



Copyright © IEEE.
Citation for the published paper:

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of BTH's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by sending a blank email message to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

From Offshore Outsourcing to Offshore Insourcing: Three Stories

Nils Brede Moe
SINTEF ICT
Trondheim, Norway
nilsm@sintef.no

Darja Šmite
Blekinge Institute of Technology
Karlskrona, Sweden
darja.smite@bth.se

Geir Kjetil Hanssen
SINTEF ICT
Trondheim, Norway
ghanssen@sintef.no

Abstract — Most large software companies are involved in offshore development of some sort, and now the trend is that small and medium sized companies are also going global. While empirical research suggests that offshoring are not always successful, evidence explaining the reasons for terminated collaborations is scarce, especially for small and medium sized companies. In this paper we explore the stories of three Scandinavian medium-sized software companies that have terminated their offshore outsourcing relationships and changed to offshore insourcing arrangements. The main reason for termination was disappointing low quality of the software delivered, being caused by insufficient domain knowledge, high turnover and a lack of motivation among the remote and external developers. We apply the theory of single-loop and double-loop learning in order to explain why the companies failed to correct the experienced problems, and the theory of escalating commitment to explain why the companies did not correct the failing course of action earlier. Finally we describe the change in the choice of the principle sourcing strategy from outsourcing to insourcing.

Keywords – *global software engineering, global software development, offshoring, insourcing, outsourcing, single-loop learning, double-loop learning, escalating commitment, empirical study*

I. INTRODUCTION

Software companies nowadays outsource, insource, expand through acquisitions, and build partnerships and joint ventures. Offshoring, which means that software development occurs in a different country, has matured over the recent years. However, far from all offshoring relationships are successful. Experience reports provide conflicting results balancing from announcements of success to total failure, with a majority being dedicated to problem reports [24], suggesting that offshore collaborations will always be challenged.

Offshore outsourcing (sub-contracting to third party vendors from other countries) is one major trend within offshoring, and often motivated by the opportunity to reduce development cost (see e.g. [4, 6]). However, results from industrial surveys indicate that the number of terminated outsourcing contracts is high [2]. Also, more current investigations demonstrate that traditional reliance on offshore outsourcing has recently changed in favor of insourcing (collaboration within the company boundaries) [22, 23].

While this transition from offshore outsourcing to offshore insourcing has been more carefully studied in information systems research, the topic has gained little attention in the field of software engineering. In fact, systematic reviews in the field of global and distributed software development suggest that offshore insourcing as a business model is least researched [20, 24].

Notably, while research on offshoring is often centered on large-scale software development in large sized companies [3], most software companies, in fact, are small or medium sized (SMEs). Software companies with fewer than 50 employees even in the USA reaches 94% [8]. Further, because offshore collaborations are enabled through a variety of different forms, and initiated for different reasons [23], the choice of sourcing strategies significantly depends on the context [21, 24], and since SMEs often follow different business strategies than large companies, there is a need for case studies with focus on specific needs of SMEs [3].

Motivated by the need for research on small and medium sized companies moving from offshore outsourcing to offshore insourcing, we have identified the following research questions:

RQ1: What are the reasons for terminating offshore outsourcing relationships?

RQ2: What are the reasons for switching from offshore outsourcing to offshore insourcing?

In this paper we study three product companies involved in small-scale software development in an attempt to understand the reasons for their initial engagement in offshore outsourcing and sequential transition to insourcing. The aim is not to explore which sourcing strategy is the best, but what worked, what did not work, and for which reasons in the particular contexts of the cases studied.

The rest of the paper is organized as follows. In Section II we present an overview of research related to the termination of sourcing contracts and theoretical background of organizational learning that we further apply for its explanatory power. In Section III we present an empirical study design and the background of the three cases. Results from the empirical studies are provided in Section IV, followed by a discussion in Section V. Finally, Section VI concludes the paper with the summary of the major findings.

II. RELATED WORK

A. Outsourcing and Insourcing Software Work Offshore

There are four main types of sourcing arrangements with offshore outsourcing as an important trend [23]. However, while a number of studies have been published on outsourcing performance that measure efficiency, user and business satisfaction, service quality, and cost reduction, comparison of these studies revealed conflicting results [17]. One explanation is that suppliers over-promise and under-deliver [15]. These findings can be the main reasons for why researchers [22, 23] have found that the previous popularity of outsourcing in global software development has started trending toward insourcing and partnerships [ibid] (See Fig.1).

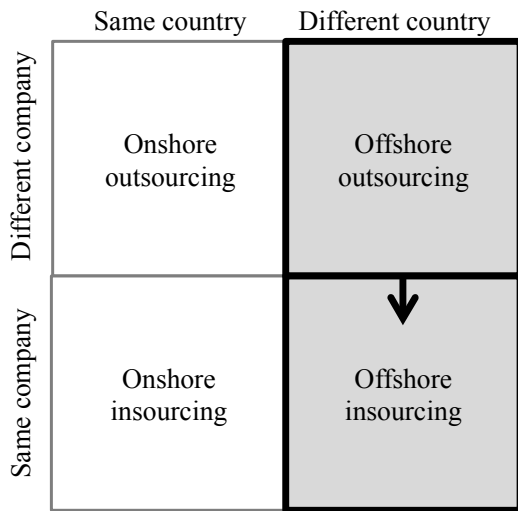


Figure 1. Types of sourcing arrangements

Although studies on offshore outsourcing performance have conflicting results, offshore outsourcing failure stories are rarely told. There are however some studies from the field of Information Systems. For example, Barney et al. [2] refer to two surveys on offshore outsourcing, where one reported that 30% and another that 50% of the companies being involved in outsourcing cancelled their contracts. The main reasons for terminating the outsourcing contracts was that cost savings did not eventuate, as well as the need for protecting their intellectual property.

While, research on the comparison of outsourcing and insourcing to date is scarce [24], we found one empirical study by Prikladnicki et al. [21]. Five different projects applying different strategies were studied, and the authors claim that it is easier to follow traditional development lifecycles in outsourcing projects, while insourcing projects are easier for iterative and incremental development [ibid]. They also conclude that the simplest level for process definition and execution is supported by onshore insourcing strategies, while offshore outsourcing depends on a high level of formalization.

B. Organizational Learning

Findings from an empirical investigation suggest that instead of blaming suppliers for poor performance, a sourcing relationship requires joint improvement efforts to evolve [21]. In other words, to succeed with offshoring software companies need to learn how to improve their offshore processes. One can argue that commitment to learning rather than to any offshoring model is more important to succeed with offshoring. For this reason we rely on the theory of learning as defined by Argyris and Schön [1] for understanding what affects organizations when they try to improve their offshoring relationships or choose to terminate their sourcing contracts.

Argyris and Schön distinguish between what they call single and double-loop learning in organizations. Single-loop learning is to change practice as problems arise in order to avoid the same problems in the future. For example, management often engages in single-loop learning by monitoring development costs, software quality, sales, client satisfaction, and other indicators of performance to ensure that the organizational activities remain within established limits, keeping the organization “on course”. In single-loop learning; if outcomes of actions are not met, the actions are changed slightly to achieve the desired results. It is a feedback loop from observed effects to making some changes or refinements that in turn influence these effects.

Double-loop learning, on the other hand, is when time is taken to understand the factors that influence the effects, and the nature of this influence, called the governing values. It is about using the problems being experienced to understand their underlying causes, and then taking some action to remedy these causes. One example is what happens when low quality is detected in a delivery. Correcting the errors itself can be seen as single loop learning, but if something is done with whatever caused the errors to be introduced (e.g. providing training to developers because of missing competence), that is considered double-loop learning. The changes based on this type of understanding will be more thorough.

Existing research indicates that some organizations learn, experientially and often painfully, how to manage outsourcing [15] while others decide to terminate their contracts, e.g. as referred by Barney et al. [2]. In this paper we will rely on the concept of single and double loop learning to understand how and why three companies terminated their offshore outsourcing contracts and moved to offshore insourcing, and whether these lessons learned may help other companies to identify, avoid or escape from unfitted offshore collaborations.

III. STUDY DESIGN

In this paper we look into three SME product companies to illustrate and discuss reasons for why companies terminate offshore outsourcing relationships in favor of offshore insourcing. The reason for choosing these companies was that they were all part of research projects where the researchers participated and fulfilled the selection criteria which made it possible to answer the research questions — a) a small or medium sized company, and b) a history of a terminated sourcing relationship. The cases were also chosen for the availability of rich empirical material. All three cases are Scandinavian SMEs terminating or not renewing their

outsourcing contracts in Asia. A summary of the cases and an overview of data collected can be found in Table 1.

This study is a multiple case study as described by Yin [26]. According to Yin, case studies are the preferred research strategy “...when a «how» or «why» question is being asked about a contemporary set of events over which the investigator has little or no control.” (ibid, p. 9). The three stories told are taken from three case studies, one lead by each of the authors.

In this study we relied on data triangulation as defined by Yin. By using data triangulation our findings and conclusion are likely to become more convincing and accurate. Yin suggests six major sources of evidence when performing data triangulation. We relied on:

- documentation,
- interviews,
- direct observation

We will now briefly describe the case companies. A more detailed description of the cases and their domain is given in the result section.

Dolphin. The story of Dolphin is taken from an ongoing longitudinal study of distributed development that started in 2009. The main objective of the study is to diagnose and improve existing offshore collaborations. As a part of this research we retrospectively collected experiences from the previous outsourcing attempts. The data collected includes formal and informal discussions during group meetings with the unit managers at the onshore office of the company, observations from current collaborations collected during an offshore site visit and documented in meeting minutes, as well as a few dedicated interviews with product managers, and development unit managers at the company.

Guppy. The story of Guppy is taken from a longitudinal study of the organizations development, going on from 2004 to 2009 [10]. The main objective of the study was to track the adoption of the agile software development method Evo and the following effects on the organization and its development processes. An important part of this study was to understand how the offshore collaboration and the change in sourcing strategy affected the development of the organization. Data was collected as observations of customer meetings and marketing events, as interviews with various roles such as developers, team managers and system architects and by reviewing product plans.

Nemo. The story from Nemo is taken from studying several agile projects involved in global software development, first with India and later with China in the period from 2007 until 2011. The data collected for this study includes interviews of project members from all sites, project managers, and people from the business department. Also several meetings (planning, review and retrospective) were observed. Two Indian outsourcing projects have been described in more detail in Hole and Moe [12]. The outsourcing relationship with the Indian company that was terminated and the following insourcing relationship with the Chinese department of Nemo are described in more detail in Barney et al. [2].

TABLE I. CASES AND DATA COLLECTION

	Dolphin	Guppy	Nemo
Number of developers	~40	~60	~100
Headquarter	Sweden	Norway	Norway
Outsourcing destination	India	Vietnam	India
Work structure	One team working on one project	One team working on one module	Virtual team developing modules
Insourcing destination	Russia	Russia	China
Team structure	Several teams working on several modules	Several teams working on several modules	Virtual team developing modules
Study length	2009-2012	2004-2009	2007-2011
Interviews	8	3	11
Observations and informal dialogues	9	6	10
Group interviews	1	1	3
Documents	1	2	1

In the next section, the companies and context are described in a narrative as described by Langley [16] to achieve an understanding of what took place in the companies studied. The narratives have been validated with representatives from the companies. Empirical results are then discussed in the light of Argyris and Schöns theory of learning [1].

IV. THREE STORIES OF TRANSITION FROM OUTSOURCING TO INSOURCING

A. Dolphin

Company profile. Dolphin is a medium-sized business unit wholly owned by a major multinational corporation producing software intensive products with the main development centers located in Sweden. The types of systems developed by Dolphin are complex products for process automation and control, in which software is being only a part of the system. Systems development undergoes a rigorous set of processes that involve coordination of work among multiple departments. Some projects are applying agile methods with dedicated product owners steering the evolution of each system, and several development teams supporting this. Although software development is not the company’s primary business, the amount of software embedded in its products grows every day. For pure cost reasons and inability to employ people in the high-cost countries, the corporate management decided to engage in offshore development.

Outsourcing to India. Although the first experience with offshoring goes back to 1995 when Dolphin hired two engineers in Russia, a serious attempt to embrace the benefits of globalization was done in 2001, when the company contracted one of the largest consultancy companies in India. When Dolphin made the decision, a market investigation was performed and potential outsourcing partners were evaluated, one of the criterions being the CMM (Capability Maturity Model)[19] certification. Out of a final list of 2-3 companies, Dolphin chose the biggest supplier, which they believed, would provide the best quality. This outsourcing experience became the first endeavor of this kind both to management and developers. The sub-contracted work included non-core competencies, in particular software development, testing and bug fixing, but no ownership for independent pieces of software was transferred. This means that outsourcing in the Dolphin's case led to distributing development tasks across two locations.

Because of the small scale of outsourced projects the supplier company in India gave low prioritization to Dolphin's contract resulting in a tremendous high turnover among employees. Development managers complained that every time they visited India, they would see new faces. This however triggered a negative loop – once the people were trained for the work in the project they would leave and new developers would require training again. Additionally, the levels of commitment to the work outsourced were unsatisfactory. Swedish employees felt that their Indian developers did not really involve themselves in the project, which was one of the reasons for the failure.

Socio-cultural and temporal distance was also seen as problematic. Cultural clashes caused misunderstandings, which were difficult to solve over the time zone differences. In combination with the inability to influence employee turnover, lack of commitment, and the inherent problems the projects never gained the expected benefits to justify the inconveniences of the distance. After three years of suffering from poor software quality and poor efficiency, the ongoing outsourcing contract was terminated and it was decided to change the sourcing strategy and expand within the already established subsidiary in Russia.

Transition to insourcing in Russia. Transition from outsourcing to insourcing started small and scaled into a new offshore office with 20 developers within four years. It required much more support than initially anticipated. In their offshore insourcing relationship Dolphin now had to take care of many administrative concerns associated with running an office themselves. On the other hand, it enabled a better control over the offshore location and ability to influence their experiences. Although an internal aversion to trust offshore engineering was seen within the business units after the failed outsourcing contract, other impediments on the engineering level were fewer.

Learning from experiences of a high turnover in the outsourcing relationship, Dolphin invested resources and implemented incentives and control mechanisms to avoid employee turnaround in their insourcing relationship. Recruitment was now seen as strategic practice, and was taken care of by employing an experienced leader at the offshore site. To increase motivation and cultivate commitment Dolphin

transferred responsibility for the entire maintained products to the offshore site. Today insourcing relationship is acknowledged for its benefits, despite a few challenges that still exist. The lessons learned from Dolphin's experience of transition from outsourcing to insourcing were formulated by the development manager as follows: "*If you are going to go down the path of transferring long term product development work – go with insourcing to a site where you can steer your success criteria rather than having some of those variables locked for you*". In Dolphin's case, the outsourcing vendor locked staff turnover, competence level and engineering culture.

B. Guppy

Company profile. Guppy is a medium-size software company that develops, maintains, and markets a business and market analysis software package. They have become the market leader in the high-end segment of the market they serve, with some of the world's largest market research agencies on the client list. Currently Guppy employs about 260 people, including more than 60 developers. The main office is located in Norway, which houses the main section of the development department as well as the top management and various support services such as operations, technical support, sales, training and others. The rest of the organization is distributed internationally with development departments and sales and other support services in other parts of the world.

Outsourcing to Vietnam. Due to growth and problems in recruiting more developers locally in Norway, in 2004 Guppy hired 10 developers at a Vietnamese consultancy company. In order to establish good communication with this offshore team one Norwegian, being married to a local, took the role as team leader at the office of the consultancy company in Vietnam, taking on a customer-on-site type of role. The team was maintaining parts of the code for one of the main modules in the product and was at that time the first and only external group of developers in the company. The rest of the development organization in Norway had some concerns regarding latency in communication due to time-zone differences but the management did not see this as a major problem. The offshore team worked for about a year, which was the release cycle duration at that time. However, approaching the release deadline, it became obvious that the quality of the code was so poor that some features developed in Vietnam had to be dropped from the release.

The low quality of the code being produced by the Vietnamese team led to the decision to terminate the contract with the outsourcing partner. All the work was back-sourced to the main office in Norway. After the following release, an in-house team of development and system architecture experts had to spend a lot of effort on fixing the code and the design to bring it back to an acceptable quality level. According to one of the experts, this drop in quality had never happened before or after the failing outsourcing attempt. In a search for an explanation, the Norwegian developers raised a handful of issues: First, according to one of the team leaders, the externals had a lack of corporate spirit that the in-house developers in Norway felt they had developed from working on the product development for several years. Also, the external developers did not show the same pride in the product. Second, the remote

developers did not have the same knowledge and understanding of the underlying business domain that the product were supporting. Third, the follow-up and quality assurance of the Vietnamese work was not good enough (even with a local inhouse representative working with the developers on a daily basis). Forth, there was also a ‘culture-crash’, to use the term of one of the Norwegian developers; *Vietnamese developers did not dare to tell if they didn't understood the problem and the tasks they were assigned which created misunderstandings and consequently low quality code.* The code quality problem became even more evident towards the end of the release period when integration started. Some code was even considered too low to go into the next release and were discarded. This situation lead to the decision of cancelling the outsourcing relationship and bring all development back to the main office in Norway.

Transition to insourcing in Russia. Due to the growth of the size and complexity of the product and the growing market for it, the need for hiring more experienced developers became even more pressing. Based on the experience with the unsuccessful attempt of outsourcing development and an opportunity of buying a competitor, Guppy made a strategic decision to integrate both their software product as well as the people with their own organization. The organization grew to more than 250 employees, now including two offshore development departments in Russia. The costs of realizing the first phase of this strategy were much higher than setting up an outsourcing arrangement, but the results shows that keeping development insourced, yet offshore, have helped Guppy to avoid the problems they experienced in their previous outsourcing relationship when someone outside the organization was given the responsibility of developing parts of their product. In order to ensure a common culture in the distributed software development departments, the company now invest in continuous training of developers at the offshore sites and in the alignment of software development process between on- and offshore sites. These efforts have been expensive when it comes to time and money, yet the result is a high level of quality and engagement in the product development, which was absent when work was outsourced.

C. Nemo

Company profile. Nemo is a Norwegian SME, producing specialized software for the engineering domain. Nemo sells mass-market software and also writes customer specific software on a contract-basis.. China has recently become an important market for its products. Nemo has very low staff turnover; less than 10% per year in its software division. Most of the people working in the software development department have been trained as engineers (2/3 of the staff) rather than professional software developers (1/3 of the staff) but the proportion of software developers is increasing. In addition to conducting software development work in the main office in Norway, Nemo also conducts software development in its offices in Eastern Europe and the UK. These are wholly owned subsidiaries of Nemo, however they were not a part of this study. While Nemo had some experience from global software development they have never engaged in outsourcing.

Outsourcing to India. Three software development projects were set for collaboration with a development team in an Indian consulting firm. The project manager in Norway mediated communication between the Indian developers and the Norwegian developers. The expected savings of employing the outsourcers were not realized, because they were not getting the features or the quality that was expected. These problems led Nemo to trial modifications, first just to the routines used to collaborate with the outsourcer and then both the routines and the resources used. Nemo trialed making more complete specifications, reviewing and commenting on the code developed in India to give the Indian developers feedback about how they could develop code that was more in line with Nemo's goals and coding conventions. However as a product owner in Norway explained: *“We then started sending them work-packages specified in detail, but we realized it would be a too big job to do this for each work package.”* There was a perception among some of the Norwegian developers that this change in the routines had actually made things worse in the Norwegian office. One team leader commented: *“Our people felt like they were spending basically all their time writing work orders and writing code for these guys through [email]”*.

Nemo then trialed more drastic changes to the project structure that it hoped would address the communication and coordination problems, by changing not just the routines but also the resources employed. The lead developer from one of the teams in India was brought to work in Norway. Having the team leader collocated with the Norwegian developers considerably changed her characteristics as a resource. Now communication between Norway and India was conducted using the team lead as the intermediary. One team leader said: *“This improved the situation a lot. The productivity increased while she was here. The important issue is to communicate with only one person.”*

There was a general perception that this considerably improved their ability to communicate with the Indian developers and coordinate their work. While this change did lead to higher satisfaction with the outsourcing relationship, Nemo felt that performance related to the software developed was still insufficient to justify continuing the relationship. The quality of the software was not good enough. In retrospective we found out that most of the code delivered by the Indians over a period of three years was not used or had been removed from the final products. One executive felt that the business model of the outsourcer was one reason for problems with the quality. He said: *“It took us quite some time to understand their business model. We started out the work with individuals and they have this idea of having one programmer and taking ten from the street and hope that things work out. First it took us a long time to understand that was how it works. Then we tried to make up relationships with people with no understanding and knowledge and just from school...”*

Those interviewed at Nemo felt that high turnover among the outsourcer's employees meant that the outsourcer's developers did not have sufficient time to acquire the human capital related to the engineering domain and the complicated software architecture. Which again affected the quality. At the end of the outsourcing relationship, one executive interviewed came to recognize that the combination of domain knowledge and software development expertise was one of the company's sources of competitive advantage. This was not considered

when the decision was made to outsource development to India.

Transition to insourcing in China. Eventually it became clear that the situation was untenable and that the outsourcing effort would have to be terminated. Nemo had commitments related to the release of the software that the outsourcers were working on therefore they could not terminate the relationship immediately, since it would jeopardize these commitments. Instead, it was decided that the outsourcers effort with India would continue for another year. No new projects would be assigned to India. At the same time, the expansion of the Chinese office made back sourcing part of the previously outsourced software development and offshoring some development work to its captive office in China an attractive alternative to the outsourcing. Nemo incorporated what it learnt about conducting global software development into its relationship with its office in China. By sending experienced engineers to China it facilitated the Chinese. The Norwegian expatriates also arrived in China with a network of contacts at Nemo to call to resolve problems when they arise. This allows them to quickly resolve problems that the expatriates are not able to resolve themselves immediately and far more expeditiously than the outsourcers could. The Chinese office soon experienced some success stories.

The Chinese developers are also offered attractive conditions ensuring low turnover, which made it possible for Nemo to plan how to increase the competence of the experienced Chinese developers even more.

D. Summary

We have now presented the story of how and why Nemo, Guppy and Dolphin changed their sourcing strategy from offshore outsourcing to offshore insourcing. In Figure 2 we illustrate the course of sourcing decisions and how these relationships evolved. While two of the companies had some experience with global software development, none of the companies had prior experience with outsourcing. All three targeted a satisfying outsourcing relationship; however, the course towards successful offshoring diverged from their initial plans.

In case of Dolphin, the company decided to outsource from Sweden to India. Dissatisfied with the relationship several improvements were initiated, but after three years of failure to achieve satisfactory outcome, the contract was terminated and the company decided to expand its insourcing relationship in Russia. The latter became a successful relationship, ongoing for more than ten years. The course of events in Figure 2 is: Outsourcing – Dissatisfying relationship – Relationship not terminated – Improvements – Dissatisfying relationship – Termination of outsourcing – Insourcing – Satisfying relationship.

Similarly to Dolphin’s case, Nemo initiated outsourcing from Norway to India, which led to dissatisfying results. After three years of failing attempts to improve the relationship, the contract was then terminated. Later the company decided to insource to China, which turned into a satisfying relationship that is ongoing for four years. According to Figure 2 the course of events is the same as in the Dolphin’s case: Outsourcing – Dissatisfying relationship – Relationship not terminated –

Improvements – Dissatisfying relationship – Termination of outsourcing – Insourcing – Satisfying relationship.

In the Guppy case, the company outsourced work from Norway to Vietnam, but in the light of poor quality after the first release cycle (about one year) they decided to terminate their relationship and backsource the development in-house. One year later the company decided to establish an insourcing relationship through acquiring a Russian company. The relationship is ongoing for more than five years. The course of events according to Figure 2 is: Outsourcing – Dissatisfying relationship – Termination of outsourcing – Insourcing – Satisfying relationship.

While all companies ended up with a satisfying sourcing strategy, they still are working to improve their offshore insourcing relationship.

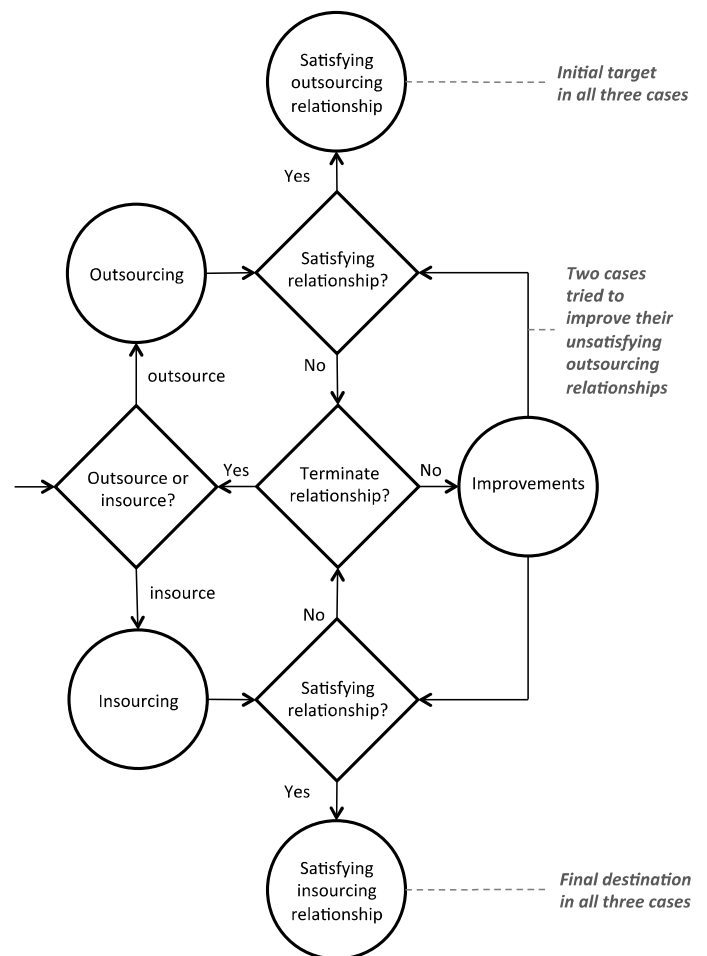


Figure 2. Evolution of sourcing relationships

In the next section we discuss why none of the companies achieved the initial target, satisfying outsourcing relationship, and how the companies came to the final satisfying insourcing relationship.

V. DISCUSSION

We have described three companies that changed their offshoring arrangements from outsourcing to insourcing. We now discuss the cases in light of our research questions:

- What are the reasons for terminating offshore outsourcing relationships?
- What are the reasons for switching from offshore outsourcing to offshore insourcing?

Despite the differences between the three organizations studied, all three cases share some similar experience (see Table II). Parallels can be drawn with respect to the reasons for outsourcing in the first place, reasons for ending the contracts and further switching to insourcing.

A. Reasons for terminating offshore outsourcing relationships

All three companies studied terminated their offshore outsourcing relationships because they were not able to solve the problems related to the offshore outsourcing arrangement, which they rapidly experienced. Common for all companies was that software being developed had too low quality to go into production. Two of the companies tried to improve their relationships, while one company ended the contract after the first year of collaboration without even trying to improve the outsourcing relationship. To answer our first research question, it is important to understand why the three companies experienced problems and why they were not able to solve them. Further we will discuss how the lack of motivation in the remote company and a lack of outsourcing experience are related to the reported problems. Finally we will discuss why two of the companies delayed their decision on terminating the relationship. In the next sections we will therefore discuss four important themes:

1. Failure to improve the relationships, due to inability to solve the underlying problems,
2. The impact of low motivation,
3. The impact of missing outsourcing experience,
4. Reasons for delaying the termination of the failing relationships.

Not solving the underlying problem. Dolphin and Nemo both initiated improvement programs to improve the code quality. New working practices were introduced, and more resources were allocated for improving communication and coordination. The working practices in Nemo included e.g. switching to agile practices, introducing coding conventions, and implementing new quality assurance QA routines.

Communication and coordination was improved by spending more money on travel and developers using a lot of time communicating directly between the sites. In the case of Nemo, one Indian Scrum master was even moved to Norway for a longer period. While the efforts of the two companies improved the situation to some degree, problems continued. The main reason was that the remote developers were missing important domain knowledge. The second major reason was the high turnover among the Indians in the case of Dolphin and Nemo. When a developer finally became productive, he or she would move to a new job or project within the outsourcing vendor company. Working on a product over time is necessary to build the knowledge of a particular business domain, which became hard due to the constant turnover.

The strategy of how the Dolphin and Nemo tried to improve the outsourcing relationship can be understood as a feedback loop from observed problems to making changes or refinements, which in turn influence these problems, hence cases of single-loop learning. However, failing to solve the real, underlying problem, which was related to the incomplete competence of the remote developers, can be seen as failing to do double-loop learning. The main reason for not being able to solve the real problem was related to the lack of influence over the outsourcing vendors, which confirms previous views that third-party vendors provide the least cooperation and control, while captive centers usually provide most [13]. In our cases, neither Dolphin nor Nemo were able to motivate the remote consultant companies to solve the competence and turnover-related problems. We also observed that outsourcing contracts for small customers and large suppliers created a misalignment of interests since there is no benefit for the vendor to invest more than the minimum in the work. In our study, the projects were small, and therefore probably not important enough for the offshore companies, which acted in self-interest mostly motivated by short-termed monetary goals. Being small and thus having low influence is therefore found to be one barrier of making fundamental changes in an offshore outsourcing project, e.g. a barrier to do double loop-learning which often requires radical measures.

In contrast to the two other companies, Guppy decided not to implement any corrective actions when they experienced the problem of low code quality close to the release of the new version of their product. Unlike Dolphin and Nemo they abruptly cancelled the whole outsourcing arrangement after one year and brought back the development to the central development department in Norway. In that respect, we can see this radical decision as a case of double-loop learning as they directly addressed the underlying problem of the “missing corporate spirit” and the low knowledge of the business domain by taking both under control.

TABLE II. PROBLEMS AND THEIR SYMPTOMS

Dolphin	Guppy	Nemo	Symptoms	Problems	Dolphin	Guppy	Nemo	Casual dependencies
X	X	X	Low quality	High turnover	X		X	Unsolved due to lack of control
X	X	X	Inefficiency of work	Culture clashes	X		X	Targeted by improvement efforts
		X	Expected savings not achieved	Time zone differences	X			Inherent problem
				Lack of commitment	X	X		Unsolved due to lack of control
				Lack of domain knowledge		X	X	Unsolved due to lack of control and high turnover
				Poor communication		X	X	Targeted by improvement efforts
				Increase in specification needs		X	X	Unsolved due to inability to increase domain knowledge

The impact of low motivation. In addition to small contracts being one reason for the inability to implement actions to solve the competence problem, another explanation is a lack of motivation among the remote developers. All three companies started small: neither transferred any ownership in the products they developed nor provided other motivation than monetary reasons. Lack of challenge, responsibility and affiliation might have caused this low motivation and low commitment in the offshore sites. There exist several theories that can help to understand the role of motivation amongst developers [9]. For example, the *role of affiliation* (part of McClelland's Need Theory) might be particularly relevant in cases with external developers. Motivation usually evolves from the feeling of being part of a team with a corporate identity, something inherently missing in an outsourcing arrangement. Thus it appears that offshore developers employed by the outsourcing vendor could be as little attracted by the contracted work, as their corporate management was attracted by the small contracts they received.

The impact of missing outsourcing experience: All three organizations considered outsourcing parts of their software development to someone outside their own organization in order to benefit from more developers with lower cost. This confirms the current view that companies outsource because "it is cheaper, and skilled labor is easier to find" [5]. Notably, all three Scandinavian companies chose to outsource to Asian suppliers without any prior offshore outsourcing experience, which might indicate that the companies took offshore outsourcing benefits for granted. A study on outsourcing by Hirschheim and Lacity warns that senior executives promote sourcing as the preferred vehicle for reducing the costs, and ignore threats to service quality and technical competence, which might be somewhat esoteric aspects from the top management perspective [11]. The outsourcing experiences shared in this paper are no different. Our findings are also in accordance with DeLone et al., who claim that the reason of failure in global projects is mainly related to the lack of awareness of the unique threats [7]. All three companies did not consider the challenge related to knowledge, domain knowledge, high turnover and poor motivation when initiating the offshore relationship.

Reasons for delaying the termination of the failing relationships. While the reasons for not being able to solve the problems in the outsourcing relationships and subsequently terminating offshore outsourcing contracts are evident, one question remain: Why did Nemo and Dolphin stay in a offshore outsourcing relationships for so long? One explanation can be that the companies experienced escalation of commitment [18]. Escalating of commitment can be defined as an increasing commitment to a failing course of action [25]. The decision makers in the companies first allocated resources to start an offshore outsourcing relationship, and then even more resources in the hope of improving the relationship. They continuously received feedback that they have not yet reached that goal, however continued to invest more resources. One manager in Dolphin described why they did not terminate earlier: "we believed for a long time we could get it to work". According to escalating commitment theory [14], individuals

tend to continue to commit to a course of action in order to self-justify the correctness of an earlier decision to pursue a particular course of action. In other words, people do not like to admit to themselves or others that a previous decision was wrong.

B. Reasons for switching from outsourcing to insourcing when offshoring

While all companies realized that they were not receiving the expected benefits from their offshore outsourcing relationship, and therefore terminated the cooperation, they still perceived offshoring as an important strategy for the future. To answer the second research question we will now discuss how the companies' previous experiences resulted in new offshoring relationships.

Because the companies now realized the importance and the difficulties of gaining access to competent and motivated developers in an offshore outsourcing relationship, their perception of how to organize offshoring changed. In our multiple case study, we found the use of two different approaches: 1) creating new or exploiting existing sites owned by the company, and 2) acquiring other companies and then integrating them into the corporate culture. Nemo was motivated by the importance of being close to the Chinese market, and since they already had a support office in China they choose approach 1). Dolphin had tried sub-contracting work to two developers in Russia and decided to expand by establishing an office, thus choosing approach 1) as well. Guppy wanted to grow by including new technology developed by another firm competing with their product and integrate the remote developers into their own organization, and therefore choose approach 2).

Existing norms and rules were challenged and changed in these companies, and new ways of working were found. This approach to organizational learning is understood as double-loop learning [1], because the organizations were discussing if they were doing the right things and then changed their offshoring arrangement. The companies were not only motivated by the access to highly skilled low cost developers, but also the need of gaining access to a new market and new technology.

The companies knew from recent experience that building a new lasting model based on offshore insourcing would require potentially huge long-term investments that inherently are avoided in an outsourcing arrangement. One example was that Nemo moved two highly skilled architects for three years from Norway to China. Also it takes a long time to build up an effective remote department from scratch, and to merge two companies with different cultures, technologies and processes that was the case for Guppy. To make sure Nemo got the right developers with the right competence when recruiting in China, they hired software developers with the same competence as in Norway (mathematicians and physicists). Similarly Dolphin recruited a knowledgeable and experienced offshore site manager, who was given the responsibility for the further recruitment to ensure good quality of the new employees. Additionally, product responsibility in the Dolphin case was gradually transferred to the offshore site to increase the motivation and commitment.

Switching to an offshoring insourcing arrangement allowed all companies to address the problems they experienced from their outsourcing attempts, such as maintaining control over recruitment, training and commitment. This positively affected the challenge regarding turnover because developers at the offshore sites were better motivated and had the feeling of affiliation with the corporate culture.

VI. CONCLUSIONS

In response to “What are the reasons for terminating offshore outsourcing relationships?” we conclude that outsourcing relationships were terminated because of the disappointing quality of the software being produced, which was caused by insufficient domain knowledge, high turnover and a lack of motivation among the remote developers. Also the various improvement programs, merely addressing the symptoms, were not helping because the fundamental problems were not solved. Once identified, the companies further realized that the real problems could not be solved within the existing relationships, because of unbalanced size of the collaboration partners, the relatively small projects not being prioritized by the remote companies, and unwillingness or inability to transfer the ownership of the software being developed to external parties.

To succeed with offshoring, a company needs to make sure the offshoring arrangement enables finding a balance between optimizing current processes (single-loop learning) and doing fundamental changes (double-loop learning). If that is not possible, a company should not be involved in offshoring in general, or outsourcing in particular, as in our three cases.

Although we cannot undoubtedly conclude that outsourcing was a wrong strategy, we attribute several important problems to the type of contract chosen. First of all, we argue that when small customers outsource to large suppliers, it is difficult to achieve conditions of mutual interests. Thus offshore vendors might act against the customers’ interest e.g. by prioritizing larger contracts and relocating their expert developers. Another problem is related to the limited control over the vendors, which becomes a hindrance for on demand recruitment, termination, protection, motivation or education of offshore developers.

We also observed that two of the companies fought with symptoms for several years instead of addressing the true causes of their problems. This was probably to self-justify the correctness of an offshore outsourcing decision and because of unwillingness to admit the failure. We thus encourage early engagement in identifying the underlying problems (double-loop learning), as it might save the unnecessary investments and painful failures, like we see with the Guppy case.

In response to “What are the reasons for switching from offshore outsourcing to offshore insourcing?” we conclude that offshore insourcing helped to address many challenges experienced in outsourcing, although some of them remained. This was enabled by a larger control over recruitment, motivation and leadership. Further, in our study the new model gave access to new technology and a new market.

While all companies improved their situation by turning from offshore outsourcing to offshore insourcing, the companies still experienced some challenges in their sourcing relationship. However those problems are not part of this

paper, require more data collection, and will therefore be reported in future research.

The second direction for future research is related to the further identification of factors specific for offshore outsourcing and offshore insourcing, and those that are generally related to utilization of both sourcing strategies. The third direction of future work may also focus on differentiating not only between insourcing versus outsourcing, but also concrete types of work, scope and context of global projects and their suitability for different sourcing strategies, as suggested by Prikładnicki and Audy [20] and Smite and Wholin [23]. As for us, we will continue analyzing additional cases of failed offshore outsourcing relationships and the role of single- and double-loop learning in the way these contracts were terminated.

ACKNOWLEDGEMENTS

This research is funded partly by SINTEF ICT, and the Swedish Knowledge Foundation under the KK-Hög grant 2009/0249 and project BESQ+ grant 2010/0311. We are grateful to Hamish Barney from the University of New South Wales Sydney (UNSW), who participated in the data collection at the Nemo case, and to the reviewers of this paper who gave us valuable feedback.

REFERENCES

- [1] C. Argyris and D.A. Schön, *On Organizational Learning II: Theory, Method and Practise*, Addison Wesley, Reading, MA, 1996.
- [2] H.T. Barney, N.B. Moe, G. Low, and A. Aarum, “Indian Intimacy ends as the Chinese Connection Commences: Changing Offshore Relationships.,” in *Third Global Sourcing Workshop*. Keystone CO, USA, 2009.
- [3] A. Boden, B. Nett, V. Wulf, and I.C. Soc, *Coordination practices in distributed software development of small enterprises*, 2007.
- [4] A.A. Bush, A. Tiwana, and H. Tsuji, “An empirical investigation of the drivers of software outsourcing decisions in Japanese organizations,” *Information and Software Technology*, Publisher, city, 2008, pp. 499-510.
- [5] E. Carmel and P. Tija, *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*, Cambridge University Press, NY, 2005.
- [6] G. Dedene and J.-P.D. Vreese, “Realities of Off-Shore Reengineering,” *IEEE Softw.*, Publisher, city, 1995, pp. 35-45.
- [7] W. DeLone, et al., “Bridging global boundaries for IS project success,” in *Proceedings of the 38th Hawaii International Conference on System Sciences*. Hawaii, 2005.
- [8] M.E. Fayad, M. Laitinen, and R.P. Ward, “Software engineering in the small,” *Communications of the ACM*, Publisher, city, 2000, pp. 115-118.
- [9] T. Hall, N. Baddoo, S. Beecham, H. Robinson, and H. Sharp, “A Systematic Review of Theory Use in Studies Investigating the Motivations of Software Engineers,” *ACM Transactions on Software Engineering and Methodology*, Publisher, city, 2009, pp. 1-29.
- [10] G.K. Hanssen, “A Longitudinal Case Study of an Emerging Software Ecosystem: Implications for Practice and Theory,” *Journal on Systems and Software*, Publisher, city, 2012, pp. 12.
- [11] R. Hirschheim and M. Lacity, “Reducing information systems costs through insourcing: Experiences from the field,” in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, Vol Vi: Organizational Systems and Technology Track, *Proceedings of the Annual Hawaii International Conference on System Sciences*, Watson, H.J., Ed., 1998, pp. 644-653.

- [12] S. Hole and N.B. Moe, "A Case Study of Coordination in Distributed Agile Software Development," *Software Process Improvement*, 2008, pp. 189-200.
- [13] G. Höfner and V.S. Mani, "TAPER: A generic framework for establishing an offshore development center," in *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE)*, 2007.
- [14] M. Keil, J. Mann, and A. Rai, "Why software projects escalate: An empirical analysis and test of four theoretical models," *MIS Quarterly*, Publisher, city, 2000, pp. 631-664.
- [15] M. Lacity, L. Willcocks, and J. Rottman, "Global Outsourcing of Back Office Services: Lessons, Trends and Enduring Challenges," *Strategic Outsourcing: An International Journal*, Publisher, city, 2008, pp. 13 – 34.
- [16] A. Langley, "Strategies for Theorizing from Process Data," vol. 24: *Academy of Management*, 1999, pp. 691-710.
- [17] A.S. Lee and R.L. Baskerville, "Generalizing generalizability in information systems research," *Information Systems Research*, Publisher, city, 2003, pp. 221-243.
- [18] F.C. Lunenburg, "Escalation of Commitment: Patterns of Retrospective Rationality," *International Journal of Management, Business, and Administration*, Publisher, city, 2010, pp. 1- 5.
- [19] M.C. Paulk, C.V. Weber, and M.B. Chrissis, "The Capability Maturity Model for Software," in *Elements of Software Process Assessment & Improvement*, El Emam, K. and N.H. Madhavji, Eds. Los Alamitos, California.: IEEE Computer Society, 1999, pp. 3-22.
- [20] R. Prikladnicki and J.L.N. Audy, "Process models in the practice of distributed software development: A systematic review of the literature," *Information and Software Technology*, Publisher, city, 2010, pp. 779-791.
- [21] R. Prikladnicki, J.L.N. Audy, D. Damian, T.C. de Oliveira, and I.C. Soc, *Distributed Software development: Practices and challenges in different business strategies of offshoring and onshoring*, 2007.
- [22] S. Sahay, B. Nicholson, and S. Krishna, *Global IT Outsourcing: Software Development across Borders*, Cambridge Univ. Press, 2003.
- [23] D. Smite and C. Wohlin, "VOICE OF EVIDENCE A Whisper of Evidence in Global Software Engineering," *IEEE Software*, Publisher, city, 2011, pp. 15-18.
- [24] D. Smite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empirical Software Engineering*, Publisher, city, 2010, pp. 91-118.
- [25] B. Staw, "Knee-deep in the big muddy: A study of escalating commitment to a chosen course of action," *Organizational Behavior and Human Performance*, Publisher, city, 1976, pp. 27-44.
- [26] R.K. Yin, *Case study research: design and methods*, Sage, Thousand Oaks, Calif., 2002.