization may to some extent be considered an outcome of organizational mimicking and normative pressures.

Given that this area of inquiry is relatively under-researched, we set forth to perform an exploratory cluster analysis on interlocked company dyads. Relying on board interlock data from the Finnish Trade Registry and a custom survey on the Finnish software industry, we examine what kinds of common patterns exist in the data. In particular, we investigate whether international activities and growth aspirations are related to interlocks with internationalized firms. In the following sections we discuss our sample and data collection, elaborate on the analysis method, and describe the contents of the clusters identified. We then discuss the implications and need for future research in light of different management research streams.

2 Sample and Data Collection

The exploratory analysis was performed using survey data from the Finnish software industry as the primary data source. The survey is part of an on-going effort to track developments of the Finnish software industry. The data for this paper came from the survey's 2009 run, which is explained in detail in [15]. In total, there were 584 full answers and several hundreds of partial responses. The financial data of the survey corresponds to the financial year of 2008. Information on different types of external finance corresponds to the situation in the summer of 2009.

The data on board members were obtained from Asiakastieto Ltd., which aggregates and sells information from the otherwise publicly accessible Finnish Trade Register. The board data was obtained as a listing that contained a person's name, date of birth, board position (or CEO), company code, and the duration of the position. The board data were entered into an SQL database, from which the annual snapshot data of board interlocks obtained for December 31, 2008.

The data set for this study comes from the intersection of firms that both answered the primary survey in 2009 and which were interlocked through their boards. The Finnish software industry is characterized by a large number of small and medium-sized firms and a small VC market [15]. Therefore, the board networks are very sparse altogether. As a result, we obtained data on 64 directed interlocks. Due to the symmetry of the network, this implies 32 unique undirected interlocks.

3 Analysis

Since our aim was to classify the different board interlocks, we used a cluster analysis as our data analysis method. Probably the best summary of the challenges in using this method in management research is given by Ketchen and Shook [16] and more generally by Hair and his colleagues [17]. Generally, cluster analysis relies on calculating a similarity measure between each case in the data and then grouping the cases according to this measure.

Due to the exploratory nature of the study, we only relied on hierarchical clustering. Since we did not have any preconceptions of the data, we used the most generic and commonly used average linkage clustering. Choosing the similarity measure was more complex. Due to the skewness of some of the variables, we rejected the Euclidean measure and choose one of the directional measures. Since the data contained both binary and continuous variables, we considered correlation similarity measure most appropriate. Moreover, it has the advantage of being relatively straightforward to interpret. Prior to clustering all variables were standardized. To our surprise, no substantial outliers were found through examination of the dendrogram and we decided to run the final analysis with the full data. All analyses are carried out with Intercooled Stata 10.1. In this short paper we decided to use only an explorative method, so the results should be considered tentative at this stage. Moreover, we did not do any detailed robustness checks.

4 Results

The mean values of the company-specific measures are presented in Table 1 and grouped by the cluster IDs. Highlighting has been added to visualize the relative differences between table cells.

Rapidly expanding international ventures (Cluster 1). The first cluster is formed by mostly internationalized and rapidly growing young companies that are interlocked with each other. Most notably, the average growth rate is higher than for any other cluster[1], and there is a modest amount of VC, and no angel or public sector finance.

[1] Medians 23% and 22% are also higher than for the rest.

Table 1. Results

Cluster	Side of dyad	1	2	3,4	5	6
Firm age	A	6,6	14,9	11,5	9,3	12,0
	B	6,8	14,5	7,8	11,6	10,3
Revenue	A	3 649 133 €	1 812 107 €	3 685 808 €	1 121 302 €	908 910 €
	B	3 301 873 €	1 758 551 €	814 232 €	1 178 792 €	1 088 162 €
Annual revenue growth	A	62 %	14 %	37 %	48 %	5 %
	B	58 %	14 %	3 %	26 %	32 %
Has international revenue	A	100 %	94 %	88 %	0 %	50 %
	B	86 %	88 %	75 %	100 %	0 %
Plans to internationalize	A	0 %	0 %	0 %	71 %	50 %
	B	7 %	0 %	25 %	0 %	100 %
Has private sector VC finance	A	36 %	12 %	63 %	29 %	0 %
	B	29 %	12 %	50 %	14 %	25 %
Has public sector VC finance	A	0 %	0 %	13 %	0 %	0 %
	B	0 %	0 %	100 %	0 %	0 %
Has angel finance	A	0 %	53 %	38 %	14 %	25 %
	B	0 %	53 %	88 %	14 %	25 %
Number of interlocks		14	17	16	7	8

In light of the metrics applied, this group of companies can be considered the most successful ones in the sample.

*Early-stage international ventures (Cluster 2).*This cluster has many similarities with the first cluster, except that the companies are smaller, they have a high frequency of angel investment, and growth has not yet taken off. It may be that this group of companies very much represents the group of companies in cluster 1, but at an earlier stage. If so, this would indicate that the role of angel finance is still required of the companies to reach a place in cluster 1. The formation of this cluster may be explained by for instance, angel networks. Striking is the absence of public sector VCs, which on average are active in this stage.

Private and public sector VC dyads (Clusters 3,4). These clusters contain a mix of new ventures at different stages that are interlocked. The companies have clear differences in their sizes. The smaller counterparts are without exception backed by public sector VCs and almost without exception have angel finance. The large counterparts often have private sector VC. Both sides of the dyad often very often have international revenue, and the larger counterparts grow very fast. Unlike the two previous groups of companies, the "glue" that forms these interlocks most probably comes from private sector VCs due to their high and most even distribution, and the fact that public sector VC is not related to any other cluster (restricting evidence that they relate to interlocks in isolation).

International and non-international dyads (Cluster 5,6). Clusters 5 and 6 are very near to each other, and the network dyads are formed by two different types of companies. The other side of the dyad comprises of firms that have rapid growth and typically have plans to internationalize, but do not yet have any international revenue. Roughly a quarter of the companies have VC finance, of which none comes from the public sector. The companies on the other side of the dyad often have international revenue, but also have slightly lower growth. Basically none of them have VC.

5 Discussion and Conclusions

Firstly, it should be noted that the cluster analysis performed here should be considered initial, and its purpose was to uncover some of the general patterns present in the data to guide further inquiry. This said, we discuss the potential implications to various research streams. All in all, this study makes a contribution to board of directors research in the SME context.

The *"Rapidly expanding international ventures"* cluster by its very existence would indicate that successful companies are interlocked with other successful companies. This counters arguments, that boards in SMEs are merely formalities, and that only VCs force them to become more active and professional. One possible explanation could be that interlocked boards function better in a value-adding service and resource dependency roles [18]. Interlocked boards may also provide firms legitimacy by spreading institutionalized practices from other parts of the industry [cf. 23], which may lead to positive performance implications. A corollary would be, that the financial and non-financial contributions of VCs and angel investors [cf. 23] can alternatively be obtained somehow by having a well-linked board.

The *International and non-international dyads (Cluster 5,6)* provides anecdotal evidence that the will to internationalize would spread over board interlocks. The two remaining groups of firms provide some but less interesting findings. The most interesting contribution of the *"Early-stage international ventures"* cluster relates to the role of angel investors in networking, as we consider it the most likely explanation for the existence of this cluster. [21] reports that few angels are high-profile and active with their investments, whereas most are extremely passive and unprofessional. The presence of this cluster would more likely attribute to the former category, as the firms in the cluster have an above average internationalization rate [15], which may be a signal of non-financial value added and more professional selection practices. "*Private and public sector VC dyads"* is a very expected result.

This study leaves many open questions mostly due to the difficulty in establishing causal relationships. The most fundamental issue relates to self-selection. Improving this analysis by utilizing responses from previous surveys can be conducted to some extent. In addition, by including indirect board interlocks, the number of observations can be increased. Regression analysis should be applied to control for the size of the company, as larger companies tend to be more international and interlocked. Determining the causal relationships between generic constructs in more detail would prove the generalizability of these results to other regulatory and cultural contexts.

References

[1] Jensen, M.C., Meckling, W.H.: Theory of the firm: Managerial behavior, agency costs and ownership structure. Journal of financial economics 3, 305–360 (1976)

[2] Daily, C.M., Dalton, D.R.: The relationship between governance structure and corporate performance in entrepreneurial firms. Journal of Business Venturing 7, 375–386 (1992)

[3] Fiegener, M.K.: Determinants of board participation in the strategic decisions of small corporations. Entrepreneurship Theory and Practice 29, 627–650 (2005)

[4] Gabrielsson, J., Huse, M.: Outside directors in SME boards: a call for theoretical reflections. Corporate board: role, duties & composition 1, 28–37 (2005)
[5] Gabrielsson, J., Huse, M.: The venture capitalist and the board of directors in SMEs: roles and processes. Venture Capital-An international journal of entrepreneurial finance 4, 125–146 (2002)
[6] Huse, M.: Boards of directors in SMEs: A review and research agenda. Entrepreneurship & Regional Development 12, 271–290 (2000)
[7] Borch, O.J., Huse, M.: Informal Strategic Networks and the Board of Directors. Entrepreneurship: Theory and Practice 18 (1993)
[8] Van den Heuvel, J., Van Gils, A., Voordeckers, W., Agoralaan-building, D., Gebouw, D.: Board roles in small and medium-sized family businesses: Performance and importance. Corporate Governance: An International Review 14, 467–485 (2006)
[9] George, G., Wood Jr., D.R., Khan, R.: Networking strategy of boards: implications for small and medium-sized enterprises. Entrepreneurship and Regional Development 13, 269–285 (2001)
[10] Johanson, J., Vahlne, J.E.: The Internationalization Process of the Firm: A Model of Knowledge Development and Increasing Foreign Market Commitments. Journal of International Business Studies 8, 23–32 (1977)
[11] Oviatt, B.M., McDougall, P.P.: Toward a Theory of International New Ventures. Journal of International Business Studies 25 (1994)
[12] DiMaggio, P.J., Powell, W.W.: The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields. American Sociological Review 48, 147–160 (1983)
[13] Davis, G.F.: Agents without Principles? The Spread of the Poison Pill through the Intercorporate Network. Administrative Science Quarterly 36, 583–613 (1991)
[14] Westphal, J.D., Seidel, M.L., Stewart, K.J.: Second-Order Imitation: Uncovering Latent Effects of Board Network Ties. Administrative Science Quarterly 46, 717–747 (2001)
[15] Rönkkö, M., Mutanen, O., Koivisto, N., Ylitalo, J., Peltonen, J., Touru, A., Hyrynsalmi, S., Poikonen, P., Junna, O., Ali-Yrkkö, J., Valtakoski, A., Huang, Y., Kantola, J.: National Software Industry Survey 2008. Helsinki University of Technology, Espoo (2009)
[16] Ketchen, D.J., Shook, C.L.: The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique. Strategic Management Journal 17, 441–458 (1996)
[17] Hair, J.F., Anderson, R., Tatham, R.L., Black, W.C.: Multivariate Data Analysis. Prentice Hall, Upper Saddle River (2006)
[18] Zahra, S.A., Pearce, J.A.: Boards of directors and corporate financial performance: A review and integrative model. Journal of Management 15, 291 (1989)
[19] Palmer, D.A., Jennings, P.D., Zhou, X.: Late adoption of the multidivisional form by large US corporations: Institutional, political and economic accounts. Administrative Science Quarterly 38, 100–131 (1993)
[20] Politis, D.: Business angels and value added: what do we know and where do we go? Venture Capital 10, 127–147 (2008)
[21] Shane, S.: Fool's Gold?: The Truth Behind Angel Investing in America. Oxford University Press, USA (2008)

Entrepreneurial Challenges in a Software Industry

Nina Koivisto and Mikko Rönkkö

Aalto University
{nina.koivisto,mikko.ronkko}@tkk.fi

Abstract. In this paper, we present two studies investigating the entrepreneurial challenges faced by companies in the Finnish software industry. The first study is based on telephone interviews conducted during the National Software Industry Survey in 2008. The second study was done in collaboration Growth Forum initiative in 2008. We collected data during three large seminars and numerous working group meetings. Qualitative analysis of the data from both studies suggest that growth is a major challenge for the firms and that managers' growth motivation and managerial capabilities are the most important requirements for successful growth.

Keywords: Entrepreneurship, Entrepreneurial Challenge, Software Industry.

1 Introduction

A large share of the previous research on firm growth has focused on positive determinants of this phenomenon but challenges of growth have received less attention [see e.g., 1]. In this paper we study how entrepreneurs at various stages of firm development view the challenges of growth. Although the research is based on the accounts of entrepreneurs rather than direct empirical observation of the challenges that the firms face, we believe that the study makes a two-fold contribution. First, the views presented by the entrepreneurs can act as a proxy measure for the actual challenges. Second, some challenges can be socially constructed [2] perceptions and perceptions and attitudes are known to affect the entrepreneurial process consequently firm growth [3].

The empirical context of the present paper is the Finnish software industry. Some have argued that this market faces unique challenges on both national and industry level [4]. Particularly, the small home market and the further issues created thereof can affects how firms can growth. Three major challenges can be identified: First, the small market means small venture capital market. Second, the small and relatively unknown home market can cause legitimacy problems when expanding abroad. Third, firms are forced to diversify either horizontally by broadening the offering or expand internationally in earlier phase, since the number of potential customers is limited in the home markets. Hence the present study sheds light on the challenges faced by companies in these particular circumstances.

2 Literature Review

Entrepreneurial firms tend to develop not continuously, but through stages, and each stage is characterized by different issues and challenges [3, 5-7]. Although there are

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 169–174, 2010.

numerous different growth stage models, these all share similarities. After the initial opportunity identification and planning stage, the firm founders face three problems of constructing resources, gaining legitimacy, and acquiring new customers. In essence, the challenge in the commercialization stage can on a high level considered from moving from idea and incorporated business to a viable firm with initial product and customer. After the initial product is ready, the early growth stage sees the challenge of productization and forming a real organization to replace the entrepreneurial team. After the early growth, the later growth means usually diversification either geographically or by introducing new products. In this stage the firm starts to grow beyond the small business stage and the challenges are more often related to organizational "growth pains" and international expansion, if the firm chooses this track. We will now present each of these stages and the challenges in more detail.

The first stage in the stage models is concerned with the founding of the company, launching the initial products and services and achieving the first sales. This stage can be called conception and development [6] or formation [8]. We can summarize this stage as turning a venture or idea into a viable business entity. In our categorization, this stage encompasses also what some models present as a separate commercialization stage [8]. Typical challenges in this stage are related to resources, legitimacy, and markets. The firm suffers from liability of newness [9], which is manifested through economic inefficiencies due to lacking routines and lack of credibility in the markets because of lack of customer references. The single biggest market related challenge is achieving the first customer reference [10]. Particularly, if the company is selling a software product that is business critical, being unable to demonstrate that the software is actually used somewhere successfully is a huge obstacle for acquiring new customers. The lack of human, social, financial, physical, technology, and organizational resources can mean that the company is hard-pressed in competing in the markets since resources are considered to be the source of competitiveness.

The focus in the early growth stage is primarily on learning how to make the product work well and on how to produce it beyond the prototype or first version developed during the previous stage. During this stage the business establishes itself as a sustainable business with a commercially feasible product and/or marketing approach [8]. For a software company, the challenges in the initial stage are moving from the first reference customer to a broader customer base [10], productization of the initial prototypes and standardization of the offering [11], and establishing a marketing function [12]. Generally, the challenges revolve around turning the initial often customer-specific solution to a more productized version to enable scalability of the business.

After the early growth stage, the firm enters into later growth [8]. A key challenge at this point is that the firm grows beyond the size that can be controlled through leadership of the initial founders. The firm hence needs a management system with layers of control, structures, and routines. One notable challenge at this stage is that the founders might not understand when it is time to step aside and hire management that already has experience from running larger organizations [13].

Although some firms start as born-global and go international during the initial product launch stage or early growth, often internationalization constitutes a separate growth stage. However, a large number of firms follow more traditional models where they first construct a resource based in the home markets and then expand internationally or follow the existing customers from the home markets when expanding internationally. The

challenges created by international expansions are numerous [14] and constitute a research stream of their own.

3 Research Methods and Empirical Study Design

The empirical data for this paper come from two studies. The first study was a telephone survey of a random sample of firms that had responded of the National Software Industry Survey [15]. Since the purpose was to study the entrepreneurial challenges we chose an open-ended approach. Open questions tend to produce a lot of non-response in mail surveys and thus we chose telephone interviews as the data collection method. A sample of 71 companies was drawn from a list of companies that had responded to a mail survey under this project. The telephone interviews included three questions that asked the respondents to elaborate on the challenges, strengths, and weaknesses of the company. These data were analyzed by coding the responses on a spreadsheet. We used open coding to identify different challenges and then grouped these codes into a hierarchy. The initial codings were performed by two researchers to eliminate the potential bias caused by a single person analyzing the responses. After coding of the data, we linked it with demographic information provided by the main mail survey.

The second study was conducted during the Growth Forum '08 project. Growth Forum is a national project by the Finnish Software Entrepreneurs' Association and Microsoft Oy (the Finnish subsidiary of Microsoft corporation) to foster the growth of Finnish software SMEs. The work of this study was done in three large seminars and numerous working group meetings to which the authors of this paper took part actively. Most of the data come from the five working groups running in parallel and engaging approximately 70 participants, of whom most are senior executives of software firms. The purpose of the working groups was to identify the challenges faced by the Finnish software industry and to propose solutions that would help both the individual firms as well as the entire industry to overcome these challenges. In total the working group had over 20 meetings that lasted from hour and a half up to three hours.

Additionally, we separately interviewed persons that we considered as key people in the working groups. These people were a Technology Director from public listed company, a Managing Director from a privately held company, a Deputy Managing Director from a privately held company, a Chairman of a privately held company, a Partner Venture capitalist, a Research manager from a research institute, a Managing director from privately held company, and a Strategic director from privately held company. The mean length of these interviews was 70 minutes.

The data from this second study were analyzed in the following way: All field notes and interview transcripts were first read to familiarize with the data. Each discrete opinion on the determinants of growth of the industry was recorded in a spreadsheet. In the second stage, all identified opinions were grouped into categories.

4 Results

Majority of the Finnish software firms start as small and also remain small [15]. In other words they never grow. One of the possible reasons for this, as explained by

several informants of the second study, is the small domestic market. However, in the global business environment with the Internet and its possibilities, this is most probably not the sole explanation. One possible reason for low growth is the low level of growth aspirations of the founders of the firms or their unwillingness to grow the business considering the increased level of risk associated. Similar findings of low levels of willingness to grow have been reported also in other studies [3]. In all, the working groups identified managers' growth motivation to be one of the top three most important requirements for successful growth of software companies. Also managerial capabilities for growth management were identified as key factors - this factor was rated one of the most critical ones having effect on successful growth.

To further understand the challenges faced in different lifecycle stages, we analyzed the coded telephone responses by cross-tabulating them by firm age class in Table 1. Although this is an imperfect proxy for development stage, it was the best data that we had available. The biggest challenge for firms is growth, particularly for firms that are in the conception and development stage or are already in the stability stage (see in Table 1). On the positive side, the fact that firms emphasize growth as their challenge means that this is something that they consider important, but on the negative side it means that firms are finding it difficult to grow. One of the key results of the analysis of the interview data from the second study that is not reflected in the telephone interview data is that firm founders and CEOs must understand that their role will change as the firm grows, and this is often difficult to accept. Managers in the Finnish software industry have often a high level of technical competence but lack managerial experience, and this becomes evident as the firm grows. Particularly when a firm expands abroad, hiring a professional management team to bring both experience and personal networks into the firm should be seriously considered.

Availability of skilled personnel was identified as a more contemporary challenge and was also linked to the growth challenge. If talented people are not available and thus cannot be recruited, a firm cannot grow. It is not only about finding the right people, but also about having the people with the right competences in the right positions. Since particularly small businesses have limited resources, the most talented people should work in the most critical parts of the business.

Table 1. Challenges by firm age class

	Age				
Challenge	<2	2-5	6-10	>10	Total
Customer acquisition	2	2	2	0	6
Personnel	1	3	4	4	12
Internationalization	1	3	1	1	6
Growth	6	4	3	6	19
Marketing	1	1	2	2	6
Sales and marketing	1	1	2	4	8
Recruiting	1	3	4	1	9
Resources	1	2	1	3	7
Total	12	16	15	17	60

Sales and marketing was mentioned as the third most important challenge on the firm-level, especially for firms in later development stages. Marketing and sales models usually involve decisions about market segmentation, the target market, and presenting and making the products available for customers in a way that attracts them. Researchers in both marketing and software engineering have argued that managerial competence in marketing is currently at a rather premature stage in many software companies where technology development has traditionally been the central area of interest among managers [16]. Since the sample of this study is dominated by small and to some degree relatively young companies, it is not surprising that this challenge of mainly smaller firms is prevalent also in our sample.

Lack of resources was raised as the fourth most important challenges in the study, particularly for firms that are in the conception and development or commercialization stage. This closely parallels what the participants in the second study referred to as the challenge of small firm size. Other challenges that are worth mentioning are internationalization and customer acquisition, of which the latter faced more by the smaller companies. Internationalization challenges go hand-in-hand with growth challenges, and were, in some cases, seen as particularly a challenge in international marketing. Building international sales channels is not only managerially demanding, but also takes a huge amount of resources. Since this upfront investment must be paid before any sales revenue is realized, firms often seek venture capital or other external funding in this stage.

Previous studies have argued that the Finnish software industry is predominantly technology oriented [16]. In sum, challenges investigated through telephone interviews, seminars, and working groups provide partly support for this idea. Fortunately, challenges and importance of marketing and growth are well realized by the companies and several firms seem to actively tackle these problems. Indicating that the industry can be evolving away from technology orientation.

5 Discussion

The results of the telephone interview and working group study seem to be roughly in line with the current theorizing on firm growth and the challenges in the growth process. Since the biggest challenges in software industry for SME firms seems to be growth, and especially the low level of growth aspirations of the founders of the firms to grow the business, this is an issue that should be studied more in the future.

References

1. Davidsson, P., Achtenhagen, L., Naldi, L.: Research on small firm growth: A review (2005)
2. Berger, P.L., Luckmann, T.: The Social Construction of Reality. Blackwell Publishing Ltd., USA (2002)
3. Wiklund, J., Patzelt, H., Shepherd, D.: Building an integrative model of small business growth. Small Business Economics 32, 351–374 (2009)

4. Kontio, J., Rönkkö, M., Mutanen, O., Ahokas, M., Junna, O., Ali-Yrkkö, J., Touru, A., Ruotsalainen, S., Heikkonen, M., Martikainen, O., Mickelsson, M., Rahkonen, A., Puttonen, V., Niemi, P., Salminen, J., Eloranta, E., Fredrikson, N., Koivunen, A., Tikka, T., Maisala, T.: Kasvufoorumi 2008. Ohjelmistoyrittäjät ry, Helsinki (2008)
5. Hanks, S.H., Watson, C.J., Jansen, E., Chandler, G.N.: Tightening the Life-Cycle Construct: A Taxonomic Study of Growth Stage Configurations in High-Technology Organizations. Entrepreneurship: Theory & Practice 18, 5–30 (1993)
6. Kazanjian, R.K.: Relation of Dominant Problems to Stages of Growth in Technology-Based New Ventures. The Academy of Management Journal 31, 257–279 (1988)
7. Gilbert, B.A., McDougall, P.P., Audretsch, D.B.: New Venture Growth: A Review and Extension. Journal of Management 32, 926–950 (2006)
8. Dodge, R.H., Robbins, J.E.: An empirical investigation of the organizational life cycle model for small...
9. Freeman, J., Carroll, G.R., Hannan, M.T.: The liability of newness: Age dependence in organizational death rates. American Sociological Review, 692–710 (1983)
10. Ruokolainen, J., Igel, B.: The factors of making the first successful customer reference to leverage the business of start-up software company—multiple case study in Thai software industry. Technovation 24, 673–681 (2004)
11. Äijö, T., Kuivalainen, O., Saarenketo, S., Lindqvist, J., Hanninen, H.: Internationalization Handbook for Software Business. Technopolis Ventures Oy (2005)
12. Alajoutsijärvi, K., Mannermaa, K., Tikkanen, H.: Customer relationships and the small software firm a framework for understanding challenges faced in marketing. Information & Management 37, 153–159 (2000)
13. Mutanen, O., Rönkkö, M.: Growth Challenges of Small Finnish Software Firms – Comparing Theory and Practice (2008) (presented at)
14. Touru, A., Rönkkö, M.: Internationalization Challenges of Small Finnish Software Firms – Comparing Theory and Practice (2008) (presented at)
15. Rönkkö, M., Ylitalo, J., Peltonen, J., Koivisto, N., Mutanen, O., Autere, J., Valtakoski, A., Pentikäinen, P.: National Software Industry Survey 2009. Helsinki University of Technology, Espoo (2009)
16. Ruokonen, M., Hätönen, J., Lindqvist, J., Jantunen, S., Marjakoski, E., Hurmelinna-Laukkanen, P.: Global network management - ideas and tools for ICT firms to succeed in international network management. Lappeenranta University of Technology, Lappeenranta (2008)

Looking at Internationalization of a Software Firm through the Lens of Network Theory

Marko Forsell

Oulu Business School, University of Oulu
marko.forsell@oulu.fi

Abstract. It is argued here that the network theory can better explain success and give a more concrete view of the internationalization than stage models of internationalization. The resulting business network in a foreign country has more effect than the chosen internationalization mode. Main conclusions are that the actors involved in the everyday actions in a foreign office and their interconnectedness with the rest of the company are important. The participation of a multi-divisional company's board of directors does not have consistent effect on the success of internationalization.

Keywords: Internationalization, network theory, software, software firm.

1 Introduction

The business world is globalizing. The most important drivers behind globalization are declining trade and investment barriers, and technological change. The Finnish ICT industry is very dependent on foreign markets due the smallness of Finnish home markets and the global competition in the ICT industry. Currently, one prominent way of looking at the business environment is to envision it as a network [1, 2]. A network is a structure where a number of nodes (business units) are related to each other by specific threads (relationships) [2]. These business relationships usually involve a number of managers who coordinate the activities of the different firms, and these relationships develop through social exchange process.

When a firm becomes part of a network, it is its resulting network position that determines its operation options [2]. Håkansson and Ford [3] identify three network paradoxes when firms become part of a business network: (a) the network position creates both opportunities and limitations, (b) a firm can influence another firm through relationship, but at the same time it is influenced by this relationship, and (c) a firm can control a network, but it is also controlled by the network. Gadde et al. [4] build their view of strategizing in industrial networks on these three network paradoxes and suggest that to handle these paradoxes a firm needs a strategic orientation applied in three dimensions of the industrial network: *resources*, *activities*, and *actors* (see also [5]). The *resource dimension* looks beyond the resources of an individual firm. The truly important resources are the business relationships that a firm has with its customers, suppliers, and other interest groups. The second dimension is the *activity dimension*. Interaction is one of the basic activities of a firm;

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 175–180, 2010.

a firm exchanges products and services, and organizes operations within and between organizations. Activities beyond the boundaries of a firm create a chain of activities between many firms. Building the activities between companies can create efficiency that cannot be pursued by one company alone or by markets. The third dimension is the *actor dimension*. The actors in the network posses the resources and make the activities [4].

The ICT industry operates globally and a growing firm must consider internationalization at some point. The main reasons for internationalization can be summed up as something a firm seeks. A firm can be: (a) a natural resource seeker, (b) a market seeker, (c) an efficiency seeker, and (d) a strategic asset or capability seeker [6]. Usually, the internationalization mode varies from wholly-owned affiliate to exporting and having a distributor in a foreign country. Leonidou and Katsikeas [7] reviewed 11 internationalization models and concluded that from the operation point of view all models were stage models and the developmental pattern was towards greater involvement in foreign markets. One of the most cited internationalization models is Johansson and Vahlne's 1977 model [8]. However, Johansson and Vahlne observed that business networks play important roles in internationalization and later updated their model [9]. In this revisited Uppsala model, 'internationalization is seen as the outcome of firm actions to strengthen network positions by what is traditionally referred to as improving or protecting their position in the market' [9, p.13]. Coviello and Munro [10] studied four small software companies and their internationalization. One of their main conclusions is that even the internationalization of small software firms can be better understood by combining incremental and network perspectives of internationalization.

2 Research Question and Method

Our research question is: "How are the network theory's forces of *actors*, *resources*, and *activities* related in the internationalization of a software firm?" To determine this we utilize the case study approach. Easton [11] states that one kind of case research method with a realist epistemological orientation is to use the existing theory to examine how the causal powers already identified act contingently in different situations. The reason for selecting the case study method is that the author has worked for more than six years at the case company in question and has firsthand knowledge and understanding of it and the decisions it has made. This should provide a deeper understanding of the context of the decisions [12].

The case company is a large, internationally operating multi-divisional company. During the period from 2006 to 2007, it used three different modes for internationalization. The significant point about these internationalization steps from the research point of view is that: (a) they are all aimed at the same goal, i.e. growth, (b) they were all made by the same company roughly at the same time, that it, the context of the decisions was the same, (c) the decisions were made by the same management, and (d) they all were different types.

These points help us to delve into these different modes and gain a deeper understanding of the internationalization process. It is seldom that the same company at the same time with the same goals and the same management make these kinds of

decisions. We look at each internationalization mode through the important dimensions of *resources*, *activities*, and *actors* of the business network theory. But first we introduce the case company.

3 Case Company

The case company (FIN) was established in 2003 by combining two existing firms that were themselves established in 1989 and 1998. At the end of 2008, the case firm employed more than 700 people with a turnover of about 45 M€. The time period this study looks into is from the beginning of 2006 to the end of 2008. During that time FIN had two divisions: industry and ICT. This study focuses on the ICT division. During the time period, FIN's ICT division took three major steps to internationalize: (a) establish a wholly owned firm in the Czech Republic, (b) acquire a major holding of a software company in Sweden, and (c) set up joint venture in Romania.

The Czech Republic firm (CZE) was established in the beginning of 2006. This company is wholly owned by FIN. The managing director appointed to CZE had previous experience with the country and had lived and worked there. Additionally, he had some connections with the ICT companies there. The idea was to acquire new customers in the Czech Republic and, more widely, in Europe. Initially, however, the business model was only to have sales operations in CZE, the software services being taken care of in Finland. Only after gaining a better foothold of local customers would CZE start hiring its own software professionals. During the first year and a half, it became apparent that it was more difficult to handle sales in the Czech Republic and Europe than was anticipated. The CZE was transformed into a sales office for FIN's industrial division.

FIN's approach to Swedish markets was different. FIN scouted for a suitable software firm for acquisition. After finding a suitable firm, a majority of its stocks was acquired, and in this way FIN's Swedish operations (SWE) were established in 2007. Initially, there were about 30 software professionals working for SWE. SWE already operated in the same industry as FIN's ICT division, but with different customers. SWE's goal was to increase its market share locally. During the first years, the growth of the company was substantial: more than 30% in turnover and personnel. However, SWE's operating profit dropped almost to 0% at the end of 2007. It opened up two offices in Sweden but was later forced to shut one down. Also, it had difficulties opening up new markets for its ICT sector. But at the end of 2008 it was clearly profitable and had a good track record in increasing turnover, and 62 software professionals were employed there.

The Romanian firm (ROM) was established in October 2007. It was a joint venture between FIN and a Romanian firm, with FIN the majority owner. The local firm also operated in the ICT industry, but with different customer focus. The Romanian partner firm had very good relationships with the local university, ICT firms, and local customers. The idea was to set up ROM as a cost-effective production site for software services that were mainly required by FIN's local customers. During the first operating year (2007), a Finnish project manager set up software processes that were required from FIN's customers. The local partner recruited local software professionals. After the first year, a second project manager was sent to ROM with

the main goal of setting up the processes required to coordinate work between ROM and FIN's offices. The Romanian office employed 50 software professionals after two years of operation and operated efficiently, creating value for FIN's total offering in the ICT division.

4 Results

4.1 Resources

CZE's business relationships relied heavily on the Finnish managing director's existing network, which he had created several years earlier while working in the country. CZE's operations followed FIN's ICT division's operations fully. It was realized during the first 18 months that the business relationships that were earlier created did not producing enough sales for further commitment to the Czech markets. The customers in the Czech Republic did not need the software services that were offered.

SWE had an existing customer base and contacts with its local interest groups. SWE continued to hire people, and it acquired new markets. SWE also opened up new offices to serve bigger markets. All this was done without much involvement from FIN because SWE prospered independently. The result was, however, that relationships with FIN did not evolve, and there was not much information exchange. SWE continued to operate as a separate firm. SWE's systems were not integrated to FIN's, for example, and they only reported their financial results through 'excel-sheets'.

ROM's local partners had contacts to universities, and they were well connected for the purposes of getting employees. FIN supplied customer contacts and also sent a Finnish project manager to set up software processes so as to ensure they would meet its main customers' expectations. ROM was ramped up from scratch and the focus was always on being a cost-efficient service provider for FIN's customers. All systems were created with efficiency in mind, and only the most cost-effective operations were adapted from FIN's operations. If services were more efficiently produced locally, then the local producers were used, e.g. for computer purchases local connections were used. Only the most important data and financial connections were created without such a mindset.

4.2 Activities

CZE's activities were tightly linked with FIN's ICT division. CZE's activities and procedures closely followed those in FIN, and the managing director knew FIN's business customs well since he had been working for FIN for several years. CZE was supposed to be responsible for the activities of working as a sales office for the Czech markets; the other work would be done in FIN's local offices.

SWE had its own activities in all operations. This was only natural since it was already an operating firm. Not much integration between FIN and SWE was made.

ROM's software processes closely followed FIN's processes since this was what was expected by customers. However, recruiting and administrational operations were built up by local Romanian partners. This division of responsibilities worked

nicely and it fulfilled the requirements of having a cost effective site to serve FIN's customers.

4.3 Actors

CZE was established by FIN's ICT division, and FIN's board was not heavily involved with this. CZE's managing director had previous experience in working in the Czech Republic. CZE's board of director's was comprised of FIN's ICT division's managing director but none of FIN's board of directors. CZE was more or less seen as an extension of FIN's ICT division and as just another office. The existing relationships in the Czech Republic did not open up sales.

FIN did not have existing relationships in the ICT sector in Sweden. Hence, SWE was acquired. SWE nominated a new board of directors as soon as the majority of the company was bought by FIN. The new board of directors consisted of two of SWE's original owners, one member from FIN's board of director's and the managing director FIN's ICT division. The initial idea of buying SWE was proposed by FIN's ICT division, but FIN's board was also heavily involved in the operation. SWE's management was not changed and it continued doing its work as it has done before. No integration in resources or activities was made.

FIN did not have any existing connections in Romania. During the search for a firm to be acquired, a suitable Romanian firm was found. During the negotiations it became apparent that the firm was not for sale and that it had a little bit different focus than FIN anticipated. However, the people from the Romanian firm warmed to the situation and they became interested in establishing a joint venture with FIN. Ramping up ROM was driven by FIN's ICT division and there were mild involvement by FIN's board. The CZE's managing director was nominated as the managing director of the Romanian office. He nominated a Finnish project manager to set up software processes in ROM. Tight cooperation between the Finnish PM and a local Romanian partner was very fruitful, and they were able to create good, deep relationships between FIN and ROM.

5 Discussion

It is our opinion that stage models or maturity models alone cannot explain the movements made by the case company. The business network theory explains why these movements were made and, at least with hindsight, one can see that the end result is what network theory postulates However, there is no ideal internationalization mode. It would be tempting to conclude that acquisition and joint venture are correct ways to enter a new market, but this is not necessarily so. It is the interdependencies of relationships which allow new opportunities to emerge. Thus it is more important to visualize and determine the networks resulting from selected entry mode. However, we know that relationships in business networks are invisible to those who are not active in the network [1] and that information asymmetry exists about the quality of the partner's business network. Also, it was interesting to notice that the role of the board was quite different in all cases, and that the mild involvement in the Romania office did not affect its success. So, at least in this case,

we cannot conclude that it is important to have deep commitment from the top. This should put even more emphasis on the actors who use resources and conduct activities in business networks.

However, this research has limitations which should be borne in mind. First, the case company is a Finnish company and all of the described internationalization cases happened in Europe, so e.g. psychic distance does not play a large role in these relationships. Second, this study relies on the author's experience and knowledge about the case company and the described internationalization cases.

References

1. Håkansson, H., Snehota, I.: No business is an island: The network concept of business strategy. Scandinavian Journal of Management 5, 187–200 (1989)
2. Ford, D., Gadde, L.-E., Håkansson, H., Snehota, I.: Managing Business Relationships, 2nd edn. Wiley, Wiltshire (2003)
3. Håkansson, H., Ford, D.: How should companies interact in business networks. Journal of Business Research 55, 133–139 (2002)
4. Gadde, L.-E., Huemer, L., Håkansson, H.: Strategizing in industrial networks. Industrial Marketing Management 32, 357–365 (2003)
5. Håkansson, H., Snehota, I.: Analysing business relationships. In: Håkansson, H., Snehota, I. (eds.) Developing Business Relationships in Networks, pp. 24–49. Routledge, London (1995)
6. Dunning, J.H., Lundan, S.M.: Multinational Enterprises and the Global Economy, 2nd edn. Edward Elgar, Cheltenham (2008)
7. Leonidu, L.C., Katsikeas, C.S.: The export development process: An integrative review of empirical models. Journal of International Business Studies 27(3), 517–551 (1996)
8. Johansson, J., Vahlne, J.-E.: The internationalization process of the firm: A model of knowledge development and increasing foreign market commitments. Journal of International Business Studies 8(1), 23–32 (1977)
9. Johanson, J., Vahlne, J.-E.: The Uppsala internationalization process model revisited: From liability of foreignness to liability of outsidership. Journal of International Business Studies, 1–21 (2009)
10. Coviello, N., Munro, H.: Network Relationships and the Internationalisation Process of Small Software Firms. International Business Review 6(4), 361–386 (1997)
11. Easton, G.: Case Research as a Methodology for Industrial Networks: A Realist Apologia. In: Naudé, P., Turnbull, P.W. (eds.) Network Dynamics in International Marketing. Elsevier Science, Oxford (1998)
12. Gummesson, E.: Qualitative Methods in Management Research, 2nd edn. Sage Publications, Thousand Oaks (2000)

Goals of Software Vendors for Partner Ecosystems – A Practitioner´s View

Karl Michael Popp

SAP AG, Corporate Development
Dietmar-Hopp-Allee 16, 69190 Walldorf Germany
karl.michael.popp@sap.com
http://www.drkarlpopp.com

Abstract. There is literature about large software vendors building ecosystems around themselves [1,2]. This paper looks at goals they try to achieve with partner ecosystems.

Keywords: Business ecosystems, software industry, partnership models, goals of partnership models, partner programs.

1 Software Ecosystem Overview

In a networked economy, companies formulate an ecosystem strategy as a part of their corporate strategy [3, 4]. In their ecosystem strategy, they include goals to be achieved leveraging their ecosystem. An economic **ecosystem** is a set of companies that exchange products or services to serve a common goal or to achieve higher levels of individual goals. More importantly, economic principles and strategies apply [4-6]. Often companies in an ecosystem are aligned along value chains or form around a value chain that serves the same set of customers, e.g. the customers of a large software vendor like Microsoft or SAP. While other definitions exist [1, 2, 4], in this paper a **software ecosystem** is an economic ecosystem that forms around one specific software vendor.

For the purpose of simplification, the focus is on the companies that interact with the software vendor or the software vendor´s customers in the following ways: they **sell or license products or provide services to the software vendor´s customers**. These products or services might be related to or integrated with the software vendor´s products or services, they **sell or license the software vendor´s products**, e.g. as **value added resellers** (VAR), they **sell services to the software vendor**, to the customers or to the software vendor´s partners, they **license or subscribe to the software vendor´s products** for internal use or for inclusion in their own products, they **license software to the software vendor** (suppliers), and last but not least, companies can sell intellectual property to the software vendor. That means they are potential **candidates for acquisition** by the software vendor.

By definition, the software vendor is also part of the software ecosystem (Fig. 1). Usually software vendors apply different strategies and tactics for different parts of

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 181–186, 2010.

the software ecosystem. These strategies and tactics must be aligned with the business model of the software vendor. Fig. 1 shows the types of ecosystems addressed separately by a software vendor. The **customer ecosystem** contains existing and potential customers of the software vendor. The **partner ecosystem** [10] contains software partners, which are other software vendors and system integrators, which provide implementation services for the software vendor´s solutions. Another ecosystem is the supplier ecosystem, which is not in the focus of this paper.

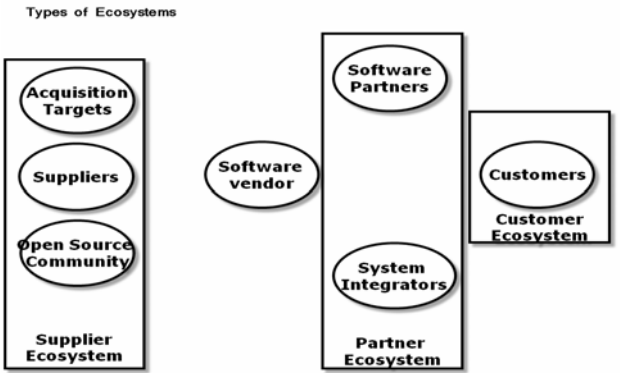

Fig. 1. Players in the software ecosystem and types of ecosystems

2 Partner Ecosystem Goals

In a simplified view of reality, a software vendor´s ecosystem activities are targeted at the following goals [2]: Financial, customer related, product related, network effect related and market related goals.

2.1 Financial Goals

In contrast to non-profit ecosystems [11], software ecosystems have financial goals, which usually target at monetizing on the ecosystem. Monetizing on the ecosystem means **creating or increasing revenue streams** by leveraging the ecosystem. Revenue streams can come from partner program and community fees and increased product revenue due to positive sales effects from the ecosystem. Fig. 2 shows some examples of revenue streams to be created or increased in an ecosystem.

Partner program fees are the entry ticket fees for partners into partner programs. The tricky thing is: if the fees are higher than the anticipated value of the benefits, partners will not join the partner program. So software vendors should carefully define and handle fees associated with the partner programs to make sure partners are not deterred from joining the partner programs.

Product revenue from the ecosystem can come from reselling products or from services fees or from license fees for using your products or from revenue shares. Software partners can also help increase license revenue from customers by promoting the software vendor´s solution in a new market or industry.

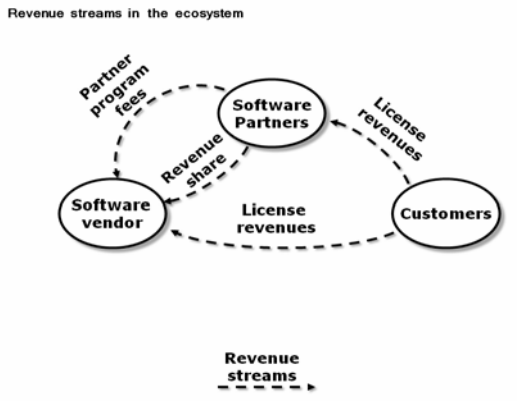

Fig. 2. Examples of revenue streams in the ecosystems

How can a software vendor **save cost** by leveraging the ecosystem? Cost savings mainly result from outsourcing the software vendor´s activities to the ecosystem, from educating the ecosystem and from leveraging the ecosystem as multiplier for product knowledge. Outsourcing to the community means that the community can take over tasks of the software vendor. The following tasks are usually (at least partially) outsourced to the community: presales activities, sales activities, product distribution, post-sales activities like support, answering product-related questions and providing additional, complementing products.

2.2 Customer Related Goals

An ecosystem is a place for an existing customer or a prospect to build trust in the software vendor´s solutions. Trust is also a key ingredient for customer retention. Trust is built between the customers and a software vendor by the sheer number of participants in the customer ecosystem, the number of partners offering solutions and services for the software vendor´s solutions, and information from existing customers shared in the ecosystem.

Attract new customers: Customer communities can start a viral effect to attract new customers. This happens when existing customers spread the news about their use of solutions or when the lock-in of existing customers extends to the customer´s ecosystem of suppliers and customers. The reason is simple: to lower the integration cost between suppliers, the customer and the customer´s customer it makes sense to use identical technology and applications. If the customer has enough market power, he might even force its suppliers and customers to use identical technology and applications.

Increase the stickiness of solutions: With a high number of partner solutions based on or integrated with the software vendor´s solutions, the stickiness of the software vendor´s solution increases. This means that the **switching cost** for the customer to switch to a competitor´s solution increases. But beware: this is only true, if the

interface between the partner solution and the software vendor´s solution is not sufficiently standardized.

2.3 Product Related Goals

Software vendors try to leverage their software ecosystem to reach the following product goals.

Strengthen market presence: Software vendors can choose diversified partnership strategies for regional and local markets. They do so by leveraging the software partners market expertise, market access as well as access to new and existing customer accounts. The partnership strategies differ by presence and strength of the software vendor and its partners in the specific markets.

Strengthen software vendor´s offerings: The portfolio of solutions of a software vendor always has whitespaces and adjacent, complementing solutions, which are not offered by the software vendor itself. Complementing offerings from partners can be positioned in these whitespaces or adjacent areas. This leads to a more complete solution offering for the customers, combining the software vendor's products with partner products. This might lead to a customer perception of a complete offering for the customer´s business problem.

Innovate and co-innovate solutions: A sound innovation strategy of a software vendor contains a mix of internal and external innovation. External innovation happens outside of the company walls of the software vendor, mainly by partners, but also by customers and system integrators [12]. Co-innovation means that two companies are working together to innovate [13]. A partner can build a solution based on a software vendor´s platform or solutions. In this case, the engineering work and cost resides with the partner. Software vendors usually provide some sort of certification to prove the interoperability of the solutions.

2.4 Network Effect Related Goals

Ecosystems can be used to fight competitors. The reasoning behind that is based on network effects [6, 7, 8]. The **network effect** says that the value of a product increases with the number of customers of that product. Network effects can be direct or indirect. A **direct network effect** comes from compatibility between products. Within a network of customers, all customers using the same software have low integration cost. An **indirect network effect** comes from the assumption that widespread adoption of a product also leads to a large number of adjacent solutions and partners.

More customers and more partners in your ecosystem make it harder for a competitor to compete against you. For customers, a large ecosystem of customers and partners provides a variety of solutions with low integration cost and promises low interoperability costs since integration cost for two customers using the same software is minimal.

Maximize ecosystem gravity: Maximizing the gravity of the ecosystem means increasing the number of participants in the ecosystem. The hope is that the more gravity the ecosystem has, the more it attracts new members for the ecosystem. Let´s have a look at two types of ecosystems to see what this means for the software vendor.

Customer ecosystem gravity is measured by the number of customers in the ecosystem. A large customer ecosystem has a lot of advantages, from generating license and maintenance revenue to providing references and pilot customers. A large customer ecosystem can lead to a positive network effect, such that partners and customers of an existing customer might become a customer as well.

Partner ecosystem gravity is measured by the number of partners in an ecosystem. On one hand, large software vendors try to engage as many partners as possible; on the other hand they try to get the focus of the partner on their products, not on the competitor´s products.

To ensure maximum participation, it is vital to keep the entry barriers to the ecosystem low. Microsoft´s barriers to become an entry level partner (called Registered Partner) are very low (just enter your data on a website) and there are no fees attached with this partnership. In addition, each partner gets a cheap package of licenses for Microsoft products. These activities serve additional goals: make every partner a customer and maximize retention in the ecosystem.

Maximize retention of participants in the ecosystem: To maximize leverage of the ecosystem, it might make sense to attract as many participants as possible and to retain them. Retention of participants in the ecosystem is based on two important factors: incentives and lock-in. Incentives are e.g. marketing opportunities and customer access for partners as well as an attractive offering for customers in an ecosystem.

Software vendors create partner lock-in by deepening the integration of partner solutions with the solutions of the software vendor. At the same time, the software vendor creates lock-in for customers that use the software vendor´s solutions integrated with partner solutions.

2.5 Market Related Goals

Software vendors try to influence markets with their ecosystem strategy. Standardizing markets, extending the market reach and creating new markets are good examples for a software vendor´s goals relating to markets.

Standardize markets: One goal is important for software vendors: Homogenize the offering and the demand in a market. Markets can become more homogeneous by establishing standards. Standardization can be driven by vendors, partners and customers alike. A common belief is that a more homogeneous market creates a bigger revenue opportunity for software vendors. But standardization also lowers the entry barriers for competitors. Let´s assume in a standardized market the competition and the market size grows. It makes sense to enter this market, if your market share is equal or larger in a standardized market.

Extend market reach: A software vendor can extend its market reach by leveraging partner´s skills, knowledge and products as well as partner´s market access and market coverage. A partners´ access to local, regional or vertical markets is a tempting business opportunity for a software vendor. The partners may position the software vendor´s products in that market, they can refer customers from that market to the software vendor or they can act as a value added reseller to sell the software vendor´s products into that market.

Create new markets and communities: A software vendor can start a new ecosystem and thus a new market. Let us have a look at an example for creating a customer ecosystem. In its effort to sell more software to business users, SAP has created the **business process expert community** (BPX). Positive effects are: SAP can learn from the community and can apply targeted marketing mechanisms at the community.

References

1. Messerschmitt, D.G., Szyperski, C.: Software Ecosystem: Understanding an Indispensable Technology and Industry. MIT Press, Cambridge (2003)
2. Meyer, R.: Partnering with SAP. Books on demand, Norderstedt (2008)
3. Zhang, R., Zhang, Y.: New strategic mode: strategic ecology management. In: IEEE, Proceedings of 2005 International Conference on Services Systems and Services Management. Proceedings of ICSSSM 2005, pp. 47–51. IEEE, Los Alamitos (2005)
4. Iansiti, M., Levien, R.: Keystones and Dominators: Framing operating and technology strategy in a Business Ecosystem. Harvard Business School, Boston (2004)
5. Buxmann, P., Diefenbach, H., Hess, T.: Die Software-Industrie: ökonomische Prinzipien, Strategien, Perspektiven. Springer, Heidelberg (2008)
6. Shapiro, C., Varian, H.R.: Information rules: a strategic guide to the network economy. Harvard Business School Press, Boston (1998)
7. Varian, H.R., Farrell, C., Shapiro, C.: The Economics of Information Technology: An Introduction (The Raffaele Mattoili Lecture). Cambridge University Press, Cambridge (2004)
8. Buxmann, P.: Network Effects on Standard Software Markets: A Simulation Model to Examine Pricing Strategies. Publications of Darmstadt Technical University, Institute for Business Studies, BWL, p. 36497 (2001)
9. Jansen, S., Brinkkemper, S., Finkelstein, A.: Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. In: Proceedings of the First Workshop on Software Ecosystems, pp. 34–48 (2009)
10. Popp, K.M., Meyer, R.: Profit from software ecosystems: Professional Edition, Business Models, Ecosystems and Partnerships in the Software Industry. Books on demand, Norderstedt (2010)
11. Stoll, J., Edwards, W.K., Mynatt, E.D.: Interorganizational coordination and awareness in a nonprofit ecosystem. In: Proceedings of the 2010 ACM conference on Computer supported cooperative work, pp. 51–60. ACM Press, New York (2010)
12. Nambisan, S., Sawhney, M.: The global brain: your roadmap for innovating faster and smarter in a networked world, pp. 49–54. Pearson, Upper Saddle River (2008)
13. Davenport, T.H., Leibold, M., Voelpel, S.: Strategic management in the innovation economy: strategy approaches and tools for dynamic innovation capabilities. Publicis, Erlangen (2006)

Anticipating Success of a Business-Critical Software Project: A Comparative Case Study of Waterfall and Agile Approaches

Marko Ikonen and Pekka Abrahamsson

University of Helsinki, Finland
{marko.ikonen,pekka.abrahamsson}@cs.helsinki.fi

Abstract. A business decision to abort projects with little or no chance at succeeding should be made as early as possible. The research on success of software engineering projects is fragmented and unorganized, which makes anticipating outcomes difficult and possibly error prone. This short paper offers a preliminary insight into success factors related to project outcomes that can be found at the midpoint of the development projects. We conducted a comparative case study where eight software development projects used the waterfall development method and four projects agile software development approaches as their primary development vehicle. Due to the explorative nature of the research, we conducted these in university settings. The results reveal that signs at project failure can be seen in the middle of the projects.

Keywords: anticipation, success factor, software engineering project.

1 Introduction

Standish Group's CHAOS reports have repeatedly shown dramatic problems in running software projects successfully regardless of the type of software being developed. While strong critique has been aimed at the validity on these report results [7,11], we can argue that the results are indicative enough to warrant serious attention. In the case of a software project failure, damage is caused not only to the project itself but to the products, services, and other business which are dependent on the project. Not enough emphasis has been placed on increasing the ability to anticipate the project outcome as early as possible.

Software development of the 2010s is still very much people-oriented. At a team level, people management requires leading skills, such as motivating, empowering, supporting and truly caring for people, which affects project results [3,16]. However, even a project with unmotivated teams can be successful. In other words, a single viewpoint is likely to be insufficient when trying to determine the level of success. While the skills of a project manager and the group dynamics are critical, so is the direction of a project. Technical competence without support from the organization, skills in coordination, communication, or influencing possibilities is risky. A business decision to abort projects with little or no chance of succeeding should be made as early as possible.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 187–192, 2010.

This short paper addresses this problem domain of coordination, communication and cohesion, and offers a preliminary insight into success factors related to project outcomes that can be found at the midpoint of the development projects. Research on success regarding software engineering projects is scarce and fragmented. Critical success factors (e.g. [15]) in software have, however, been identified but only *a posteriori*. The same applies to software process improvement studies (e.g. [1,5]). Our initial findings from an earlier study [10] suggest anticipating project success with high accuracy is possible. In this study, we extend our research to include modern agile software engineering projects using the Scrum method.

2 Preliminary Research Model for Anticipating Project Success

When developing a model to determine a project's success, the following two issues need to be addressed: (1) the level of subjectivity in performing the analysis and (2) which model will be selected as the backbone for the new model. As stated, the research in this area is fragmented and unorganized. Kitchenham et al. [13] illustrate this. None of the existing models fit our purposes perfectly. Several isolated relationships can be deduced from the literature between a "project success" and, for example, "actions" of the development teams. These relationships come from several sources with a wide difference in the context where they were identified. We decided to opt for a different strategy. We treat "all" existing models as equals and compile a comprehensive mega-model as our preliminary research model. To meet these needs we have built an all-inclusive questionnaire designed to include the majority of success relationships drawn from different reference disciplines. The description of the research model can be found in [9]. For the purposes of the study we present the preliminary research model in Figure 1 with the reference disciplines, theories used and their expected impact. Due to the very limited space, the references are excluded.

3 Study Setting

The study of [10] showed a possibility of predicting a software engineering project success, based on a small, qualitative sample. While understanding that the dynamics beyond traditional process models is important, we conducted a comparative experiment to find out whether anticipation is plausible for modern, Scrum-based projects. In the experiment, we used the same measurement and analysis techniques as in the study of [10], except we focused on project success only.

Our goal is to reveal the key questions (i.e. the most expressive power to reveal driving and restraining elements) regarding Scrum-based projects. Scoring for this was done by aggregating the answers of each member within each project. Then, these project-specific answers were scored by comparing them with the

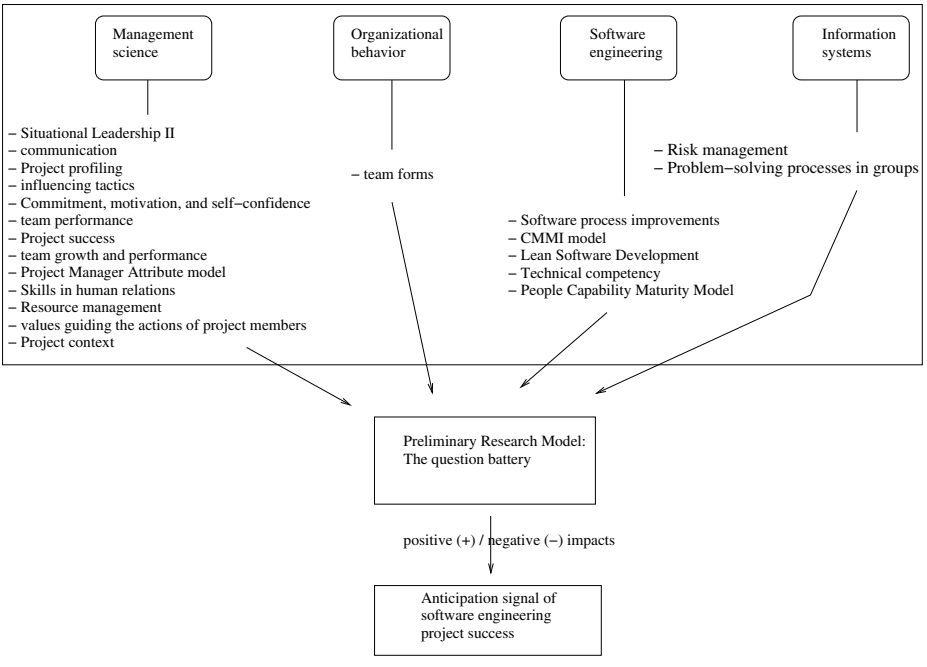

Fig. 1. The preliminary research model with the reference disciplines and theories

answers of other projects, question by question. If we perceived something to have a positive effect on the project success, it increased the project's *driving score* (DRI) by 1 to 7 points. Similarly, issues perceived as a negative effect increased the *restraining score* (RES) by 1 to 7 points. An answer can increase both driving and restraining scores.

The data gathering started in autumn 2007 and spanned eight quartiles. Totally, 61 project members were interviewed from twelve projects. In order to determine whether a particular project was successful, after a project ended, the customers of each project estimated the success in cooperation with the supervisors. Such a determination for success covers Shenhar's first and second dimensions: project efficiency and impact on customer (see [17]). Due to a possible evaluation bias, an extra evaluation, called an expert evaluation described in [10] and performed similarly, was used to compensate them.

4 Results

We explored how accurately we can anticipate the success of Scrum development projects. In this and subsequent analyses we keep continuously comparing the waterfall method results equally to enable continuous reflection on the differences. Table 1 presents the scaled scores from the interview sessions and their correlation to the success. Success, in its turn, is presented in Table 2a (see the

Table 1. The scaled scores of driving and restraining elements for the projects: the waterfall-based projects (marked with *w*) of the original study and the Scrum-based projects (marked with *s*) of our study. The correlations of these scores to the project success scores (i.e. *avg* in Table 2a) are presented.

	project												
element	w1	w2	w3	w4	w5	w6	w7	w8	s9	s10	s11	s12	**correlation**
DRIw	.53	.22	.16	.42	.03	.45	.42	.36					.87
DRIs									.63	.48	.42	.39	.13
RESw	.86	.66	.75	.92	.39	.80	.86	.86					.93
RESs									.78	.68	.61	.82	.93

Table 2. (a) The evaluations of project success after the projects made by expert evaluator and the projects' boards (including the customer and the supervisor). *w* refers to waterfall, *s* to Scrum-based project. *avg* is the average of the two success scores and is used for the correlation calculations in Table 1. *stdev* represents standard deviation. **(b)** The top 10 lists of the scored questions with the scores (scr). The codes relate to the preliminary research model (see Chapter 4 in [9]).

project	expert	board	**avg**	stdev
w1	.83	.90	**.87**	.05
w2	.63	.61	**.62**	.01
w3	.57	.95	**.76**	.27
w4	.90	.84	**.87**	.04
w5	.37	.27	**.32**	.07
w6	.83	.96	**.90**	.09
w7	.87	.82	**.85**	.04
w8	.60	.82	**.71**	.16
s9	.76	.75	**.76**	.01
s10	.70	.80	**.75**	.07
s11	.62	.76	**.69**	.10
s12	.73	.84	**.79**	.08

(a)

DRIs		*RESs*		*DRIw*		*RESw*	
code	score	code	score	code	score	code	score
CO-03	24	PP-05	26	CO-03	22	CO-03	21
CO-04	24	CM-03	14	RS-05	17	CM-02	11
PG-13	22	PI-08	14	PG-13	16	PP-05	11
PI-10	22	CO-03	12	CM-02	14	MF-06	9
CO-12	20	PG-30	12	MF-06	13	CO-04	8
CO-02	18	CO-05	10	PI-10	13	CO-21	8
PG-01a	18	RS-06	10	CO-13	11	RS-06	8
MF-17	16	IP-01	8	PG-11	11	CO-12	7
PI-08	16	PP-03	8	PG-14	11	RS-01	7
RI-04	16	CO-12	6	CO-02	10	PI-10	6

(b)

avg column). The driving element scores for Scrum projects (*DRIs*) do not correlate with project success since the correlation value is only .13. Nevertheless, the driving scores for waterfall projects (*DRIw*) correlate strongly (.87). Moreover, a strong correlation between the restraining elements for Scrum projects (*RESs*) and project success (Table 2a) exists (.93). In addition, restraining scores for waterfall projects (*RESw*) have as high correlation (.93) as *RESs*.

The average score *avg* of the two success scores in Table 2a was used in calculating the correlations (Table 1) between success and the scores conducted from the interview.

The strong value of correlation between the restraining scores and success of Scrum projects (the correlation for *RESs* in Table 1) is considered a promising signal of an accurate anticipation ability regarding project success. The questions

with their codes are presented in [9]. Table 2b shows the ten most accumulated scores for positive drivers (*DRIs*, *DRIw*) and for negative restraining elements (*RESs*, *RESw*). Each of the lists is ordered based on the number of scores they received. The *RESs* and *RESw* columns in Table 2b are the most appealing due to their strong correlation (Table 1).

5 Summary

We maintain that anticipation provides a valuable tool for business critical software projects. Success, in terms of project efficiency and impact on the customer, can be seen after a project ends. However, an economical tension requires a business decision to abort projects with little or no chance to succeed as early as possible. Without anticipation, these decisions cannot be made until the end.

We found key questions which revealed the most restraining elements in the projects evaluated in the study. For the waterfall projects, both the drivers (*DRIw*) and restrainers (*RESw*) equally explained the project success whilst only the restraining elements worked to anticipate success in Scrum projects (*RESs*). We found that the less restraining elements there are, the more successful the project will be. This is the case with both waterfall and Scrum-based projects since the correlations for both (*RESw* and *RESs*) were .93. Hence, the key questions for anticipating success in Scrum projects are found in the *RESs* column in Table 2b. The five most powerful ones are PP-05, CM-03, PI-08, CO-03, and PG-30. In waterfall projects, the most useful questions in addition to CO-03 and PP-05 are CM-02, MF-06, CO-04, CO-21, and RS-06 (see the *RESw* column in Table 2b.). These findings are very similar to the literature, as shown, for example, in [2,3,4,6,12]. Lack of communication skills, oral as well as written, of experienced engineers in business has been found to be a commonplace problem [14].

Despite the promising results presented in this paper, they cannot validate the findings comprehensively. An obvious limitation to the validity of the proposed findings is the use of students in this study as study subjects. However, it is quite well established that when one seeks to establish a trend, the use of students is quite acceptable [18]. Höst et al. [8] concluded that students are indeed relevant when considering experimentation in software engineering. We do not maintain that our findings are one-to-one with industry but rather that, given the specific circumstances, we indicate that there may be a trend explaining project success or failure when a particular set of indicators are searched for.

While the fragmented area of success in SE projects still lacks a holistic model capable to explain project success comprehensively, the results above are encouraging for pursuing the development of such a model. Once successful, this is a significant cost-saving opportunity for software businesses when applying the model in practice. The new and revised model visible in Table 2b serves as a mini-model for practitioner use already at its current state.

References

1. Abrahamsson, P.: Measuring the success of software process improvement: the dimensions. In: Proceedings of EuroSPI 2000 -conference (2000)
2. Addison, T., Vallabh, S.: Controlling software project risks: an empirical study of methods used by experienced project managers. In: SAICSIT 2002: Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, pp. 128–140. South African Institute for Computer Scientists and Information Technologists (2002)
3. Situational Leadership® II - The Article. The Ken Blanchard Companies (2001)
4. Boehm, B.: Software engineering economics. Prentice-Hall, New Jersey (1981)
5. Conradi, R., Fuggetta, A.: Improving software process improvement. IEEE Software 19(4), 92–99 (2002)
6. Curtis, B., Krasner, H., Iscoe, N.: A field study of the software design process for large systems. Communications of the ACM 31(11), 1268–1287 (1988)
7. Glass, R.L.: The Standish report: does it really describe a software crisis? Communications of the ACM 49(8), 15–16 (2006)
8. Höst, M., Regnell, B., Wohlin, C.: Using students as subjects - a comparative study of students and professionals in lead-time impact assessments. Journal of Empirical Software Engineering 5(3), 201–214 (2000)
9. Ikonen, M.: Working Toward Success Factors in Software Development Projects. Number 2009-19 in Series of Publications C. Department of Computer Science. University of Helsinki, Helsinki (2009)
10. Ikonen, M., Kurhila, J.: High-impact success factors in capstone software projects. In: SIGITE 2009: Proceedings of the 10th ACM SIGITE conference on Information technology education. ACM, New York (2009)
11. Jørgensen, M., Moløkken-Østvold, K.: How large are software cost overruns? critical comments on the standish group's chaos reports. Information and Software Technology 48(4), 297–301 (2006)
12. Keil, M., Cule, P.E., Lyytinen, K., Schmidt, R.C.: A framework for identifying software project risks. Communications of the ACM 41(11), 76–83 (1998)
13. Kitchenham, B., Budgen, D., Brereton, P., Turner, M., Charters, S., Linkman, S.: Large-scale software engineering questions – expert opinion or empirical evidence? IET Software 1(5), 161–171 (2007)
14. Norback, J., Hardin, J.: Integrating workforce communication into senior design. IEEE Transactions on Professional Communication 48(4), 413–426 (2005)
15. Reel, J.: Critical success factors in software projects. IEEE Software 16(3), 18–23 (1999)
16. Robbins, S.P.: Essentials of organizational behavior. San Diego State University, Prentice Hall (2002)
17. Shenhar, A.J., Levy, O., Dvir, D.: Mapping the dimensions of project success. Project Management Journal 28(2), 5–13 (1997)
18. Tichy, W.: Hints for reviewing empirical work in software engineering. Journal of Empirical Software Engineering 5(4), 309–312 (2000)

Monitoring Social Media: Tools, Characteristics and Implications

Mikko O.J. Laine[1,2] and Christian Frühwirth[1]

[1] Software Business Lab, Aalto University, Otaniementie 17, Espoo, Finland
[2] Scandinavian Consortium for Organizational Research, Stanford University, CERAS 531, Stanford University, Stanford, California 94305-3084, USA
{mikko.laine,christian.fruehwirth}@tkk.fi

Abstract. The proliferation of social media is opening up new possibilities for companies to increase their awareness of openly expressed consumer sentiment about themselves and their competitors. In this paper a wide variety of software tools designed for monitoring social media are reviewed. Definitions for relevant concepts are given. Several tool characteristics are evaluated including analyzable social media types and language support. The main finding is that the tools offer a wide coverage of social media, but vary significantly in other characteristics with notable room for improvements. Therefore, careful judgment should be exercised when selecting the right tool for a particular use.

Keywords: social media, monitoring, tools.

1 Introduction

Social media has transformed the broadcasting nature of conventional media into media where users themselves are creating as well as consuming the content [1, 2, 3]. People connect, socialize, share and form virtual communities through social media, all the while generating content such as text, pictures and videos that are accessible by everyone on the Internet. Tools offered by social media websites such as Flickr, Facebook, and Twitter allow for users to engage in a wide spectrum of various social interactions [2, 12]. These user interactions form active dialogues between users and often offer an intimate account into their thoughts, opinions and sentiments.

Companies traditionally engage in complex market research activities to find out what customers think about their products, brand or competitors such as customer surveys, focus group interviews or competitor analyses [4]. Now, the open nature of social media has opened up an unprecedentedly wide access to these customer sentiments. This is strengthened by the social media's trend of customers already used to voicing and communicating their opinions to others. Furthermore, much of the dialogues between social media users leave a lasting record, meaning that they can be searched, traced backwards and analyzed [3, 5]. In addition, technologies of the social media websites allow for following nearly instantaneous announcements of any updates on the contents using mainly RSS-feeds.

But how can the amassing historical dialogues and in-gushing updates of these websites be surveyed and followed to derive meaning and benefit for companies

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 193–198, 2010.

about their brand, products and services? To service exactly these needs, various software tools offered over the Internet have surfaced.

In this paper we review a wide selection of available software tools for monitoring social media. We assess several attributes and functionalities of these tools. Particular attention is paid to the languages the tools can be applied to. The language aspect will surely be of interest to companies interested in localization, operating in local markets, serving a particular language-restricted consumer segment (such as elders or developing countries) as opposed to just the general common English language of the Internet.

The review and analyses offered by this paper might, first of all, inform companies interested in monitoring social media of these tools and their strengths and weaknesses. Producers of the tools engaged in this particular type of software business may learn about some of the areas of weaknesses in these products for further product development or be offered an insight into the prevalent competition. From a scientific perspective, we offer introductory insight into these tools that might give ideas to researchers and thus spur further studies on related topics.

1.1 Defining the Relevant Concepts

As social media is a very recent phenomenon, it still lacks a formal definition [6]. While attempts at defining it have been made [1, 3], new and popular terms in general – such as social media – are often are used as buzzwords without exactly knowing what the terms actually mean. This results in interchangeable and incorrect use of the terms and in general confusion. In order to clarify the used concepts and to convey a unified understanding for the readers, we define here some of the popular terms for the context of this paper.

Media, often also referred to as the mass media, are the tools and channels to relay information and data (content) [13]. Media include newspaper and magazines, electronic analog media include radio and television, and electronic digital media include computer networks such as the Internet. In the older forms of media, content is generated centrally and broadcast to the media's consumers.

Social media can then be defined as the tools and channels to relay content mainly generated by the consumers themselves. It is a form of electronic digital media that involves the end-users in such way that the content is produced by the users for the users using highly accessible technologies [3, 5]. This change has been enabled by the constantly-developing technologies used over the Internet [11, 12].

Companies offering social media tools and services frequently go by the same name as the service itself, as is the case with, for example, Facebook and Twitter. Furthermore, websites of these companies (later social media websites) often provide the main access to these services for the consumers, while other access can be offered e.g. with specific mobile applications [4].

People engaging in social media form social networks [7, 8] and virtual communities [9, 10] amongst themselves. Social networks comprise of people and the ties between them characterized by one or more type of interdependency such as communication, friendship or exchange. These ties can be formed via social media or by other means. [7, 8] Some authors make a distinction between social networking websites (e.g. Facebook) and social media websites (e.g. YouTube) [cf. 1]. However, we feel this distinction to be fairly superficial, as these holistic services regularly exhibit characteristics and functionalities of both categories.

Finally, tools for monitoring social media are (most times) software services offered over the Internet to filter and analyze the textual content produced by and in social media. The tools find content based on user-defined keywords. The tools incorporate multiple functionalities, such as analyses of volume, source, author, keyword, region and sentiment, and then reporting these analyses conveniently and in a graphical fashion. Most often the tools are based upon reading and aggregating multiple freely accessible RSS-feeds of updates from the social media websites.

2 Social Media Monitoring Tools

The selection of tools was formed by discussing the matter with industry experts and using their suggestions, and by doing searches with Google using keywords such as "social media monitoring tool" and "social media listening tool". From these suggestions and search results, the most often-mentioned or otherwise popular tools were selected. Therefore, the selection is far from being neither objective nor exhaustive, but should represent a wide-enough range of known and popular tools to get a general understanding on what is being offered and to draw simple generalizations.

The tools were initially selected and reviewed during September 2009, and some updated in February 2010. In total, 24 tools were selected. Table 1 shows the selected social media monitoring tools. The tools listed first are commercial tools of which some allow free use trials with limited functionality. Tools starting from Google are completely free, and some of the commercial tools base their functionality on them.

Table 1. Social media monitoring tools selected for analysis

Name of tool	Company	URL
BrandsEye	Quirk eMarketing	http://www.brandseye.com/
BuzzLogic	BuzzLogic	http://www.buzzlogic.com/
BuzzMetrics	Nielsen	http://en-us.nielsen.com/
Cision Social Media	Cision	http://us.cision.com/products_services/
FindAgent	Hapax	http://www.findagent.co.uk/
Maestro Platform	TNS Cymfony	http://www.cymfony.com/Solutions/
MyReputation	ReputationDefender	http://www.reputationdefender.com/myreputation/
Online Rep.Monitor	Distilled	http://reputation.distilled.co.uk/
Radian6	Radian 6 Technol.	http://www.radian6.com/cms/home/
Scout Labs	Scout Labs	http://www.scoutlabs.com/
Sentiment Metrics	Sentiment Metrics	http://www.sentimentmetrics.com/
SM2	Alterian	http://www.alterian-social-media.com/
Trackur	Trackur	http://www.trackur.com/
truCAST suite	Visible Technolog.	http://www.visibletechnologies.com/products.html
Google Alerts	Google	http://www.google.com/alerts/
Google Blogsearch	Google	http://blogsearch.google.com/
Google Insights	Google	http://www.google.com/insights/search/
Google Reader	Google	http://google.com/reader/
Google Trends	Google	http://www.google.com/trends/
--"-- for Websites	Google	http://trends.google.com/websites/
Link Checker	Seo Pro	http://seopro.com.au/free-seo-tools/link-checker/
Twitter Search	Twitter	http://search.twitter.com/
Technorati	Technorati	http://technorati.com/

We wanted to incorporate also non-commercial tools to see what they can do compared to the commercial ones.

The assessment criteria were selected intuitively by initially going through the tools and looking at some of the functionalities they offer. As the focus of the study is on the monitoring functionalities, we did not evaluate any engagement functionalities. Engagement, however, is a powerful next step from mere monitoring. Ultimately, we decided to assess the tools regarding the following criteria: cost of the tool, languages it understands, various types of sources that are aggregated, and whether it can do geographical, historical and sentiment analyses.

3 Results and Discussion

Table 2 on the next page shows the results with the analyzed tool next to the assessment criteria. If a cost range is presented, it means that there are various pricing alternatives available with varying functionalities between this range. N/a indicates that the information was not readily available in the website or a tool trial. Extent-column lists the general feel of complexity about the tool and its features. For the review, free trial accounts were created and demo webinars attended where available.

From the results table we can see that most of the tools offer functionality to aggregate feeds from all the various types of available social media. Some of the tools, however, are limited in terms of functionality, such as Trackur, which basically only lists the found information. Then again, the more complex tools that offer a wide variety of functions often operate only specifically in a predefined set of languages. In addition to this, they are significantly more expensive.

Language support in many cases seems to mean that only the results in supported languages are shown and all other languages are simply ignored. This will cause serious considerations with companies wanting to analyze specific languages. One of the rare exceptions that do not ignore results from unsupported languages even though all of the (sentiment) analyses would not work is Alterian's SM2.

The free tools offer only one functionality or insight into one media type. For example, Google Insights and Google Trends monitor only Google's search terms. In addition, the outputted data is normalized, so any easy analysis is impossible. The free tools, however, offer a quick and easy way to do very simple monitoring.

Due to their offered functionality sets, some of the commercial tools seem to be more applicable to active monitoring and some are more suitable for analyzing historical data. This is important when a high-hype company or product that generates a lot of discussion considers these tools compared to a low-hype company or product where social media output is likely to be much more limited.

Many reviewed tools experienced technical challenges. It is thus evident that many of the tools are still in their infancy. Future is bound to bring stability and overall quality to these tools as the industry matures while social media keeps growing and becoming even more popular. We also find that thorough information about the products was not available or easily accessible on many providers' websites. This is very paradoxical considering the operational setting and could be greatly improved.

The limitations of this paper are obvious. The list of reviewed tools is by no means exhaustive. Other tools could have been picked and they might represent the population of available tools better. The assessment criteria were limited and the assessments

Table 2. Results for the assessed social media monitoring tools

Name of tool	Cost $	Languages	Forums/News/Media	Blogs	Microblogs	Geography	History	Sentiment	Extent
BrandsEye	1-350	n/a	X	X	X	X	X	X	Simple
BuzzLogic	n/a	n/a	X	X	X	n/a	n/a	X	Med/Compl
BuzzMetrics	n/a	5	X	X	X	X	X	X	Med/Compl
Cision Social Media	n/a	8	X	X	X	X	X	X	Med/Compl
FindAgent	n/a	Some	X	X	X	n/a	n/a	X	Medium
Maestro Platform	n/a	11	X	X	X	X	X	X	Med/Compl
MyReputation	10-15	All	X	X	X	-	X	-	Simple
Online Rep. Monitor	8-500	n/a	X	X	X	-	X	-	Simple
Radian6	600 up	10	X	X	X	X	For fee	X	Complex
Scout Labs	99-749	English	X	X	X	X	X	X	Medium
Sentiment Metrics	400 up	10	X	X	X	X	X	X	Medium
SM2	n/a	All	X	X	X	X	X	X	Med/Compl
Trackur	18-297	All	X	X	X	-	6 mon	Manual	Simple
truCAST suite	n/a	12	X	X	X	X	X	X	Complex
Google Alerts	free	All	X	X	X	-	-	-	Simple
Google Blogsearch	free	All	X	X	-	-	X	-	Simple
Google Insights	free	All	-	-	-	X	X	-	Simple
Google Reader	free	All	X	X	X	-	-	-	Simple
Google Trends	free	All	-	-	-	X	X	-	Simple
-- "-- for Websites	free	All	-	-	-	X	X	-	Simple
Link Checker	free	All	-	-	-	-	-	-	Simple
Twitter Search	free	All	-	-	X	-	X	X	Simple
Technorati	free	English	-	X	-	-	X	-	Simple

themselves were made at some parts using subjective interpretation. Nevertheless, we believe that this review offers insight into the nature of available solutions and allows for very basic generalization.

4 Conclusions

In this paper we reviewed a multitude of software tools available over the Internet for monitoring social media. The main finding is that the commercial tools offer good coverage of the types of analyzable social media, but they vary significantly in terms of price, language support and technical sophistication. The free tools can be used easily to analyze one specific type of functionality or media and are a viable option for simple monitoring. In conclusion, discretion should be exercised when trying to match a tool for a specific need.

This study has been a first scratch at the world of monitoring social media. Further research should concentrate on a specific area on these tools, such as usability, and solidly relate the findings to the existing theory base and thus offer a deeper theoretical perspective. In addition, beyond the mere attributes of the tools, further research should be applied to a sample problem setting to assess the tools' viability in practice. Furthermore, it should be investigated how these tools can be used in taking the next logical step: engage the monitored social media.

References

1. Kim, W., Jeong, O.-R., Lee, S.-W.: On social web sites. Information Systems 35, 215–236 (2010)
2. Thackeray, R., et al.: Enhancing promotional strategies within social marketing programs: Use of Web 2.0 social media. Health Promotion Practice 9(4), 338–343 (2008)
3. Social media, http://en.wikipedia.org/wiki/Social_media
4. Kotler, P., Keller, K.L.: Marketing management, 13th edn. Prentice Hall, Englewood Cliffs (2008)
5. What is Social Media? http://webtrends.about.com/od/web20/a/social-media.htm
6. Xiang, Z., Gretzel, U.: Role of social media in online travel information search. Tourism Management 31, 179–188 (2010)
7. Social network, http://en.wikipedia.org/wiki/Social_network
8. What is Social Networking? http://webtrends.about.com/od/socialnetworking/
9. Laine, M.O.J.: Bibliometric analysis and systematic review of management literature on virtual communities. In: Proceedings of the 9th European Academy of Management Conference (EURAM), Liverpool, UK, May 11-14 (2009)
10. Miller, K.D., Fabian, F., Lin, S.-J.: Strategies for online communities. Strategic Management Journal 30(3), 305–322 (2009)
11. What is Web 2.0, http://oreilly.com/web2/archive/what-is-web-20.html
12. Hansen, D.L., et al.: Do you know the way to SNA? A process model for analyzing and visualizing social media data (forthcoming)
13. Media, http://dictionary.reference.com/browse/media

FLOSS-Induced Changes in the Software Business: Insights from the Pioneers

Juho Lindman, Risto Rajala, and Matti Rossi

Aalto University School of Economics, Information Systems Science,
Runeberginkatu 22-24, 0010 Helsinki, Finland
firstname.lastname@hse.fi

Abstract. Companies that build their offerings with Free/Libre Open Source Software (FLOSS) communities have evoked fundamental changes in the operating environment of software firms. However, prior literature has not paid sufficient attention to how the managers of software firms perceive these changes and the impact of FLOSS activity on their business. This study investigates the perceptions of the entrepreneurs and senior managers in Finnish software firms regarding these issues. Based on narratives obtained from discussions with the managers, we group the findings into four categories that provide insight into the ongoing changes in the software industry.

1 Introduction

The Free/Libre Open Source Software (FLOSS) phenomenon has received increasing attention among IS-researchers as an important aspect of the information economy and an essential consideration for all software companies (Fitzgerald 2006). FLOSS collaboration has delivered high quality mainstream applications such as the Linux operating system and the Apache Web Server (Raymond, 1999; Mockus et al., 2002; Fitzgerald, 2006). In sum, FLOSS development represents an important phenomenon that deserves to be studied further (Feller 2001; Lerner and Tirole, 2001). Prior research on making commercial use of FLOSS has focused mainly on pointing out that managers must take care when adopting FLOSS (see, for example, Lerner and Tirole, 2001; Ven et al., 2008). In this paper, we focus on individual companies and pose the research question: *"How do software entrepreneurs perceive FLOSS-driven changes in their business?"* We address this question empirically through a qualitative inquiry and a narrative analysis.

2 Related Research

The benefits of open innovation are widely accepted in the software development community (e.g., Von Hippel and Von Krogh, 2003; Henkel, 2008). In its broadest sense, software innovation refers to research and development (R&D) activities that involve intellectual capital, physical products, and processes in software production

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 199–204, 2010.

(Vujovic and Ulhoi, 2008). Chesborough (2003) observes that strategic innovations have been regarded typically as a company's most valuable competitive assets, which also serve as barriers to entry by competitors. Service-dominant logic (Vargo and Lush, 2004) describes a significant transition in business in terms of the use of resources. It considers resources in the development and delivery of offerings as operand resources (those on which an operation or act is performed) and operant resources (those that act on other resources). FLOSS development depends to a great extent on resources that are external to the firm. In FLOSS businesses, resources are accessed through collaborative relationships (Dahlander and Magnusson, 2008).

Fitzgerald (2006) argues that the emergent forms of FLOSS have a strong commercial orientation in the product development, delivery, and support processes. Fitzgerald (2006) has labeled these forms as OSS 2.0. In many cases, voluntary cooperation-based collective action systems involve some form of public or semipublic good (Heckathorn, 1996; Monge et al., 1998). According to the definitions by Olson (1965) and Udéhn (1993), public goods offer participants in networks collective benefits that are (a) non-excludable, in that they are available to all network partners, and (b) jointly supplied, in that partners' uses of the goods are non-competing.

3 Methodology

A narrative approach is considered a feasible research strategy in this study, as it is well suited to investigate a phenomenon within its real-life context (White, 1981). There are several noteworthy examples that suggest how to use the narrative approach to research FLOSS (e.g., McDaniel, 2004; Szczepanska et al., 2005; Brown and Jones, 1998; Alvarez and Urla, 2002). The narrative approach coaches the members of a certain organization to frame their understanding of social reality and their place in it in a discursive manner (Phillips and Hardy, 1997).

Table 1. Cases

Case 1	Tripod (pseudonym) is a small Finnish FLOSS company that specializes in developing collaborative learning and knowledge management software, related training, and consultancy. Its revenue model is based largely on service contracts with public organizations.
Case 2	OurDB (pseudonym) is a large Finland-based international firm that specializes in relational database management systems and related services. The company's revenue model is based on dual licensing, which means it provides both FLOSS and proprietary versions of its software.
Case 3	Yoga (pseudonym) is a small Finnish entrepreneurial firm that focuses on consultation and FLOSS development. The firm specializes in combining relational databases and e-mail management tools.
Case 4	Nemesis (pseudonym) is a small Finnish company that specializes in FLOSS-based Web services, especially content management systems, related services, and online support.
Case 5	Tulip (pseudonym) is a small software development unit belonging to a branch of a large multinational corporation. It produces FLOSS software systems for interoperability testing of the concerns equipment of the main product lines. The unit is located in Netherlands and employs three persons.

We selected five FLOSS software companies to discover how the managers in these companies perceived and described the changes in their operating environment. The companies were selected on the basis that their product and service offerings were built on FLOSS. The method used for data collection was semi-structured interviews. Respondents were (senior) managers responsible for the company's strategic decision making. We conducted one to four interviews with each of the respondents from the selected firms over a 4-year period from 2004-2008. To gain a rich understanding about the organizations in their contexts, we interviewed the whole staff then employed by Tripod (3 people), Yoga (1 person), and Tulip (4 persons). Conversely, OurDB and Nemesis are larger companies, so we limited our discussions to the CEOs and CTOs of those companies. In total, we conducted such interviews for twelve respondents. The average duration of the interviews was about 2 hours, with durations ranging from 1.5 to 3 hours.

4 Discussion

The openness of innovation activity is a key theme in commercial FLOSS development. This is evident in the respondents' accounts of companies' innovation processes. The responses we received depict a fundamental difference between the open and closed innovation paradigms.

"I think the architecture of participation that is embedded in the open source philosophy is a superior innovation method. And it is not limited to software. Look at Wikipedia. It just so happens that software developers were the first ones to adopt it in the modern world. The simple fact that everything you create is open for scrutiny by anyone else is a strong incentive to produce good stuff from the start. And the meritocracy of open source leads to faster innovation and thereby better innovations. It is a Darwinian system where over time, the best solutions will emerge." (CEO, OurDB)

"It is possible to create this kind of a joint project only if you let people see that their response has some effect on the software. ---

There was a lot to do with our software before it was ready, but we opened in a very early stage. We were able to give plausible promise and thus received a lot of valuable feedback. This resulted in a quite different end product." (Respondent from Tripod)

"Our solutions are made for the customers, not for ourselves. We want to build a working solution, but we want the customers to sit down with us, so we can do it on the users' terms. --- We believe that it is not enough for us to provide open source software. In our opinion, customer should also have open access to the actual work process. --- Not only through external communication, but also in internal collaboration. We want to get the customers' messages heard." (CEO, Nemesis)

The open innovation form embodies working together with numerous partners and various members of the FLOSS community. Our narratives underscore that through FLOSS activity, firms open their innovation processes in order to benefit from the knowledge and the innovation capacity of diverse OSS communities. The theme that FLOSS offered increased customer involvement was raised several times.

"We would never have gained 5 million users to our database product without acting according to the principles of the open source software community. Since we first released our software under an open license, we have gathered feedback … development ideas, problem descriptions and solutions … and responded to all possible initiatives from the user community to develop the product with the skillful individuals using the product." (CEO, OurDB)

"At OurDB we love users who never pay us money. They are our evangelists. No marketing could do for us what a passionate user does when he tells his friends and colleagues about the software. Our success is based on having millions of evangelists around the world. Of course, they also help us develop the product and fix bugs." (CEO, OurDB)

We find that while the software providers that rely on the goods logic in a closed innovation paradigm (traditional mass-market oriented software firms) aim at software delivery though distant, transactional customer relationships and focus on the development of standardized, homogeneous services for multiple users, companies that focus on OS development strive to empower their customers to engage in software co-development.

The main advantage of FLOSS is the external contribution made by users and developers. Harnessing this innovation potential would allow the production of software and services that would be appreciated by users. The interviewees viewed that external competencies are becoming increasingly important.

"The vast community of [our OurDB product] users and developers is what drives our business." (CEO, OurDB)

"We have five million server installations in use worldwide. Around them there are small 'software ecosystems.' There are books and articles written, lectures held, courses taught, and applications developed around our products. This community of volunteers is our most important asset. Yet, it is difficult to define." (CEO, OurDB)

"Having a large user group, you get higher product quality because more people us software in different situations and provide feedback." (Project manager, Tulip)

To conclude, the ability to utilize external resources and capabilities is recognized as one of the key factors in remaining competitive in the software business. As public goods, FLOSS-based platforms, components, and applications shift the focus from development of proprietary innovations to use of the goods and knowledge that are publicly available.

A vital consideration in FLOSS activity is how it changes the means of value capture in software business. It is apparent that the interviewees saw the future of the software business as being in services rather than in commoditized products. The respondents agree that proprietary software cannot compete successfully for long in the same market as a similar FLOSS product:

"---the business will have a fierce price war, where profits disappear. ---

"We started by not focusing on profits at all. Instead, we focused on boosting the use of the software. --- The vast community of our users and developers is what drives our business. Then we sell our offerings to firms – those who need to scale and cannot afford to fail. The enterprise offering consists of certified binaries, updates and upgrades, automated DBA services, 7x24 error resolution, etc.

-You pay by service level and the number of servers. No nonsense, no special math.
-Enterprise software buyers are tired of complex pricing models (per core, per CPU, per power unit, per user, per whatever the vendor feels like that day)—models that are still in use by the incumbents." (CEO, OurDB)

The narratives provide reason to conclude that when software is distributed freely, the traditional revenue sources are waned, and firms are compelled to develop novel revenue models that may be based on services and bound only indirectly to the distribution of software licenses.

5 Conclusions

Table 2 summarizes the findings concerning the impact of FLOSS on software business through dimensions of: 1) the impact of FLOSS on the innovation process of software firms; 2) user involvement; 3) the use of external resources in software firms' operations; and 4) the revenue model as part of software firms' business models.

Table 2. Identified FLOSS-induced changes in the software business

Dimensions	Impact
Innovation process	FLOSS-based software development urges software innovators to open up their innovation processes.
User involvement	FLOSS activity emphasizes user involvement in software development and delivery.
External resources	FLOSS activity emphasizes access to external capabilities rather than the ownership of resources.
Revenue models	FLOSS-based public goods change the revenue models of software firms.

This work has some important implications. First, the findings highlight the ways FLOSS affects the software business. FLOSS-based software development forces software innovators to open up their innovation processes and it emphasizes user involvement in software development and delivery. FLOSS activity emphasizes access to external capabilities rather than internal resource ownership. finally FLOSS-based public goods change the focus of competition in the software business from product-dominant to service-dominant operations. It closes traditional sources of revenue, and compels firms to develop new revenue models based primarily on services. For scholars interested in the software industry, the findings provide interesting avenues for further research.

References

Alvarez, R., Urla, J.: Tell Me a Good Story: Using Narrative Analysis to Examine Information Requirements Interviews During an ERP Implementation. The DATA BASE for Advances n Information Systems 33(1), 38–52 (2002)

Brown, A., Jones, M.: Doomed to Failure: Narratives of Inevitability and Conspiracy in a Failed IS Project. Organization Science 35(1), 73–88 (1998)

Chesbrough, H.: Open Innovation: How Companies Actually Do It? Harvard Business Review 81(7), 12–14 (2003)

Dahlander, L., Magnusson, M.: How do Firms Make Use of Open Source Communities? Long Range Planning 41, 629–649 (2008)

Feller, J.: Thoughts on Studying Open Source Software Communities. In: Russo, N.L., Fitzgerald, B., DeGross, J.I. (eds.) Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective, pp. 379–388. Kluwer, Dordrecht (2001)

Fitzgerald, B.: The Transformation of Open Source Software. MIS Quarterly 30(4), 587–598 (2006)

Heckathorn, D.: The Dynamics and Dilemmas of Collective Action. American Sociological Review 61(2), 250–277 (1996)

Henkel, J.: Champions of revealing—the role of open source developers in commercial firms. Industrial and Corporate Change 18(3), 435–471 (2008)

Lerner, J., Tirole, J.: The open source movement: Key research questions. European Economic Review 45, 819–826 (2001)

Meyer, P.: Episodes of collective invention, working paper 368, US Department of Labor, Bureau of Labor Statistics, Washington, DC (2003)

Mockus, A., Fielding, R., Herbsleb, J.: Two case studies of open source software development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology 11(3), 309–346 (2002)

Monge, P.R., Fulk, J., Kalman, M.E., Flanagin, A.J.: Production of Collective Action in Alliance-Based Interorganizational Communication and Information Systems. Organization Science 9(3), 411–433 (1998)

Olson, M.: The Logic of Collective Action. Harvard University Press, Cambridge (1965)

Phillips, N., Hardy, C.: Managing multiple identities: Discourse, legitimacy and resources in the UK refugee system. Organization 4(2), 159–186 (1997)

Raymond, E.S.: The Cathedral and the Bazaar, Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly & Associates, Inc., Sebastopol (1999)

Szczepanska, A.M., Bergquist, M., Ljunberg, J.: High Noon at OS Corral: Duels and Shoot Outs in Open Source Discourse. In: Feller, J., Fitzgerald, B., Hissam, S.A., Lakhani, K.R. (eds.) Perspectives on Free and Open Source Software. MIT Press, Cambridge (2005)

Udéhn, L.: Twenty-Five Years with the Logic of Collective Action. Acta Sociologica 36, 239–261 (1993)

Vargo, S.L., Lush, R.F.: Evolving a services dominant logic. Journal of Marketing 68, 1–17 (2004)

Ven, K., Verelst, J., Mannaert, H.: Should You Adopt Open Source Software? IEEE Software, 54–59 (May/June 2008)

Von Hippel, E., von Krogh, G.: Open Source Software and the "Private-Collective," Innovation Model: Issues for Organization Science. Organization Science 14(2) (March-April 2003)

Vujovic, S., Ulhøi, J.P.: Online innovation: the case of open source software development. European Journal of Innovation Management 11(1), 142–156 (2008)

The Case for Software Business as a Research Discipline*

Mikko Rönkkö, Aku Valtakoski, and Juhana Peltonen

Aalto University
{mikko.ronkko,aku.valtakoski,juhana.peltonen}@tkk.fi

Abstract. Software industry has served as an important empirical setting for research in management. Subsequently, some scholars have proposed that software business constitutes or is poised to become its own research discipline or topic area. We argue that this is without solid grounding: First, there is a lack of evidence behind this claim, and some of the favoring arguments are simply fallacious. Second, there seems to be a misunderstanding of what constitutes a research discipline. Third, proponents of this thesis apparently ignore much of extant research in relevant disciplines and other research fields. In our view, the case for a software business discipline has been fueled by knowledge transfer problems between researchers primarily identifying with software business and mainstream management research. We conclude that software business does not constitute a discipline of its own and it is highly unlikely that this will ever happen.

Keywords: Software business, scientific progress, academic discipline.

1 Introduction

Recently some researchers have called for the formation of a distinct discipline, field, or research area to be founded around software business. Scholars have attempted to identify "The position of software business as a field of research" [1, 2] and argue that "An international software business research community focusing on these and many other important issues is gradually forming" [3]. Moreover, some have even gone as far as arguing that they had already founded such a research area: "Expert panel will publish 'Volendam Manifesto' that will define the core of the research area. This will be used as a corner stone for the founded research area ..." (our translation) [4]. This new "area" is being justified by filling knowledge gap between software engineering and business management: "…it seems evident that there exists a major body of knowledge at the intersection of software engineering and business management that is beyond these two generic disciplines and thus specific to software business." [5]. The main argument of the authors of these lines seems to be that software business needs to and can become a distinct research discipline or research area of its own

* A longer paper presenting the same argument in more elaborate form is available as report number 1/2010 in the Working Paper Series of the Software Business Laboratory (ISSN 1795-6978).

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 205–210, 2010.

right. Although these cited sources are all from Finland, the founding of the International Conference on Software Business indicates that these ideas have broader appeal.

One of the main problems with the arguments for software business as a discipline, field, or research area, is that the authors presenting these arguments do not define what exactly they mean by these terms. In this paper we use the word *discipline* to refer to a system of research having a distinct paradigm [6] consisting of topics, a theory base, and methods. Furthermore, we define *topic area* as a system of research with a shared topic but operating under existing disciplines regarding theory and methods. We believe that together these two concepts cover most of the uses of the terms discipline, field, and area in the papers advocating the distinctiveness of software business research.

This paper focuses mainly on the arguments advocating software business as a distinct discipline, which we think are without merit. Our main conclusion in this paper is that research efforts would be better spent by focusing on software business issues through the lenses of current disciplines rather than focusing on developing and advocating a new discipline.

2 Problems in Justifying a Software Business Discipline

The main argument for software business as a distinct discipline seems to be that software business as a phenomenon is so unique that it should constitute a scientific discipline of its own right. The first line of this argumentation suggests that the current research disciplines are somehow unable to effectively describe and explain the phenomenon of software business and thus a new one is needed. The second line of argumentation seems to state that software business as a phenomenon is simply so unique that no other empirical phenomenon resembles it, in which case the existing theories are not useful in describing software business at all. In other words, the first argument states that current disciplines attempt to but are inefficient in explaining software business and the second argument that the existing disciplines ignore software business altogether. In both cases, the burden of proof resides on the person presenting the argument, but thus far none of the papers have explicated the logical reasoning or empirical observations that would warrant concerns on the adequacy of the existing disciplines.

Third line of argumentation, although one that cannot be found in the reviewed papers, but that has been presented on several occasions in informal discussions, is that the current research papers in traditional disciplines are not useful for a practicing manager. We will start by discussing the first two arguments and leave the third argument to the end of the paper.

The strongest case against the first argument is the fact that, there exists a large number of academic papers, some of which are even important to their respective disciplines, which are placed in the empirical context of software industry. Therefore, we must conclude that the existing disciplines and theories frequently employed there can be used to explain phenomena related to software business. Second, this argument seems to be a victim to the fallacy of confusing empirical context with theory. Theory is an explanation why two constructs are related in a particular way [7]. The argument

that "software business has not been researched in these empirical terms" mixes two levels of abstraction. Yes, we grant that there exists very little research that builds theory specifically related to software business. However, theory development happens at a higher level of abstraction; if theory does not mention concepts found in software business, it does not mean that the theory could not be used to explain the phenomena in this context.

The second line of arguments for the establishment of software business research discipline is based on the uniqueness of software as a phenomenon. Some proponents of software business discipline suggest that software as an artifact is distinct from any other artifact and that this uniqueness of the artifact gives rise to organizational, management, or business phenomena unseen in other sectors of economic activity. First of all, software as an artifact is not that unique. Many other industries deal with similar intangible goods [8]. Granted, software does have some distinctive characteristics from these "passive" information goods. Yet, for the most part, the same basic economic principles apply to software as for other information goods. Furthermore, it is also somewhat unclear how the proposed special characteristics of software could give rise to unique organizational phenomena in comparison to other information goods or other knowledge-intensive goods or services.

One particularly popular form of the uniqueness of software argument builds on the complexity of software artifacts. This arguments – and virtually every other argument based on the so-called special characteristics of software – are an example of deductive fallacy if not an outright *illicit major*. In this case flawed logic can be refuted on two fronts. First, the person presenting the argument would first need to justify that there exists a rule that when something is e.g., sufficiently complex, existing theories do not apply. Second, the existence of counter examples of existing theories being applicable and having predictive validity illustrate that at least some of the premises in the argument are false.

In sum, this far we have not seen "the smoking gun" indicating that the current disciplines are inadequate for explaining phenomena in the software industry. What then would be needed for a software business to constitute a discipline? Let us consider this from the perspective of work by Thomas Kuhn [6], who introduced the concepts of paradigm and normal science to the discourse on the history of science. Paradigm in this context refers to the accepted sets of principles and practices on which science builds including "law, theory, application, and instrumentation" [6]. Within a period of normal science a discipline is typically dominated by a single paradigm. If empirical anomalies that current theories cannot explain are identified, the discipline enters into a crisis that is eventually resolved by introducing a new paradigm that explains the existing phenomena and the anomalies better than the previous paradigm.

To constitute a discipline, research on software business would need to define its paradigm including research subjects, sets of topics, and theories. Additionally, if a new paradigm for software business were to succeed, it should demonstrate superior explanation when compared to the current paradigms in for example economics and sociology. Comparing this with the current research on software business, we argue that while the set of research subjects (software and software business) is relatively well-defined, there are serious problems with both the topics and theories of software business. To become a distinct discipline, research would need to develop an independent system of theories that are related to both the topics and subjects of research.

The third argument for software business as a distinct discipline attacks the practical relevance of current research in the actual (other) disciplines. The argument has two forms with the first one being that there are numerous academic papers in the discipline of management that are not useful to managers of software companies. The second form is that no existing academic paper presents a comprehensive model of how a software business should be run. First, we suggest that these are *argumentum ad ignorantiam* and based on partial or missing knowledge of management research and what constitutes a research discipline. Second, we also argue that proposal for a software business discipline reveals a knowledge transfer problem between management researchers and those that identify software business as their primary discipline or topic area.

The reason that a single article might not be useful for a practicing manager is due to focus and specialization that are required for efficient scientific research, but that can lead to difficulties in understanding the articles if the reader lacks proper background knowledge. However, since knowledge needs to be communicated not only with researchers, but also to practitioners and other audiences, science has developed a double-loop model of knowledge creation and transfer [9]. This model, shown in Figure 1, implies that there are two distinct discussions to which a scholar participates. The first loop operates within a scholarly community, efficiently creating knowledge that conforms to established paradigms through discourse in academic journals and conferences. In the second loop, the results of research are communicated to practitioners and students through teaching and consulting as well as publishing in books and magazines like *Harvard Business Review* or *IEEE Software*. Following the wrong loop or having insufficient understanding of the current modus operandi causes knowledge transfer problems.

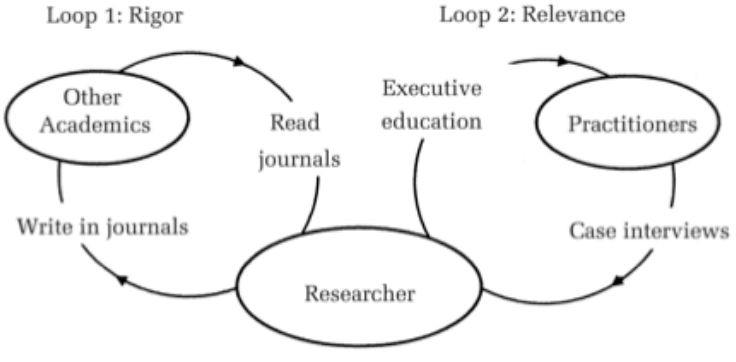

Fig. 1. Double loop model of knowledge creation [9]

The lack of relevance argument is sometimes also presented in the form of downplaying the value of the first loop and advocating software business as a purely applied "discipline", whose purpose is not at all to participate in the theory development-theory testing game, but to construct models that can be directly communicated to the second loop. The exact definition of this constructive research is elusive and it is difficult to find methodological texts pertaining to this approach. However,

generally this approach is presented as synonymous or as a combination of approaches called design science and action research. Following this approach, researchers construct artifacts often called e.g. frameworks. We do not suggest that constructing such models is not useful – in fact, the consulting industry mostly operates on these kinds of activities. What we argue is that unless a model presents an interesting and testable theoretical argument or provides sufficient empirical evidence for the validity of the model with the appropriate reporting of the research setting and methods, the result does not advance science not matter how widely they are applied.

Another problem with the model building – particularly with general models – is that generally there is no one right way to manage even a simple manufacturing firm. The concept of scientific management that proposed this was introduced almost a century ago [10], but the approach has been abandoned in favor of contingency theory: the organization must fit its environment, environments differ, and hence there cannot be general prescriptive models.

Hence the arguments for lack of relevance argument boils down to either a misunderstanding about the purpose of the current disciplines that address software business or a lack of appreciation about what is possible with scientific method.

6 Discussion and Conclusions

In this paper, we have reviewed the arguments passed for establishing an independent research discipline for software business. Our review indicates that these arguments are often based on fallacious supporting arguments and evidence, misunderstandings about what actually constitutes a scientific discipline, and general ignorance about the state of research in actual disciplines and the applicability of these disciplines' results to research on software business phenomena. In conclusion, we argue that no worthwhile arguments have been published for the establishment of software business research discipline.

We conclude the paper by discussing software business as a topic area. The usefulness of a software business topic area can be considered from two different perspectives. First, for any question to be most effectively studied within software business topic area, the context should provide higher relevance for the question. Some people have argued software product development as such topic [11], but this is a rare example of area where software-specific topics can be found. Another way to look at the topic area is whether a set of papers are better combined into a discourse by the empirical context or phenomena. We argue that the latter provides a more natural fit.

What value then, has software business research outside of being an empirical context for other disciplines? There is potentially value in software business as an agenda for education or societal impact or as a bridge making theory from management and other social sciences more accessible for software engineering researchers [12].

Our paper has several direct implications for current and aspiring software business researchers. We argue that the idea of software business as a distinct discipline should be abandoned. For the researchers currently working in the software business area, the position of the proposed topic are should more clearly presented in relation to existing areas and disciplines. Without this, software business research cannot be built on by researchers operating in other areas leading to isolation and then eventually

irrelevance of the research community. If this proves impossible on the level of topic area, then at least it allows better positioning of individual studies. For aspiring software business researchers, the present paper and the references presented enable more informed decision on where to position oneself in the field of sciences.

References

1. Mäkelä, M., Mutanen, O.: Research in software business: implications of the special qualities of software as a good. In: Proceedings of Engineering Management Conference, 2005 IEEE International, pp. 780–783 (2005)
2. Mäkelä, M.M.: Software Business: Position as a Field of Research and Avenues for Scholarly Contributions. Presented at the 14th International Conference of the International Association for Management of Technology (IAMOT), Vienna, Austria, May 22 (2005)
3. Käkölä, T.: Software business models and contexts for software innovation: key areas for software business research. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS), Big Island, Hawaii, USA (2003)
4. Ohjelmistoliiketoiminnan tutkimuksen tulevaisuus: Volendamin manifesti, tulossa, http://www.swbusiness.fi/portal/67/?id=16499
5. Kontio, J., Ahokas, M., Pöyry, P., Warsta, J., Mäkelä, M., Tyrväinen, P.: Software Business Education for Software Engineers: Towards an Integrated Curriculum. In: 19th Conference on Software Engineering Education and Training Workshops (CSEETW 2006), Oahu, HI, USA, p. 5 (2006)
6. Kuhn, T.S.: The Structure of Scientific Revolutions. University of Chicago Press, Chicago (1970)
7. Bacharach, S.B.: Organizational theories: Some criteria for evaluation. Academy of Management Review 14, 496–515 (1989)
8. Shapiro, C., Varian, H.R.: Information rules: a strategic guide to the network economy. Harvard Business School Press, Boston (1999)
9. Vermeulen, F.: I Shall Not Remain Insignificant: Adding a Second Loop to Matter More. Academy of Management Journal 50, 754 (2007)
10. Taylor, F.: The principles of scientific management. The Echo Library, Teddington (2006)
11. Xu, L., Brinkkemper, S.: Concepts of product software: Paving the road for urgently needed research. In: First International Workshop on Philosophical Foundations of Information Systems Engineering, LNCS. Springer, Heidelberg (2005)
12. Rönkkö, M., Frühwirth, C., Biffl, S.: Integrating Value and Utility Concepts into a Value Decomposition Model for Value-Based Software Engineering. In: Product-Focused Software Process Improvement, pp. 362–374 (2009)

How Can Academic Business Research Support the Finnish Software Industry?

Nina Koivisto

Aalto University School of Science and Technology
Nina.Koivisto@tkk.fi

Abstract. This paper describes preliminary results of an interview study on business research regarding the software industry in Finland. For the study, we interviewed 30 people in 4 positions in Finnish universities. Overall, adopting a long-term approach in studying a given question was considered to be the essential strength of research done in universities. Obtaining funding for such research was considered to be a major weakness. There also appears to be a mismatch between policy organizations and what researchers wish to research, and that research should be more centrally themed. Several trends in the software industry were considered worthy areas of inquiry from a practitioner viewpoint, however not as appealing from an academic perspective. The respondents also considered that there are no "national level" challenges that would be specific to the software industry. Obtaining finance for commercializing and productizing inventions that emerge from basic research was considered a problem, though Finland has an extensive publicly supported innovation system.

Keywords: Research, Software Industry.

1 Introduction

The software industry is currently undergoing fundamental structural changes that challenge traditional products and services business models. This includes blurring the border between of content and software and increasing use of cloud resources. Motivated by these trends, an inquiry was conducted on how academic research is positioned to study these changes. The study was conducted in Finland, because it enabled studying the Finnish public innovation system, which has come under increasing scrutiny recently. The study of [1] concluded that Finnish research, with the exceptions of medicine and agriculture, does not meet standards that are considered world class. In particular, Finnish research units suffer from the over-diversification of resources into small and dispersed research units. An earlier assessment by the Academy of Finland also reached similar conclusions [2]. A low degree of international collaboration and low quality of publications were also identified as key challenges.

This paper seeks to find answers in what kind of research are universities at their best, what kinds of present and future trends and phenomena have been observed in the software industry that also form interesting targets for research, and what kinds of

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 211–216, 2010.

methods and researchers should these phenomena ideally be studied. After this section, I will present preliminary results of the interview study starting from the research methods and empirical study design. In the third chapter I will focus on results (trends of the software field and technically oriented trends). The paper is concluded by a section discussing the trends and problems identified.

2 Research Method

The data for this paper were collected in conjunction with the Growth Forum '09 project [1]. Growth Forum is a national project headed by the Finnish Software Entrepreneurs' Association that aims to spark public discussion on the significance of the software sector to the Finnish economy and to communicate information on its needs and opportunities to policy makers. The author was involved in the research work group of the Growth Forum that had representatives from six Finnish Universities. This included four professors, three researchers and a senior lecturer. The workgroup is working on a more comprehensive report on the same topic.

We sought to target informants who were involved in conducting or managing research regarding the software industry from various disciplinary backgrounds. To limit the sample the group decided to target individuals involved in conducting or managing research that studies software companies where a business perspective is dominant. In addition, we included people involved in research that does not explicitly target the software industry's business aspects, but can be considered generalizable to this context.

To communicate the sampling criteria, the work group prepared an interview process description that included a description of types the informants to be targeted. The document was later circulated for comments in the workgroup, and some changes were made to its content. The sampling strategy can be considered a snowballing approach. The university representatives identified an individual to conduct the interviews in their university and proximate universities that were not involved in the Growth Forum project. In two cases, the representative was the same person as the interviewer. This included the author, who conducted the interviews at Helsinki School of Technology and University of Turku.

Before proceeding with any of the interviews, these individuals created a shortlist of their contacts that match the above criteria (with some minor exceptions noted) that was reviewed together with the author. The amount of interviews was limited from three to six in each university to maintain a balance in the sample. In some cases it was agreed that the list be extended after gaining experiences from a few interviews. In these cases, a similar discussion round took place to discuss the match of the interviewees with the sampling criteria.

As a result 30 interviews were conducted in 8 universities. The interviewees included 17 professors, 2 docents, 5 researchers, 3 lecturers, and 3 other titles. The universities represented in this paper are Lappeenranta University of Technology, University of Turku, Turku School of Economics, Aalto University School of Science and Technology, Aalto University School of Economics, University of Tampere, Tampere University of Technology, University of Jyväskylä, and University of Oulu. Because of the tight schedule of the project, we had to exclude rest four universities.

The informants represented fields of information systems, technology and innovation management, venture capital, entrepreneurship, international business, strategic management and marketing. The data were mainly collected during the fall 2009 by structured face-to-face or telephone semi-structured interviews and they were audio recorded. The interview questions are listed in Table 1.

The NVivo 8 software package was used to analyze the data. Themes were picked out of categories and representative quotes selected from interviews. The research themes in software business found by interviews are: computer science, technology & innovation management, venture capital, entrepreneurship, international business, strategy and leadership, and marketing. According to these interviews we formed an overview that is presented in this article.

Table 1. Interview questions

Q1: Doing research in universities: a) In what kind of research are universities at their best? b) What kind of research should not be conducted in universities? c) Where should this kind of research be done?
Q2: What kinds of present and future trends and phenomena have you observed in the software industry that also forms interesting targets for research? Please pay special attention to the following: a) ”technically-oriented” trends, that have the potential to shape the industry’s structure (e.g. cloud computing, clean tech, ubiquitous computing) b) The challenges and opportunities of software companies from small developed countries (like Finland) to expand outside their borders and become major global players
Q3: To the items identified in the previous question: a) What kinds of methods and researchers should study these phenomena in an ideal case? b) Is your research group planning to do the aforementioned kind of research?

3 Results

According to the interviews, university research is at its best when it is longitudinal (8/30 statements), wide range (6), and basic research (5); when looking at ‘the big picture’ and combining perspectives from several fields. Basic research also brings in sustained and credible understanding, especially when focusing on case studies (2) in different sectors of business and strategic analyses that are based on understanding causal connections. “Focusing on a particular problem profoundly, and with a long period of time, is a central strength”, says one interviewee. The fact that the university does not get finance (2) to do it was regarded as probably the biggest problem. ”Universities are best at blue sky basic research, but they are not doing that. It benefits all industry, everything else is irrelevant. Constructive research is okay, if you have to fill in gaps”, says one interviewee.

It was also often stated that research is at its best in different technologies (7), their properties, software development, and software production. The people's behavior and interdisciplinaries (6) are not as strong and it is also reflected in the results. ”Research that takes the interaction between people into consideration in the product development process is also needed”, a scholar from Lappeenranta says. “The user based points of view should be taken into consideration in more studies”, says another scholar from Lappeenranta.

Technical research can be more easily utilized from the industrial companies’ point of view. So it is interesting to think what (direct) advantage the research of the university has or from whose point of view benefits should be thought of. Academic research must have a strong theoretical foundation but also a link to the practice. ”It is difficult to find professors that give any value to the fact that the results should be exported to the business world into the right environment, to have a real impact. The research results are taken to other researchers, not to the real environment”, an interviewee criticizes.

Matters from which a model, method or conformity to law cannot be generalized in universities should not be studied. It would not be worth to consult, to evaluate or to conduct market research, make reports (10), surveys (8) or short shortsighted commissioned research in universities. The consultant companies, the Technical Research Centre of Finland, Research Institute of the Finnish Economy (the branch reports) and the polytechnics (the operative processes of the business) should make them. ”Practical problems that can be quickly solved should be left to the industry. The university should look for the answers to the future until 50 years on”, an interviewee describes. It was hoped that the companies used the university as partners in cooperation (5) in the (research) development. The research, teaching, the social influencing, and cooperation with the companies are extremely important issues to the universities. It was experienced that university research is at its strongest when it is cooperating with companies in the long-term. Universities should attempt to maintain and develop long-term relationships companies.

The interviewees listed interesting research subjects: to clarify how the entrepreneurial attitude could be promoted, those methods with which more companies would be made to grow bigger and still remain Finnish, to document best practices/ operations models/strategies connected to the overcome of the aforementioned challenges, availability of the (risk) finance, and utilizing embedded software to branch out to new industries.

3.1 Trends of the Software Field

The trends of the software field were considered extremely important from the point of practice but not so interesting in the academic point of the research subject. On the other hand, the limits of the software field were considered unclear; how does the software field join other fields. Strategy and management scholars think the challenge of the future in the software field will be regeneration in the future. She wanted research also to 'look in the mirror' afterwards; the conclusions cannot be based only on the fact that beforehand it would be estimated what kind of effects the new systems could have.

Internationality is an important part of the development of the ICT field. The development must be carried out on international forums in the cooperation with researchers and other developers, and the examining should not be concentrated in a national context only. The international networks must be utilized so that the best practices are brought to Finland, and the brain drain would not totally remove know-how to the countries of the cheaper labor. "How start-up financing effects the growth and internationalization of a company, interaction with the customer through a network point of view, and both the significance and developing of the confidence are interesting research subjects", a scholar of international business lists.

An interesting trend is a social point of view in software development. The earlier research has not taken into consideration that the software is made among people to other people. “The questions that are related to the people's organizing are interesting,” state several interviewees also.

3.2 Technically Oriented Trends

The technically oriented trends have to do with examining new opportunities and how various new possibilities can be adapted. These trends should be taken into consideration more effectively in business research, and communication between different parties will be needed considerably more.

Some of the strong technological trends according to interviewees are: cloud computing (12 /30 interviewees), ubiquitous computing (9), open source (6), green IT (4), and SaaS (3). Especially open source, innovations, Web 2.0, SaaS, and cloud computing are considered important because they change the structure of the operations and business models. SaaS means global competition and specialized, small niche offerings.

The other interesting trend was ubiquitous computing, the spreading of the Internet into every place. It is believed that it opens new, big business opportunities and many service concepts in people's lives. However, at the same time it was stated to be more difficult to see which schedule, for example bio IT will spread ubiquitously.

Other important research areas according to interviewees are: the research of the service-service, the convergence and mobilization of the medium, the increasing role of the social medium, the emerging software business in other branches, and how open code, innovations and Web 2.0 change business models.

An interviewee describes: "The biggest change that is taking place is a certain kind geographical tailoring. Different geographical locations specialize in different matters, for example, China specializes in electronics manufacturing, more than two thirds of the semiconductors are made in Taiwan. One can think that the value chain of the company spreads around the world: IT-coders are in India, call centers in Estonia, Hardware collecting in China and the main market in the USA. An interesting situation to the managing director who sits in Finland: what is done here, and how the whole ‘big picture’ is conducted?" She continues: “New kinds of know-how, ability to lead must be found, by which none of these I would have learned at school”.

4 Discussion and Conclusions

The most significant change that is due to the trend will not be technical but the connection to the company structure of the field and to the value networks. This will

provide numerous opportunities to the software companies. It is interesting to see what role national competitive advantages of Finland, such as high education level and telecommunication know-how, will act in this development. In any case as the central infrastructure required by the software goes "to the cloud", increasingly smaller companies can deliver technical solutions to an increasingly wider group of different customers. The efficient marketing and control of trademarks form the competitive advantage in the future. So the Finnish competences are required to transfer to the areas in which they have traditionally been at their weakest. The rareness of the companies succeeding in Web 2.0 can be considered as a serious symptom of the situation in Finland. In the software field there is big potential and a need to become even more internationalized. This requires better business know-how. The research and teaching of management and business indeed have big potential to be changing the direction of the future.

One of the central problems on the basis of the interviews is finding the financiers and a long-term commitment to the research. The companies of the software field are small in size. From Finland the infrastructure of the small companies is missing.

The research is important and utilization of its result a vital condition, but as rector Teeri crystallized about Aalto in the occasion of Growth Forum 10.11.2009: "It is not the task of the Aalto to commercialize the know-how of Aalto, the university organization is not a sales organization".

Interesting future work would find out how companies respond to this research.

References

1. Kontio, J., et al.: Kasvufoorumi 2009. Technical report. Helsinki: Ohjelmistoyrittäjät ry (2009), http://www.ohjelmistoyrittajat.fi/files/documents/kasvufoorumi09_loppuraportti.pdf
2. Oy, T.: #Evaluation of the Finnish National Innovation System (Full Report) (2009)
3. Akatemia, S.: Suomen tieteen tila ja taso 2009 (2009), http://www.aka.fi/fi/A/Tiedeyhteiskunnassa/Tutkimuksen_arviointi/Suomen-tieteen-tila-ja-taso/

Software and Standards in an Emerging Domain

Mirja Pulkkinen, Denis Kozlov, and Jan Pawlowski

University of Jyväskylä, Department of Information Systems and Computer Science
{mirja.k.pulkkinen,denis.y.kozlov,jan.m.pawlowski}@jyu.fi

Abstract. This paper considers the software market in the field of e-Learning and Technology Enhanced Learning (TEL) through observing standardization developments in the domain. A model of software market evolution suggests that observations on development and deployment of standards in the domain indicate the status on the market. As the model holds, in a mature phase, level of standardization is high and the number of competing standards low. The result of the study is, there appears to be a market, but a heterogeneous one, with hesitation on dominant designs and an overall modest level of standards adoption. Content standardization enables the (re)use of the learning content in multiple formats with diverse learning management systems (LMS), virtual learning environments (VLEs), learning platforms, HRM and administration systems. A number of initiatives to develop specifications or standards exist. Competing standards or specifications are found for most target areas. Efforts to establish common frameworks and reference models have emerged. Such overall consensus frameworks will support developments towards content interoperability in this vertical as they have in other domains.

Keywords: Software market evolution, content, standards, e-Learning, TEL.

1 Introduction

Together with adoption of technology support for teaching and learning, specific software and systems for the domain of education and training have emerged. Research companies like Gartner and Forrester were frequently mentioning e-Learning connected to hype in the mid 00's. Breaking into the second decade of the 21st century, this software domain has permeated its position as one of the software verticals watched by the market researchers. Forrester reports on building of a growing community around e-Learning[1]. Web catalogues that list software providers with SaaS or ASP business model[2], the cutting edge in the software industry, seem to provide evidence that the e-Learning vertical did not disappear with the hype: significant numbers of companies are listed under the search terms *education, e-Learning, teaching, training*. The software domain of learning, education and training is not only in the interest of education sector, i.e. schools or educational institutions. For knowledge

[1] For Information & Knowledge Management Professionals "Learn From A Community: The eLearning Guild" A Forrester report by Claire Schooley, June 24, 2008.
[2] SaaS Showplace, saas-showplace.com; ASP News www.aspnews.com

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 217–222, 2010.

work, there is a constant need to build new knowledge and life-long learning. Employee on-the-job learning and training as well as organizational learning in private businesses are supported with e-Learning platforms.

This paper investigates the specifications and standards in the market vertical of software for education: learning/teaching, training and assessment, focusing on systems and applications designed for managing and utilizing learning content. The paper is based on the studies on standards in an EU best practices network, ICOPER www.icoper.org, which conducts work on the improvement of e-content interoperability in LMS and applications, through harmonization and support for content standards development for competency-based learning.

Section 2 gives a brief overview on the theory of software market evolution and standards evaluation. Section 3 presents the data, collected in the ICOPER project, and the current status of development and deployment. Section 4 presents an analysis of the material, which is discussed in Section 5, with some concluding remarks.

2 Theoretical Background

2.1 Software Market Evolution

The theory of software market evolution (Tyrväinen et al 2008), sees the emergence of a market for software in a domain of systems and applications as the sum of four factors:

- Enabling technologies to build applications and systems for the domain
- Potential users develop demand for such systems: the acquisition of them becomes feasible and desired.
- As software supports processes, two more factors are related to the processes:
    - Processes generate the requirements for the software and systems. The process development emerges together with the systems implementation.
    - The software applications and systems have interfaces to other systems, (software as well as hardware), and other processes. (Tyrväinen et al 2008, Tyrväinen, Mazhelis et al. 2009). Process harmonization and simultaneously an emerging common understanding of the activities in the domain drives the evolution.

New solutions offered to a market of users will have success when all these factors contribute to successful implementation and use, and the users of the systems and applications gain benefit from the use. Patterns of technology acceptance and industry evolution repeat themselves quite independently of the industry or domain (Tyrväinen, Mazehlis et al. 2009)

Four stages can be observed in the evolution of a software market domain: 1) Innovation phase with in-house development of bespoke systems, 2) Productization phase with spin-off companies taking over the development as well as established software houses expanding to the new domain with software products, 3) Transition phase with a shake-out of both a diversity of standards and companies in the market, and finally, 4) Service and variation phase, where the processes, applications and interfaces follow one or few dominant standards and the ICT providers go over to competing with diversified services (Tyrväinen et al. 2008).

In this study, the focus is on the factor of standardization as a post hoc indicator of developments in a segment. Education domain deploys mostly mainstream operating systems and network protocols. The interfaces to look at are data formats (as content files of learning resources or learner information, the meta-data on content, learner etc.) and the application programming interfaces (APIs) in the platforms (LMS, VLEs) used. The questions to be answered are, *does a significant education/learning software market vertical exist, and how well developed is it*?

2.2 Standards Evaluation

Cooper (2009) provides an extensive set of criteria to measure fitness e-Learning / TEL standards. According to Cooper, there are six groups of indicators of fitness: 1) Business case, i.e. suitability of a standard for implementation in a business case; 2) Purpose and intended use, i.e. whether a standard has a clearly stated purpose and whether it emerged from a stable practice or theory; 3) Context, i.e. whether a standard is mature and has proven effective in specific situations; 4) Provenance and Governance, i.e. the quality and credibility of the originator and the institution maintaining the standard; 5) Technical aspects of the standard: platform independence, architectural requirements; 6) Personal requirements: privacy, ethics.
If e-Learning/TEL standards are found fitting, the standardization level, and thus the software market maturity in a domain should increase. Relevant in estimating the evolution stage according to recent studies is (Tyrväinen, Mazhelis et al. 2009):

1. Emerging domain specific standards, and maybe competing standards, if a market has emerged.
2. Adoption of standards, i.e. availability of examples of effective and wide use if the market is evolving.
3. Shake-out and emerging of dominant standards.

3 Research Methodology and Data

The study is conducted as a qualitative analysis of e-Learning standards on two sources. Besides the information provided by accredited standardization bodies (CEN, IEEE, ISO/IEC), and consortia working on the area of e-learning and TEL, (e.g. ADL www.adlnet.gov, IMS Global Learning Consortium www.imsglobal.org/, JISC/ CETIS http://jisc.cetis.ac.uk/), a meta-analysis is conducted on the initial analysis of standards in the ICOPER project. Duval (Duval 2004) provides an overview of the activities in standardization in the field, and e.g. JISC/CETIS has taken on publishing regular reviews:

http://wiki.cetis.ac.uk/Eductional_Technology_Standards_Reviews.

The ICOPER project providing material for our analysis is organized around 6 work areas: 1) Outcome-based learning with focus on competencies, 2) Learning design, 3) Content reuse, 4) Delivery of open content, 5) Student assessment and course evaluation, and 6) Services integration. Each of these delivered an initial report on their work area, with review of the standards relevant in the respective area. The reports D1.1, D2.1, D3.1, D.4.1, D5.1 and D6.1 are to be found in the project space in the project collaboration support environment www.educanext.org

4 Evaluation of Learning Technology Standards

The complete list resulting from the initial analysis in the project (Kozlov et al. 2009) shows 25 specifications or standards (at various stages of a standard life cycle) considered in some way relevant in e-Learning and TEL domain. In the first analysis phase, this list is scanned for items that are related to the core processes in the domain (teaching/learning processes or processes directly supporting these). Table 1 shows the remaining specific standards, their adoption rate, and competing standards.

Table 1. Analysis of existing domain specific specifications and standards

Standard /Specification	Observations	Possible competition	Background
LOM Standard for Learning Object Metadata, IEEE 1484.12.1-2002	Significant adoption	CEN MLO, DC SCORM	Merged: IMS Learning Resource Metadata LRM
SCORM Sharable Content Object Reference Model (4th Edition) Version 1.1 ADL 2004	In specific use (US DoD) high adoption	LOM; IMS LD; CEN MLO	
IEEE RCD Reusable Competency Definitions, IEEE 1484.20.1, 2007	Some adoption (Draft standard)	HR-XML, CEN MLO	Merged: IMS Reusable Definition of Competency or Educational Objective RDCEO
IMS LD Learning Design, IMS GLC 2003	Some adoption	SCORM	
IMS QTI Question and Test Interoperability IMS GLC 2005	Moderate adoption	DocBook, FML, QAML, SuML	
ECTS Information Package/Course Catalogue MLO Application Profile, CEN CWA 16076:2010		SCORM	
MLO-AD Metadata for Learning Opportunities – Advertising, CEN WS-LT CWA 15903:2008		IEEE LOM, HR-XML	
IT Learning, Education, Technology QM, ISO/IEC 19796-1:2005	Local adoption		
HR-XML, Schema for Human Resource; HR-XML Consortium 2007	Significant adoption in private sector	IMS GLC ePortfolio, CEN-MLO	

In observations on background (on the right in the Table 1 above), it can be seen that work towards a specification successfully contributed to an emerging standard. IMS GLC conducts projects also on further areas, e.g. services as Organized and distributed digital learning content (Common Cartridge - CC), Applications, systems,

and mash-ups (Learning Tools Interoperability - LTI) and Learner information: privileges and outcomes (Learning Information Services – LIS) and ePortfolio.

In almost all cases there are competing approaches (Table 1). IMS QTI has some potential rivals with the questions and answers assessments functionality (e.g. DocBook, FML, QAML, SuML) in non-specific approaches. Extending the overlaps to standards non-specific to the e-Learning or Technology Enhanced Learning (TEL) domain the situation is even more complex. E.g. the Dublin Core (DC) metadata standard is widely applied.

In addition to specifications and standards, common frameworks (as in other domains) are emerging. The **FREMA framework** on assessment and evaluation[3] is quite well developed, due to the applicability of automation in the assessment area. The emerging International **eFramework**[4] is an attempt to cover the field from the service orientation viewpoint. However, an established common framework of reference does not exist.

5 Discussion and Conclusions

The observations in our study tell that a vertical of *e-Learning/TEL software* does exist on the market, and it is evolving. The market at this point is rather heterogeneous, but development of specifications and standards is pursued by authorities like EU, as well as active industry consortia. The EU required approach, *competence driven education*, induces learning outcome based teaching and learning and pushes the introduction of specifications for learning outcomes and other related targets. Work is being conducted on defining e.g. competency profiles and learning opportunities. Overall, adoption of e-Learning and TEL specific standards or specifications, as reported in the ICOPER initial studies, is in early phases. With maybe the exception of Learning Object Metadata, LOM, the rate of deployment of standards is rather modest. IMS QTI for testing/assessment has gained broader support. A number of dedicated consortia, e.g. IMS Global Learning Consortium and ADL (Advanced Distributed Learning, the provider of SCORM) and many less known ones, conduct parallel work towards standardization. Accredited (ISO/IEC, IEEE, CEN/ISSS) and national standardization bodies entertain initiatives for accreditation of standards.

Educational institutions, being government supported, are confined by government funding, policies and guidelines in their strategies for adopting educational systems and software. Therefore, the evolution of this software market vertical might not follow exactly the pattern found in other verticals. By observing standards, however, support for the software market evolution model is found also in this domain. Lack of coherent frameworks for the domain to build a common understanding on the area, as found an essential collaboration vehicle in other domains, may be a factor that hampers the evolution. Emerging frameworks like the e-Framework and the ICOPER Reference Model IRM may help to alleviate the versatility, and support further standardization as well as the software business activity in this domain.

[3] http://www.frema.ecs.soton.ac.uk/
[4] http://www.e-framework.org/

Acknowledgements

This paper has been written in the context of the ICOPER Best Practice Network (http://icoper.org), which is co-funded by the European Commission in the eContentplus programme, ECP 2007-EDU-417007. The paper only states the opinions of the authors.

References

CEN (2008), http://www.cen-ltso.net/Main.aspx?put=1042

Cooper, A.: Evaluating Standards – A Discussion of Perspectives, Issues and Evaluation Dimensions, JISC CETIS April 14 (2009), http://blogs.cetis.ac.uk/adam/2009/04/14/the-problem-with-evaluating-standards/

Duval, E.: Learning Technology Standardization: Making Sense of it All. ComSIS 1(1) (February 2004), http://www.comsis.org/ComSIS/Volume01/InvitedPapers/ErikDuval.htm

HR-XML Consortium. XML Schema for Human Resource XML (2007), http://www.hr-xml.org/hr-xml/wms/hr-xml-1-org/index.php?id=E00DA03B685A0DD18FB6A08AF0923DE01392

IEEE, IEEE 1484.12.1-2002. Learning Object Metadata (2002)

IEEE, IEEE 1484.20.1/Draft 8 Draft Standard for Learning Technology – Data Model for Reusable Competency Definitions (2007), http://ltsc.ieee.org/wg20

IMS, IMS Learning Design Specification, version 1 (2003), http://www.imsglobal.org/learningdesign/

IMS, IMS Question & Test Interoperability Specification, version 2.1 (2005), http://www.imsglobal.org/question/

ISO, ISO/IEC 19796-1:2005 - Information technology - Learning, education and training - Quality management, assurance and metrics - Part 1: General approach (2005)

Kozlov, D., Pulkkinen, M., Pawlowski, J.: D7.1. ICOPER Reference Model IRM. Conceptual Model. ICOPER ECP 2007 EDU 417007 (2009)

SCORM 2004, Advanced Distributed Learning, 4th edn. (2004), http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20Edition/Documentation.aspx

Tyrväinen, P., Warsta, J., Seppänen, V.: Evolution of Secondary Software Businesses: Understanding Industry Dynamics. In: León, G., Bernardos, A., Casar, J., Kautz, K., DeGross, J. (eds.) IFIP International Federation for Information Processing, Open IT-based innovation: moving towards cooperative IT transfer and knowledge diffusion, vol. 287, pp. 281–401. Springer, Boston (2008)

Tyrväinen, P., Mazhelis, O.: Vertical Software Industry Evolution. In: Analysis of Telecom Operator Software. Physica-Verlag, Heidelberg (2009) (A Springer Company)

Workshop on Global Outsourcing of Software Development

Nazmun Nahar

University of Jyväskylä, Finland

Global outsourcing is a pertinent issue in today's business world. Increasingly, software and non-software organizations from private and public sectors are attempting to develop some or all of their software through global outsourcing in order to enhance innovation, reduce development time and cost, improve quality and productivity, adapt to changing global economic and market conditions, to name a few. Global outsourcing is composed of near shore outsourcing (the software development work is done by the service provider in a nearby country) and offshore outsourcing (the development work is done in a very far away country). Global outsourcing is gaining significant importance, as it can provide various important benefits to both the outsourcer and service provider, if it is successful. However, most of the organizations fail to obtain the expected benefits, as global outsourcing is highly complex and risky. Some global outsourcing projects fail totally.

Topics of interest:

- Different strategies for global outsourcing
- New types of outsourcing business models
- The key challenges associated with each particular global outsourcing business model and their practical solutions
- Global outsourcing business models innovation
- Managing multi-sourcing engagements
- Approaches to develop outsourcing relationships with the service providers of least known or unknown destinations
- Strategies for long-lasting outsourcing relationships
- New types of outsourcing partnership formation
- Requirements analysis in distributed software development
- Virtual teams management in distributed software development
- Intellectual property rights protection
- Mechanisms for conflicts management
- Roles of and impacts on cloud computing in global outsourcing
- Knowledge engineering, discovery, transfer and management in global outsourcing

The workshop provided a forum for interactive discussions of current and emerging topics from the field of global outsourcing of software development. This workshop was mainly designed for academics, scholars and practitioners. High-quality research and practice papers, including theoretical, empirical and analytical studies on global outsourcing of software development from the service provider's and client's viewpoints enriched the workshop.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, p. 223, 2010.

Workshop on Competencies for the Globalization of Information Systems in Knowledge-Intensive Settings

Jan vom Brocke[1], Franz Lehner[2], and Jan M. Pawlowski[3]

[1] University of Liechtenstein
[2] University of Passau, Germany
[3] University of Jyväskylä, Finland

Software development has become a global task for the past decades. Outsourcing and offshoring solutions have been discussed frequently, in both research and in industry. Export is the main driver for a society's and corresponding individual's wealth. As more and more organizations apply outsourcing and offshoring practices, new ways have to be found to compete in this rapidly changing environment. Additionally, the digital divide is becoming a central issue, determining the relation of developed and developing countries.

To deal with both issues, stakeholders need to be enabled to successfully compete on this global marketplace. This includes new key competencies for individuals and organizations, such as intercultural management / communication. Moreover, organizations and individuals need support to acquire and continuously enhance those skills. It is one of the main concerns of the next generation of information systems to take this global dimension into account. Besides other technological innovations, cultural awareness is a key success factor for the future.

The workshop discussed innovative solutions for global information systems, in particular for systems supporting knowledge intensive processes, such collaboration of virtual (project) teams, knowledge management or processes in the service industry. We focused on deriving competencies for stakeholders in this setting. We aimed at discussing state-of-the-art solutions for information systems, IS development as well as methodological aspects encouraging the discourse on this emerging field of research. The key issue were competencies for stakeholders to participate successfully in the process of IS globalization.

Topics of interest:

- Management of distributed services and organizational implications
- Global knowledge management
- Globally distributed education
- Competencies for the globally distributed worker
- Applications and cases for globally distributed work
- Support systems for organizations, teams and individuals
- Awareness and trust for global teams
- Social software for international collaborations

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, p. 224, 2010.

Tutorial: SaaS Business – Theory and Practice

Antero Järvi[1], Tuomas Mäkilä[1], Jussi Nissilä[2], and Jussi Karttunen[2]

[1] University of Turku, Finland
[2] Turku School of Economics, Finland

Description. Software-as-a-Service (SaaS) will affect the business of most IT and software companies, as well as the client organizations' IT function. Yet, software buyers and even software vendors are still somehow unfamiliar with the exact definition of the Software as a Service. Is it a whole new business model, an adjustment to revenue and distribution model, or just a new deployment technology? It appears that SaaS is not just a technical issue, but affects profoundly the way businesses provide and use software.

The goal of this tutorial was to clarify the main concepts of Software-as-a-Service, giving the audience tools to answer an important question from their own standpoint: How will Software-as-a-Service change software business? The issues were presented using two dimensions: (i) technology vs. business perspectives and (ii) software providers vs. buyers' viewpoint.

The tutorial consisted of the following parts:

1. Introduction to the SaaS. We presented SaaS definition and related concepts, like Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).
2. SaaS Value Creation. SaaS creates value for different stakeholders. We discussed the benefits of the SaaS model and analyzed the truth of these claims, e.g.: Lower Total Cost of Ownership (TCO), more flexible IT management, easier entry to new markets, and tapping to competitive business ecosystems.
3. SaaS How-to. In this more practice oriented part we explained how to utilize SaaS model in different business scenarios: How to buy SaaS? How to manage SaaS solutions as a part of companies' information systems? How to launch a SaaS solution? How to do business with SaaS?
4. SaaS in Finland. The attendants of the ICSOB 2010 had the possibility to hear first about the characteristics of the Finnish SaaS market, and challenges and possibilities the SaaS products bring to Finnish companies.

Target audience. Business and development executives of software companies, CIOs of companies interested to utilize the SaaS solutions and researchers with initial interest on the topic.

Tutorial organizers. SaaSify Your Business -research group (Funded by TEKES Verso program) investigates SaaS as a global and Finnish phenomenon. Group consists of researchers with both academic and industrial backgrounds. The tutorial was designed and held by personnel with years of experience in university level teaching based on the current industrial interests.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, p. 225, 2010.

Tutorial: Applying Statistical Research Methods in Software Business

Mikko Rönkkö and Jukka Ylitalo

Aalto University, Finland

Description. Currently the researchers that include software business as one of their topic areas come from very different paradigms, but often build their research on management theories, such as the resource based view. Since statistical analysis is the predominant method in more established management disciplines, it is important for aspiring software business researcher to understand statistics to be able to evaluate the quality of these studies. Additionally, understanding statistical analysis broadens the scope of the research questions a researcher can work with.

The focus of the tutorial was on both the theory and application of statistical analysis in software business research. In contrast to many textbooks on statistical analysis that first present theory and then proceed to application, the tutorial was structured so that we first presented a practical research problem and then explained how statistics can be used to assess the problem. Thus the emphasis of the tutorial was more on practical perspective, and theory came to play only when we discussed how statistical analysis could lead to false conclusions and how to avoid common problems.

The tutorial was consisted of the following parts:

1. Philosophy of statistical research and basics of statistical inference
2. Types of data and measurement
3. Regression analysis
4. Structural equation modeling

Target Audience. The tutorial was designed so that no prerequisite knowledge of statistical inference or statistical analyses was required. During the session we covered all the basics required. The session was aimed for both researchers that have either none or very little background on statistics or for those that have used statistical analysis before, but would need a refresher on the principles and theories of statistical studies. The tutorial included examples with Stata and R. Structural equation modeling was presented with Mplus and R.

Tutorial Organizers. Mikko Rönkkö is a research manager and a doctoral candidate at Aalto University, School of Science and Technology, Software Business Lab. He specializes in statistical research and has currently several papers using statistics as well as papers about statistical methods in review.

Jukka Ylitalo is a researcher at Software Business Lab in Aalto University, School of Science and Technology. He has a master's degree in systems and operations research and has specialized in statistical analysis. He has taught statistical analysis in Helsinki University of Technology for several years.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, p. 226, 2010.

Tutorial: Creating Productive Global Virtual Teams – Developing Effective Collaboration across Cultures and Time Zones

Donald R. Chand

Information & Process Management, Bentley University, USA

Description. In today's global economy, knowledge workers have to collaborate on a daily basis with colleagues around the world. While travel may be preferred for certain tasks, the bulk of global collaboration is based on technologically-mediated communication. Thus, a new challenge for organizations is in developing the skills, capacities, and attitudes that will make working across cultures and time zones successful.

Our research discovered bits and pieces of how innovative managers facilitated the development of camaraderie, trust and knowledge sharing among the members of global work teams separated by distance and culture. We have packaged these bits and pieces of insights and skills into best practices for building and maintaining trust and effective communication in virtual teams, as well as how to build organic communities.

Unlike traditional cross-cultural training, where the focus is cultural differences and what not to do, this tutorial centered on how relationships are built and sustained in virtual spaces, and how using "everyday ethnography" provided the skills to achieve effective collaboration in global virtual teams.

Content of the tutorial

- Global Work Models: i) Bridge team collaborative engagement, ii) Integrated globally-distributed team
- Barriers to Global Work: i) Cultural differences, ii) Limitations of computer-mediated communication, iii) Loss of proximity, iv) Difficulty of building trust, v) Lack of work visibility, vi) Organizational politics
- Building Ethnographic Skills: i) Unpacking the layers of culture impacting global work, ii) Unpacking the limitations of computer-mediated communication, iii) Unpacking organizational politics
- Overcoming Barriers to Global Work: i) Personal relationship building, ii) Virtual community building - learning from the open source work environment

Target audience. The target audiences for this tutorial were professionals who work with team members located overseas, managers who administer the work of offshore vendors, tomorrows' global leaders, and academics interested in global virtual teams.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, p. 227, 2010.

Industrial Session: Software Business Innovation Track

Jyrki Kontio

R & D-Ware

Software industry is going through a major transition everywhere in the world. On one hand the recent recession and increased price competition from emerging economies in Asia have caused changes in where software is being developed. Outsourcing has become much more prevalent - almost a standard part of many software development projects. Customers are more demanding and providers need to develop better applications faster and at lower cost.

On the other hand, significant changes are taking place in technology platforms used and types of applications being developed. Cloud computing, Software as a Service and gaining popularity of smart phones are changing where computing takes place and social media as well as content sharing applications are changing what users do with their devices.

Technological discontinuity, changes in consumer behavior and likely changes in market structures are taking place at the same time. These trends will cause challenges to many companies but they will also create opportunities for innovation for those who dare.

Traditionally software organizations have seen innovation as a technical issue: innovations have been considered technological advances that give companies an edge over competition. While technical innovations are still relevant, increasing share of successful innovations take place outside the technical domain. iPhone was not so much of a technical innovation but a usability innovation and systemic innovation; Salesforce.com's compelling feature is the possibility to buy software functionality as a service; Facebook hooks people by helping them develop their social network online. Innovation in software industry is now driven by business innovations - innovative business models, partnerships, service types, processes and novel combinations of products, services and content.

The Software Business Innovation Track of the International Conference on Software Business was intended for sharing practical industry experiences and innovations in the area of software business. It collected practitioners, industry visionaries and software scholars together to discuss and share insights and experiences in software business innovation.

Our aim was to boost the innovativeness of software organizations everywhere and to gain deeper understanding of new forms of innovations - as well as the innovation process itself - so that we could identify and deliver more innovative services, products and content to the marketplace.

One of the known catalysts for innovation is the interaction of people with different backgrounds and experience. That was exactly the role of the Software Business Innovation Track.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, p. 228, 2010.

Author Index