

Articulation Work in Small-Scale Offshore Software Development Projects

Alexander Boden

Institute for Information Systems
University of Siegen
Hölderlinstr. 3
D-57068 Siegen
+49 (0)271 740-4763

Bernhard Nett

Fraunhofer Institute for Applied
Information Technology
Schloss Birlinghoven
D-53754 St. Augustin
+49 (0)2241 14-1598

Volker Wulf

Institute for Information Systems
University of Siegen
Hölderlinstr. 3
D-57068 Siegen
+49 (0)271 740-2910

alexander.boden@uni-siegen.de bernhard.nett@fit.fraunhofer.de volker.wulf@uni-siegen.de

ABSTRACT

The paper describes coordination practices of small and medium enterprises (SME) that outsourced parts of their software development to offshore vendors. Our case study in a German company shows that the geographic distribution of knowledge and expertise requires continuous informal coordination across the teams. Such practices can be described as a combination of formal coordination and informal *articulation work* in the sense of Anselm Strauss' theoretical framework. Both are closely interwoven and connected to the use of media such as coordination tools and instant messengers. We conclude that SME risk losing their core competencies when neglecting articulation work and hint at mismatches between scientific recommendations and actual development practices of SME.

Categories and Subject Descriptors

J.4.4 [Computer Applications]: Social and behavioral sciences – sociology.

General Terms: Management, Human Factors, Theory.

Keywords: Offshoring, Nearshoring, Articulation Work, Ethnography, Coordination, CSCW, SME

1. INTRODUCTION

Offshore-outsourcing (abbrev. *offshoring*) is an issue of growing importance for small and medium enterprises (SME). An increasing number of German SME engage in offshoring of software development to regions in Eastern Europe [1]. SME are often described as being strongly dependent on a flexible use of coordination methods, work distribution and hierarchies. Small team sizes allow for a high degree of flexibility and customer retention which are often regarded as being the core business advantages of SME especially compared to large competitors [2].

The maintaining of flexibility in the context of offshore software development is confronted with problems of the temporal and

spatial distribution of cooperative work as well as cultural and language-related aspects [3]. At the same time, agile development methods of SME are highly dependent on exhaustive communication and team interaction and thus especially vulnerable to offshoring-related conditions [4]. Since empirical evidence is still sparse we want to contribute to the discussion of offshoring by presenting an ethnographic field study that centers on SME-specific aspects of cooperative work in the context of offshoring. In doing so, we draw on Anselm Strauss' *articulation work* framework [5] and center on formal and informal coordination practices, which play an important role for cooperative software development [6].

2. ARTICULATION WORK

Articulation work has been widely used as a framework for ethnographic research into collaborative work environments and is regarded as a core concept of CSCW studies [7]. Strauss describes articulation work as “the coordination of lines of work, (...) accomplished by means of the interactional processes of working out and carrying through of work-related arrangements” [8]. Articulation work is needed to regulate the distribution of tasks: who does what, when, where, how, with which quality, until when etc. Yet, articulation work is different from coordination: while the latter governs distributed tasks, articulation work manages the consequences of the related distributed nature of work [9]. Thus, articulation work comprises also important aspects of self-organization and their integration into formal models of work distribution [10].

With regard to geographically distributed software development of SME it is important to note that small local teams often are able to articulate their work by means of everyday social interaction. Thus, articulation work often remains invisible and is not acknowledged as being part of the work itself [11]. However, when complexity of project work and geographical distance increase, usual social interactions may no longer be sufficient. Thus, the need for efficient articulation evolves as an additional challenge and the actors are confronted with a new and sometimes unexpected complexity.

In order to articulate distributed project work, actors need access to appropriate means of communication. Thus, being able to engage in informal communication seems to be a key success factor of distributed software development [3]. However, with geographic distance, vital informal communication becomes less frequent and can pose obstacles for efficient articulation work. Proposed solutions usually center on two different strategies:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHASE'08, May 13, 2008, Leipzig, Germany.
Copyright 2008 ACM 978-1-60558-039-5/08/05...\$5.00.

reducing the need for frequent informal (usually by means of formalization and standardization) or technically supporting informal communication, for example by supporting concepts like awareness, or by a targeted combination of various communication channels [11]. Usually this is to be accomplished by using specialized tools like groupware applications, which are proposed for the support of articulation work in distributed work environments [12].

Considering the high importance of SME for the German software industry it seems to be important to gain more insight into distributed cooperative work practices and the related use of tools. As articulation work is strongly dependent on situated discursive practices of cooperating developers it can be analyzed only by taking a practice-based perspective.

3. RESEARCH METHODS

We started our research with an exhaustive analysis of the literature on offshoring, covering discourses of various communities of practitioners and scientists. Based on our findings, relevant research questions were identified which centered on articulation work in form of coordination and communication in SME as well as general assumptions of German developers and managers concerning offshoring of software development.

To collect data, we drew on a triangulation of several ethnographic research methods, comprising semi-structured interviews, participant observation and artifact analysis. For the participant observation, two German SME were visited for a period of ten working days each. A third participant observation was conducted at the Russian partner company of one of our sample companies. During our stay we had many opportunities to observe local and distributed articulation processes in the context of meetings, individual work situations and cooperative tasks. Several informal interviews were conducted and we were allowed to analyze artifacts such as e-mails, chat protocols, internal work papers and whiteboard sketches. The findings were documented by means of field notes and photos which were taken during the research.

For the analysis of the collected data we drew on Glaser's and Strauss' Grounded Theory [14]. After each research step, the transcripts of the material, both field notes as well as interview data, were scrutinized. Our aim was to identify articulation processes in the context of the work trajectories and to recognize their meaning for the actors. Data was coded during a process that consisted of several stages. At first, we composed categories based on the findings in the collected data. Then these categories were related to each other and evolved during the further research.

Our findings were correlated to the literature with a focus on articulation processes during software development. Thus, we concentrated on relations between formal work organization and the actual work processes, which were not considered as mechanical "performance" of formal specifications but as creative reaction to situated work contexts [14]. In order to relate our results to a broader context we also conducted semi-structured interviews with thirteen managers and developers of additional German SME, two interviews with one representative of an IT industry association and one representative of a large German

company as well as four interviews with Eastern-European offshoring vendors.

4. CASE STUDIES

The companies in our field study were German SME that engage in offshoring of parts of their software development. They will be referred to as Alpha and Beta in the following descriptions.

Company Alpha provides data processing products and services in the field of statistic and documentation. Most of the approximately 20 employees of the company are software developers who work in several teams on different projects. The products comprise databases, documentation and presentation systems used by cultural establishments like archives or museums, the services are offered around the use and adaptation of these products. Since the mid-1990s, the company has been employing four software developers in Tomsk, Siberia. The basis for this decision was an internship of a competent Russian developer, who still works for the company. Based on this positive experience, the decision to engage in offshoring was taken and further offshore developers were hired. The first project aimed at the reengineering of an existing software product and was led independently by an offshore project leader who directly reported to the German manager. The offshore project leader also was responsible for the communication with the German customers, providing support and investigating bug reports. Despite unexpected delays in the development, the offshoring was expanded to several other projects which involve a closer cooperation between German project leaders and offshore developers.

Company Beta offers a standard software solution (developed in two different branches) for process modeling and services in the field of process management. The management is based in Bonn. Four offshore developers in Saint Petersburg, who are managed by a German project leader in Berlin, carry out the software development. Seven further employees at the office in Berlin provide testing and support. According to the German manager the decision for offshoring mainly aimed at the reduction of development costs. Due to personal contacts to a Russian developer, the company founded a branch office in Saint Petersburg in 2002. At first, the new Russian employees were invited to Germany. They were able to take over the software development after a couple of months. Since then, up to eleven developers have worked for the offshore company which is nowadays owned by the Russian senior developer. Recently, Beta has reduced the size of the offshore team to four as a reaction to problems of coordination and poor efficiency.

5. RESULTS

Both companies develop software in close cooperation between German project leaders and Russian developers, who act as extended team members. During our participant observation we were able to observe several stages of offshore projects; the related coordination practices and deployed tools will be presented here in form of three short examples.

5.1 Planning of Offshore Projects

It is the general strategy of both companies to let the German project leaders handle the contact with the customers. The project leaders then have to "translate" the customer's needs into

specifications for the offshore developers. For the planning phase both companies prefer to meet face-to-face whenever possible.

During our participant observation in Alpha we were able to accompany a German project leader visiting Tomsk for the planning of a new project. The project leader had already negotiated with the customer based on rough estimates; now he wanted to evaluate the new product in more detail. In several meetings the project leader, the assigned Russian developer and the Russian team leader discussed features, technical obstacles and the estimated workload of the project. In doing so, the project leader mainly tried to explain the customer's needs to the offshore developers. These, in turn, offered technical solutions for the actual implementation. For the discussion, a whiteboard was used and the developer also took down notes into a diary. The results of the meetings were formulated in a Microsoft Word document with detailed specifications (written by the Russian Team leader) as a basis for the development and a rough project plan for the customer prepared in Microsoft Excel (written by the German project leader).

Company Beta handles the specification of features in a similar way. Actually the German project leader should write specifications and store them into a central Lotus-Notes database. However, as this is not always possible due to time constraints (especially in the past when the small German team had to write specifications quickly enough to keep eleven offshore developers busy), Russian developers now often have to write their own specifications which are explained to them by the German project leader preferably during personal visits.

5.2 Maintenance of Software Development

Bug-tracking systems serve in both companies for the cataloguing and assignment of bugs during the development and maintenance of the software. Alpha uses the open-source system Mantis, while Beta relies on the tool SQA. Company Beta follows a clearly formalized procedure of handling bugs. When a German tester finds a bug (or a customer reports one to the support team), he tries to reproduce it and writes a standardized bug description. The bug is then stored into the bug-tracking system and assigned to an offshore developer, who in turn will do the bug-fixing. When this is done, the developer will mark the bug as "fixed", thus assigning it back to the responsible tester, who will investigate the results and mark the bug either as "closed" or reassign it.

While the formal handling of bugs clearly aims at the reduction of communication (and at providing an overview of the state of the project work) it became obvious during the participant observation that the use of bug-tracking systems is often accompanied by the ample use of instant messengers. Thus, if a German tester finds a bug and puts it into the database he will at first contact a Russian developer and draw his attention to the problem. The offshore developer then has time to investigate the error and make some comments concerning the technical reasons or the severity of the problem. The tester uses such information to estimate the relevance of the bug together with the project leader for the whole development process and to decide how much effort should be put into the fixing. The results of the discussion then are communicated back to Russia before the bug is actually stored and assigned using the database.

Instant messengers are also used to track the progress of the project work. In theory, the German project leader of Beta should be able to get an overview of the state of the development by investigating the bug-tracking systems (where open and closed bugs are stored) and the specifications in the Lotus-Notes database (where developers should track their work progress and encountered problems in form of diary-like records). However, participant observation revealed that in practice the systems are often not up-to-date. As Russian developers have to track the changes for themselves and often not do so, the Project leader has to request the status of work regularly and remind the developers to update the information in the systems.

5.3 Inspecting Results

When projects are approaching finalization, both companies prefer to meet face-to-face again. Due to the fact that the last days before the final product is finished often involve a great deal of coordination, all practitioners reported personal visits as the best way to deal with the many small bugs and problems that are expected. During the projects, intermediate results like finished new features often are inspected by prototypes or using desktop sharing, as was observed in company Beta.

Beta uses Sametime, an add-on for Lotus-Notes that offers communication functionality of an instant messenger and also allows desktop sharing. Thus, the project leader was able to take control of the mouse and screen of the Russian developer and test the newly implemented features in this way. The inspection was accompanied by a chat discussion and took nearly three hours. The subjects of the discussion were the feature specifications in the database, which were worked off feature by feature. In the chat the project leader commented the new features and posed questions regarding their expected functionality based on the specifications in the database. He also asked the developer about the progress of other features, and reminded him to track his work in the product database.

6. DISCUSSION

Our field studies provide insights into the role of articulation work for offshore development projects of German SME. Despite the use of formal coordination tools like bug-tracking systems and central feature databases, informal communication and discussion are important parts of everyday project work and mainly take place using instant messengers. The conjointly discussed estimates concerning the relevance and technical reason of errors as well as the effort of fixing them are essential for the deployment of bug-tracking systems [cf. 15]. Similarly, the shared understanding that results from joint discussions during personal visits are important for producing specifications and project estimates as well as for the implementation in later stages of the projects. These findings correlate with related work from the literature that stresses the importance of informal face-to-face communication for cooperative software development [5, 16].

Despite emerging aspects of offshore software development [17] our findings also hint to an inherent dilemma of distributed teams: The spatial distribution of teams can also lead to an allocation of competences which are needed to write technical specifications and bug descriptions. As the offshore team conducts the whole software development of Beta, German testers apparently lack the competence to judge the possible technical reason for encountered

bugs, thus relying on informal communication with the responsible developers abroad. The offshore developers, on the other hand, are not able to assess the importance of certain features or bugs because they are not as familiar with the customer's needs as the German project leaders or testers.

At the same time, writing more ample documentation, tracking the progress of development and updating bug databases seems to be problematic. Thus, we often observed situations where the German project leader had to remind the offshore developers to update their status reports. The maintaining of proper and exhaustive documentation is work-intensive and requires active efforts of all team members. As time constraints often seem to keep developers from updating the systems, the SME allow their developers a greater deal of autonomy, thus trading off trust against control. However, the resulting higher relevance of informal articulation work is regarded as being beneficial as it allows for a greater deal of flexibility and is reported to augment the motivation of the employees [cf. 18]—a positive aspect, that was usually not considered beforehand

7. CONCLUSION

Our findings suggest that articulation work can be an important factor for SME that need to maintain a high level of flexibility even if engaged in offshoring of software development. Thus, companies can benefit from aiming at a balanced relation between formal coordination and informal articulation work instead of unilaterally trying to formalize their development processes.

It seems to be constructive to support the articulation practices we observed by means of adequate tools. However, the SME in our sample rather rely on flexible communication media like instant messengers, while specialized groupware tools are not commonly used. Thus, our observations hint at mismatches between scientific recommendations and the practices of software development in German SME that should be explored further.

We are currently expanding our data basis to get a more detailed understanding of articulation practices in SME. During our interviews and observations many practitioners in our sample considered using open and extendable IDE like Eclipse for future projects. It will be interesting to observe if and how developers actually adapt the groupware-plugins that are currently available or in the process of development (like IBM's Eclipse-based collaboration platform JAZZ), and how these technical changes will effect future development practices.

8. REFERENCES

- [1] Aspray, W., Mayadas, F. and Vardi, M. Y. 2006. Globalization and Offshoring of Software: A Report of the ACM Job Migration Task Force. ACM.
- [2] BMBF 2004. Analyse und Evaluation der Softwareentwicklung in Deutschland. 2000, DOI=<http://www.isi.fhg.de/publ/downloads/isi00b69/software.pdf>
- [3] Herbsleb, J. D., Paulish, D. J. and Bass, M. 2005. Global Software Development at Siemens: Experience from Nine Projects. In Proceedings of the International Conference on Software Engineering (St. Louis).
- [4] Ramesh, B., Cao, L., Mohan, K. and Xu, P. 2006. Can distributed software development be agile? Communications of the ACM 49, 41-46.
- [5] Strauss, A. L. 1988. The Articulation of Project Work: An Organizational Process. The Sociological Quarterly 29, 163-178.
- [6] Redmiles, D.; van der Hoek, A.; Al-Ani, B.; Hildebrand, T.; Quirk, S.; Sarma, A.; Silva Filho, R. S.; de Souza, C. and Trainer, E. 2007. Continuous Coordination: A New Paradigm to Support Globally Distributed Software Development Projects. Wirtschaftsinformatik 49, S. 28-38.
- [7] Schmidt, K. and Bannon, L. Taking CSCW Seriously: Supporting Articulation Work. Computer Supported Cooperative Work (CSCW): An international Journal, 1, 1 (1992), 7-40.
- [8] Strauss, A. L. 1993. Continual permutations of action. Aldine de Gruyter.
- [9] Star, S. L. and Strauss, A. L. 1999. Layers of Silence, Areas of Voice: The Ecology of Visible and Invisible Work. Computer Supported Cooperative Work 8, 9-30.
- [10] Schmidt, K. and Simone, C. 1996. Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. Computer Supported Cooperative Work: The Journal of Collaborative Computing 5, 155-200.
- [11] Damian, D. D., Lanubile, F. and Mallardo, T. 2006. The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations. In Proceedings of the International Conference on Software Engineering (Shanghai).
- [12] Crabtree, A., O'Neill, J., Tolmie, P., Castellani, S., Colombino, T. and Grasso, A. 2006. The Practical Indispensability of Articulation Work to Immediate and Remote Help-giving. In Proceedings of the International Conference on Computer Supported Cooperative Work (Banff).
- [13] Strauss, A. L. and Corbin, J. M. 1998. Basics of qualitative research: techniques and procedures for developing grounded theory. Sage Publications.
- [14] Suchman, L. A. 1987. Plans and situated actions. The problem of human-machine communication. Cambridge University Press.
- [15] Fitzpatrick, G., Marshall, P. and Phillips, A. 2006. CVS Integration with Notification and Chat: Lightweight Software Team Collaboration. In Proceedings of the International Conference on Computer Supported Cooperative work (Banff).
- [16] Olson, G. M. and Olson, J. S. 2000. Distance Matters. Human-Computer Interaction 15, 139-178.
- [17] Boden, A., Nett, B. and Wulf, V. 2007. Coordination Practices in Distributed Software Development of Small Enterprises. In Proceedings of the International Conference on Global Software Engineering (Munich), 235-244.
- [18] Nonaka, I. and Takeuchi, H. 1995. The knowledge-creating company: how Japanese companies create the dynamics of innovation. Oxford University Press.