

# Enabling Agile Offshoring with the Dual-Shore Model

Joachim Sauer  
C1 WPS GmbH  
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany  
js@c1-wps.de

**Abstract:** Conventional offshoring with requirements engineering onshore and implementation offshore is not very agile. Problems often arise in the transfer of requirements and knowledge and in the quality of the developed system. Agile offshoring may be a solution for these problems. But its application involves risks. In this paper common challenges of collaboration in agile offshoring projects are presented. Based on these and reflections on dimensions of distribution and cultural differences, three cooperation models for offshoring are analyzed from a knowledge sharing perspective. Thereby we show how dual-shore offshoring with onshore and offshore development offers a stable organizational foundation for agile offshoring.

## 1 Characteristics and challenges of agile offshoring

The offshoring of application development is an important part of global software development. But many offshoring projects suffer from problems resulting from a reduced communication bandwidth and not very agile development with long feedback cycles. This ultimately leads to inappropriate applications and failed projects [CT05].

The application of agile methods in offshoring projects promises to improve this situation. Agile offshoring (AO) means the application of agile methods and practices in offshore software development projects. Agile offshoring thus is a special case of agile global software development (AGSD).

Agile methods strive for improving flexibility and quality. Requirements are not considered to be carved in stone but may change and lead to adjustments in the developed system. Open communication, incremental releases, frequent feedback and simple, clear processes are core values of most agile methods. Many advantages of agile methods are especially desirable in distributed projects, e. g. early discovery of shortcomings and defects, intense involvement of the later users and better insight into the project progress. Well-known approaches for agile offshoring include Distributed Extreme Programming (DXP) [KJCL02] and Distributed Scrum [SVBP07]. They carry over the fundamentals of frequently used agile methods to distributed projects.

Unfortunately, real agility seems hard to realize in offshoring projects. A main problem lies in the distribution and diversity of the project members that hinders the adoption of interdisciplinary teams and informal communication. We recently carried out an investigation to assess further common challenges of agile offshoring. For this purpose, experience

reports that were published on major agile conferences were analyzed. The depicted challenges were categorized using open coding – a practice from grounded theory [Pan96]. First results of this study have been published in [Sau06].

Here we concentrate on the challenges that are most important for the collaboration of distributed teams. The following issues were identified:

**Team spirit:** Personal contact amongst project members is reduced in consequence of the geographical and temporal distribution. This makes it difficult to cultivate a common team spirit.

**Common vision:** The same reasons also hamper the establishment of a common vision – a set of shared values and principles that facilitates the cooperation.

**Dealing with diverse cultures:** Cultural differences show up on different levels: organizational structures and working habits, country-specific characteristics, and individual behaviors. They can lead to misunderstandings and frustration if they are not dealt with in proper form.

**Language problems:** Another frequent source of misunderstandings is the language. Although all participants of the experience reports' projects spoke English as common language, lacking language skills, different pronunciation, missing body language and different meanings of some words were problematic.

**Informal communication:** Communication during coffee breaks or informal meetings is important for a pleasant atmosphere and mutual friendship and trust. This is hard to accomplish in distributed teams.

**Cooperation and coordination:** All experience reports name cooperation and coordination of activities more or less as main problem areas of distributed teamwork. Issues that could be resolved easily in a colocated environment can take much longer in global software development.

**Knowledge transfer:** Knowledge distribution is critical in global teams. Common knowledge has to be explicitly build up and maintained. This can be challenging as fluctuation of employees is much higher in offshoring countries. The process of introducing changes and news should be planned carefully.

These issues should be addressed in research and practice to improve the outcome of agile offshoring projects.

## 2 Our research objective

Research of communication challenges in distributed projects concentrates on two areas. The literature on CSCW (computer supported cooperative work) deals with different

communication needs and the effects of distance [OO00]. Another important area of research are tools for supporting distributed teams. These can be grouped in the categories communication (e. g. video conference systems, e-mail), common information spaces (e. g. wikis), workflow-management (workflow management systems) and workgroup-computing (e. g. calendars, group editors, meeting support systems) [TSMB95].

In this paper we concentrate on organizational structures of distributed teams. Three different cooperation models are compared in search for better communication in agile offshoring projects. “A **cooperation model** is a usage model that describes and regulates a cooperative situation either explicitly or implicitly.” [Zül04, p. 342] It mainly determines the distribution of the project members to the various project sites. The analyzed models are direct and indirect offshoring as conventional cooperation models for offshoring and dual-shore offshoring as agile solution for some of their shortcomings.

The discussion is based on the identified challenges of agile offshoring, three case studies of real-world projects (one using action research, two interview-based) [KS07] and reflections on dimensions of distribution and cultural differences in offshoring.

We distinguish between six different dimensions of distribution that were derived from the literature ([Gum06], [ECWP03], [AP99]). **Geographic distribution** is the most discussed one that separates offshoring from other forms of cooperative software development. **Temporal distribution** is typically caused by separated time zones, but also by different work rhythms or part-time work. Outsourcing leads to a strong **organizational distribution** between client and service provider. Dimensions not specific to offshoring are **distribution among stakeholder groups** (team members have different roles and functions, education and knowledge), **project distribution** (the project is split into several subprojects that have to be synchronized), and **personal distribution** (team members working on several projects in parallel).

The other focus of this analysis are **cultural differences**. As offshoring projects are characterized by teams working on different continents, team members have their roots in different cultural backgrounds. Cultural differences can hinder the knowledge transfer between teams, can lead to misunderstandings and problems and prevent the building of trust and a common team spirit [KSW04]. They are not limited to national cultures. In software development projects, people from many backgrounds have to work together. They act in different roles with different beliefs and goals: mainly the customer (typically split in business and IT department), analysts, programmers, testers, and project managers.

### 3 Direct and indirect offshoring

The first-generation offshoring model is known as **direct offshoring**. The customer directly interacts with the offshore service provider. This model typically uses a strict separation of tasks: the customer determines the requirements, the service provider does the implementation. This model’s roles and communication channels are depicted in figure 1.

The customer’s IT and business departments have to agree on the requirements and the specification (1). The project manager (PM) and tester (T) typically belong to the IT de-

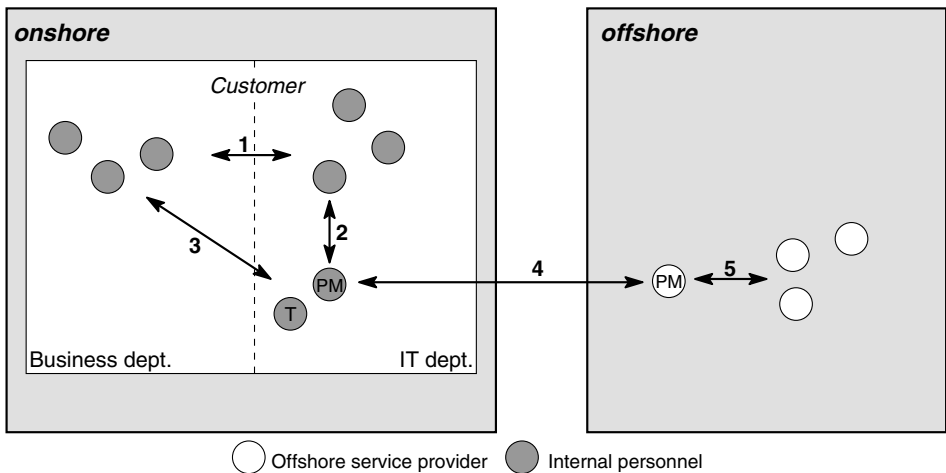


Figure 1: Direct offshoring

partment. They discuss organizational and technological issues with the IT department (2) and business issues with the business department (3). Problems due to different stakeholder groups and work cultures may arise. The most burdened communication channel is the one between the project managers from the customer and the offshore service provider (4). It serves as the only onshore-offshore bridge as well as the interface between requirements and implementation. The project managers have to deal with several challenges in the areas of geographic, temporal and organizational distribution and cultural differences. They have to manage the knowledge transfer progress with few context information and possible awareness difficulties. The software development process at the offshore location is more or less hidden from the customer (5).

The success of this model depends on the correct transfer of the requirements and domain knowledge to the offshore service provider. The geographical and temporal separation as well as cultural differences and language problems can hamper the cooperation between the customer and the offshore service provider severely. The customer has to overcome communication difficulties by himself. If problems arise, only contact persons from the offshore service provider are available.

**Indirect offshoring** has been developed to avoid problems of direct offshoring. In the following, only new communication channels and important changes in comparison with direct offshoring are described. See figure 2.

Instead of directly interacting with the offshore service provider, the customer employs a local external partner who has experience with software development in offshoring. Its personnel works at the onshore location with the customer's IT and business department (2, 3). The external project manager takes over the important part of communicating with the offshore service provider (5), reducing the need of direct interaction between the customer

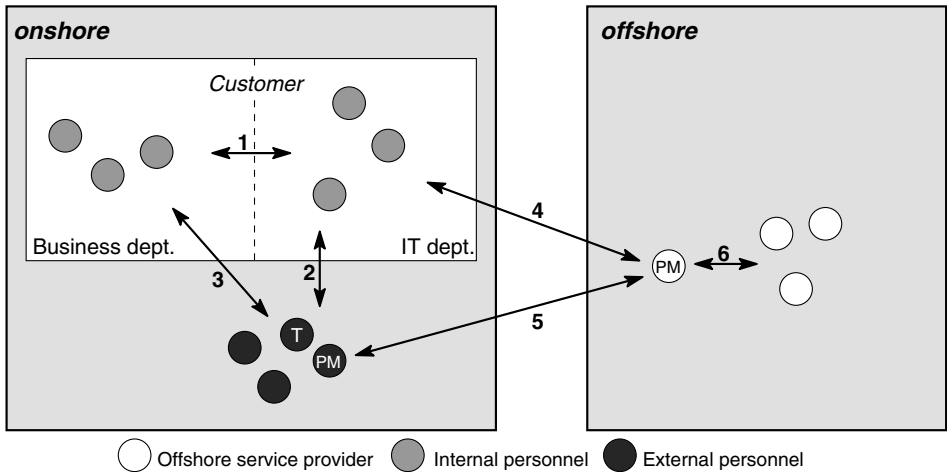


Figure 2: Indirect offshoring

and the offshore service provider (4). This set-up does not change the distribution and cultural challenges and possible overload of the onshore-offshore channel.

The success of this model depends mainly on management, language, cultural and overall skills of the external personnel. If they manage to effectively mediate between the customer and the offshore service provider, many communication problems that would otherwise affect the project can be resolved.

#### 4 Dual-shore offshoring with onshore and offshore development

Dual-shore offshoring is based on the insight that the transfer of requirements and knowledge over considerable distances between people with different cultural backgrounds is very hard. With dual-shore offshoring, employees of the offshore service provider alternate between working offshore and onshore. Optionally, onshore personnel works for a while offshore. This exchange is not limited to developers. It also makes sense to let onshore domain experts and business analysts pass on their knowledge directly at the offshore location.

In our understanding of dual-shore offshoring, development teams work onshore and offshore, coordinating their work and shifting tasks dynamically. Onshore and offshore employees work in mixed teams in close cooperation, directly sharing business and technological knowledge. See figure 3.

With this model, many issues can be resolved without involving the other shore (7, 8). The important onshore-offshore channel is relieved as communication takes place between or with the support of members of the same company, eliminating organizational and cul-

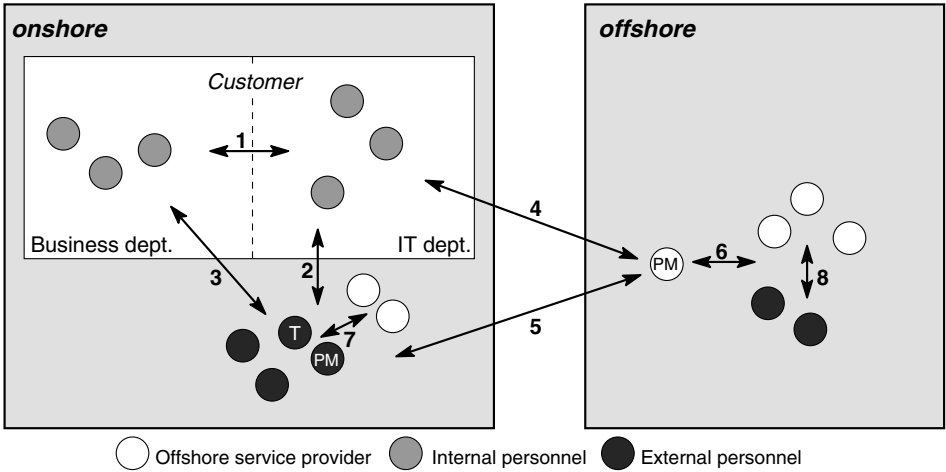


Figure 3: Dual-shore offshoring with onshore and offshore development

tural differences (5). Furthermore, business requirements can be translated onshore into tasks easier understood by developers. Communication can take place from developer to developer, focusing on technological issues.

By now, several companies apply the dual-shore model in agile offshoring projects, including Zensar Technologies [Win05], NIIT Technologies [Ste05] and Valtech (Duo-Shore, TwoShore) [MS03].

## 5 Comparison and conclusion

The direct and indirect offshoring cooperation models should only be employed in small or technology-focused projects. In these projects, the requirements can be unambiguously specified and they as well as the environments remain stable. The offshore service provider should be chosen carefully. Under these circumstances, these conventional forms of offshoring offer the potential of considerable cost reduction due to extensive use of offshore resources. Indirect offshoring is especially advisable if the customer has no prior experience with offshoring. The risk can be further reduced at the expense of higher costs if the external partner is willing to take on part of the responsibility.

Dual-shore offshoring is well suited for agile offshoring projects for the development of complex application software. This comes mainly due to two factors:

1. The relief of the onshore-offshore channel through mixed teams that cooperate directly.

2. Onshore and offshore development that overcomes the separation of requirements engineering and implementation.

Our empirical, qualitative studies have shown that dual-shore offshoring offers many advantages in the areas of communication, coordination and cooperation: easier team building and mutual understanding, better requirements and knowledge transfer, less misunderstandings, more frequent feedback and better controllability for the project manager. On the other hand, dual-shore offshoring with onshore and offshore development imposes the need of global software development with distributed teams. This can introduce new risks and increase the efforts needed for personnel, coordination, travels and working abroad. The advantages and disadvantages of the cooperation models have to be carefully considered for every single project with its specific setting and requirements.

While cooperation models offer instructions for the basic set-up of the project and the cooperation between the teams, they have to be complemented by a detailed concept for everyday communication and work division. Future research work in this area should focus on finding such a suitable concept for dual-shore offshoring to improve agility and outcome of offshoring projects.

## References

- [AP99] Josef Altmann and Gustav Pomberger. Kooperative Softwareentwicklung: Konzepte, Modell und Werkzeuge. In August-Wilhelm Scheer and Markus Nüttgens, editors, *Electronic Business Engineering: 4. Internationale Tagung Wirtschaftsinformatik 1999*, pages 643–664. Physica-Verlag, Heidelberg, 1999.
- [CT05] Erran Carmel and Paul Tjia. *Offshoring Information Technology – Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press, 2005.
- [ECWP03] J. A. Espinosa, J. N. Cummings, J. M. Wilson, and B. M. Pearce. Team boundary issues across multiple global firms. *Journal of Management Information Systems*, 19(4):157–190, 2003.
- [Gum06] Dorina C. Gumm. Distribution Dimensions in Software Development Projects: A Taxonomy. *IEEE Software*, 23(5):45–51, 2006.
- [KJCL02] Michael Kircher, Prashant Jain, Angelo Corsaro, and David L. Levine. Distributed Extreme Programming. In Michele Marchesi, Giancarlo Succi, Don Wells, Laurie Williams, and James D. Wells, editors, *Extreme Programming Perspectives*, chapter 44, pages 553–568. Pearson Education, 2002.
- [KS07] Andreas Kornstädt and Joachim Sauer. Mastering Dual-Shore Development – The Tools & Materials Approach Adapted to Agile Offshoring. In Bertrand Meyer and Mathai Joseph, editors, *Software Engineering Approaches for Offshore and Outsourced Development, First International Conference, SEAFOOD 2007, Zurich, Switzerland, February 5–6, 2007. Revised Papers*, volume 4716 of *Lecture Notes in Computer Science*, pages 83–95. Springer, 2007.
- [KSW04] S. Krishna, Sundeep Sahay, and Geoff Walsham. Managing cross-cultural issues in global software outsourcing. *Commun. ACM*, 47(4):62–66, 2004.

- [MS03] Brad Murphy and Tim Snyder. TwoShore – The Valtech Agile Global Delivery Model, Part 1 – for the Project / Program Director. Whitepaper, Valtech Technologies, 2003.
- [OO00] G. M. Olson and J. S. Olson. Distance Matters. *Human-Computer Interaction*, 15(2&3):139–178, 2000.
- [Pan96] Naresh R. Pandit. The Creation of Theory: A Recent Application of the Grounded Theory Method. *The Qualitative Report*, 2(4), December 1996.
- [Sau06] Joachim Sauer. Agile Practices in Offshore Outsourcing – An Analysis of Published Experiences. In *Proceedings of the 29th Information Systems Research Seminar in Scandinavia, IRIS 29 – Paradigms Politics Paradoxes, August12–15, Helsingoer, Denmark*, 2006.
- [Ste05] Rolf Stephan. Kommunikation und Wissenstransfer – Schlüsselfaktoren für erfolgreiche Offshore Projekte. In Heinz-Josef Hermes and Gerd Schwarz, editors, *Outsourcing. Chancen und Risiken, Erfolgsfaktoren, rechtssichere Umsetzung*, chapter 11, pages 213–234. Haufe, 2005.
- [SVBP07] Jeff Sutherland, Anton Viktorov, Jack Blount, and Nikolai Puntikov. Distributed Scrum: Agile Project Management with Outsourced Development Teams. *hicss*, 0:274a, 2007.
- [TSMB95] Stephanie Teufel, Christian Sauter, Thomas Mühlherr, and Kurt Bauknecht. *Computerunterstützung für die Gruppenarbeit*. Addison-Wesley, Bonn, 1995.
- [Win05] Martin Winkler. Mit dem Dual Shore Delivery Model Sprach- und Kulturbarrieren bei IT-Offshoreprojekten überwinden. In Reiner Clement, Andreas Gadatsch, Martin Kütz, and Jens Juszcak, editors, *IT-Controlling in Forschung und Praxis – Tagungsband zur 2. Fachtagung IT-Controlling, Sankt Augustin, 21. und 22.02.2005*, number 13, pages 127–136, 2005.
- [Zül04] Heinz Züllighoven. *Object-Oriented Construction Handbook: Developing Application-Oriented Software with the Tools & Materials Approach*. dpunkt.verlag. Copublication with Morgan-Kaufmann, 2004.