

Kompetenzentwicklung in kleinen Unternehmen der Softwarebranche

Zur Praxisorientierung im Software-Engineering

Marco Durissini, Bernhard Nett und Volker Wulf

Fraunhofer Institut für Angewandte Informationstechnik, Sankt Augustin

Zusammenfassung

Der folgende Beitrag stellt eine empirische Studie zur Arbeitspraxis in der Software-Entwicklung vor, die auf der Basis von Experteninterviews mit Entwicklern aus kleinen Unternehmen der Softwarebranche im Rahmen des Forschungsprojekts ViSEK (Virtuelles Software Engineering Kompetenzzentrum) durchgeführt wurde. Im Mittelpunkt der Gespräche standen Fragen zu Praxisproblemen der Kompetenzentwicklung sowie zur Brauchbarkeit von Konzepten des Software Engineering in kleinen Unternehmen. Die Studie soll aufzeigen, wie in der Praxis kleiner Unternehmen der Softwarebranche Kompetenzen und Wissen gesichert werden und welche Rolle in diesem Zusammenhang Konzepte des Software Engineering haben.

1 Einleitung

Die neuste Ausgabe des Standish Chaos Report (2004) verdeutlicht die enormen Probleme, mit denen die Praxis der Software-Entwicklung noch immer behaftet ist. Im Vergleich zur letzten Ausgabe sind die dokumentierten Ergebnisse noch schlechter ausgefallen. So können nur noch 28% der Software-Entwicklungsprojekte als erfolgreich abgeschlossen gewertet werden. 18% der Projekte werden dagegen ohne Ergebnis abgebrochen während mehr als die Hälfte aller Softwareprojekte erheblich den Zeit- und Budgetrahmen übersteigen. Mangelnde Einbeziehung von Nutzern wird als wichtigster Grund für das Scheitern von Software-Entwicklungsprojekten genannt.

Vor diesem Hintergrund erscheint mehr als 35 Jahre nach Begründung des Forschungsfeldes „Software Engineering“ eine Reorientierung des Forschungsfokusses bedenkenswert. Bisher war das Forschungsparadigma im Software-Engineering stark als theoretisch motivierte Setzungen, als Formalisierung von Prozessen, bestimmt. Dabei kam u.E. der empirischen Erforschung tatsächlicher Arbeitspraktiken menschlicher Akteure in Software-

Entwicklungsprozessen eine zu geringe Bedeutung zu. Auch die Relevanz der Praktiken der Nutzer von Softwareartefakten wurde unterschätzt.

Darüber hinaus hat das Software-Engineering zu lange auf die Software-Entwicklung in Großunternehmen und Großprojekten fokussiert. Die dort entwickelten Prozesse, Methoden und Tools scheinen aber nicht ohne weiteres auf die Arbeit in kleinen und mittelständischen Unternehmen übertragbar zu sein. Letzterer Typus von Unternehmen stellen aber die bei weitem größte Zahl der in Deutschland tätigen Unternehmen der Softwarebranche dar (vgl. BMBF 2000).

Unter dem Terminus „Benutzer-orientiertes Software-Engineering“ verstehen wir ein an der Arbeitspraxis orientiertes Vorgehen in der Software-Technik. Die Praxisorientierung bezieht sich auf Arbeitsprozesse der Software-Entwicklung, -Aneignung und -Nutzung. Durch den Bezug auf empirisch erforschbare Praxis kann im Verhältnis zwischen Software-Engineering und Software-Ergonomie die Dichotomie zwischen Entwicklung und Nutzung überwunden werden. Im Gegensatz zur bisher dominierenden Orientierung der Software-Technik impliziert die hier vorgeschlagene Praxisorientierung eine enge Verbindung von „Verstehen und Gestalten“ (vgl. Winograd & Flores 1988).

Im Folgenden soll eine Fallstudie zur Arbeitspraxis in der Software-Entwicklung präsentiert werden. Diese Studie wurde im Rahmen des Forschungsvorhabens VSEK (Virtuelles Software Engineering Kompetenzzentrum) durchgeführt. VSEK zielt auf eine Stärkung des Software-Engineering in Deutschland ab mit einem speziellen Fokus auf kleine und mittelständische Unternehmen (vgl. Nett & Wulf 2005). Das zunächst auf die Bereitstellung einer Software-Engineering Wissensdatenbank abzielende Projektangebot wurde in der Folge durch begleitende Transfer- und Vernetzungsaktivitäten ergänzt (vgl. Kalmar & Wulf 2003). Die hier vorliegende Studie wurde zur Ermöglichung einer empirisch basierten Abstimmung der Transfer- und Vernetzungsangebote des VSEK-Projekts auf industrieübliche Wissensprozesse unternommen.

2 Vorgehen

In der hier darzustellenden Studie wurden Wissensprozesse in kleinen Unternehmen der Softwarebranche untersucht. Unter Wissensprozessen werden dabei alle Arten von wissensbezogenen Aktivitäten, z.B. die Aneignung, Nutzung und Sicherung von Wissen, verstanden (vgl. Hasenkamp & Rossbach 1998). Es wurde geprüft, welche formellen und informellen Lernpraktiken in kleinen Unternehmen der Softwarebranche vorfindbar sind, auf welche Wissensressourcen sich diese stützen und wie Wissen vor dem Hintergrund der betrieblichen Praxis bewertet, gesichert und vermittelt wird. Zudem galt es zu untersuchen, welche Rolle Software Engineering im Rahmen der identifizierten Wissensprozesse spielte. Dabei waren Softwareentwickler von besonderem Interesse, weil sie in den Unternehmen die Akteure sind, welche die Praxis der Software-Entwicklung am unmittelbarsten prägen.

Die Studie basiert auf teilstrukturierten Experteninterviews, weil diese gute Einblicke in Strukturzusammenhänge und Wandlungsprozesse von Handlungssystemen bieten und über

die Insider-Erfahrungen spezifischer Status- und Interessengruppen informieren. Des Weiteren können sie als Sondierung für weitere Forschungen für die Orientierung und Strukturierung dienen (vgl. Meuser & Nagel 1991).

In der Zeit von April bis August 2003 wurden insgesamt 17 Interviews in kleinen Unternehmen der Softwarebranche durchgeführt. Zwei Unternehmen waren im weiteren Berliner Umkreis angesiedelt, die übrigen im weiteren Kölner Raum. Die Interviews fanden vor Ort bei den Unternehmen statt. Hinsichtlich der Mitarbeiterzahl waren die Unternehmen wie folgt verteilt: 10 Unternehmen hatten weniger als 10 Mitarbeiter, 5 Unternehmen zwischen 10 und 100 Mitarbeitern und 2 Unternehmen zwischen 100 und 200 Mitarbeiter.

Alle Interviews wurden aufgenommen und anschließend transkribiert. Die Auswertung der Transkriptionen erfolgte durch Paraphrasierung der forschungsrelevanten Aussagen. Anschließend wurden die Paraphrasen zu thematischen Clustern zusammengefasst und mit den leitenden Forschungsfragen vergleichend analysiert (vgl. ebd.). Im Folgenden sollen ausgewählte Ergebnisse zur Praxis wissensorientierter Arbeit in der Software-Entwicklung hinsichtlich ihrer Bedeutung für das Software-Engineering und die Kompetenzentwicklung kleinen Unternehmen der Softwarebranche dargestellt werden.

3 Ergebnisse der Studie

3.1 Software-Engineering in kleinen Unternehmen

Nach Meinung der meisten Interviewten wird der Einsatz neuer Methoden des Software-Engineering in kleinen Unternehmen dadurch erschwert, dass sie für die Realisierung großer Entwicklungsprojekte konzipiert wurden und nicht auf die Bedürfnisse kleiner Projekte zugeschnitten sind. Zudem wurde Software-Engineering als Streiten um konfligierende Konzepte wahrgenommen: „Man ist noch weit von einem grundlegenden Formalismus des Software Engineering entfernt, obwohl man bereits seit 30 Jahren daran arbeitet.“

Die Einführung neuer Methoden setzt Veränderungen der eigenen Arbeitsabläufe voraus. Z.T. erfordern sie auch neue kostspielige Entwicklungstools. Die Kunden aber kennen sich selten mit neuen Methoden aus. Von daher können sie auch nicht den Entwicklungsaufwand nachvollziehen, der damit verbunden ist, und den sie bezahlen müssen.

Für die Interviewten ergab sich damit aus ihrer Kundennähe, insbesondere aus dem direkten Einwirken der Kunden auf die Softwareproduktion, ein starker Anpassungsdruck, der jedoch umso heterogener ist, je mehr Kunden ein Unternehmen hat. Unter diesen Umständen können sich Regeln, auch wenn sie nützlich wären, für die weitere Unternehmensentwicklung, beim nächsten Kunden als Risiken entpuppen, als starres Konzept, das keine Lösung, sondern Probleme bringt.

Keiner der Interviewten berichtete von einem formalisierten Softwareprozess in seinem Unternehmen. Es wurde z.T. sogar argumentiert, dass eine zu dezidierte Orientierung des Softwareentwicklungsprozesses an einem Software-Engineering-Konzept zu einem Ausschluss von Kunden mit davon abweichenden Vorstellungen führe und Software-Engineering mithin die Marktchancen eines Unternehmens beeinträchtige.

Die angewandten Methoden und Vorgehensmodelle konnten nicht eindeutig im Sinne bekannter Modelle identifizieren werden. Es wurde zumeist, je nach Projektkontext, Kundenerwartungen und Entwicklerpräferenzen ein Mix angewendet, bestehend aus verschiedenen Elementen aktueller Vorgehensmodelle. In der Regel wurden Entwicklungsprozesse iterativ angegangen. Dabei wurden auch Impulse aus neuen Methoden aufgegriffen, wie beispielsweise Rapid Prototyping oder UML-Diagramme zum Entwerfen von Systemarchitekturen.

Wenig überraschend war ein allgemeines Interesse an neuen Methoden von Seiten der Interviewten. Doch dieses theoretische Interesse wurde der Praxis gegenübergestellt, in der es wenige Umsetzungsmöglichkeiten gebe. Software-Engineering liefere „keine Patentrezepte“. Der positive Aspekt, der sich aus dieser Negation ergab, wurde deutlich in der Aussage, dass sich in der betrieblichen Praxis kaum Regeln etablieren ließen, sondern dass sie geprägt sei von einem „Mischmasch“ „von Trends, die kommen und gehen“.

Das Fehlen allgemeiner Regeln wurde nicht als Freiraum für Kreativität empfunden, sondern als Parcours, auf dem „jede Menge Detail-Regelungen“ z.T. noch unterhalb der Unternehmensebene (Abteilungen und Projektteams) beachtet werden müssen. Enge Kundenvorgaben wurden als Behinderung betrachtet, die den Gestaltungsspielraum der Entwickler einengen; die Softwareproduktion wurde als ein Feld beschrieben, in dem statt einer allgemeinen Struktur eine Unüberschaubarkeit besonderer Strukturen vorherrsche. Diese Unübersichtlichkeit, so der Tenor, könne nur durch Kommunikation bewältigt werden

3.2 Kompetenzanforderungen in kleinen Unternehmen

Die Interviewten stellten durchweg die Bedeutung intensiver Kontakte zu den Kunden heraus. Diese wurden als umso wichtiger gesehen, als die Produkte meist keine Standardsoftware darstellten, sondern Anpassungen oder Erweiterungen eines Softwareprodukts für die besonderen Gegebenheiten ihrer Kunden. Für Entwickler bedeutete dies zunächst, dass Kommunikationsfähigkeiten wichtiger wurden, mit deren Hilfe sie Vagheiten produktiv wenden und Arbeit iterativ gestalten lernen mussten. Entsprechend bedeutete dies für die Beteiligten, dass sie lernen mussten, im Verlauf von Prozessen deren Erfordernisse wahrzunehmen, um sich darauf einstellen zu können. Fortwährend gelte es zu verstehen und zu „sehen, was der Kollege tut“, wurde formuliert. In der Alltagspraxis wurde weitgehend selbst organisiertes Arbeiten mit unvollständiger Dokumentation und Information als Normalsituation wahrgenommen.

Aus Entwicklersicht waren insbesondere die Kundenerwartungen massiv gestiegen. So verlangten die Kunden zunehmend Implementierung, Wartung und Erweiterung von Softwareprodukten selbst umsetzen zu können. Gleichzeitig nahmen die

Qualitätserwartungen und der Kostendruck zu, was hohe Anforderungen an die Beschäftigten mit sich brachte. Zum einen galten diese der Funktionalität und Nutzbarkeit der Software. Zum anderen wurde das Verständnis für die Kundenwünsche zum immer zentraleren Element der Qualitätsorientierung.

Da die Implementierung neuer Produkte nicht selten mit einer zumindest partiellen Restrukturierung von Arbeitsabläufen beim Kunden verbunden ist und technische Gestaltungsentscheidungen oft weit reichende, mitunter ungewollte Implikationen haben, lassen sich Produktplanung beim Entwickler und Nutzungsplanung beim Kunden nicht völlig trennen. Dies verstärkt entsprechende Abstimmungserfordernisse und die Notwendigkeit von Kundenkontakten. „Der Kunde weiß nicht, was er will“, kennzeichnet eine Seite der Wahrnehmung dieser „symmetry of ignorance“ in der Kunden-Produzenten-Beziehung, die mangelnde Kenntnis des Produzenten über die Verhältnisse beim Kunden die andere (Fischer 1999). Es handelt sich dabei aber nicht allein um eine kognitive Differenz, sondern auch um ein Feld von Interessenkonflikten.

Wenn aber Produktentwicklung und Unternehmensentwicklung so miteinander verbunden sind, bedeutet dies für Mitarbeiter in kleinen Unternehmen der Softwarebranche, dass sie im Projektgeschäft in der Lage sein müssen, ein hohes Maß an Selbständigkeit an den Tag zu legen, komplexe trans-disziplinäre Zusammenhänge schnell verstehen und sinnvoll berücksichtigen zu können. Darin kommt eine partielle Auflösung zwischen Management- und Ausführungsfunktionen zum Tragen, die durch den hohen Kommunikationsbedarf der Arbeiten gefördert wird.

Damit wird zum einen die Funktion eines reinen ‚Cod(ier)ers‘ durch den iterativen Charakter der Anforderungserhebung und die technischen Substitutionsmöglichkeiten sowie durch mögliche Auslagerungen in Billiglohnländer¹ in Frage gestellt. Zum anderen ergibt sich damit der Bedarf an immer selbständiger arbeitenden Projektmitarbeitern, sei es als Projektleiter für zuarbeitende ‚Cod(ier)er‘, sei es als die verschiedene Rollen integrierenden Allrounder. Letztere waren vor allem in sehr kleinen Firmen oft gefragt.

Im Hinblick auf die geforderten Kompetenzen bedeutet dies, dass Fähigkeiten zum selbständigen Arbeiten und Entscheiden von großer Bedeutung sind. Weiterhin ergibt sich jedoch gerade aus der geforderten Selbständigkeit die Erfordernis disziplinierter und loyaler Kooperation und von Teamfähigkeit. Die Aushandlungsabhängigkeit vieler Arbeitsumstände gerade in kleinen Unternehmen der Softwarebranche setzt entsprechende Fähigkeiten „sich verkaufen zu können“ voraus, scheint aber von den Beschäftigten durchaus im Sinne eines Arbeitsstolzes als Gestaltungsfreiraum geschätzt zu werden (Heidenreich & Töpsch 1998): Viele Interviewte stellten ihre berufliche Existenz als Professionalisierung ihres Hobbys dar und grenzten sich unter Verweis auf ihren Spaß an der Arbeit explizit von traditionellen Karrieremustern ab.

Zum Verständnis der Kundenwünsche wird nach Angabe der Interviewten zunehmend erforderlich, deren Implikationen verstehen zu können, so dass Domänenkenntnisse

¹ Drei der 60 Unternehmen, die wir um ein Interview baten, hatten ihren Entwicklerstamm ins Ausland verlegt (Tschechien, Russland, Vietnam).

zunehmend von Bedeutung für Entwickler werden. Neben diesen allgemeinen Kenntnissen über das Feld treten schließlich die Fähigkeiten, die speziellen Kontexte beim jeweiligen Kunden zu erschließen. Die im Kontext soziotechnischer Technikgestaltungsansätze entwickelten, oft beteiligungs-basierten Methoden empirischer Forschung halten deshalb – wenn auch oft in reduzierter, mitunter gar unbewusster Form – Einzug in die Praxis von Entwicklern in der Softwarebranche und stellen entsprechend zunehmend wichtige Kompetenzen dar.

3.3 Lernen in kleine Unternehmen

Bei der Informationssuche wurden Bücher meist nur als Referenz- und Nachschlagewerke genutzt; Fachzeitschriften aus Zeitgründen nur selten gelesen. Nur wenige Entwickler zogen sich zeitweilig mit Büchern zurück, um sich systematisch in neue Themen einzuarbeiten (die Möglichkeit dazu zu haben kann unter dem Zeitdruck, unter dem Entwickler stehen, bereits ein Privileg darstellen).

Hingegen wurde das Gespräch mit den Kollegen am Arbeitsplatz von den meisten Interviewten als fruchtbarste Informationsquelle beschrieben („Multiplizierungseffekt durch die Kollegen“). Auch das Internet wurde als Informationsquelle intensiv genutzt und war aufgrund seiner Vielfältigkeit und Schnelligkeit erstes Referenzmedium. Unübersichtlichkeit und Unzuverlässigkeit schränkten jedoch den Informationswert des Internets jedoch stark ein.

Interessanterweise konnten Foren und Newsgroups diese Nachteile in der Regel nicht überwinden: Zwar galten sie als hilfreich zum schnellen Lösen von konkreten Entwicklungsproblemen, doch auch als unübersichtlich und unstrukturiert. Zudem setze ihre Nutzung eine kontinuierliche und z.T. aktive Teilnahme voraus, die für die meisten Interviewten aus Zeitgründen nicht möglich war. Foren und Newsgroups wurden deshalb oft während Ausbildung oder Studium stark genutzt, gerieten in der Berufspraxis jedoch schnell in den Hintergrund, insbesondere im Hinblick auf die aktive Beteiligung.

Herstellerforen (z.B. die Seiten von SUN zu Java) waren eine Ausnahme, da sie auch nach Abschluss von Studium oder Ausbildung stark frequentiert werden. Eine Erklärung dafür könnte sein, dass es nach Angaben der Interviewten oft nicht bloß um Informationen, sondern um belegbare Entscheidungsgrundlagen geht. Diese Gewährleistungsfunktion kann bei der Informationssuche im Hinblick auf Fragen des persönlichen Expertenrenommées bis hin zu finanziellen Problemen im Rahmen zwischenbetrieblicher Gewährleistung von Belang sein.

In Einzelfällen wurden nicht-öffentliche Infotheken (Bulletin-Boards) zwischen Partnerunternehmen zum Erfahrungsaustausch von den Unternehmen selbst aufgebaut. Andere Unternehmen boten ihren Kunden Supportforen nicht allein zur Entlastung der Support-Abteilung an, sondern auch zur Sammlung von relevanten Informationen für die Wartung und Weiterentwicklung der Software. Eine einfache, aber für Marketingverantwortliche sicherlich hochwertige Quelle war die ‚Feature-Request‘-Datenbank.

Solche Einrichtungen allein garantieren noch keine Nutzung, erhöhen jedoch die Wahrscheinlichkeit, dass bereits gemachte Erfahrungen nicht verloren gehen. Die Interviewten fanden solche soziotechnisch konzipierten Einrichtungen attraktiv und sehr sinnvoll, weil sie kaum Anschaffungskosten verursachen, leicht zu handhaben und vor allem nicht an formelle organisatorische Abläufe gebunden sind. Interessanterweise betrachteten sie diese soziotechnischen Lösungen jedoch nicht als Wissensmanagement.

Innerbetrieblich geschah das Einbringen von „Methoden über Köpfe“: das bedeutet, dass es innerhalb der kleinen Unternehmen zwar wenig formale Fixierungen von Rollen und Verantwortlichkeiten gab, aber durchaus gewisse Spezialisierungen. Mitarbeiter stellen aus dieser Sicht „Portfolios an Kompetenzen“ dar. Oft wird dann erst auf den Projekttreffen deutlich, welche Kompetenzen für einen gegebenen Kontext erforderlich sind: entsprechend wird dann das Projektteam (re-)organisiert. Aus den Angaben der Interviewten wird nicht klar, ob die Ursache für die Unabschätzbarkeit von Kompetenzen vor Projektbeginn auf die völlige Unterschiedlichkeit der Entwicklungsprozesse oder das Unvermögen der Unternehmen zurückzuführen ist, Ähnlichkeiten im Prozess zu erkennen und ggf. zu nutzen. Kompetenzmanagement wird von daher vor allem relevant bei der Aufstellung geeigneter Projektteams.

3.4 Problemfelder der Kompetenzentwicklung

Von den Befragten wurden im Zweifelsfall etablierte Communities gegenüber anderen Informationsquellen präferiert, wenn es um gesicherte Informationen über Personal, Märkte, Trends oder Akquisitionsmöglichkeiten ging. Informelle Treffen und Kontakte wurden deshalb umso wichtiger eingeschätzt – auch im Kundenkontakt: Durchgängig wurde darauf hingewiesen, wie wichtig ein belastbares Vertrauensverhältnis zum Kunden einzuschätzen sei. Verstärkte Kommunikation wurde deshalb vom Gros der Entwickler als einzig langfristig Erfolg versprechende Strategie aufgefasst.

Doch obwohl „firmeninterne Kommunikation immer wichtiger“ wurde, kamen die Interviewten zu dem Schluss: „Mehr Treffen wären gut“. Der wachsende Bedarf an Kommunikation führte so zu zunehmender Zeitknappheit: wo es vormals noch interne Entwicklertreffs zum allgemeinen Erfahrungsaustausch jenseits des Projektgeschäfts gab, da waren aus Zeitmangel nur noch die aufgabenbezogenen Projektmeetings und die ‚Coffee Corners‘ als informelle Treffpunkte geblieben. Und während persönliche, vertrauensvolle Beziehungen als immer wichtigerer Wettbewerbsvorteil gesehen wurden, wurde nach Meinung der Interviewten der Aufbau solcher Beziehungen und Communities immer schwieriger. Kommunikation als ‚Lösung‘ konnte mithin selbst zum ‚Problem‘ werden und die Nachhaltigkeit des weitgehend individuell organisierten Kompetenzmanagements in den kleinen Firmen der Softwarebranche gefährden.

Staudt und Kley (2001) betonten die zentrale Bedeutung von informellem und selbst gesteuertem Lernen im Arbeitsprozess für die betriebliche Weiterbildung. Weiß (2000) gab in diesem Kontext den Zeitanteil informeller Weiterbildungsaktivitäten, den die Befragten als weiter steigend einschätzten, bereits 2000 mit 46 % an. Unsere Studie lässt heute gerade in den kleinen Unternehmen der Softwarebranche eine noch höhere Prozentzahl erwarten.

Damit drängt sich der Verdacht auf, dass dies weniger einer Zunahme entsprechender Aktivitäten als einer Abnahme alternativer Qualifizierungsbemühungen geschuldet ist.

Nach Meinung der Interviewten ist der domänenspezifische Kompetenzerwerb von stark zunehmender Bedeutung; er könne aber nur im Geschäftsbetrieb selbst realisiert werden – in dem bei zunehmender Konkurrenz die Möglichkeit zur Qualifizierung jedoch prekär werden kann. Während die bereits skizzierte Studie des BMBF (2000) durchschnittliche Zeitaufwände von 3-10 Tagen je Beschäftigtem und Jahr für die institutionalisierte berufliche Weiterbildung benannte, dürften solche Werte in den kleinen Unternehmen, aus denen die Interviewten dieser Studie stammen, kaum erreicht werden, denn entsprechende Nachfragen nach betrieblichen Qualifizierungsmaßnahmen wurden durchweg ausweichend beantwortet. Stattdessen wurde der Nutzen von Qualifizierung gerne pauschal in Frage gestellt.

So wurde etwa verlangt, dass Qualifizierungsmaßnahmen als ‚Enabler‘ Zugang zu neuen Kunden oder Geschäftsfeldern eröffnen müssten: in dieser nur scheinbar dynamischen Perspektive wird jedoch nicht berücksichtigt, dass innovatives Qualitätsmanagement auch schon für die Sicherung des Kundenstammes von Bedeutung ist: den Interviewten war klar, es „schläft die Konkurrenz nicht“. Als weitere pauschale Entschuldigung diente oft der Verweis auf den Ressourcenmangel in kleinen Unternehmen, welche der Nachfrage- und Markttransparenz auf der Anbieterseite gegenübergestellt wurde („zu viele zu teure Angebote mit unklarem Nutzen“).

Ein anderer Teil der Aussagen zur Mitarbeiterqualifikation adressierte das Qualifizierungsproblem von einem aktuellen Struktur-Entwicklungstrend her, den man als zunehmende Domänendifferenzierung der kleinen Unternehmen der Softwarebranche interpretieren kann: „Die Spezialisierung nimmt zu“. Diese zunehmende Spezialisierung wird neben der Zunahme der Bedeutung der ‚soft skills‘ noch für einen anderen Effekt im Hinblick auf Mitarbeiterqualifikation verantwortlich gemacht: „Domänenspezifisches Wissen wird immer wichtiger“. Deshalb erfolgt die Einarbeitung neuer Mitarbeiter domänenspezifisch und situativ. Dabei gibt es oft keine andere Möglichkeit, als „von erfahrenen Kollegen zu lernen“.

Was aber, wenn kein einschlägig erfahrener Kollege dafür zur Verfügung steht? Und wie sollen arbeitslos gewordene Entwickler sich qualifizieren, wenn Lernen nur im Kontext der Arbeit erlernbar ist? Obwohl ein Großteil der Interviewten von ähnlichen Problemen selbst betroffen sein dürfte, werden solch allgemeine Fragen (wenn überhaupt) nur implizit angesprochen.

4 Schlussfolgerungen

Wir haben Ergebnis einer explorativen empirischen Untersuchung der Lern- und Arbeitspraxis in kleinen Unternehmen der Softwarebranche dargestellt. Diese Ergebnisse bedürfen einer weiteren Absicherung durch die Einbeziehung weiterer Unternehmen und die Heranziehung zusätzlicher Untersuchungsmethoden. Nichtsdestotrotz weisen bereits die vorliegenden Ergebnisse darauf hin, dass sich Lern- und Arbeitspraxis in kleinen

Unternehmen erheblich von der in Grossunternehmen der Softwarebranche unterscheidet und vom Software-Engineering bisher noch kaum erreicht wurden. Daraus lässt sich schließen, dass für kleine Unternehmen spezielle soziotechnische Maßnahmenbündel zur Förderung ihrer Wissensprozesse entwickelt werden müssen. Insofern stellen diese Ergebnisse eine wichtige Grundlage zur Weiterentwicklung der Wissensdatenbank und der Vernetzungs- und Transfermaßnahmen im Vorhaben „Virtuelles Software-Engineering Kompetenzzentrum (VSEK)“ dar (Kalmar & Wulf 2003).

Die Veränderungen der Wissensprozesse in den kleinen Firmen der Softwarebranche machen zum einen Probleme etablierter Lerninstitutionen deutlich, zum andern zeigen sie die Problematik der Gestaltung und Förderung neuer institutioneller Arrangements, die den emergenten Formen des Lernens im Feld besser gerecht werden. So reicht Vernetzung alleine nicht aus, um neue Möglichkeiten inter-organisationalen Lernens wie z.B. *legitimate periphere Partizipation in Communities of Practice* (Lave & Wenger 1991; Wenger 1998) zu schaffen. Von daher bedarf es von der hier vertretenen soziotechnischen Perspektive her Forschung zu diesbezüglichen Bedingungen in der Softwarebranche.

In der sich dynamisch entwickelnden Softwarebranche wird Wissen nicht immer *erst* theoretisch erzeugt, und *dann* umgesetzt, sondern beide Wissensprozesse können in der betrieblichen Praxis verbunden sein, mitunter sogar invers ablaufen. Die Wissenschaft muss deshalb in zunehmendem Maße nicht nur die Aufgabe des Lehrens, sondern auch die des Lernens bewältigen. Zugleich wird deutlich, dass insbesondere der Bereich der berufsbegleitenden Weiterbildung ein ebenso problematischer wie wichtiger Bildungsbereich für die Softwarebranche ist.

Jenseits der hier untersuchten Wissensprozesse zeigt diese Untersuchung das Potential eines an der Arbeitspraxis orientierten Forschungsparadigmas im Software-Engineering. Nur wenn es gelingt, ein hinreichendes Verständnis für die Komplexität und Situiertheit menschlicher Arbeit in den verschiedensten Teilbereichen der Softwarebranche zu entwickeln, können geeignete Prozessmodelle, Methoden und Werkzeuge zur Unterstützung menschlicher Arbeit entwickelt werden. Das Software-Engineering könnte an dieser Stelle in methodologischer Sicht auf langjährige Erfahrungen im Bereich Computer-unterstützter Gruppenarbeit zurückgreifen. Dort hat sich die Verknüpfung von ethnographischen Methoden mit Aktionsforschungsansätzen als angemessenes Forschungsparadigma erwiesen (Suchman 1987; Hughes et al. 1994; Wulf & Rohde 1996).

5 Literatur

- BMBF (Hrsg.) (2000): Analyse und Evaluation der Softwareentwicklung in Deutschland. Endbericht an das Bundesministerium für Bildung und Forschung, GfK Marktforschung, URL: http://www.iese.fhg.de/pdf_files/evasoft_abschlussbericht.pdf, 12.08.2004.
- Fischer, Gerhard (1999): Social Creativity, Symmetry of Ignorance and Meta-Design. In: Proceedings of the third Creativity & Cognition Conference, New York 1999, S. 116-123.
- Hasenkamp, Ulrich; Roßbach, Peter (1998): Wissensmanagement. In: WISU, Nr. 8-9, S. 956-964.

- Hughes, John; King, Val; Rodden, Tom; Andersen, Hans (1994): Moving out from the Control Room: Ethnography in System Design. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 1994), S. 429 - 439.
- Huysman, Marleen; Wulf, Volker (Hrsg.) (2004): Social Capital and Information Technology. Cambridge, MA: MIT-Press.
- Lave, Jean; Wenger, Etienne (1991): Situated learning: legitimate peripheral participation. Cambridge: Cambridge University Press.
- Ralf Kalmar; Volker Wulf (2003): Das virtuelle Software-Engineering Kompetenzzentrum (ViSEK). In: Proceedings der sechsten internationalen Tagung Wirtschaftsinformatik (WI 2003) am 17. - 19.9. 2003 in Dresden, Bd. II, Heidelberg: Physica-Verlag, S. 987-1006.
- Meuser, Michael; Nagel, Ulrike (1991): Experteninterviews – vielfach erprobt, wenig bedacht. In: Garz, D. Kraimer, K. (Hrsg.): Qualitativ-empirische Sozialforschung. Opladen: Westdeutscher Verlag, S. 441–471.
- Nett, Bernhard; Wulf, Volker (2005): Wissensprozesse in der Softwarebranche: Eine empirische Perspektive auf klein und mittelständische Unternehmen. In: Gendolla, P.; Schäfer, J. (Hrsg.): Wissensprozesse in der Netzwerkgesellschaft, Bielefeld: transcript Verlag, S. 147 – 168.
- Probst, Gilbert J.A.; Raub, Steffen; Romhardt, Kai (1998): Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Frankfurt am Main: Dr. Th. Gabler Verlag.
- Standish Chaos Report (2004), Ergebnisse zitiert nach:
www.computerworld.com/managementtopics/management/project/story/0,10801,97283,00.html
- Staudt, Erich; Kley, Thomas (2001): Formelles Lernen – informelles Lernen. Erfahrungslernen: wo liegt der Schlüssel zur Kompetenzentwicklung von Fach- und Führungskräften? In: Berufliche Kompetenzentwicklung in formellen und informellen Strukturen. QUEM-Report, Heft 69, Berlin, S.227-275.
- Suchman, Lucy A. (1987): Plans and situated action. The Problem of Human-Maschine Communication. Cambridge: Cambridge University Press.
- Wenger, Etienne (1998): Communities of Practice: Learning, Meaning, and Identity. Cambridge: Cambridge University Press.
- Winograd, T.; Flores, F. (1988): Understanding Computers and Cognition. 3. Aufl., Norwood, NJ: Addison-Wesley.
- Wulf, Volker; Rohde, Markus (1995): Towards an Integrated Organization and Technology Development. In: Proceedings of the Symposium on Designing Interactive Systems, 23. - 25.8.1995. New York: ACM-Press, S. 55 – 64.