

COPING WITH CULTURAL AND MATURITY INEQUALITY IN OFFSHORE OUTSOURCING: IS MINIMIZING INTERACTION THE SOLUTION?

Hertzum, Morten, Roskilde University, Universitetsvej 1, Building 43.2, 4000 Roskilde, Denmark, mhz@ruc.dk

Pries-Heje, Jan, Roskilde University, Universitetsvej 1, Building 43.2, 4000 Roskilde, Denmark, janph@ruc.dk

Abstract

Many companies consider and undertake outsourcing of their software-development activities. Often information systems development is outsourced to vendors in different cultures or with a different level of software-process maturity. Recommendations for managing such outsourcing arrangements typically involve that client and vendor should increase interaction, learn more about the culture of the other part, communicate more, form partnerships, and the like. We have studied a client that did the opposite with a successful outcome. Based on a case study we lay out the story of how interaction between client and vendor on purpose was minimized. What mechanisms were used? What worked and what did not? We conclude that minimizing interaction can be a viable strategy to follow when clients face large cultural and maturity inequality in offshoring their software-development activities.

Keywords: Offshore outsourcing, Culture, Maturity, Minimal-interaction strategy, Extra costs.

1 INTRODUCTION

Information systems (IS) outsourcing can be defined as the practice where an organization purchase goods or services “that were previously provided internally” (Lacity et al. 1993). Many companies consider and undertake outsourcing. It has been estimated that global software outsourcing is going to reach \$260 billion by 2009 (Vitharana et al. 2007). The documented benefits of IS outsourcing have been reduction of time and cost, increased quality, improved business performance, and increased ability to concentrate on the core of the business (McFarlan et al. 2004). Recently, outsourcing to another country – so-called offshoring – has become popular because large numbers of technically skilled developers are being educated in countries such as India where wages are low compared to Western Europe and North America, in which skilled developers are in short supply. Offshore outsourcing is defined as “the subcontracting of an activity by a client organization to an independent service provider working from an overseas destination” (Vlaar et al. 2008) or simply as “inter-country outsourcing” (King et al. 2008).

While geographic distance is, thus, a defining characteristic of offshore outsourcing, the challenge is not geography as such but overcoming communication bottlenecks, knowledge asymmetries, psychological dissociation, and socio-cultural differences. General recommendations to manage collaboration at a distance include establishing common ground at the outset and distributing tasks such that only loosely coupled tasks are allocated to different sites (Olson et al. 2000). For offshore outsourcing two specific challenges are that the client and the vendor often have very different cultures and are at very different levels of maturity with respect to their software development processes. To handle these challenges it is frequently recommend to increase interaction, learn more about the culture of the other part, communicate more, form partnerships, or the like (Bhat et al. 2006; Hendry 1995; Krishna et al. 2004).

In this study, we analyse a company that did the opposite, namely minimized interaction between client and vendor, and with a successful outcome, in the sense that they just decided to renew their contract with a large Indian vendor; on the same terms as for the last four years. This approach is contrary to the prevalent recommendations in the literature, and we consider it of interest to study this anomaly. In doing so our research question is: Is minimal interaction between client and vendor a way to overcome cultural and maturity inequality in offshore outsourcing?

The next section sets the context for our study by describing related work on offshore outsourcing, culture and maturity. Section 3 accounts for the method we used in our empirical study, and Section 4 introduces the client and vendor organizations. Section 5 analyses the client’s minimal interaction approach to offshoring and identifies the mechanisms established to succeed with this approach. Section 6 discusses how the minimal interaction approach has affected the client’s software-development activities and at what costs.

2 OFFSHORE OUTSOURCING

Carmel and Agarwal (2002) propose four stages of offshore sourcing of information systems development. An organization is at the first stage if it is still an offshore bystander. While there may be a variety of reasons for remaining at this stage, more and more organizations choose to proceed to one of the subsequent stages. Organizations at the second stage have started experimenting with offshoring, for example through pilot projects, and often their motivation for offshoring is unavailability of onshore developers rather than a proactive focus on offshore possibilities. Loaded wages for skilled Indian developers at 30-50% of onshore wages in Western Europe and North America is an important motivator for offshoring, but rarely realized at this stage. One reason for this is that the stage is transitional; when some experience has been gained and cost savings start to occur, organizations move to the next stage. At the third stage organizations are characterized by a proactive cost focus. A typical recommendation at this stage is to restrict offshoring to non-core and structured

tasks, such as construction based on detailed specifications (e.g., Cusick et al. 2006). Often, onshore project managers receive targets specifying that a certain percentage of the developer hours on their projects should be offshore developer hours. At least one study finds that onshore staff tended to perceive offshore developers as cheap worker-bees who could be ordered around (Levina et al. 2008). At the fourth stage organizations no longer view offshoring as simply a source of low cost work but have adopted a proactive strategic focus. The strategic objectives pursued at this stage include access to new markets and offshoring of entire projects from requirements through to support, also of projects involving innovation and new product development. The best offshore practices described by Bhat et al. (2006) appear to be directed solely at this fourth stage and focus on achieving shared goals, culture, processes, and responsibilities for client and vendor.

While the vast majority of research approaches offshoring from the client's point of view, some studies do investigate vendors' views of offshoring (e.g., Bhat et al. 2006; Oza et al. 2005). A theme common to client and vendor research is asymmetries in knowledge and experience. These asymmetries concern, among other things, the business domain, typically with the client in possession of business knowledge and the vendor less so (e.g., Levina et al. 2008), and development processes, typically with vendors that have more structured development processes than clients (e.g., Oza et al. 2005). Technical knowledge exhibits another type of asymmetry in that the vendor typically employs a large pool of technically skilled developers while this resource is scarcer onshore.

The top risks associated with offshore outsourcing include lack of top management commitment, miscommunication of requirements, inadequate user involvement, failure to manage end user expectations, and poor change control (Iacovou et al. 2008). These risks do not appear to be specific to offshoring but rather to apply to information systems development in general (cf. Schmidt et al. 2001). Some of the top risks associated with offshoring are, however, specific to offshoring, including language barriers, lack of offshore project management know-how by the client, lack of technical or business know-how by the offshore team, and failure to consider all costs (Iacovou et al. 2008). Many client organizations have expected cost reductions from their offshoring arrangements due to the lower offshore wages but have not fully understood all the costs involved in outsourcing (Barthélemy 2001). Dibbern et al. (2008) identify specification costs, design costs, knowledge-transfer costs, coordination costs, and control costs as the five main categories in which clients face extra costs when projects are offshored. The five categories of extra costs relate to the less effective possibilities for communication between client and vendor and the resulting degradation in their mutual awareness of each other's work and day-to-day activities.

2.1 The role of culture

Hofstede (2001) defines culture as "the collective programming of the mind that distinguishes the members of one group or category of people from another". Hofstede's work shows that even within a single organization, national groups of employee exhibit different cultural characteristics. These characteristics have been specified in terms of five cultural dimensions: power distance, uncertainty avoidance, individualism/collectivism, masculinity/femininity, and long-/short-term orientation (Hofstede 2001). It appears that managers in organizations chronically underestimate the magnitude and importance of cultural differences (Hofstede et al. 2005).

Prior surveys indicate that national culture is a leading cause of problems in the offshoring of services (Metters 2008). Metters (2008), for example, refers to a survey where 60 executives involved in offshoring information technology (IT) services cited "cultural differences" as the most important problem in relation to offshoring. Also, Terdiman and Berg's (2001) framework for evaluating a potential offshoring country has "cultural issues" as one of three main areas. The interest in nearshoring is another indication that similar cultural characteristics, such as ways of doing business, are considered important to outsourcing decisions (Carmel et al. 2007). In contrast to nearshoring, offshoring typically implies profound cultural differences between client and vendor. Never the less one can find countries that may be remote but culturally close such as Great Britain and Australia.

In the concrete Denmark and India (the client and vendor countries of this study) differ along several of Hofstede's cultural dimensions but in particular with respect to power distance – defined as the extent to which the less powerful members of institutions and organizations expect and accept that power is distributed unequally. In India power distance is very high, in Denmark very low (Hofstede 2001). Consequently, in the Danish business culture, rank and title are less important than in India where hierarchical forms of behaviour are expected. In Denmark subordinates are expected to speak up and offer suggestions; in India superiors and seniors enjoy more respect, and decisions tend to be top-down. This affects, for example, communication styles and ownership of results (Schomer 2006).

Recommendations for handling cultural differences in offshoring arrangements include facilitated communication sessions (Dubé et al. 2001), building consensus on norms for meetings and deadlines (Paré et al. 1999), and other efforts to establish a shared culture (Bhat et al. 2006).

2.2 Maturity and software process improvement

Maturity models are used to improve the performance of organisations, processes, technology, and people. The Capability Maturity Model (CMM) is a framework characterizing a five-step path for software process improvement (Paulk 1995). The path describes key processes and goals at each of the five levels. An organization has to meet the goals at one level to reach the next. For example, to go from the basic level 1 where behaviour is characterized by being ad-hoc and intuitive to level 2, you need to achieve the goals incorporated in six key process areas: requirements management, sub-contractor management, project planning, project tracking, quality assurance, and configuration management. CMM became so popular that a large number of other models using the same five-step path were invented, including People-CMM, Integrated Product Development CMM, Systems Acquisition CMM, and Testing Maturity Model. Finally a large number of the CMM models were summoned in CMM-integrated – or just CMMI (Ahern et al. 2003; Chrissis et al. 2003).

In relation to offshoring, it is noteworthy that India has embraced CMMI. Four countries in the world have used the CMMI model extensively: Australia, India, Japan, and the US (IndiaExpressGroup 2003). The highest level of maturity is level 5, and a few years ago 80% of all the companies in the world at level 5 were from India (Mohnot 2003).

In Denmark only one or two companies have reached level 5 (cf. Pries-Heje et al. 2008), and the majority of Danish companies are at level 1. Thus, when Indian and Danish organizations enter into offshoring arrangements there may be huge maturity inequalities between them.

3 METHOD

Our empirical study is a case study based on interviews in one Danish organization. The case study is single-case and embedded (type 2) according to the typology by Yin (1994, p. 39). We have not obtained data from the vendor. Thus, the empirical data are restricted to a client-side perspective on offshoring. One of the authors has worked with the organization since 2003, and has carried out several assessments and training sessions in the organization from 2003-2007. We believe that it is fair to claim that this author has extensive knowledge on how software development is carried out in the organization.

Concerning offshore outsourcing, however, the case study reported here took place in 2008 and was carried out by both authors. We conducted an initial interview with three staff involved in the client's offshoring at the managerial level. During this interview we got an overview of the client's offshore-outsourcing history and identified seven persons for in-depth interviews. The interviewees comprised persons involved in or responsible for (1) the start-up of the offshoring activities, (2) the entire course of offshoring activities, (3) the offshoring contract, (4) the offshore development centre, (5) concrete offshoring projects and certification of offshore staff, and (6) improvement of the client's development processes; and (7) an offshore coordinator recently returned from a long-term placement at the vendor.

The seven in-depth interviews were loosely structured by an interview guide addressing:

- The offshoring arrangement between client and vendor, and its evolution
- Client-vendor interactions at the levels of the offshore agreement, projects, and individual staff
- The creation of a project identity in projects involving offshoring
- The coordination of such projects
- Initiatives undertaken to facilitate offshoring and the lessons learned from them
- Issues relating to differences in the cultural background of onshore and offshore staff

In addition, the interviewees were asked to reflect upon the factors critical to the client's experience with offshoring. This part of the interviews was based on a walkthrough of Iacovou and Nakatsu's (2008) ten-item list of top offshoring risk factors.

The interviews were conducted in meeting rooms at the client's premises, except one interview which for practical reasons was conducted at the authors' university. The initial interview was documented in written notes; the in-depth interviews were audio-recorded, and subsequently an extensive written record of the main points was produced. The written record included selected quotes, but the interviews, which lasted 1-2 hours, were not transcribed verbatim. The interviews were analyzed by reading through the written records several times, noting issues stated in individual interviews and patterns emerging across interviews. These issues and patterns were then grouped into themes, resulting in the analysis presented in this paper.

4 THE EMPIRICAL SETTING

The client is a Danish organization in the financial sector. It has approximately 850 employees, some 450 of which are directly involved in the development of IT systems. The client has for 40 years developed and hosted services for the Danish banks, particularly with regard to payment solutions. The financial sector is characterized by high volumes of safety-critical transactions and, thereby, a need for efficient and secure systems. Moreover, the financial sector is dynamic with changes in national and, increasingly, international legislation forcing revisions of systems, with mergers and acquisitions among banks necessitating integration or redesign of systems, and with considerable competition among providers of financial services creating a continual pressure for the development of new services.

After an early, unsuccessful attempt with outsourcing in the late 1980s, the client refrained from further attempts during the next decade. In 2000 the client started offshoring to India, and in 2002 they started working with their current vendor. The vendor is an Indian software-development organization, which employs over 8000 software developers and has years of experience as an offshore-outsourcing vendor of financial and other services. While the vendor has been certified at CMM level five since 2002 and CMMI since 2006, the interviewees estimate that the client is at CMM level 1 or 2. The collaboration between the client and the vendor has been going on for six years, and it has been decided to renew the current contract when it expires by the end of 2008.

The client's rationale for entering into an offshoring relationship was to increase its capacity. This is stated by several interviewees, who also state that thanks to this increase in capacity the client has been able to carry through projects it would otherwise have been unable to take on.

5 OFFSHORE OUTSOURCING WITH MINIMAL INTERACTION

When setting up an offshore arrangement some interaction is required to negotiate the terms of cooperation, write a contract, and start working together (Willcocks et al. 2006). In the phase following – the operating phase it has been called (Cullen et al. 2006) – there also needs to be some interaction; the salient question is: how much? At one end, we can talk about minimal interaction. That is, just enough to make things work. One could say that minimal interaction is about paying as little transaction cost (Williamson 1979) as possible. Minimal interaction also entails that as few changes as

possible are made in the client's and vendor's internal processes. It should be noted that when one reduces transaction costs the remaining interaction will look as being more intensive at particular contact points. At the other end, we can maximize interaction trying to come as close together as possible. This may involve more communication, more learning about the culture of the other part, trying to level or balance maturity, forming a partnership, and maybe even blur the distinction between a client and a vendor.

5.1 Keeping distance

An illustrative example of minimal interaction is project A in the Danish organisation we are studying. This was the first project the client offshored to the vendor. The project, which lasted three years, consisted of converting an existing system to another platform. That is, the existing system in itself comprised a complete and, by definition, fully accurate specification. Such a task involves little analysis and design compared to programming. This characteristic of project A was the main reason it was chosen for offshoring, and it implied that the client could specify the project very accurately and very easily. This made the project very suited for the client's minimal interaction strategy because minimal interaction could be attained at low risk.

Also, project A was only economically feasible for the client if it could be offshored. The project showed that the vendor had the technical knowledge required to make the conversion. Very few errors were detected during testing, and some of them turned out to be errors in the "specification"; that is, hitherto unnoticed errors in the old system.

After having completed project A, it was decided to set up a more permanent relationship between the client in Denmark and the vendor in India. It was at this point that the idea of minimized interaction really came into play. A manager says: *"The point of departure is that they are vendors. They are not employees. They are a vendor like an external company we cooperate with. The idea was to establish it out there [i.e., at the vendor], so that they can maintain their culture and keep working the way they are used to; and people here [i.e., at the client] work in their way. Actually, reducing the need for intercultural interaction to as little as possible was part of what I was trying to accomplish."*

5.2 Exchanging people

The client has made use of two mechanisms for exchanging people to accomplish offshoring projects while maintaining minimal interaction. Both mechanisms involve intensive interaction but for selected people and selected periods of time. First, offshore developers have been on placements at the client to work with their onshore colleagues. This mimics how new onshore IT developers acquire business knowledge, but in addition to improving the offshore developers' business knowledge it has also facilitated the general relationship between onshore and offshore staff. However, the placements require that onshore staff has the necessary time for communicating and interacting with the offshore developers; and the placements temporarily cancel the economical effect of offshoring because the offshore developers get onshore wages while they are onshore.

Second, the client has placed an offshore coordinator at the vendor. The few onshore employees who have had this position have been at the vendor on long-term placements. The offshore coordinator has a mediating role involving frequent phone contact with client staff, with whom they are well connected, and participation in project meetings with vendor staff. Collaboration between the offshore coordinator and vendor staff is face-to-face, thus avoiding the limitations of communication and collaboration at a distance and providing more opportunities for becoming aware of cultural and maturity issues in need of attention. Periodic onshore visits have been necessary for the offshore coordinators to maintain their network among the client staff. Moreover, the vendor may occasionally feel that the presence of the offshore coordinators transgresses the client-vendor boundary.

Apart from these two mechanisms, an effort has been made to motivate offshore developers to work for the client for a longer period of time. In the Indian offshore-outsourcing industry it is customary

that IT developers move into the management ranks after only a couple of years as developers. This is very different from the career path of Danish developers, who often work a decade or more as developers in the same business area. This cultural difference threatens the client's minimal interaction strategy, because the continuous renewal of offshore developers implies that most of them will have insufficient business knowledge. The client has therefore aimed to make their relationship with the vendor sufficiently interesting for offshore developers to make it attractive for them to stay for a long period of time. The onshore placements of offshore developers have been effective in this regard.

5.3 Exchanging knowledge

The client has set up business courses at the vendor. The courses have been run by visiting onshore staff and by some of the offshore developers that have been on onshore placements. In some areas of the client's business, the courses form an entire certification program, which ensures that offshore developers have a basic understanding of the business area for which they develop systems. While the offshore developers are not at a level of business understanding comparable to the onshore staff, their improving business understanding increases their ability to work autonomously and decreases the amount of interaction they need to have with the client.

The courses and certification programs are an attempt to exchange knowledge in concentrated packages and to several offshore developers at a time. This is considered preferable to frequent ad hoc interactions, which are complicated by the geographical separation. Extensive ad hoc interaction is also seen as time consuming, especially to the client, and therefore as contrary to the intention of shifting work from the client to the vendor. A manifestation of this is that a single point of entry has been enforced when offshore staff needs to communicate with onshore quality-assessment staff. This has been decided to protect the majority of the onshore staff from becoming engaged in too many, time-consuming communications. In this case it appears that the client has been more concerned with not making offshoring unpopular among its onshore staff than with providing the offshore staff with access to needed knowledge and the opportunity to gradually build a network.

Restricting access to needed knowledge in order to minimize interaction creates problems because it prolongs the period during which business knowledge is unevenly distributed between client and vendor. As an example, the present assessment of project B – a large, ongoing offshore project – is that it has been hard to strike the proper balance between technical and business development. The project requires both technical and business knowledge, but the uneven distribution of knowledge entails that the vendor, which is involved in the project with a massive 300-400 person years, often has only the technical knowledge. In working on project B, the vendor proceeds on the basis of its technical knowledge and remains unaware of some of the issues that might warrant business considerations. The client is at too great a distance from the vendor's work to spot such issues and has not been able to specify them up front. As a consequence opportunities for business considerations are unintentionally bypassed, and project B becomes to an excessive extent about technically reprogramming a system.

5.4 Developing software in two places with minimized interaction

Today the client uses the vendor in India on a regular basis. Project B provides an interesting example. This project consists of converting a standalone system into a service available to many systems. The client considers such moves toward a more service-oriented architecture crucial to enable reuse across systems and to enable flexible assignment of the development of individual services to onshore or offshore groups with the ability and capacity to take them on. A main reason for completely reprogramming the system is, however, that the existing system has evolved over a long period, and due to extensive changes in the staff working on the system nobody any longer has a comprehensive overview of the programming code. Moreover, the documentation is not trusted to be up-to-date. Thus, it has become exceedingly difficult and costly to make revisions of the system (cf. Naur 1985). The capacity and lower price of the offshore vendor compared to onshore developers make it feasible to solve these difficulties by reprogramming the system from scratch.

However, turning a system into a service is not merely a technical task but also requires considerable business understanding to know the applications to which a service is relevant and the differences in what these applications require from the service. To overcome this challenge the client decided to apply use cases (Cockburn 2000; Jacobson et al. 1992). At first, some of the offshore developers that had been on placements at the client, but had returned to India, were asked to lead the writing of use cases in India. It was agreed to use a writing style with four abstraction levels with the first being mostly business oriented and the fourth very technically oriented. But when the results came in, the more business oriented use-case levels just consisted of pointers to lower levels, and the client discovered that it took way too much effort to review the very detailed technical use cases at the fourth level. Thus, that way of dividing work did not minimize interaction. In a second round, business staff at the client was taught to write use cases. These upper-level use cases were then given to the vendor who wrote the technical levels. This proved to minimize interaction much better.

6 DISCUSSION

Cultural differences between client and vendor are an inherent characteristic of offshore outsourcing. For offshoring to India it is also common that the vendor's development processes are at a higher maturity level than the client's development processes (Levina et al. 2008; Vlaar et al. 2008). The case investigated in this study concerns whether such inequalities can be handled by minimizing the interaction between client and vendor.

6.1 Costs and challenges of minimal interaction

In the section above we mentioned project A as an example of minimizing interaction. While the project was successful in the sense that the offshoring arrangement produced a high-quality system, it was restricted in the sense that only a modest part of the activities of a full project were performed by the vendor. The entire project A was offshored to the vendor, but project A was special in the sense that it consisted almost entirely of programming. In this sense project B is a better example of the client's minimizing interaction strategy, because a larger amount and variety of development activities were offshored to the vendor. Figure 1 shows how work has been divided between the client and the vendor. In the beginning the bottleneck of this model was in the middle. It was difficult to offshore enough coding activities to the vendor and test the quality of the produced code. Today the bottleneck has moved to the front and back ends of the process. About 400 people are ready to work at the vendor site, starting from business oriented use cases and delivering integrated code ready for acceptance. The hard part now, tells a manager at the client, is to get the business people to write enough, high-quality use cases – that is, to decide and specify how they want the business processes to be.

The main limitation of the client's approach has been that in order to minimize interaction with the vendor it has become necessary to perform considerable extra work. This work is required to enable the vendor to take on tasks in spite of its limited business knowledge. The extra work consists of preparing tasks for offshoring, preparing the vendor for working in the client's application domain, and assessing the quality of the vendor's work. In the terminology of Dibbern et al. (2008) this extra work corresponds to specification costs, knowledge-transfer costs, and control costs. While the knowledge-transfer activities and certification programs are intended to gradually make it possible for the client to offshore also the specification of systems and the assessment of work products, the currently offshored activities are somewhat biased toward programming. Thus, the client is succeeding in offshoring programming but, at least currently, at the cost of extra work on other activities. Compared to previous, onshore development the client's activities have shifted toward the start and end of the development process, see Figure 1.

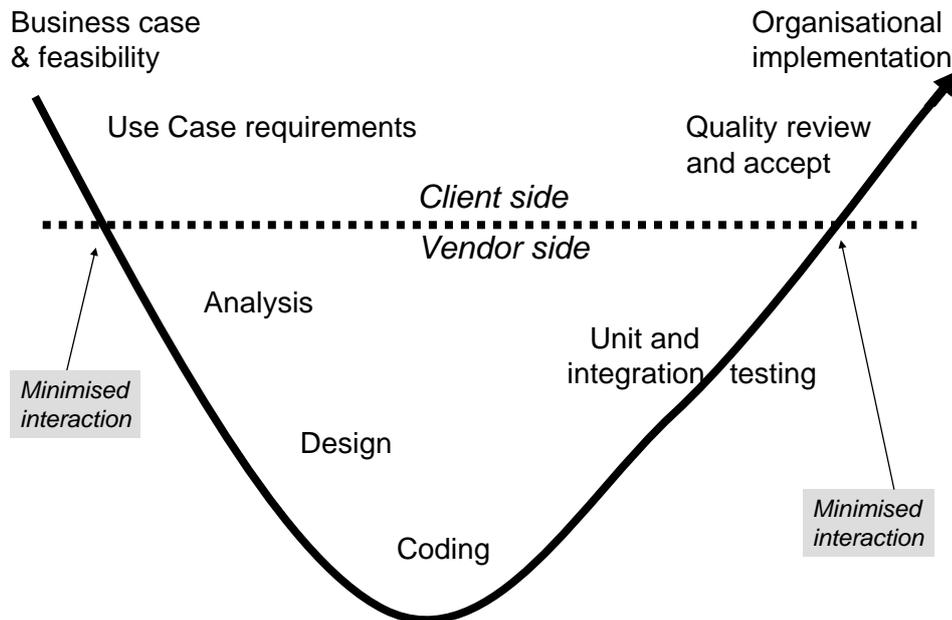


Figure 1. Software development activities at Client and Vendor.

This shift has important consequences for the client. First, it implies that the client is to a considerable extent doing work in order not to have to do work. The amounts of extra work have not been fully foreseen, and cultural differences entail that the extra work is perceived differently by client and vendor. For example, the vendor organizes activities partly from the implicit perspective that hours are cheap and capacity large, but this perspective is defective when some of the hours (e.g., control activities) are to be performed by the client. It is an ongoing learning process to identify and reduce areas of extra work but also to realize that offshoring is increasing the amount of some of the client's tasks. Second, the extra work may exceed the capacity of the client staff and thereby prevent the client from offshoring as much work as the vendor would be able to perform. While the bottleneck that initially motivated the client to offshore was perceived as a shortage of programming capacity, it may now emerge as a shortage of capacity to specify systems and control work products. This way, the uneven distribution of business knowledge may be the factor that limits the client's minimal-interaction approach to offshoring, making a reduction of the knowledge asymmetry central to continued success with this approach. Third, the tasks of the client staff are changing. This implies that client staff increasingly needs a different mix of competences with more focus on business understanding and abilities to facilitate the formulation of requirements, the transformation of requirements into system specifications, and the follow-up on whether developed systems match business requirements. Some client staff may welcome this change of focus; others may be reluctant to give up time for programming in favour of activities at which they feel less proficient and comfortable.

6.2 Conway's law

Conway's law (Conway 1968) states that "organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations." Thus, the communication bottleneck between client and vendor in offshoring arrangements will lead to system designs that reproduce this structure. Conway concludes that flexibility of organization is central to effective design. Flexibility is needed to be able to adjust the organizational structure to a system

architecture that matches the needs of the use situation. Because designers' understanding of these needs will likely evolve during the development process, flexibility of organization is required throughout the development process, not just when projects are set up.

Conway argues that especially for large systems the required flexibility is rarely present and that the structures of large systems therefore tend to disintegrate during development. This disintegration is the result of a three-step process. First, when designers realize that a system will be large they are tempted to assign too many people to the project. This temptation is exacerbated by access to a large pool of development staff, as is typical in offshoring. Second, in a large project the communication paths must be restricted in order to avoid that communication consumes all people's time, as exemplified by the single point of entry enforced between offshore staff and the client's quality-assessment staff. This causes the communication structure to disintegrate. Third, Conway's law ensures that the disintegration of the communication structure will be reproduced in the system structure, which therefore also disintegrates. This argument appears pertinent to offshoring because the client gets access to the vendor's large pool of development staff and because the communication between client and vendor is already restricted by their physical separation (e.g., Herbsleb 2007; Herbsleb et al. 1999).

In projects A and B we can clearly explain part of what we see by using Conway's law. Project A, for example, complied with Conway's law by reproducing the organizational separation between an onshore group with business knowledge and an offshore group with technical knowledge in the system: The system was completely rebuilt technically but remained completely unchanged functionally (as it was planned). This was not a problem for project A itself, because it was the client's first offshore project and a lot was learned from it. However, project B gives some indication that the client's aim of eventually offshoring entire projects from requirements to implementation is still hampered by the uneven distribution of business knowledge. This makes communication about business knowledge a central bottleneck because it leads to missed opportunities in the offshore-developed systems.

7 CONCLUSION

In October 2008 while we were making interviews in the Danish organisation it was decided to renew the contract with the vendor without changing anything but the ending-date of the contract. Thus, the way the interaction had worked between client and vendor was considered a success. We have shown in this paper that the strategy followed was one of minimizing interaction. Concretely, we have shown how keeping distance, exchanging people, and exchanging knowledge can be used to develop software in a way that minimizes interaction. It should be noted, however, that achieving minimized interaction requires a lot of work. It is not a cheap solution; the price (and time) invested by the client was larger than expected.

We conclude that minimizing interaction can be a viable strategy to follow when clients and vendors face large cultural and maturity inequality in offshore outsourcing.

Acknowledgements

This work has been supported by the Danish Agency for Science, Technology, and Innovation through its funding of the SourceIT project. We wish to thank our colleagues in the SourceIT project for inspiring discussions and master-thesis students Sofia Knudsen and Magnus Hansen for their support in documenting the interviews. Special thanks are due to the interviewees for their openness and their willingness to take part in our study in spite of their busy schedules.

References

- Ahern, D. M., A. Clouse, et al. (2003). CMMI Distilled: A Practical Introduction to Integrated Process Improvement, Addison-Wesley.
- Barthélemy, J. (2001). "The hidden costs of IT outsourcing." MIT Sloan Management Review 42(3): 60-69.
- Bhat, J. M., M. Gupta, et al. (2006). "Overcoming requirements engineering challenges: Lessons from offshore outsourcing." IEEE Software 23(5): 38-44.
- Carmel, E. and P. Abbott (2007). "Why 'nearshore' means that distance matters." Communications of the ACM 50(10): 40-46.
- Carmel, E. and R. S. Agarwal (2002). "The maturation of offshore sourcing of information technology work." MIS Quarterly Executive 1(2): 65-77.
- Chrissis, M. B., M. Konrad, et al. (2003). CMMI: Guidelines for Process Integration and Product Improvement, Addison Wesley.
- Cockburn, A. (2000). Writing Effective Use Cases, Addison-Wesley Professional.
- Conway, M. E. (1968). "How do committees invent?" Datamation 14(4): 28-31.
- Cullen, S., P. B. Seddon, et al. (2006). Managing the sourcing process: A life cycle perspective. Chapter 2. Global Sourcing of Business and IT Services. L. P. Willcocks and M. C. Lacity. Houndgrave, UK, Palgrave Macmillan.
- Cusick, J. and A. Prasad (2006). "A practical management and engineering approach to offshore collaboration." IEEE Software 23(5): 20-29.
- Dibbern, J., J. Winkler, et al. (2008). "Explaining variations in client extra costs between software projects offshored to India." MIS Quarterly 32(2): 333-366.
- Dubé, L. and G. Paré (2001). "Global virtual teams." Communications of the ACM 44(12): 71-73.
- Hendry, J. (1995). "Culture, community and networks: The hidden cost of outsourcing." European Management Journal 13(2): 193-200.
- Herbsleb, J. D. (2007). Global software engineering: The future of socio-technical coordination. Proceedings of the 2007 International Conference on Software Engineering: Future of Software Engineering. Washington, DC, IEEE Computer Society: 188-198.
- Herbsleb, J. D. and R. E. Grinter (1999). "Architectures, coordination, and distance: Conway's law and beyond." IEEE Software 16(5): 63-70.
- Hofstede, G. (2001). Culture's Consequences: Comparing Values, Behaviors, Institutions, and Organizations Across Nations. Second Edition. Thousand Oaks, CA, Sage.
- Hofstede, G. and G. J. Hofstede (2005). Cultures and organizations: Software of the mind. New York, McGraw-Hill.
- Iacovou, C. L. and R. Nakatsu (2008). "A risk profile of offshore-outsourced development projects." Communications of the ACM 51(6): 89-94.
- IndiaExpressGroup. (2003). "Software and systems firms embrace CMMI." Express Computer - India No. 1 IT Business Weekly 20030310. Retrieved November 21, 2008, from <http://www.expresscomputeronline.com/20030310/indtrend1.shtml>.
- Jacobson, I., M. Christerson, et al. (1992). Object-oriented software engineering: a use case driven approach, Addison-Wesley.
- King, W. R. and G. Torkzadeh (2008). "Information systems offshoring: Research status and issues." MIS Quarterly 32(2): 205-225.
- Krishna, S., S. Sahay, et al. (2004). "Managing cross-cultural issues in global software outsourcing." Communications of the ACM 47(4): 62-66.
- Lacity, M. C. and R. A. Hirschheim (1993). Information Systems Outsourcing; Myths, Metaphors, and Realities, John Wiley & Sons, Inc. New York, NY, USA.
- Levina, N. and E. Vaast (2008). "Innovating or doing as told? Status differences and overlapping boundaries in offshore collaboration." MIS Quarterly 32(2): 307-332.
- Lewin, A. Y. and C. Peeters (2006). "Offshoring Work: Business Hype or the Onset of Fundamental Transformation." Long Range Planning 39: 221-239.

- McFarlan, F. W. and B. DeLacey (2004) "Outsourcing IT: The Global Landscape in 2004, Harvard Business School." Volume, DOI: 9-304-104
- Metters, R. (2008). "A case study of national culture and offshoring services." International Journal of Operations & Production Management 28(8): 727-747.
- Mohnot, N. (2003). "Why "India Inside" Spells Quality." Monday, October 27, 2003. Retrieved November 21, 2008, from <http://dqindia.ciol.commakesections.asp/03102703.asp>
- Naur, P. (1985). "Programming as theory building." Microprocessing and Microprogramming 15(5): 253-261.
- Olson, G. M. and J. S. Olson (2000). "Distance Matters." Human-Computer Interaction 15(2&3): 139-178.
- Oza, N. V. and T. Hall (2005). "Difficulties in managing offshore software outsourcing relationships: An empirical analysis of 18 high maturity Indian software companies." Journal of Information Technology Case and Application Research 7(3): 25-41.
- Paré, G. and L. Dubé (1999). Virtual teams: An exploratory study of key challenges and strategies. Proceedings of the 20th International Conference on Information Systems. New York, ACM Press: 479-483.
- Paulk, M. C. (1995). "The Evolution of the SEI's Capability Maturity Model for Software." Software Process Pilot issue: 3-15.
- Pries-Heje, J., J. Nørbjerg, et al. (2008). The Road to High Maturity: How the first Danish company reached CMMI level 5 in 100 months. Software Processes & Knowledge: Beyond Conventional Software Process Improvement. P. A. Nielsen and K. Kautz. Aalborg, Software Innovation Publisher: 163-192.
- Schmidt, R., K. Lyytinen, et al. (2001). "Identifying software project risks: An international Delphi study." Journal of Management Information Systems 17(4): 5-36.
- Schomer, K. (2006). "Culture Matters: 5 Challenges India Offshore Teams Face in Working with Americans." Sourcingmag.com Retrieved 10. November 2008, 2008, from <http://www.sourcingmag.com/content/c060814a.asp>.
- Terdiman, R. and T. Berg (2001). Offshore application outsourcing. Stamford, CT, Gartner Group.
- Vitharana, P. and R. Dharwadkar (2007). "Information Systems Outsourcing: Linking Transaction Cost and Institutional Theories." Communications of the Association for Information Systems 20: 2-2.
- Vlaar, P. W. L., P. C. van Fenema, et al. (2008). "Cocreating understanding and value in distributed work: How members of onsite and offshore vendor teams give, make, demand, and break sense." MIS Quarterly 32(2): 227-255.
- Willcocks, L. P. and M. C. Lacity, Eds. (2006). Global Sourcing of Business and IT Services. Houndmills, Basingstoke, Hampshire, UK, Palgrave Macmillan.
- Williamson, O. E. (1979). "Transaction-Cost Economics: The Governance of Contractual Relations." The Journal of Law and Economics 22(2): 233.
- Yin, R. K. (1994). Case Study Research. Design and Methods. 2nd Edition. SAGE, Thousand Oaks, CA, USA.