

Offshoring in kleinen und mittleren Unternehmen der Software-Industrie

Alexander Boden, Bernhard Nett und Volker Wulf

Um ihre Wettbewerbsvorteile nicht zu verlieren, sind kleine und mittlere Unternehmen (KMU) der Software-Industrie beim Offshoring auf ein hohes Maß an Flexibilität angewiesen. Die häufig für Offshoring empfohlene Formalisierung der Softwareentwicklung ist daher für KMU nur mit Einschränkungen möglich. Um die praxisrelevanten Probleme deutscher KMU im Kontext von Offshoring besser zu verstehen, untersuchen wir diese mittels qualitativer empirischer Forschungsmethoden. Wir zeigen anhand einer Einzelfallstudie, dass KMU im Rahmen von Offshoring häufig gezwungen sind, ihren Offshore-Entwicklern mehr Autonomie einzuräumen, als ursprünglich gewünscht wurde. Die geographische Verteilung von Kompetenz und Wissen erfordert dabei ständige informelle Abstimmungsprozesse zwischen den Teams. Diese so genannte *Artikulationsarbeit* kann – als Pendant zu formellen Formen von Koordination – offenbar durch formale Koordinationswerkzeuge (etwa Fehlerdatenbanken) unterstützt, aber nicht ersetzt werden. Für KMU der Software-Industrie kann es daher beim Offshoring notwendig sein, auf ein synergetisches Verhältnis zwischen formeller Koordination und informeller Artikulationsarbeit zu achten, statt die Entwicklung einseitig zu formalisieren.

Inhaltsübersicht

1. Herausforderungen an KMU beim Offshoring
2. Bedeutung von Artikulationsarbeit
3. Methodik
4. Fallstudie eines kleineren Standardsoftware-Anbieters
 - 4.1 Unternehmenshintergrund
 - 4.2 Arbeitsteilung zwischen den Teams
 - 4.3 Artikulationspraktiken in geographisch verteilter Zusammenarbeit
 - 4.4 Diskussion
5. Notwendigkeit informeller Koordination
7. Literaturverzeichnis

1. Herausforderungen an KMU beim Offshoring

Offshore-Outsourcing – kurz Offshoring – ist in den letzten Jahren auch für kleine und mittlere Unternehmen der IT-Branche immer mehr zum Thema geworden. Unter Outsourcing versteht man in der Regel den Bezug von Vorprodukten oder Services von einem externen Anbieter, der im Fall von Offshoring meist in einem Niedriglohnland ansässig ist. Mittlerweile verfolgt eine wachsende Zahl deutscher KMU Offshoring-Strategien in Form geographisch verteilter Softwareentwicklung [Dibbern & Heinzl 2006]. Die bisherige Forschung zu Offshoring in der Softwarebranche beschäftigte sich dabei vor allem mit Großunternehmen und Fragen der Standardisierung und Formalisierung der gesamten Geschäfts- und Softwareentwicklungsprozesse [BITKOM 2005].

Aufgrund ihrer geringen Größe und der oft auf starke Kundenbindung ausgerichteten Geschäftsmodelle setzen KMU jedoch häufig auf sehr flexible und informelle Formen von Koordination und Kooperation. Gerade Soft-

warentwicklung benötigt ein hohes Maß an Reaktionsvermögen, um unerwarteten Entwicklungen dynamisch begegnen zu können. Flexibilität und Kundenbindung sollen daher als wichtige Wettbewerbsfaktoren von KMU gegenüber größeren Konkurrenten auch beim Offshoring gewährleistet bleiben. KMU verzichten dabei häufig bewusst auf die Etablierung formalisierter Prozessreifegrade, die als zu kostenintensiv und zu inflexibel empfunden werden [BMBF 2004]. Der alternative Einsatz agiler Entwicklungsmethoden ist jedoch erheblich von intensiver Kommunikation und Team-Interaktion abhängig und kann durch entsprechende Probleme im Rahmen von mit Offshoring besonders stark betroffen sein [Ågerfalk & Fitzgerald 2006]. Die Aufrechterhaltung von Flexibilität und Kundenorientierung beim Offshoring stellt KMU daher vor große Herausforderungen, da zu den üblichen Problemen verteilter Kooperation insbesondere solche hinsichtlich zeitlich-räumlicher sowie kulturell-sprachlicher Differenzen hinzukommen [Herbsleb et al. 2000].

Dennoch sind solche praktischen Problemstellungen beim Offshoring von Seiten kleiner und mittlerer Unternehmen bisher nur wenig von der Forschung adressiert worden [vgl. Dibbern et al. 2004]. Daher fokussieren wir in unserem Beitrag auf die Rolle von informellen Koordinationspraktiken für die Offshore-Softwareentwicklung eines KMU der Software-Industrie. Auf der Ebene der Theorie greifen wir dabei auf Anselm Strauss' Konzept der *Articulation Work* (Artikulationsarbeit) zurück, das unterschiedliche Formen der Ad-hoc-Koordination unter anderem auf der Mitarbeiterebene umfasst, die bisher in diesem Zusammenhang wenig Beachtung erfahren haben [Redmiles et al. 2007; Lee et al. 2006].

2. Bedeutung von Artikulationsarbeit

Artikulationsarbeit beschreibt interdependente Handlungen kooperierender Akteure, die hauptsächlich unterhalb der formal geregelten Ebene geleistet werden. Artikulationsarbeit muss unter anderem zur Verteilung und Zusammenführung von Teilaufgaben verrichtet werden: wer macht was, bis wann, in welcher Qualität, unter Einsatz welcher Ressourcen etc. [Strauss 1993]. Sie unterscheidet sich jedoch von formaler Koordination dadurch, dass sie wichtige Selbstorganisations- und -reflexionselemente umfasst (beispielsweise den Umgang mit Interdependenzen, individuellen Bedeutungszuschreibungen, Perspektiven und Interpretationen von Arbeit, vgl. [Schmidt & Simone 1996]). Artikulationsarbeit beruht dabei wesentlich auf diskursiven Praktiken kooperierender Akteure und kann nur aus einer Praxisperspektive heraus deutlich gemacht werden.

In Bezug auf Offshore-Softwareentwicklung ist hervorzuheben, dass Artikulationsarbeit in kleinen, lokalen Teams durch alltägliche soziale Interaktion geleistet werden kann und häufig gar nicht als Arbeit wahrgenommen wird [Star & Strauss 1999]. Wenn jedoch Arbeitsumgebungen (etwa durch Offshoring) komplexer werden, steigen auch die Anforderungen an die Artikulationsarbeit, so dass alltägliche Fähigkeiten zur impliziten Artikulation von Gruppenarbeit oft nicht mehr ausreichen. So muss Artikulationsarbeit beim Offshoring auf mehreren Ebenen geleistet werden: sie betrifft sowohl formalisierte Koordinationsmechanismen und informelle ad-hoc Koordination als auch lokale und organisationsübergreifende Arbeitspraktiken. Trotz kleiner, überschaubarer Teamgrößen geht die Entscheidung zum Offshoring daher gerade für KMU häufig mit einer enormen Steigerung der Komplexität (und oft auch einem entsprechenden Ressourcenbedarf) von Artikulationsarbeit einher, die für den erfolgreichen Abschluss von Softwareentwicklungsprojekten geleistet werden muss.

3. Methodik

Den Beginn unserer Untersuchung bildete eine umfassende Literaturanalyse der vorhandenen Literatur zum Offshoring von Softwareentwicklung; dabei wurden Diskurse unterschiedlicher wissenschaftlicher Communities sowie der Beraterliteratur berücksichtigt. Auf Grundlage der so gewonnenen Erkenntnisse wurden relevante Forschungsfragen identifiziert, die insbesondere auf Aspekte der Artikulationsarbeit in Form informeller Koordinations- und Kommunikationspraktiken in KMU zielten. Für die Datenerhebung wurde eine Teilnehmende Beobachtung in einem Offshoring betreibenden deutschen KMU durchgeführt. Dazu wurde das Unternehmen über einen Zeitraum von zwölf Arbeitstagen besucht, wobei lokale und verteilte Artikulationspraktiken im Rahmen von Meetings, individuellen Arbeitssituationen und kooperativen Aufgaben *in situ* beobachtet werden konnten. Ergänzend zur Beobachtung wurden ausführliche Interviews mit dem Inhaber des Unternehmens sowie mit Entwicklern und Projektleitern durchgeführt sowie Artefakte wie E-Mails, Chat-Protokolle, interne Arbeitspapiere und Tafelbilder ausgewertet.

Die Ergebnisse der Untersuchung wurden in Form von Feldnotizen und Fotos dokumentiert, die während der Teilnehmenden Beobachtung angefertigt wurden. Zur Auswertung des Datenmaterials orientierten wir uns am

Grounded Theory-Verfahren nach Glaser und Strauss [Strauss & Corbin 1996]. Dazu wurde das erhobene Material jeweils im Anschluss an die einzelnen Interviews und Teilnehmenden Beobachtungen ausgewertet und im Rahmen eines mehrstufigen Prozesses kodiert. Dabei wurden zunächst auf der Grundlage der Ergebnisse Kategorien gebildet. Diese wurden anschließend miteinander in Beziehung gesetzt und im Rahmen der Analyse weiterentwickelt.

Ziel der Studie ist in erster Linie ein detailliertes Verständnis von Artikulationspraktiken im Kontext von Offshoring, die derzeit noch wenig erforscht sind. Da das in unserer Studie gewählte explorative Vorgehen sowie die Beschränkung auf ein einzelnes Unternehmen gewisse Einschränkungen hinsichtlich der Generalisierbarkeit der Ergebnisse mit sich bringen, wurden sie mit der Literatur zu Artikulationsarbeit im Rahmen von Softwareentwicklung verglichen und durch eine parallel durchgeführte Interviewerhebung mit dreizehn Unternehmen weiterer deutscher KMU ergänzt, die Software-Offshoring betreiben.

4. Fallstudie eines kleineren Standardsoftware-Anbieters

4.1 Unternehmenshintergrund

Das untersuchte deutsche KMU bietet eine Standardsoftwareanwendung für Prozessmodellierung sowie Services im Bereich Prozessmanagement an. Der Jahresumsatz des Unternehmens betrug im Jahr 2007 etwa 1 Million Euro. Die Unternehmensleitung ist mit 10 Mitarbeitern in Bonn ansässig, während die Softwareentwicklung von einer Niederlassung in Berlin mit etwa sieben Mitarbeitern durchgeführt wird. Im Jahr 2002 gründete das Unternehmen eine Zweigniederlassung in St. Petersburg, um Entwicklungskosten einsparen zu können. Dazu wurde ein russischer Entwickler aus dem persönlichen Bekanntenkreis des Unternehmers eingestellt und zusammen mit drei weiteren Entwicklern für einige Monate nach Deutschland geholt, um vor Ort in das Programm eingearbeitet zu werden. Das Offshore-Unternehmen gehört mittlerweile einem der dort tätigen Entwickler und führt zusätzlich zwei Projekte mit anderen Kunden aus den USA durch; der überwiegende Teil des Umsatzes stammt aber nach wie vor aus der Kooperation mit dem untersuchten Unternehmen. Die Offshore-Niederlassung beschäftigte nach der Gründung bis zu fünfzehn Mitarbeiter, jüngst wurde das Team nicht zuletzt aufgrund von Kostensteigerungen auf acht Mitarbeiter reduziert, von denen vier ausschließlich für das untersuchte Unternehmen arbeiten. Diese Entscheidung hing auch mit der Übernahme des deutschen Unternehmens durch eine Holding im Jahr 2006 zusammen.

Obwohl die von dem untersuchten Unternehmen gewählte Form des (Captive-)Offshoring zunächst ein Sonderfall zu sein scheint, zeigt unsere Interviewerhebung, dass die Gründung einer Zweigniederlassung im Ausland von vielen der befragten KMU der Softwarebranche gegenüber dem Einkauf von professionellen Offshoringsservices bevorzugt wird. Dies hat vermutlich damit zu tun, dass die Softwareentwicklung das Kerngeschäft der untersuchten Unternehmen ausmacht und die Unternehmen daher einen möglichst großen Einfluss auf die im Ausland beschäftigten Mitarbeiter sowie die Durchführung der Entwicklung behalten möchten. Zudem wurden etablierte Offshoringanbieter häufig als vergleichsweise teuer und damit unwirtschaftlich für kleine Unternehmen bezeichnet.

4.2 Arbeitsteilung zwischen den Teams

Die Offshore-Entwickler agieren im untersuchten Fall als erweitertes Team und sollen von der deutschen Firma aus gesteuert und kontrolliert werden. Ein örtlicher Teamleiter in St. Petersburg ist dabei verantwortlich für die termingerechte Ablieferung der Ergebnisse sowie die disziplinarische Steuerung seines Teams. Diese Konstellation scheint unter deutschen KMU durchaus verbreitet zu sein, wie unsere Interviewerhebung zeigte. Im untersuchten Fall sind die „offshore“ angesiedelten Entwickler hauptsächlich mit der Entwicklung der beiden Varianten der Standardsoftwarelösung (eine für Einzelplatzrechner und eine netzwerkfähige Variante für Arbeitsgruppen) beschäftigt, während das Projekt- und Qualitätsmanagement (QA) vom deutschen Unternehmen aus durchgeführt wird. Kundenwünsche nach neuen Produktfunktionalitäten oder auftretende Fehler sollen unter Aufsicht des deutschen Projektleiters durch die deutschen Mitarbeiter aufgenommen und spezifiziert werden. Dazu dienen zwei Koordinationswerkzeuge, die Fehlerdatenbank (Bug-Tracking-System) SQA, in der aufgetretene Fehler (Bugs) und neu zu implementierende Funktionalitäten (Features) erfasst werden, sowie eine zentrale Produktdatenbank in Lotus-Notes, die u.a. Formatvorlagen und nach Versionen geordnete Spezifikationen neuer Funktionalitäten enthält. Die Produktdatenbank dient dabei auch als eine Art Logbuch, in das die Entwickler ihre Fortschritte dokumentieren sollen. Dazu können Kommentare direkt in das Spezifikationsdokument eingefügt wer-

den. Für die direkte Kommunikation stellt das Unternehmen seinen Mitarbeitern die Lotus-Notes Erweiterung *Sametime* zur Verfügung, die das Versenden von Instant Messages und das Herstellen einer Remote-Desktopverbindung ermöglicht.

Die Arbeitsteilung zwischen den Teams betrifft dabei auch gefundene Fehler, welche die russischen Entwickler, wenn sie sie bei der Entwicklung bemerken, wie ein Kunde an die deutsche QA melden sollen, statt selbst eine Beschreibung anzulegen oder den Fehler einfach zu beheben. Die QA versucht dann, auf der Grundlage der Beschreibung den Fehler zu reproduzieren, ihn hinsichtlich seiner Relevanz zu klassifizieren, eine standardisierte Fehlerbeschreibung in SQA anzulegen und die Beseitigung des Fehlers einem Entwickler zuzuweisen. Die Beschreibungen enthalten neben einer Kurzdarstellung eine detaillierte Erläuterung innerhalb eines Textfeldes, aus der hervorgeht, welche Version und welche Variante (*Build*) betroffen sind, wie der Fehler reproduziert werden kann, und welches Verhalten des Programms eigentlich gewünscht wäre. Zur Illustration werden häufig Bildschirmfotos angefertigt und in der Datenbank abgelegt. Wenn der Fehler eine bestimmte Datei betrifft (etwa eine abgespeicherte Datei, die beim Öffnen einen Fehler produziert), dann kann diese ebenfalls in die Datenbank gespeichert werden. Wenn der zuständige Entwickler den Fehler als bearbeitet gekennzeichnet hat, testet die QA in Deutschland das Resultat und markiert den Fehler in SQA je nach Ergebnis als „geschlossen“ oder erneut als „offen“.

4.3 Artikulationspraktiken in geographisch verteilter Zusammenarbeit

Im Idealfall soll sich der deutsche Projektleiter mittels Produktdatenbank und SQA jederzeit einen Überblick über den Stand der Arbeiten verschaffen können. Im Rahmen der Teilnehmenden Beobachtung zeigte sich jedoch, dass die formale Koordination in der Praxis häufig nicht ausreicht. So sind die Daten in den Systemen nicht immer aktuell, weil die russischen Entwickler im Tagesgeschäft die Angaben nicht jedes Mal wie geplant aktualisieren. Wenn dem deutschen Projektleiter auffällt, dass sich der Status wichtiger Fehler über einen längeren Zeitraum hinweg nicht verändert hat, fragt er deshalb informell per Instant Messenger den entsprechenden Entwickler nach dem Fortschritt der Arbeit.

Insgesamt spielen Instant Messages für die Artikulation der alltäglichen Zusammenarbeit eine wichtige Rolle, was sich im Rahmen der Interviewerhebung als typisch für Offshoring deutscher KMU erwies. So werden Instant Messenger im untersuchten Fall ständig parallel zur Produktdatenbank und SQA eingesetzt. Wenn das deutsche Team etwa Fehler in der Datenbank anlegt und diese zur Bearbeitung einem russischen Entwickler zuweist, wird dieser zusätzlich per Instant Messenger auf die Dringlichkeit der Aufgabe hingewiesen. Dabei werden häufig auch Rahmenbedingungen geklärt und Informationen vermittelt, die nicht immer aus den formalen Beschreibungen in der Fehlerdatenbank hervorgehen. Diese informelle Form der Besprechung dient dabei nicht nur zur Information des russischen Entwicklers, sondern auch dazu, gemeinsam abzugleichen, wie lange das Beheben voraussichtlich dauern wird bzw. festzulegen, wie lang es dauern darf. Das ist insbesondere dann wichtig, wenn die deutsche QA die Ursache des Fehlers nicht einschätzen kann. Daher wenden sich Tester, wenn sie einen Fehler finden, in vielen Fällen zuerst informell an den zuständigen Entwickler und machen ihn auf den Fehler aufmerksam. Dieser kann sich dann den Fehler anschauen und einschätzen, was die Ursache ist bzw. wie aufwändig das Beheben des Fehlers sein wird. Kleinere Fehler werden meist sofort behoben; wenn die Lösung zeitaufwändiger ist, entscheidet die QA in Absprache mit dem Projektleiter, wie dringend der Fehler ist und für welche Version er behoben werden soll.

In Rahmen der Feldforschung können auch emergente Verschiebungen der formalen Arbeitsteilung beobachtet werden. So gelingt es in der Praxis dem deutschen Team häufig nicht, Funktionen rechtzeitig genug zu spezifizieren, um das (vormals wesentlich größere) Offshore-Team beschäftigt zu halten. Aus praktischen Gründen wird daher die ursprünglich durch das Management gewünschte Praxis dahingehend abgewandelt, dass die russischen Entwickler ihre eigenen Spezifikationen schreiben, die anschließend vom deutschen Projektleiter kontrolliert werden. Diese Praxis erfordert umfangreiche Artikulationsarbeit, wird jedoch trotz der Verkleinerung des Offshore-Teams beibehalten. Nachträgliche Veränderungen der Arbeitsteilung wurde auch in den Interviews immer wieder beschrieben; dabei zeigte sich auch, dass es bei den gemeinsamen Lernprozessen im Rahmen des Offshoring wichtig ist, Entwicklerfeedback von der Offshore-Niederlassung ernst zu nehmen und entsprechend bei der Umsetzung von Veränderungen zu berücksichtigen.

Für den Wissensaustausch mit den russischen Entwicklern verlässt sich der deutsche Projektleiter auf regelmäßige persönliche Treffen, wozu er durchschnittlich zwei Mal im Jahr selbst nach St. Petersburg reist. Diese Praxis, persönliche Treffen für kommunikationsintensive Phasen der Entwicklung einzuplanen, wurde in den Interviews auch wegen ihrer positiven Auswirkungen auf den Aufbau von Vertrauen und gegenseitigem Verständnis immer

wieder als essentiell für den Erfolg des Offshoring beschrieben. Im untersuchten Fall finden die Besuche üblicherweise kurz vor der Veröffentlichung einer neuen Version statt, wobei gleichzeitig der *Kick-off* der nachfolgenden Version realisiert werden soll. Da ein in der Regel zweiwöchiger Aufenthalt dazu jedoch nicht immer ausreicht, nutzt der Projektleiter alternativ das Programm *Sametime* für die Herstellung einer Bildschirmverbindung. Dabei hat er in der Regel die Kontrolle über die Maus und bedient das Programm, das lokal oder auf dem Rechner eines Offshore-Mitarbeiters läuft. In diesem Rahmen probiert er dann die neu entwickelte Funktionalität aus oder beschreibt noch zu entwickelnde Funktionalitäten direkt im Programm, während der Entwickler die Prozedur verfolgt und bei Bedarf Fragen zu den Erläuterungen im Chat stellen kann. Für die Kommunikation wird der Instant Messenger von *Sametime* genutzt, als Basis für den Ablauf der Meetings nutzt der Projektleiter die Produktdatenbank, innerhalb derer er die Funktionalitäten der Reihe nach abarbeitet. Dabei kann der Offshore-Mitarbeiter am Bildschirm verfolgen, wie der Projektleiter den Status verschiedener Funktionalitäten in der Produktdatenbank ändert oder Kommentare zu deren Bearbeitung hinzufügt. Diese Online-Besprechungen finden in unregelmäßigen Abständen statt und können durchaus mehrere Stunden dauern.

Weitere Verschiebungen der formalen Vorgaben zeigen sich vor allem auch beim Einsatz von Softwarewerkzeugen. Das Offshore-Team setzt seit einiger Zeit intern statt auf SQA und *Sametime* auf eine eigene Fehlerdatenbank sowie *Google*-Webdienste (*Google Talk*, *Google Docs* etc.) für die Kommunikation. Auslöser für die Umstellung ist die Unzufriedenheit der russischen Kollegen mit der Unternehmenslösung SQA, die keine automatisierten Berichte über die geleistete Arbeit ermöglicht, die die russischen Entwickler aber einmal im Monat abliefern müssen. Daher stellten diese auf ein anderes *Open-Source* Produkt um und passten dieses an ihre Bedürfnisse an. Während die russischen Entwickler so im Bereich der Berichterstellung ihre Arbeit vereinfachen konnten, müssen als Folge davon Fehler aufwändig in zwei unterschiedlichen Systemen gepflegt werden. Die damit einhergehende Mehrarbeit muss jedoch von beiden Teams geleistet werden, wobei die russischen Kollegen dafür verantwortlich sind, die Informationen aus SQA in ihr eigenes System zu übertragen. Das deutsche Team unterstützt die russischen Kollegen dabei, indem in SQA eingetragene Fehler parallel per Email an die Bearbeiter gemeldet werden, damit diese die entsprechenden Fehler in ihre eigene Datenbank einpflegen können. Nach der Bearbeitung melden sich die russischen Kollegen wiederum per Email, und der deutsche Tester „schließt“ den Fehler in SQA. Diese Praxis wird vom deutschen Projektleiter toleriert, da die russischen Kollegen derzeit aktiv an einer Automatisierung des Datenabgleichs arbeiten. Die derzeitige Praxis stellt daher eine Art Übergangslösung dar, wenngleich das Arbeiten mit zwei parallelen Systemen vom deutschen Projektleiter grundsätzlich als problematisch angesehen wird.

4.4 Diskussion

Die Teilnehmende Beobachtung zeigte die große Bedeutung von informellen Abstimmungsprozessen für den Einsatz der Koordinationswerkzeuge in der Praxis. Diese Form der Artikulationsarbeit findet meist in Form kurzer Nachfragen statt und wird in der Regel über verschiedene Instant Messenger abgewickelt. Die gemeinsam ausgehandelte Einschätzung von Fehlern hinsichtlich Relevanz, technischer Ursache und Aufwand der Behebung dienen dann als Grundlage für das Anlegen standardisierter Fehlerbeschreibung in der Datenbank. Dabei lernt das deutsche Unternehmen erst von den russischen Kollegen, präzise Fehlerbeschreibungen zu schreiben. Einer der deutschen Entwickler begründete dies so: „Vor dem Offshoring lief alles lokal und war entsprechend knapp dokumentiert. Man konnte ja jederzeit einen Kollegen fragen. Jetzt muss alles genau schriftlich festgehalten werden, um möglichst alle Mehrdeutigkeiten auszuschließen, also alle Schritte zur Reproduktion dokumentiert und auch das erwartete Verhalten geschildert werden, da es sonst zu vielen Nachfragen kommt“ (Feldnotizen, 26. Januar 2007).

Die Notwendigkeiten des Offshoring zwangen das Unternehmen dazu, Artikulationspraktiken zu explizieren, die vorher im lokalen Team implizit und unsichtbar funktioniert hatten. Gleichzeitig führte die geographische Distanz zwischen Testern und Entwicklern jedoch auch zu einer geographischen Verteilung von Kompetenz, die für die Bewertung der in den Koordinationswerkzeugen abgelegten Informationen notwendig ist; daher ist der alltägliche Einsatz der Fehlerdatenbank SQA und der Produktdatenbank nach wie vor ständig mit Artikulationsarbeit verbunden [vgl. Gerson & Star 1986].

Die gestiegene Bedeutung informeller Artikulationspraktiken wird auch daran deutlich, dass die beiden kooperierenden Unternehmen mittlerweile unterschiedliche Systeme für die Katalogisierung ihrer Fehler und für die synchrone Kommunikation einsetzen. Aus dem Blickwinkel der formalen Arbeitsorganisation sollte der Einsatz unterschiedlicher Systeme zu erheblichen Schwierigkeiten führen. Der deutsche Projektleiter kann diese aufwändige Form der Kooperation nur deshalb tolerieren, weil die im Unternehmen etablierten informellen Artikulationspraktiken, die auch innerhalb des Systems formale Einträge mit informeller Kommunikation verbinden,

erheblich zur teamübergreifenden Transparenz des Arbeitsverlaufs beitragen. Die Produktdatenbank und SQA dienen in diesem Zusammenhang nicht in erster Linie als Mittel der Koordination, sondern tragen zur Sichtbarkeit von geleisteter und noch zu leistender Arbeit zwischen den Teams bei, deren Details und Aktualität aber häufig situativ via informeller Nachfragen eingefordert werden müssen.

Die aufwändige schriftliche Dokumentation von Arbeitsfortschritten mittels logbuchartiger Annotationen im Spezifikationsdokument wird dabei von den Entwicklern nicht regelmäßig gepflegt, weshalb der Projektleiter immer wieder nachhaken und Nachbesserungen verlangen muss. Der Versuch, durch den Einsatz von Koordinationswerkzeugen die Notwendigkeit von Artikulationsarbeit zu reduzieren, scheitert an einem grundsätzlichen Problem: das Pflegen ausführlicher und unmissverständlicher Informationen in den formalen Datenbanken ist aufwändig und beruht auf der aktiven Mitwirkung aller Mitarbeiter. Da diese aus Zeitgründen diesen Pflichten nicht immer nachkommen, verfolgt das Unternehmen die Strategie, den russischen Kollegen größere Autonomie einzuräumen, um den Aufwand der Datenhaltung zu reduzieren. Die damit steigende Bedeutung der Artikulationsarbeit wird in diesem Zusammenhang eher als Vorteil wahrgenommen, da sich das Einräumen von mehr Autonomie nach Ansicht des deutschen Projektleiters positiv auf Kreativität und Motivation der Offshore-Mitarbeiter auswirkt. Um die Sachhaltigkeit dieses Arguments zu prüfen, planen wir derzeit die Durchführung einer Teilnehmenden Beobachtung in dem russischen Unternehmen.

5. Notwendigkeit informeller Koordination

Das vorgestellte Beispiel zeigt, dass der Einsatz von Koordinationswerkzeugen eng mit Artikulationsarbeit verbunden sein kann. Die Notwendigkeit zur Artikulationsarbeit konnte im untersuchten Fall durch einen höheren Grad an Formalisierung von Fehlerbeschreibungen zwar verringert werden, die Kompetenzverteilung zwischen den Teams erfordert jedoch auch weiterhin ständige Artikulationsarbeit. Da es in der Praxis oft nicht möglich ist, alle Eventualitäten und Probleme verteilter Koordination *ex ante* formal zu regeln, können als Patentrezepte propagierte Strategien wie Prozessformalisierung und technische Koordinationsunterstützung (etwa Fehlerdatenbanken) Artikulationsarbeit demnach nicht obsolet machen, sondern nur erleichtern (oder in ungünstigen Fällen erschweren).

Dabei wurde auch deutlich, dass die Notwendigkeiten der Arbeitsteilung beim Offshoring unerwartete Eigenynamiken entwickeln können. Der hier beschriebene Fall zeigt dabei, dass fehlende Aufmerksamkeit gegenüber den Erfordernissen der Artikulationsarbeit mit Enttäuschungen und notwendigen Lernprozessen für die KMU verbunden sein kann, die durch das Offshoring gezwungen sein können, ihren Offshore-Entwicklern mehr Autonomie einzuräumen, als ursprünglich gewünscht wurde. Das geringere Maß an Kontrolle kann dabei jedoch aus Sicht der Unternehmer durchaus positive Auswirkungen auf Kreativität und Motivation der Offshore-Mitarbeiter haben.

Dieses Ergebnis deckt sich auch mit den Erkenntnissen aus unserer Interviewerhebung. Obwohl die meisten Interviewpartner auf klar definierte Rollen, Prozesse und Spezifikationen setzten, hoben sie durchweg „weiche“ Aspekte wie Vertrauen, Respekt, Sozialverhalten und besonders das persönliche Kennenlernen als zentrale Erfolgsfaktoren hervor. Der offensichtliche Konflikt zwischen dem Grad der Granularität von Daten und der Abhängigkeit von Artikulationsarbeit beim Einsatz von Koordinationssystemen deutet dabei darauf hin, dass es für KMU der Software-Industrie unter bestimmten Bedingungen sinnvoller sein kann, auf ein synergetisches Verhältnis von formeller Koordination und informeller Artikulationsarbeit zu setzen, als letztere einseitig ersetzen zu wollen. Die hier beschriebenen Aspekte verteilter Kooperation sollten daher von KMU der Softwarebranche berücksichtigt werden, die Teile ihrer Softwareentwicklung im Ausland abwickeln wollen, ohne ihre Effizienz und Flexibilität einzubüßen.

7. Literaturverzeichnis

[Ågerfalk & Fitzgerald 2006] Ågerfalk, P. J.; Fitzgerald, B.: Flexible and distributed software processes: old petunians in new bowls? In: Communications of the ACM, 49. Jg., 2006, S. 27-34.

[BITKOM 2005] BITKOM: Leitfaden Offshoring, 2005.
http://www.bitkom.org/files/documents/BITKOM_Leitfaden_Offshoring_31.01.2005.pdf, Zugriff am 7.3.2008.

in: HMD – Praxis der Wirtschaftsinformatik 265 (2009), S. 92-100.

[BMBF 2004] BMBF: Analyse und Evaluation der Softwareentwicklung in Deutschland, 2004. <http://www.isi.fhg.de/publ/downloads/isi00b69/software.pdf>, Zugriff am 7.03.2008.

[Dibbern & Heinzl 2006] Dibbern, J.; Heinzl, A.: Selective Outsourcing of Information Systems in Small and Medium Sized Enterprises. In: Hirschheim, R. A.; Heinzl, A.; Dibbern, J., (Hrsg.), Information Systems Outsourcing. Enduring Themes, New Perspectives and Global Challenges. 2. Aufl., Springer, Berlin, Heidelberg, 2006, S. 57-81.

[Dibbern et al. 2004] Dibbern, J.; Goles, T.; Hirschheim, R.; Jayatilaka, B.: Information systems outsourcing: a survey and analysis of the literature. In: ACM SIGMIS Database, 35. Jg., 2004, S. 6-102.

[Gerson & Star 1986] Gerson, E. M.; Star, S. L.: Analyzing due process in the workplace. In: ACM Transactions on Office Information Systems, 4. Jg., 1986, S. 257-270.

[Herbsleb et al. 2000] Herbsleb, J. D.; Mockus, A.; Finholt, T. A.; Grinter, R. E.: Distance, dependencies, and delay in a global collaboration. In: Proceedings of the ACM conference on computer supported cooperative work, 2000, S. 319-328.

[Lee et al. 2006] Lee, G.; Delone, W.; Espinosa, J. A.: Ambidextrous coping strategies in globally distributed software development projects. In: Communications of the ACM, 49. Jg., 2006, S. 35-40

[Redmiles et al. 2007] Redmiles, D.; van der Hoek, A.; Al-Ani, B.; Hildebrand, T.; Quirk, S.; Sarma, A.; Silva Filho, R. S.; de Souza, C.; Trainer, E.: Continuous Coordination. A New Paradigm to Support Globally Distributed Software Development Projects. In: Wirtschaftsinformatik, 49. Jg., 2007, S. 28-38.

[Schmidt & Simone 1996] Schmidt, K.; Simone, C.: Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. In: Computer Supported Cooperative Work, 5. Jg., 1996, S. 155-200.

[Star & Strauss 1999] Star, S. L.; Strauss, A. L.: Layers of Silence, Areas of Voice: The Ecology of Visible and Invisible Work. In: Computer Supported Cooperative Work, 8. Jg., 1999, S. 9-30.

[Strauss 1993] Strauss, A. L.: Continual permutations of action. De Gruyter, New York, 1993.

[Strauss & Corbin 1996] Strauss, A. L.; Corbin, J.: Grounded Theory: Grundlagen Qualitativer Sozialforschung, Beltz, Weinheim, 1996.

Stichwörter: Offshoring, KMU, Articulation Work, Ethnografische Studie, Koordination, Softwareentwicklung