

**This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.**

Author(s): Ojala, Arto; Paavilainen-Mäntymäki, Eriikka; Su, Ning; Lyytinen, Kalle

Title: Software Complexity and Organization of Firms' Offshoring Activities

Year: 2017

Version:

Please cite the original version:

Ojala, A., Paavilainen-Mäntymäki, E., Su, N., & Lyytinen, K. (2017). Software Complexity and Organization of Firms' Offshoring Activities. In S. Stigberg, J. Karlsen, H. Holone, & C. Linnes (Eds.), SCIS 2017: Nordic Contributions in IS Research (pp. 15-27). Springer. Lecture Notes in Business Information Processing, 294.
https://doi.org/10.1007/978-3-319-64695-4_2

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Software Complexity and Organization of Firms' Offshoring Activities

Arto Ojala¹, Eriikka Paavilainen-Mäntymäki², Ning Su³, and Kalle Lyytinen⁴

¹ University of Jyväskylä, Finland

² University of Turku, Finland

³ Ivey Business School, Canada

⁴ Case Western Reserve University, the USA

Abstract. How does software complexity shape software providers' offshoring tasks, and how do such firms organize their offshoring activity? These questions are important, since the global software development market is growing rapidly, offering new opportunities for software managers and entrepreneurs to distribute their activities geographically. Based on a multi-site case study of 12 software firms, we study connections between software complexity and the offshoring strategies selected. Our findings suggest that software firms select a variety of organizational structures for their offshoring activity, and that the selection is shaped by the complexity of the software in question.

Keywords: Software complexity, Offshoring, Outsourcing, Software firms

1 Introduction

Global software markets are growing rapidly [1, 2], highlighting the growing strategic importance of the software industry in the global economy. At the same time the development and distribution of software has become a global activity, and customers and software suppliers are often located in entirely different geographical locations [3, 4]. For the most part, the geographical distribution of the software poses no great problems, as the software can be delivered to customers at low cost and high speed over the Internet [5, 6]. In contrast, multiple challenges can arise from identifying requirements that are sensitive to the local context, developing software across multiple geographical sites, or providing services to maintain and run the software globally. These challenges are often related to the fact that software complexity¹ [7, 8] increases as the expansion of global operations grows [9].

When software firms specify requirements for multiple foreign customers, they must usually customize the software according to the customers' preferences and local needs, and integrate it with the customers' existing complex IT infrastructure; in accordance with this, the complexity of the software increases [8, 9, 10]. The

¹ In line with Jarke and Lyytinen [8] we refer here to external software complexity.

phenomenon overall is linked to the growing heterogeneity of the client base as the global reach expands [11]. This increases variance in requirements, gives rise to new dependencies between technology components, and generates unexpected interactions in software solutions [8, 12]. Within these interdependencies, the software complexity can vary considerably, depending on the nature and context of the software use, and the software development strategy selected. When a firm seeks to develop software for “mass-markets” it will deliberately seek to keep the complexity lower and to exercise strict control over variance in local adaptations and services, so that the software can suit a maximally wide (and preferably homogenous) customer segment [13, 14, 15]. In contrast, some software firms seek to develop “tailored” software solutions [14, 15, 16] which cater for customers' specific local requirements. This increases the complexity of the software underlying the delivered service [7, 8].

As firms develop software and related services for a growing body of foreign customers the firm's operations expand internationally [3, 11]. The international operations can be carried out using a variety of offshoring strategies, whereby the firm relocates its activities on a global scale [17, 18]. For instance, a firm may offshore-outsource its development activities to third-parties in a foreign country, or alternatively it may offshore-insource development tasks to its own foreign units [19, 20, 21, 22].

The existing offshoring literature has focused on a number of questions, in particular (i) why software organizations offshore their operations, (ii) what activities they should offshore and where, (iii) what related rationale they follow, including the structure of the decision-making process, and (iv) how the organization implements, monitors, and manages offshored activities [3, 17]. There has been less research on how software complexity, plus related operations, aligns with firms' management of offshoring operations. Hence, the aim of the research is to indicate how software complexity [8, 23] shapes the organization of software offshoring. We specifically wished to examine how offshoring firms choose alternative organizational structures as a way to manage the complexity caused by software complexity [24].

2 Literature Review

2.1 Offshoring

Two different terms, namely offshoring and outsourcing have been applied to describe how firms move their tasks and processes to other organizations. Because the usage of the terminology is sometimes vague, these terms are defined in this study as follows: Offshoring refers to moving certain activities to another country, either to a firm's own foreign unit or to a third-party located abroad. By contrast, outsourcing refers to moving some of the firm's activities to another organization, located either in the same country or in a foreign country (see e.g. [25]). In practice, offshoring can be implemented using two different options. The first option is to offshore-outsource some of the firm's activities to third parties, e.g. to foreign distributors. The use of

foreign distributors offers a low-cost access to local knowledge in a foreign country [26]. For instance, software firms can utilize distributors' knowledge of different activities (e.g. localization, customization, technical support, etc.) to better serve their foreign customers [6, 19]. Another option is offshore-insourcing, in other words, the establishment of one's own subsidiary in a foreign country; this will operate as a remote service site for the parent firm [19, 20, 21, 22]. By using a foreign subsidiary, a software firm can use its own personnel to deal with customers and local distributors. In many cases, this requires the recruitment of personnel with relevant knowledge of the target industry and customers in the target country (e.g. [27]). This approach requires more resources and includes higher financial risk, but it also increases market control and lowers transaction-related risks [28]. A firm might apply one of these options solely, or else they can be used in parallel, depending on the software developed (cf. [19]).

The main idea behind offshoring is that a firm should focus on its core competences, and that it should offshore activities that are not related to its core business [20]. Offshoring has traditionally been seen as an activity in which an organization moves parts of its manufacturing or other activities to a low-cost country (e.g. [25, 29]). In the software industry, offshoring has usually been applied to software coding, in which the coding process is moved to a country such as India, which has substantially lower labor costs [19, 30, 31, 32]. However, offshoring is increasingly seen as a strategy to attain qualified personnel [17], technical expertise [5], and worthwhile innovations [33]. For instance, there might be a lack of expertise to develop software for customers' specific needs in a target country [16]. By offshoring software development activities to another country, a firm can get an access to local knowledge and special skills [5, 17, 29, 31]. This can help the firm to develop software that meets the requirements of the local market [11, 19]. Offshoring can also be a decision involving the language and the business culture, in that the offshoring partner's language and its local business culture skills are needed in order to deal with e.g. sales, support, and localization activities (cf. [20, 34]). Altogether, offshoring may help firms to create global markets by increasing the talent pool and innovation capability of the firm [20].

2.2 Software Complexity

In the software industry, the characteristics of software can vary greatly, from highly complex software to software with low complexity [23, 35]. The complexity can involve internal complexity, referring to the type and number of dependencies within the software code, which is largely a function of the size of the code base [23, 35]. Alternatively, it can involve external complexity [8], which relates to the dependencies of the software with its development and use environments, covering for example the scope and rate of change in customers' requirements [8, 10], or changes in the market environment [7, 9]. In the context of this study, we are interested in external complexity of software and how it shapes offshoring strategies.

If software is "tailored" or "customized" according to the prior requirements set by the customers [15, 16], the software complexity increases [8, 23, 35]. This is mainly due to increases in the diversity and dependencies of the software, which in turn

increase the number of versions and functionalities that need to be developed and managed [8, 23]. When a firm develops highly complex software for foreign customers, there may be a growing need to offshore some of the development activities to countries where the customers are located. The development of complex software thus requires close cooperation with customers [13], and this means that knowledge of customers' idiosyncrasies and specific software requirements becomes an important asset [14, 36]. By offshoring labor-intensive development tasks [10] to nearby customers, a firm can get a better understanding of a foreign customer's preferences.

Conversely, software can also be developed according to the aim of keeping the software complexity low. This kind of software involves "packaged" [14] or "mass-market" [15, 16] software which is generally developed for a wide market segment [35]. This is achieved by designing software on a broad basis, using general knowledge of customers' behavior and needs [37]. General requirements can thereafter be merged and grouped, while at the same time seeking to remove or minimize any context-specific elements [13, 14, 37]. In addition, one will seek to have fewer different versions of the software, and attempt to include the same functionalities of the software in each version [23, 35]. This kind of software is easier to install; it can be downloaded from the Internet, or it can be used as a cloud service.

In practice, it can be challenging to formulate a strict division of software firms into high-complexity software developers on the one hand, and low-complexity developers on the other. For this reason, we see it as more fruitful to approach software firms as operating on a continuum, with high-complexity and low-complexity software representing opposite ends. Hoch et al. [16] call firms operating on the middle of this continuum as "enterprise solution firms." Compared to software firms developing highly complex software, such firms use a more standard modular structure in their software. The modular structure makes it possible to reuse and recombine components of the software, thereby decreasing design complexity [23, 36]. The modular structure also makes it easier to customize and localize other components according to different customer requirements [16, 36].

3 Research Methodology

The aim of the study was to identify how software complexity shapes the organization of software offshoring, and consequently the management of organizational complexity. To gain an in-depth understanding of the phenomenon, we applied an exploratory case study method [38]. This method was chosen because it is capable of encompassing empirically rich and detailed data relating to a complex and understudied phenomenon [38, 39, 40].

3.1 Data Sampling

The case firms were selected by using purposeful, theoretical sampling, as recommended by Eisenhardt [41]. We thus deemed it important that the case selection

should fit the research aim of understanding the relationship between software complexity and the offshoring strategy. The firms selected complied with the following criteria: (i) the firms had international operations; (ii) the firms provided software and related services (i.e. they were not pure service providers, as would be the case for software consulting firms); and (iii) the firms in the sample differed in the nature of their software and related service offerings; hence the sample covered a range of firms, from low-complexity to high-complexity software firms.

The firms were divided into three groups according to the complexity of the software they developed. The first group, the developers of low-complexity software, developed software aimed at mass-markets without any need for tailoring or customization. In addition, the software could be installed by the customer on a self-service basis. The second group, the developers of medium-complexity software, developed software that was broadly aimed at business users. The software had a modular software structure, and the modules were customized and/or localized in line with customer requirements. The third group, the developers of high-complexity software, developed software that was tailored according to the individual customer's requirements; thus, it called for close liaison with the customer during the requirement analysis and the installation phases. Table 1 provides detailed information on each case firm.

Table 1. Overview of the case firms

Firm	Year of establishment	Description of software	Target industry	Software complexity	Number of interviews
Firm A	1988	Data-security software	Consumers and diverse industries	Low-complexity software	2
Firm B	2000	Cloud gaming software	Network operators and consumers	Low-complexity software	16
Firm C	1998	Mobile gaming software	Mobile operators and consumers	Low-complexity software	3
Firm D	1995	Firewall software	Diverse industries	Low-complexity software	3
Firm E	1990	Data-in-transit security software	Diverse industries	Medium-complexity software	3
Firm F	1966	3D modeling software	Building and construction industry	Medium-complexity software	5
Firm G	1991	Network analysis software	Mobile phone manufacturers and network operators	Medium-complexity software	4
Firm H	2006	3D modeling software	Furniture industry	Medium-complexity software	10

Firm I	2008	Identity and access management software	Diverse industries	High-complexity software	5
Firm J	1998	Virtual design and modeling software	Mobile phone manufacturers, mobile operators	High-complexity software	3
Firm K	2006	Risk management software	Financial sector	High-complexity software	7
Firm L	1992	Video codec software	Mobile phone manufacturers	High-complexity software	10

3.2 Data Collection

The data for this study were collected within three different projects conducted between 2004 and 2015. The final round of the interviews was conducted in 2014 and 2015, to ensure the continued applicability of the data obtained in previous years. This final round of interviews was undertaken with all the case companies, as a means of checking the comparability of cases. Although the data were collected over a 10-year period, there were only very slight changes in the firms' software offering. Altogether, 71 semi-structured interviews were conducted for this study, with each interview lasting 30–90 minutes. The first author of this study conducted all the interviews. Most of the interviews were conducted face-to-face. Nevertheless, eight telephone interviews and one Skype interview were undertaken because of difficulties in finding a suitable time for a face-to-face interview. Interviews with the CEO or the manager responsible for foreign operations were the main source of information. However, to avoid bias from individual opinions [42, 43, 44], other employees with a variety of positions in the case firms were also interviewed.

All the interviews were recorded and transcribed verbatim, using a word processing program. Thereafter, the complete transcripts were sent back to the interviewees for review. To avoid retrospective bias [43, 45], several different types of secondary data were collected and used to validate the interview data whenever possible. The secondary data included press releases, advertising material, annual reports, industrial reports, and news articles. The interview data were compared with early records. If there were inconsistencies, these were discussed with the persons interviewed.

3.3 Data Analysis

The analysis was conducted in line with the recommendations of Miles and Huberman [46], and it consisted of three parallel activities: (i) data reduction, (ii) data display, and (iii) conclusion-drawing/verification. In the data reduction phase, the

complete transcripts from all the interviews were simplified and summarized by compiling a detailed document covering the history of each case firm. In addition, information from other sources (secondary data) was added to the written case documents. Thereafter, the case firms were categorized into three different groups according to the complexity of their software. The general procedure followed the guidelines of Pettigrew [47], who argued that arranging incoherent aspects in chronological order is essential in understanding the causal links between different events.

In the data display phase, the most important data drawn from the categories were arranged in tables. These tables included quotes from the interview data illustrating the important events in the case firms' international operations. The most relevant quotes from the interviews are included later in this paper (see the Findings section). The tables facilitated comparison of the cases, making it possible to detect similarities and differences between the case firms' international behavior and offshoring strategies.

The phase of conclusion drawing and verification concentrated on identifying the aspects that appeared to have significance for this study. In this phase, the regularities, patterns, explanations, and causalities related to the phenomena were noted. From these, it was possible to develop the constructs and theoretical logic behind the use of different offshoring strategies.

4 Findings

In all cases, the firms kept their core competences (i.e. specific skills and techniques related to core software development) in-house. However, the case analysis indicates that the firms' offshoring focused on four different sets of activities, which were affected by software complexity. These activities were: (i) localization, which was needed in order to make the software appropriate to the target market, (ii) customization according to customer-specific needs, (iii) integration of the software within the customer's IT environment, and (iv) product support. All these activities² further influenced the ways in which the firms organized their international operations. The following subsections provide detailed descriptions of how these activities were organized by the case firms. Here it should be noted once again that the categorization in the table does not represent three totally separate groups; rather, these firms represent points on a continuum, with Firm A having the least complex software and Firm L the most complex software.

4.1 The Offshoring Strategies of Developers of Low-Complexity Software

The developers of low-complexity software (firms A, B, C, and D) conducted their software development in their headquarters. Because their software was aimed at a

² Activities (i)–(ii) were related to software development for foreign customers, whereas activities (iii)–(iv) were supporting services.

wide customer segment, the needs for localization or customization were low. In fact, the decrease in the need for offshoring was substantial, since no localization work was required except in relation to language. In the software provided by firms A and B, language support for various (widely-used) languages was already included in the software within the development process; by contrast, firms C and D localized the language separately for each target country if it was deemed necessary. Firm C conducted its localization activities at headquarters, whereas Firm D offshore-outsourced and/or insourced localization to foreign distributors or subsidiaries. The software that these firms provided was easy to install, and it integrated automatically with a customer's existing IT environment, with no need for external support. Thus, customers were able to handle these activities by themselves. The Sales Manager of Firm A explained this as follows:

“Technically the software is very standard. It includes language support, which is important, especially for consumer markets. The language support is included in all versions of the software...customers can install the software by themselves and all the version updates are delivered automatically.”

The only activity that all the providers of low-complexity software offshored was customer support. Because the software was easy to learn and to understand, requiring no specific knowledge of software development, the distributors were able to handle customer support. Consequently, these firms developing the least complex software organized their global service and support activities so that the distributors took care of customer relationships and support activities, while the subsidiaries had control over the distributors. Thus, the task of the subsidiaries was to supervise the distributors, to give training related to new products and features, and to support existing distributors if they had problems with the software. In some extreme cases, the subsidiaries gave support to end-users, if the problem was something that the distributors were not able to solve.

4.2 The Offshoring Strategies of Developers of Medium-complexity Software

The firms developing medium-complexity software had a somewhat different strategy. Headquarters had the main responsibility for software development, but a substantial proportion of the development and support activities were offshored to foreign distributors and subsidiaries. The software development tasks that were offshored were mainly related to localization and/or customization work for customers in the target countries. This required local knowledge, since the localization included (in addition to the language) aspects such as the inclusion of local standards, regulations, and measurement units for the software. In addition, firms G and H customized software according to the customer's specific needs. The customization was done jointly between a foreign distributor or subsidiary and headquarters. The Executive Vice President of Firm G explained the customization needs of their network analysis software as follows:

“The customization that we do for the customers is related to their [network] controlling system; each customer has slightly different kinds of systems, protocols, and ways to stimulate networks.”

The customers of firms E, F, and G were in fact able to integrate the software, since on the whole they were familiar with the technology and the software. However, Firm H did provide integration services for its customers; this was because the software was targeted at furniture manufacturers and furniture chains, where users were not so familiar with new technologies.

All the providers of medium-complexity software organized their global service and support activities so that the distributors supported their own customers, while the subsidiaries gave support to their direct customers. In more demanding cases, when the distributors or subsidiaries were not able to support their customers, the responsibility for product support moved to headquarters. Contrary to the situation among providers of low-complexity software, there was no foreign subsidiary supervising or controlling the distributor, since the distributors were directly responsible to headquarters for their activities. The Executive Vice President of Firm F explained this as follows:

“Each subsidiary and distributor has exactly the same tasks. Even if our R&D is in Finland, each foreign unit takes care of localization for its customers. They all have an employee who works as a product manager, so we have local product management in each target country. However, we use subsidiaries in the main markets, and distributors in smaller market areas.”

4.3 The Offshoring Strategies of Developers of High-complexity Software

The providers of high-complexity software (firms I, J, K, and L) used a strategy by which headquarters had the main responsibility for software development. This was mainly because the complexity of the software made offshore-outsourcing and insourcing more difficult. Even if these firms used distributors and/or subsidiaries for marketing and sales, the high complexity of the software decreased the possibilities for offshore development or for other support activities.

All the providers of high-complexity software localized and customized their software for customers. The localization work was truly customer-centric. It included localization of different kinds of reports, user interfaces, and so on, in contrast to other types of software providers, who mainly localized language, or who made some industry-specific changes to the software. In the case firms (I, J, K, and L), localization and customization were conducted at headquarters, since the processes required knowledge that was available only in R&D units. This kind of knowledge was related to, for example, the industry platforms in which the software would be integrated (Firm L), the customers' R&D processes (Firm J), and the specific function of the software (firms I and K). The Vice President of Firm K explained this as follows, commenting in relation to the firm's finance and risk management software:

“[Related to localization and customization] Even though banking and finance activities are similar from day to day, each bank is different, and they all have different kinds of systems, different kinds of processes, and their own requirements that they want to follow when using our software.”

In all the cases in this group, technical support personnel from headquarters carried out the integration of the software with the customer’s existing IT environment. This process was, in many cases, time consuming and required a thorough knowledge of the customer’s IT environment. Because of the high complexity of the software, global support activities were organized so that headquarters had the main responsibility for product support. However, in less complex situations, subsidiaries or distributors were used to provide first-line support.

5 Discussion

According to our findings, the firms developing the less complex software offshored only a limited portion of their activities, because the software was easy to understand, install, and use with no need for localization or customization [11, 14, 16]. These features decreased the overall complexity of the software [7, 8, 23]. The only activity they typically offshored was product support. By using this strategy, the firms were able to focus on their core competence, i.e. product development in a single location.

In the case of firms E, F, G, and H, the software complexity increased as the customers indicated more requirements for localization and customization [23, 36]. Their demands generated variation in different versions of the software [8], each targeted for a certain market or customer segment. The increased complexity was dealt with by offshoring localization and customization activities to foreign subsidiaries or distributors. These units were better able to manage development tasks for local customers, as they had the local knowledge needed for the related development tasks (cf. [11]).

The firms developing the most complex software (I, J, K, and L), tailored their software for each customer separately. This required in-depth knowledge of the customers’ business processes and of the system environment in which the software was integrated [11, 14, 36]. This specialized knowledge was available only at headquarters, and it was not easily offshored to a foreign unit. Hence, the high complexity of the software decreased the possibilities to offshore the firms’ activities. Because of this, offshored activities were limited again to product-related support.

Altogether, the findings indicate that a low level of complexity decreases offshoring needs, whereas a high level of complexity inhibits offshoring possibilities. Firms developing medium-level complex software do the most offshoring, since (i) most of their localization and customization activities require local knowledge from the target countries, but (ii) the level of complexity of the software is not so high as to inhibit offshoring. Thus, the connection between software complexity and the need for offshoring is not linear.

Taken as a whole, the findings indicated that the level of software complexity had a significant effect on how the software firms organized their foreign operations.

Figure 1 illustrates three different service and support models emerging from our analysis demonstrating how software complexity affects the organization of offshoring activities. The models are abstractions arrived at by analysis of the interviews. Within the figure, solid arrows represent strong support/control, and dotted arrows demonstrate weak support/control. In the first model, depicting providers of less complex software, firm headquarters focuses on product development, while subsidiaries are used to support and supervise existing distributors. The distributors take care of product support activities when there is a need for direct contact with the customers. Hence, direct control and support is organized vertically. In the second model, which is followed by the providers of medium-complex software, the subsidiaries and distributors have equal tasks, and they comprise the main actors in dealing with customers (solid arrows). As discussed above, these tasks are related to localization/customization, integration, installation, and customer support. In this model, the firm's headquarters supports/controls subsidiaries and distributors (solid arrows); however, headquarters has only indirect contact with the actual end-users, and provides second-line support for users (dotted arrow). In the third model, followed by the providers of most complex software, headquarters takes a substantial role in controlling development and use. It has direct contact with the customers (solid arrow); in this case, the subsidiaries and distributors give only indirect support (dotted arrows).

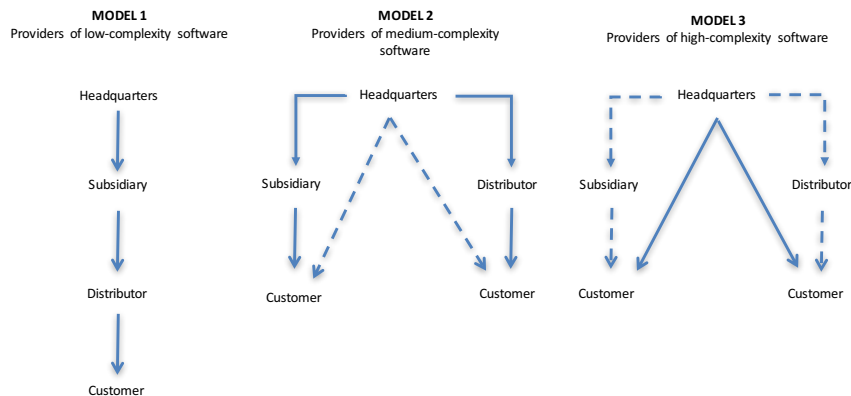


Fig. 1. Models for offshoring international service and support activities, based on software complexity.

6 Conclusions

This study examined software firms' offshoring strategies and the extent to which they are influenced by software complexity. Our results contribute to IS literature in several ways. First of all, the study shows how software complexity [7, 8, 23] shapes offshoring strategies. Low software complexity decreases offshoring needs, whereas

high software complexity limits offshoring possibilities. However, the firms that do the most offshoring tend to be those that develop software which sits between the two extremes of the continuum. This highlights the fact that the connection between software complexity and the need for more complex offshoring solutions is not linear.

Secondly, the findings contribute to an understanding of the interplay between software complexity [8, 23] and software firms' approaches to managing and coordinating global software development and distribution activities [3, 6, 48]. We show that software firms can adjust their international operations according to the underlying software complexity in their efforts to better serve their foreign customers. Even though previous literature has shown that variations in software complexity are related to different approaches in managing requirements [7, 8] or in developing software [23, 35], less attention has been paid to the connection between software complexity and the strategies related to the offshoring and organizing foreign activities after the development phase.

References

1. Gartner, 2014. Gartner Says Worldwide Software Market Grew 4.8 Percent in 2013. Press release. Ref. www.gartner.com/newsroom/id/2696317
2. Peltonen, J., Rönkkö, M., & Mutanen, O.-P. (2014). Growth Forum 2013 Summary Report. Helsinki: Ohjelmistoyrittäjät ry. Ref. <http://www.kasvufoorumi.fi/wp-content/uploads/2013/12/Growth-forum-report-2013.pdf>
3. Su, N. 2013. Internationalization Strategies of Chinese IT Service Suppliers. *MIS Quarterly*, 37(1): 175-200.
4. Su, N. 2015. Cultural Sensemaking in Offshore Information Technology Service Suppliers. *MIS Quarterly*, 39(4), 959-983.
5. Metters, R. 2008. A typology of offshoring and outsourcing in electronically transmitted services. *Journal of Operations Management*, 26(2), 198-211.
6. Ojala, A. & Tyrväinen, P. 2006. Business models and market entry mode choice of small software firms. *Journal of International Entrepreneurship*, 4(2-3), 69-81.
7. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J. & Robinson, W. 2011. The brave new world of design requirements. *Information Systems*, 36(7), 992-1008.
8. Jarke, M. & Lyytinen, K. 2015. Complexity of Systems Evolution: Requirements Engineering Perspective. *ACM Transactions on Management Information Systems*, 5(3), 11:1-11:7.
9. Sambamurthy, V. & Zmud, R.W. 2000. The Organizing Logic for an Enterprise's IT Activities in the Digital Era—A Prognosis of Practice and a Call for Research. *Information Systems Research*, 11(2), 105-114.
10. Pressman, R.S. 2004. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Higher Education.
11. Wareham, J., Fox, P.B. & Cano Giner, J.L. 2014. Technology Ecosystem Governance. *Organization Science*, 25(4), 1195-1215.
12. Hanseth, O. & Lyytinen, K. 2010. Design theory for dynamic complexity in information infrastructures: the case of building internet. *Journal of Information Technology*, 25(1) 1-19.
13. Alajoutsijärvi, K., Mannermaa, K., & Tikkanen, H. 2000. Customer Relationships and the small software firm: A framework for understanding challenges faced in marketing. *Information & Management*, 37(3), 153-159.

14. Nambisan, S. 2001. Why Service Business Are Not Product Businesses. *MIT Sloan Management Review*, 42(4), 72-80.
15. Segelod, E. & Jordan, G. 2004. The use and importance of external sources of knowledge in the software development process. *R&D Management*, 34(3), 239-252.
16. Hoch, D.J., Roeding, C.R., Purkert, G., Lindner, S.K., & Müller, R. 2000. *Secrets of Software Success*. Boston: Harvard Business School Press.
17. Levin, A.Y., Massini, S., & Peeters, C. 2009. Why Are Companies Offshoring Innovation? The Emerging Global Race for Talent. *Journal of International Business Studies*, 40(6), 901-925.
18. Manning, S., Massini, S. & Lewin, Y.A. 2008. A Dynamic Perspective on Next-Generation Offshoring: The Global Sourcing of Science and Engineering Talent. *Academy of Management Perspectives*, 22(3), 35-54.
19. Aspray, W., Mayadas, F. & Vardi, M.Y. 2006. Globalization and offshoring of software. Report of the ACM Job Migration Task Force, Association for Computing Machinery.
20. Davis, G., Ein-Dor, P., King, W. & Torkzadeh, R. 2006. IT OFFSHORING: History, Prospects and Challenges. *Journal of the Association for Information Systems*, 7(11), 770-795.
21. Jarvenpaa, S.L. & Mao, J-Y. 2008. Operational capabilities development in mediated offshore software services models. *Journal of Information Technology*, 23(1), 3-17.
22. Prikladnicki, R. & Audy, J.L.N. 2009. Comparing offshore outsourcing and the internal offshoring of software development: A qualitative study. *AMCIS 2009 Proceedings*.
23. Mens, T. 2012. On the Complexity of Software Systems. *Computer*, 45(8), 79-81.
24. Anderson, P. 1999. Perspective: Complexity Theory and Organization Science. *Organization Science*, 10(3), 216-232.
25. Carmel, E. & Tjia, P. 2005. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press.
26. Williams, C. 2011. Client–vendor knowledge transfer in IS offshore outsourcing: insights from a survey of Indian software engineers. *Information Systems Journal*, 21(4), 335-356.
27. Luo, Y. 2000. Dynamic Capabilities in International Expansion. *Journal of World Business*, 35(4), 355-378.
28. Lu, J.W. & Beamish, P.W. 2001. The internationalization and performance of SMEs. *Strategic Management Journal*, 22(6-7), 565-586.
29. Oshri, I., Kotlarsky, J., & Willcocks, L.P. 2015. *The Handbook of Global Outsourcing and Offshoring*. 3rd Edition, Palgrave Macmillan.
30. Carmel, E. & Nicholson, B. 2005. Small Firms and Offshore Software Outsourcing: High Transaction Costs and Their Mitigation. *Journal of Global Information Management*, 13(3), 33-54.
31. Lacity, M.C., Khan, S., Yan, A. & Willcocks, L.P. 2010. A review of the IT outsourcing empirical literature and future research directions. *Journal of Information Technology*, 25(4), 395-433.
32. Lacity, M.C., Solomon, S., Yan, A. & Willcocks, L.P. 2011. Business process outsourcing studies: a critical review and research directions. *Journal of Information Technology*, 26(4), 221-258.
33. Su, N., Levina, N., & Ross, J.W. 2016. The Long-Tail Strategy of IT Outsourcing. *MIT Sloan Management Review*, 57(2), 81-89.
34. Khan, S.U., Niazi, M. & Rashid, A. 2011. Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. *Information and Software Technology*, 53(7), 693-706.
35. Banker, R.D., Davis, G.B., & Slaughter, S.A. 1998. Software Development Practices, Software Complexity, and Software Maintenance Performance: A Field Study. *Management Science*, 44(4), 433-450.

36. Tauterat, T., Mautsch, L.O., & Herzwurmm G. 2012. Strategic Success Factors in Customization of Business Software. *Lecture Notes in Business Information Processing*, 114, 267-272.
37. Kabbedijk, J., Brinkkemper, S., Jansen, S. & van der Veldt, B. 2009. Customer involvement in requirements management: lessons from mass market software development. 17th IEEE International Requirements Engineering Conference, 281-286.
38. Yin, R.K. 2009. *Case study research: Design and methods*. CA: SAGE Publications.
39. Benbasat, I., Goldstein, K.K. & Mead, M. 1987. The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, 11(3), 369-386.
40. Edmondson, A.C. & McManus, S.E. 2007. Methodological fit in management field research. *Academy of Management Review*, 32(4), 1155-1179.
41. Eisenhardt, K.M. 1989. Building theories from case study research. *Academy of Management Review*, 14(4), 532-550.
42. Eisenhardt, K.M. & Graebner, M.E. 2007. Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, 50(1), 25-32.
43. Huber, G.P. & Power, D.J. 1985. Retrospective Reports of Strategic-level Managers: Guidelines for Increasing their Accuracy. *Strategic Management Journal*, 6, 171-180.
44. Myers, M.D. & Newman, M. 2007. The qualitative interview in IS research: Examining the craft. *Information and Organization* 17(1), 2-26.
45. Miller, C.C., Cardinal, L.B. & Glick, W.H. 1997. Retrospective reports in organizational research: A reexamination of recent evidence. *Academy of Management Journal* 40(1), 189-204.
46. Miles, M.B. & Huberman, A.M. 1994. *Qualitative Data Analysis: An Expanded Sourcebook*. California: Sage Publications.
47. Pettigrew, A.M. 1990. Longitudinal Field Research on Change: Theory and Practice. *Organization Science*, 1(3), 267-292.
48. Ojala, A. & Tyrväinen, P. 2007. 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.