Collaboration Patterns for Offshore Software Development

Uwe van Heesch, HAN University of Applied Sciences, Arnhem, The Netherlands

Global sourcing in IT projects becomes increasingly important in medium and large-scale software projects to keep up with growth and innovation demands of the industry. Offshore contribution in software projects can reduce cost and development time, but it comes with special challenges that need to be addressed. Apart from large geographical distance, cultural and communication differences need to be anticipated in order to tap the full potential of offshore and onshore collaboration. This paper reports on two collaboration patterns for software projects with offshore contributions, the author observed in industrial practice and in the literature.

Categories and Subject Descriptors: D.2.9 [**Software Engineering**]: Management—*Programming Teams*; D.2.9 [**Software Engineering**]: Management—*Productivity*; D.2.9 [**Software Engineering**]: Management—*Software quality assurance*

General Terms: Human Factors, Management

Additional Key Words and Phrases: Offshore Software Development, India

ACM Reference Format:

van Heesch, U. 2015.Collaboration Patterns for Offshore Software Development. jn 1, 1, Article 1 (January 2015), 10 pages.

1. INTRODUCTION

This paper presents an initial set of patterns for offshore software development. The term *offshore software development* generally refers to software projects, in which the development team is spread across multiple countries [Carmel 1999]. Outsourcing, which is sometimes confused with offshore, does not generally refer to *geographical* reallocation of services (such as software development), but to *organisational* reallocation of services to another company. Constellations, in which this company is also in a different country can be referred to as offshore outsourcing. The patterns in this paper concern offshore development, not outsourcing.

Usually, but not exclusively, offshore software development means collaboration with low-cost locations; i.e. countries, in which the wage-level is significantly lower than in the collaboration partner's location. While several countries from Asia, America, and Europe provide low-cost services for software development, according to the Gartner group, India is still the undeniable leader in offshore services (as of 2010) [Gartner 2010].

Collaboration with offshore partners offers several opportunities, among others high availability and scalability of human resources and significantly lower labor cost. But it also brings along special challenges that need to be addressed to tap the full potential of those benefits. Apart from long geographical distance, team members communicate in a foreign language (usually English), have different communication styles (e.g. minimalistic vs. comprehensive), other cultural differences (e.g. in the area of conflict management or reporting hierarchies), and different views of punctuality and quality.

The patterns presented in this paper document experiences, the author made as project manager and software architect when developing software with strong Indian contribution. To make sure that the identified phenomena are recurring patterns and not just individual experiences, this paper only reports on patterns, the author observed in multiple software projects. Additionally, each pattern refers to literature reporting on similar findings from empirical studies or experience reports.

The pattern descriptions can be used by software developers, managers and architects, who start working in medium and large-scale offshore projects. This catalog may also be helpful for companies planning to start offshore engagements. Finally, it is important to note that the patterns take the perspective of the onshore (in contrast to the offshore) partner. A similar pattern catalog could be envisioned from the perspective of an Indian software vendor, for instance.

2. USED PATTERN FORMAT

The pattern format used in this paper is based on the format used by Gamma et al. [Gamma 1995] and kept simple deliberately. The following elements recur in each pattern description:

Pattern name*.* A concise term representing the pattern's driving characteristics.

In short*.* A one-sentence representation of the pattern.

Problem*.* A description of the problem solved by the pattern and the context, in which it is applicable. Generally, all patterns can be applied in offshore software projects, but sometimes the context is more specific. The problem statement is refined by a number of factors that need to be taken into consideration when solving the problem.

Solution*.* An outline of the solution and the potential consequences that arise when applying the solution.

Related patterns*.* Existing patterns that provide similar solutions.

Related literature*.* A brief discussion of publications describing phenomena similar to the described pattern.

3. SETTING UP THE TEAM

This category contains successful patterns of team setups.

3.1 Experience Mix

In short: When staffing offshore software projects, make sure you mix experienced offshore collaborators with non-experienced developers throughout the team.

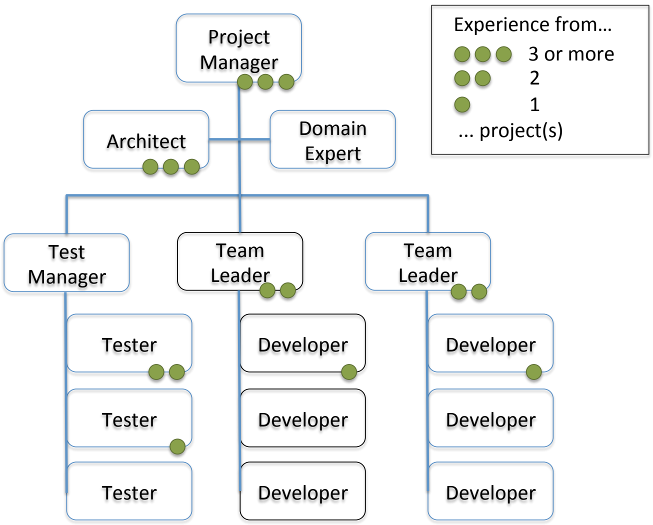

Fig. 1. Offshore collaboration experience distribution in the team

3.1.1 *Problem.* **How do you assign people to a mixed onshore-offshore team in a way that offshore collaboration experience is best exploited?** When software projects are staffed, it is a common practice to mix senior software engineers with junior software engineers. Apart from technical and general project management experience and expertise, in mixed onshore-offshore teams, the offshore collaboration experience of the team members needs to be considered. A successful team composition has to take the following forces into account:

(1) Especially when companies start getting engaged in offshore software development, only few people are experienced in offshore collaboration.

(2) Collaboration across geographical boundaries requires other ways of working (e.g. related to remote communication, travelling, used language, working in different time zones) than co-located software development. Teams need to get used to these working conditions in order to tap their full potential.

(3) You need members in your team with good technical expertise to tackle tough technical problems.

(4) Ideally, all team members have good communication skills and are open to foreign cultures. Developers who have had prejudices against offshore collaboration for a long time may not become successful team players.

(5) Only experienced team members know how to avoid many common pitfalls in offshore collaboration.

(6) Each function within a team (e.g. project manager, architect, tester, developer, domain expert) has its own special challenges in offshore projects.

3.1.2 *Solution.* Select team members in a way that existing offshore experience is spread over different roles in your project team. Figure 1 exemplifies such a distribution for a medium-size team. The project manager and architect are most experienced in offshore coordination, whereas the sub-teams can cope with non-offshore-experienced team members, as well.

In smaller teams, it is vital to have at least one experienced offshore collaborator on both shores. It is not advisable to start an offshore projects without at least a few experienced people. Ideally, people with management tasks (e.g. project managers, architects, or team leads) are already experienced in offshore collaboration. Key resources should be highly experienced already, as they have to take care of the team running smoothly. Each sub-team (e.g. working collaboratively on a task) can cope with and train multiple inexperienced team members, if two or more experienced resources are also part of the team. You only need very few highly experienced members to take over key positions in the team. The rest of the experienced team members should be evenly distributed over the team. The greatest number of positions in the team can be staffed with resources who do not have offshore collaboration experience at all. Note, that this entire distribution model is independent of shores. It applies to onshore and offshore members similarly.

Following these suggestions,

(1) You use developers who made experience in earlier offshore projects as multipliers to grow an experienced offshore resource pool in your company.
(2) The experienced team members naturally pass knowledge to the inexperienced team members, either by active mentorship, or indirectly by acting as role models.
(3) You can still assign technically skilled resources even if they don't have offshore experience already.
(4) The experienced key resources consider known pitfalls and challenges, inherent in offshore development.
(5) If each type of position has experienced, as well as in-experienced team members, the special offshore challenges in those positions are sufficiently covered.

Depending on the concrete context, the following factors may negatively impact the successful application of the solution:

(1) When a company starts getting involved in offshore development, chances are high that only very few resources have offshore experience. In such cases, having a balanced mix as advocated above is not possible.
(2) Staffing software projects is a complex task in itself. Introducing offshore experience as another factor to be considered on top of availability, costs, employment level and technical skills makes this task even more complex.
(3) Even if the team has a balanced mix of offshore experience, some team members do not have the required soft skills. They will always work better in a co-located team.

3.1.3 *Related Patterns.* The following table presents existing patterns related to (aspects of) this pattern:

Table I. : Related Patterns

Name	**Description**
TAILORED TRAINING [Lescher 2010]	Conducting a co-located tailored training at the beginning of a project is an alternative, especially if the vast majority of the team is inexperienced in offshore development.
PREPARED TROOPS [Bricout et al. 2004]	Train all resources on distributed development from the start of the project.

PREPARED COMPANY [Bricout et al. 2004]	Structure and processes in a company may need to be adjusted to support distributed teams.
CAREFULLY SELECTED [Bricout et al. 2004]	Screen team members for personality and carefully built the team over time for maximum cohesion.

3.1.4 *Related Literature.* The selection of capable and experienced team members in offshore teams is an important factor that influences the success or failure of the project [Fabriek et al. 2007][Dibbern et al. 2008]. Gopal et al. found that the number of team members with offshore experience influences the amount of rework required after releases [Gopal et al. 2002]. Experienced offshore development resources have an important role in bridging cultures [Winkler et al. 2008]. Using mixed-experienced teams as a means to spread and multiply knowledge in offshore teams is also discussed by Nicholson and Sahay, who generally analyse the issue of knowledge management in offshore software development [Nicholson and Sahay 2004].

3.2 DISSOLVE GEOGRAPHICAL BOUNDARIES

In short: When staffing offshore software projects, assign responsibilities in your team in a way that management or reporting lines regularly cross geographical boundaries.

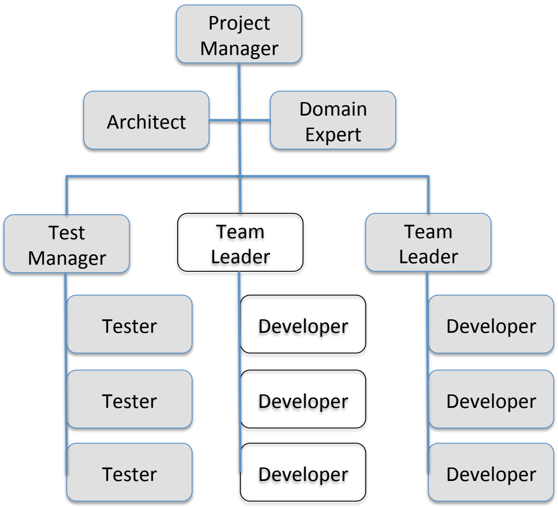

Fig. 2. An isolated offshore team (no fill) within a team

3.2.1 *Problem.* **How do you assign key roles and responsibilities in a mixed onshore-offshore project team?**

Compared to single-shore software projects, when staffing development teams with offshore contribution, the geographical distance between the team members is an important factor to be considered. A successful team composition needs to balance the following forces:

(1) As the geographical boundary between the offshore and the onshore part of the team forms a natural communication impediment, Conway's law would advocate to split the team at this boundary and organize management and reporting lines along it, thereby reducing communication overhead. One example of such a team construction can be seen in Figure 2. Although the organigram shows one software development team, the offshore team (transparent background) is isolated within this team.

(2) Co-located team members naturally and informally share information and support each other. This has not only been embraced as an important principle by the authors of the agile manifesto [Beck et al. 2001]. If the co-located offshore members of the team also form a logical sub-team with fix-scoped responsibilities, it is likely that they become an island within the project team. In that case, information exchange with the onshore team members will likely be tunnelled through the highest ranked team-member in the offshore sub-team who communicates to onshore members. This hampers exchange of important information and knowledge in the team.

(3) Isolated sub-teams may feel responsible for their tasks in the project only, instead of identifying themselves with the project as a whole. As a consequence, the onshore and offshore team divisions may shift the blame onto each other in case of problems with the integration of sub-systems. Even in absence of severe problems, the team does not tap the full potential of all team members. Naturally different persons have different skills

and competences. If the team setup contains isolated sub-teams of co-located developers, this potential is limited to the sub-team boundaries.

(4) If there are only a few team-members who communicate with both onshore and offshore team members, these team members become critical team resources. In case of illness or holidays, the project may significantly suffer from their absence, because the offshore project team is de-coupled from the rest of the team, at least in the perceptions of the developers. A replacement can only partially bridge this gap, because establishing trustful and efficient communication partnerships takes time.

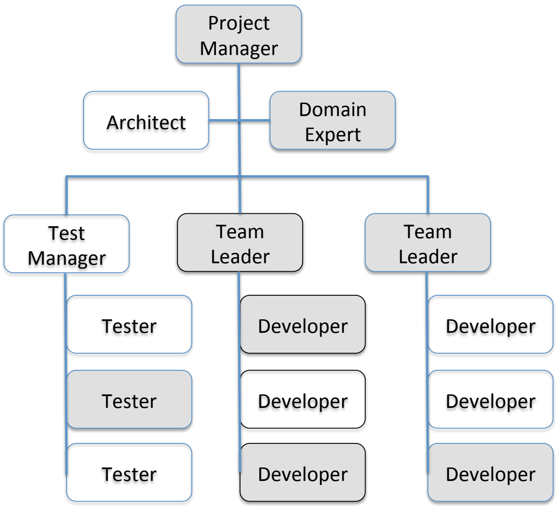

Fig. 3. One merged team with mixed offshore and onshore members

3.2.2 *Solution.* Merge the two geographical parts of your team into one unit. Especially in teams spread over multiple continents, establishing a team of co-workers that feels and acts as a whole is important. Naturally, in the beginning there will always be an *offshore team* and an *onshore team*. In order to merge the two separate units into one unified team you have to bridge the geographical distance in peoples' heads. To achieve this:

(1) Dissolve national boundaries. Organize your team in a way that geographical location is no distinguishing factor between sub-teams any more. Figure 3 shows an exemplary organigram in which national boundaries are regularly crossed in sub-teams. As a consequence, knowledge, skills, and important information are naturally spread within the team, i.e. between co-located team members and within sub-teams. Make sure that key positions are spread over both parties.

(2) Organize tasks and responsibilities in a way that responsibilities are shared across national boundaries. Assign the most suitable person to a task independent of the geographical location. Ideally none of the team members collaborates with her/his own shore exclusively.

(3) If possible, write a project governance plan collaboratively with onshore and offshore members. At least make sure that key members of both shores approve the plan to get a mutual commitment.

(4) When speaking with or about the team, always refer to the team as a whole, do not speak of the *offshore team* or the *onshore team*. Otherwise the team, but also the customer may always treat the onshore and offshore team members separately.

(5) Share success and failure with the entire team. Do no communicate things to one part of the team and then later to the other part. Rather organize a telephone or video conference to share information with members from both shores at the same time.

(6) If possible, organise a project kick-off meeting, in which all team members meet in person. People collaborate better when assigning faces and facial expressions to voices in calls.

(7) Regularly enable different team members to travel between shores. This should not only happen from onshore to offshore, but also the other way round. That way, team members get to know each others living and working conditions better, which helps significantly in building a community of trust.

Following these suggestions,

(1) Information will efficiently be shared among all team members, because offshore and onshore members meat each other regularly in sub-team meetings and working session, while information is additionally shared on the office floors.

(2) The team members are more likely to identify with success and failure of the entire project team, rather than having the notion of "we" and "them" in their minds. Skills and competences are optimally spread in the team and over tasks.

(3) Single communication proxies between shores are avoided, therefore the risk of communication bottlenecks is significantly reduced.

(4) Including team members on the other shore in all important communication requires a lot of discipline from team members. Important communication should not happen on the office floor, but in conference calls.

(5) Some people do not feel comfortable when face-to-face contact with co-workers is missing. Especially situations, in which team members have never met in person, can make it hard to build a community of mutual trust.

3.2.3 *Related Patterns.* The following table presents existing patterns related to (aspects of) this pattern:

Table II. : Related Patterns

Name	**Description**
EARLY BONDING [Bricout et al. 2004]	Spend time together face to face in social situations to improve team binding.
CROSS-SITE DELEGATION [Lescher 2010]	Delegate individuals between sites.
FACE-TO-FACE EVERY 2 MONTHS [Bricout et al. 2004]	Face to face meetings every 2 months.
DISTRIBUTED CREDIT [Bricout et al. 2004]	All team members get credited for success.
ONE PROJECT [Bricout et al. 2004]	Run the team as a single team with a single person in charge, despite not being co-located.

3.2.4 *Related Literature.* The problem of status differences in offshore collaboration is extensively studied in [Levina and Vaast 2008]. According to Fabriek et al., offshore projects can fail because geographical and organizational distance negatively influences communication and knowledge exchange between onshore and offshore project team members [Fabriek and Brinkkemper 2008]. Kwan et al. revisited Conway’s law and suggest that in software development, teams should rather be aligned at the level of development tasks and teams, instead of at the level of organization [Kwan et al. 2012]. The role of people in global software projects is also discussed by Misra et al., who advocate among others, to design a long term mobility program for onshore and offshore members of the team [Misra et al. 2013].

4. FUTURE WORK

This paper is a first effort to describe several offshore software development patterns, mined in several software projects and from the literature. We identified the following categories of offshore software development patterns, which will subsequently be codified in the future.

Setting up the team. Patterns for team structuring and the assignment of roles and responsibilities.

Defining Reporting and Meeting activities. Patterns for organising communication and monitoring in the team

Defining tasks and estimating efforts. Patterns for effort and task planning

Bridging culture. Patterns for dealing with culturally-mixed teams

Managing quality. Patterns for dealing with quality management across shores

Managing knowledge. Patterns for knowledge and skill management

Acknowledgements

Collaborating with Indian software engineers has been a wonderful and rich experience, both on professional and personal level. Travelling to India and getting involved in Indian work life and culture to me are indispensable factors for successful collaboration as one team. I am truly grateful for the wonderful people I met and made friends with.

I would also like to thank my EuroPLoP shepherd Allan Kelly for critically assessing the paper, challenging its argumentation and for providing excellent suggestions for improvement.

REFERENCES

BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J., MARICK, B., MARTIN, R. C., MELLOR, S., SCHWABER, K., SUTHERLAND, J., AND THOMAS, D. 2001. Manifesto for agile software development. http://www.agilemanifesto.org/.

BRICOUT, V., HELIOT, D., CRETOIU, A., YANG, Y., SIMIEN, T., AND HVATUM, L. 2004. Patterns for managing distributed product development teams. Tech. rep., Schlumberger Oilfield Services.

CARMEL, E. 1999. *Global software teams: collaborating across borders and time zones.* Prentice Hall PTR.

DIBBERN, J., WINKLER, J., AND HEINZL, A. 2008. Explaining variations in client extra costs between software projects offshored to india. *MIS Quarterly 32,* 2, 333–366.

FABRIEK, M. AND BRINKKEMPER, S. 2008. Reasons for success and failure in offshore software development projects. In *Proceedings of the EUROPEAN CONFERENCE ON INFORMATION SYSTEMS (ECIS).*

FABRIEK, M., VAN DEN BRAND, M., BRINKKEMPER, S., HARMSEN, F., AND HELMS, R. 2007. Improving offshore communication by choosing the right coordination strategy. Tech. rep., Utrecht University, The Netherlands.

GAMMA, E. 1995. *Design patterns : elements of reusable object-oriented software.* Addison-Wesley professional computing series. Addison-Wesley.

GARTNER. 2010. Gartner's 30 leading locations for offshore services, 2010-2011. Tech. rep., Gartner, http://www.gartner.com/resId=1491316.

GOPAL, A., MUKHOPADHYAY, T., AND KRISHNAN, M. S. 2002. The role of software processes and communication in offshore software development. *Communications of the ACM 45,* 4, 193–200.

KWAN, I., CATALDO, M., AND DAMIAN, D. 2012. Conway's law revisited: The evidence for a task-based perspective. *IEEE Software 29,* 1, 90–93.

LESCHER, C. 2010. Patterns for global development: how to build one global team? In *Proceedings of the 15th European Conference on Pattern Languages of Programs.* ACM, 6.

LEVINA, N. AND VAAST, E. 2008. Innovating or doing as told? status differences and overlapping boundaries in offshore collaboration. *MIS Quarterly 32,* 2, 307–332.

MISRA, S., COLOMO-PALACIOS, R., PUSATLI, T., AND SOTO-ACOSTA, P. 2013. A discussion on the role of people in global software development. *Tehnički vjesnik 20,* 3, 525–531.

NICHOLSON, B. AND SAHAY, S. 2004. Embedded knowledge and offshore software development. *Information and organization 14,* 4, 329–365.

Winkler, J. K., Dibbern, J., and Heinzl, A. 2008. The impact of cultural differences in offshore outsourcing–case study results from german—indian application development projects. *Information Systems Frontiers 10,* 2, 243–258.