

Inhaltsverzeichnis

Vorwort

W. Dilger, E. Keitel 3

Von Technik zu Kultur und zurück

Berührungspunkte zweier Projekte im Kontext der Chemnitzer Forschergruppe

R. Gunzenhäuser, A. Henninger, A. Sieber, G. Süß 5

Varying Cultures – A Challenge for Software Engineering?

A. Henninger, A. Sieber, W. Dilger 21

Softwareentwicklung in St. Petersburg

Erste Eindrücke

A. Sieber 37

Vorgehensmodelle, Projekttypen und Arbeitspraktiken als Bausteine zur realitätsnahen Beschreibung von Softwareentwicklung

A. Sieber, A. Henninger 51

Stil als Konzept zur Analyse von Kommunikationsarbeit

Zur Konvergenz soziologischer und linguistischer Perspektiven

J. Haase, F. Kleemann, I. Matuschek, C. Schulz-Nötzold 65

Personenverzeichnis

86

Liebe Leserinnen und Leser,

amerikanische Softwareunternehmen mit Niederlassungen in Deutschland, die Greencard für indische EntwicklerInnen¹, russische EntwicklerInnen, die zu Hause für deutsche Unternehmen programmieren – Presseberichten zufolge scheint so die heutige Softwareentwicklungslandschaft auszusehen. Einige erhoffen sich von dieser internationalen Kooperation technologische Fortschritte, während andere sich fragen, ob die unter solchen Umständen produzierte Software wohl funktionieren wird. Mit anderen Worten: Ist Softwareentwicklung ein Feld, in dem globale Kooperationen problemlos verwirklicht werden können, oder gibt es auch in der Softwareentwicklung so etwas wie Kulturen und Stile, die eine Zusammenarbeit nicht voraussetzungslos ermöglichen?

Mit dieser Frage beschäftigt sich das informatische Teilprojekt „Softwareentwicklung in der Praxis im Kulturvergleich“ im Rahmen der Chemnitzer DFG-Forschergruppe „Neue Medien im Alltag: Von individueller Nutzung zu soziokulturellem Wandel“. Die Ergebnisse dieses Teilprojekts, die in Auszügen in diesem Chemnitzer Informatik-Bericht veröffentlicht sind, legen eine positive Beantwortung dieser Frage nahe: Kulturen und Stile lassen sich bei der Entwicklung von Software finden und beschreiben. Sie stehen jedoch nur in Beziehung zu Ländergrenzen, wenn damit besondere soziale oder wirtschaftliche Bedingungen verknüpft sind, die bis in den Arbeitsalltag und das Leben der EntwicklerInnen wirken.

Die Begriffe ‚Kultur‘ und ‚Stil‘ selbst entstammen wissenschaftshistorisch gesehen nicht der Informatik. Die Kulturwissenschaften, die Soziologie und die Linguistik können jedoch auf eine Fülle von Diskussionsbeiträgen zu diesen Themengebieten verweisen. Aufgrund der interdisziplinären Zusammensetzung der Chemnitzer DFG-Forschergruppe² kann das informatische Teilprojekt jedoch auf die in diesen Fachgebieten entwickelten Kompetenzen bei der Identifikation und Analyse von Kulturen und Stilen zurückgreifen. Aus diesem Grund sind hier nicht nur Beiträge des informatischen Teilprojektes versammelt, sondern ebenso Aufsätze des kulturwissenschaftlichen, soziologischen und germanistischen Teilprojekts.

Im ersten Aufsatz „Von Technik zu Kultur und zurück“ zeigen das informatische und das kulturwissenschaftliche Teilprojekt auf, wie TechnikerInnen Kulturen und KulturwissenschaftlerInnen (Computer-) Technik im jeweiligen Kontext sehen. Beide Projekte können von den im anderen Fach vorhandenen Kompetenzen profitieren. Darüber hinaus führte der Diskurs zu einer Fragestellung, die sich nur in Zusammenarbeit zufriedenstellend lösen lässt.

Im Beitrag „Varying Cultures – A Challenge For Software Engineering?“ sucht das informatische Teilprojekt nach einem Ordnungssystem angesichts der Vielzahl der Kulturbegriffe und nach seiner eigenen Position in diesem Feld. Der Beitrag „Softwareentwicklung in St. Petersburg“ stellt

¹ Bezeichnungen von Personengruppen ohne konkreten Bezug werden mit „Innen“ gekennzeichnet. Ebenso Gruppen, zu denen weibliche und männliche Personen gehören. In allen anderen Fällen entsprechen die grammatikalischen Formen dem natürlichen Geschlecht der bezeichneten Personen.

² Zur Forschergruppe gehören zwei Projekte in der Informatik, zwei in der Soziologie, je eines in der Psychologie, Anglistik, Amerikanistik und Germanistik.

die Ergebnisse einer Vorstudie in dieser Stadt dar, die Erkenntnisse zur Struktur der Softwarebranche, den Akteuren und den Rahmenbedingungen liefert. Gleichzeitig wurden in dieser Studie auch die methodischen Schwierigkeiten für eine empirische Erschließung des Feldes deutlich.

Um „Vorgehensmodelle, Projekttypen und Arbeitspraktiken als Bausteine zur realitätsnahen Beschreibung von Softwareentwicklung“ geht es im vierten Beitrag. Hier versucht das informatische Teilprojekt, anhand eines Fallbeispiels bisherige normative Beschreibungen der Softwareentwicklung auf der Ebene der Projektorganisation mit deskriptiven Beschreibungen aus der Perspektive der Entwicklerinnen und Entwickler in konstruktiver Weise zu verknüpfen.

Eine andere Form der Beschreibung individuellen Arbeitshandelns zeigen das soziologische und das linguistische Teilprojekt auf. Sie führen in ihrem Aufsatz „Stil als Konzept zur Analyse von Kommunikationsarbeit“ das Analysekonzept ‚Stil‘ ein. Es ermöglicht die „dichte Beschreibung“ (Clifford Geertz) verschiedener Aspekte „subjektiver Arbeitsformen“ über mehrere Ebenen. Damit lässt sich sowohl der personale Arbeitsstil als auch der Kommunikationsstil einer Person explizieren.

Mit letztgenanntem Beitrag findet dieser Chemnitzer Informatik-Bericht seinen Abschluss. Die Diskussionen zu ‚Kultur‘ und ‚Stil‘ innerhalb der Forschergruppe werden jedoch mit Sicherheit weitergeführt. Wir bedanken uns bei den Autorinnen und Autoren der Beiträge für die Darstellung des aktuellen Diskussionsstandes. Martin Söns danken wir für die professionelle Unterstützung bei der Erstellung des Manuskripts. Wir profitierten bereits von der fachübergreifenden Kooperation und erhielten viele Anregungen für unsere eigene Projektarbeit. Dies wünschen wir auch der werten Leserschaft.

Evelyne Keitel & Werner Dilger

Chemnitz, im Januar 2001

Von Technik zu Kultur und zurück

Berührungspunkte zweier Forschungsprojekte im Kontext der Chemnitzer Forschergruppe¹

Randi Gunzenhäuser, Annette Henninger, Andrea Sieber,
Gunter Süß

Abstract The focus of this paper will be on the concepts of ‚culture‘ and ‚technology‘ in two traditionally unrelated fields of study, Software Engineering and Cultural Studies. These two disciplines are part of an interdisciplinary project at Chemnitz University of Technology, „New Media in Everyday Life: From Individual Use to Sociocultural Change“². The Cultural Studies project and the Software Engineering project both present their conceptual frameworks as well as their motivation toward bridging the gap between cultural and technical viewpoints respectively. In our opinion, a joint effort of these diverse disciplines can be especially productive with respect to the New Media, the study of digital products, and digital engineering.

The questions which have arisen from our cooperative interaction show that a major effort must be spent on convincing traditional scientists from both disciplines to overcome their narrow-minded concepts. Our work demonstrates that Software Engineering profits from the Cultural Studies‘ know-how regarding definitions of culture, while, at the same time, a Cultural-Studies approach benefits from the technical knowledge about programming provided by Software Engineering.

As far as the New Media are concerned, the cultures of technology and the technologies of cultures are at stake.

1 Einleitung

Die interdisziplinäre Arbeit in dem von der Deutschen Forschungsgemeinschaft geförderten Forschungsprojekt „Neue Medien im Alltag: Von individueller Nutzung zu soziokulturellem Wandel“ an der TU Chemnitz konfrontiert die beteiligten Fächer mit Fragestellungen und Methoden anderer Disziplinen. Die Arbeit in Einzelprojekten an Neuen Medien führt Fachrichtungen zusammen, die traditionell wenig Kontakt haben, wie etwa das Software Engineering und die amerikanistische Kulturwissenschaft. Im folgenden geht es um die Möglichkeiten einer produktiven Zusammenarbeit zwischen diesen beiden Fächern am Schnittpunkt zwischen Kultur und Technik. Wir stellen die spezifische Motivation beider

¹ Wir bedanken uns herzlich bei den Projektleitern, Evelyne Keitel und Werner Dilger, für die konstruktive Kritik an früheren Versionen dieses Aufsatzes. Vielen Dank auch an Angelika Hahn für ihre wertvollen Hinweise.

² The project is sponsored by the *Deutsche Forschungsgemeinschaft*, the German equivalent to the National Science Foundation (grant no. Bo 929/13-1).

Von Technik zu Kultur und

Einzelprojekte zur Zusammenarbeit dar, bevor die Projekte ihr Verständnis und ihren Zugang zu Kultur bzw. Technik aufzeigen. Abschließend erläutern wir, welche konkreten interdisziplinär zu bearbeitenden Fragestellungen sich aus den beschriebenen Forschungskontexten ergeben. Wir zeigen, dass die in der Kooperation erarbeiteten neuen Fragen traditionelle Axiome der jeweiligen Wissenschaften unterminieren können und *für beide Fächer* spannende Forschungsmöglichkeiten eröffnen.

Unser Ansatz geht von der These aus, dass sich zwei so weit auseinanderliegende Fächer wie das Software Engineering und die Kulturwissenschaft nicht nur über ihren Forschungsgegenstand austauschen, sondern auch in bezug auf ihre jeweilige Fragestellung gegenseitig befruchten können. Diese Entwicklung ist Teil eines großflächigeren Wandels der Wissenschaftslandschaft, den die Beschäftigung mit den Neuen Medien mit sich bringt: Technik und Kultur rücken immer offensichtlicher gemeinsam in den Blick.³ Digitale Produkte und Produktionsbedingungen stehen nicht nur in technischen, sondern auch in kulturellen Kontexten und wirken auf kulturelle Zusammenhänge zurück, die wiederum Einfluss auf technische Entwicklungen haben.

So wird bei der Untersuchung der Softwareentwicklung offensichtlich, dass verschiedene kulturelle Kontexte mitspielen: Aspekte formaler Ausbildung, Zielsetzungen bei der Programmierung, Formen der Arbeitsorganisation sowie technische und ästhetische Traditionen. Analysiert man etwa Computerspiele, stellt sich heraus, dass sie nicht nur von programmiertechnischen Voraussetzungen abhängig sind, sondern auch auf Spannung oder grafische Effekte hin entworfen wurden. Wer einen dieser Aspekte unberücksichtigt lässt, greift – so die Meinung vieler KulturwissenschaftlerInnen und einiger WissenschaftlerInnen im Software Engineering – zu kurz. Deshalb wird vielfach ein Umdenken in Richtung auf einen intensiveren interdisziplinären Austausch gefordert, also nicht nur ein Austausch zwischen verschiedenen Fächern, sondern auch zwischen Human- und Ingenieurwissenschaften, zwischen *humanities* und *engineering*.⁴

2 Motivationen

Das informatische Teilprojekt beschäftigt sich kulturvergleichend mit Softwareentwicklung in der Praxis. Der Arbeitsalltag von Software-EntwicklerInnen in unterschiedlichen Ländern und Regionen wird verglichen. Da in bestimmten Denktraditionen des Software Engineering subjektive und kulturelle Einflüsse auf die Softwareentwicklung postuliert werden (z. B. Floyd

³ Eine Vielfalt von kulturwissenschaftlichen Titeln zu den Neuen Medien und zum Internet beweisen, dass sich die jüngeren Kulturwissenschaften an den Neuen Medien geradezu abarbeiten. Hier dient die Technik den Kulturwissenschaften allerdings oft nur als Metapher für die Beschreibung zeitgenössischer kultureller Veränderungen. Siehe etwa den von Rob Shields herausgegebenen Sammelband *Cultures of Internet: Virtual Spaces, Real Histories, Living Bodies* oder die Aufsatzsammlung *Virtual Realities and Their Discontents*, die Robert Markley editiert hat, sowie den Band *Immersed in Technology: Art and Virtual Environments*, herausgegeben von Mary Anne Moser, die alle bereits 1996 im Titel die Brücke zwischen technischen und kulturellen Phänomenen schlagen. Umgekehrt beginnt das Software Engineering in jüngster Zeit, sich dem Kulturbegriff zu nähern, siehe etwa die Texte von Floyd und Schefe, die beide 1999 erschienen sind.

⁴ Siehe etwa die Einleitungen zu den kulturwissenschaftlichen Sammelbänden in Fußnote 3.

1999, Scheffé 1999), gingen wir davon aus, dass es hier tatsächlich um den Vergleich verschiedener ‚Kulturen‘ geht. In der Forschergruppe löste die Verwendung des Begriffs kontroverse Diskussionen aus. Wir hatten den Eindruck, dass wir uns zwischen zwei Möglichkeiten entscheiden müssen: Entweder müssen wir uns vom Kulturbegriff verabschieden, oder wir haben die Aufgabe, uns in ein riesiges Forschungsfeld mit einer langjährigen Tradition einzuarbeiten. Eine Hilfestellung bot uns das kulturwissenschaftliche Teilprojekt an, in dem Wissen über Traditionen des Kulturbegriffs und den Stand der Forschung zum Know-how gehört.

Das kulturwissenschaftliche Teilprojekt beschäftigt sich mit der Rezeption von fiktionalen Hypertexten, die digital erstellt und rezipiert werden, mit sogenannten *hyperfictions*. Der Verlauf der bisherigen Arbeit zeigt, dass die Analyse von Hyperfiktionen, von Computerkunst bzw. Computerspielen, nicht nur die Kenntnis ästhetischer Strukturen erfordert, sondern darüber hinaus technisches, im engeren Sinne programmiertechnisches Wissen verlangt, welches KulturwissenschaftlerInnen meist nicht besitzen. An diesem Punkt erweist sich die interdisziplinäre Zusammenarbeit mit der Informatik als hilfreich: Fragen zu Programmiersoftware, zum Vorgehen bei der Entwicklung von Hypertexten, zu *graphic engines* (Programmbibliotheken, die komplexe Grafikbefehle zur Verfügung stellen) oder anderen Entwicklungswerkzeugen können hier diskutiert werden. Derartige technische Rahmenbedingungen, ohne die viele ästhetische Entscheidungen und Wirkungsmechanismen nicht verständlich sind, lassen sich in Zusammenarbeit mit der Informatik ergründen.

Der Auslöser für die Zusammenarbeit mit dem jeweils anderen Fachgebiet war in beiden Fällen der Wunsch nach begrifflicher Klärung. Bei der Zusammenarbeit ergaben sich jedoch neue Herausforderungen in bezug auf das eigene Fach und die eigene Methode.

3 Erklärungen und neue Fragen

Die Suche nach Erklärungen für kulturelle Phänomene im Software Engineering (Abschnitt 3.1) und für technische Phänomene in der kulturwissenschaftlichen Rezeptionsforschung (Abschnitt 3.2) begann für beide Projekte in der eigenen Disziplin. Dabei entstanden jedoch auch Fragen, die sich nur im Austausch zwischen den Disziplinen – also interdisziplinär – klären lassen (Abschnitt 3.3).

3.1 Das Teilprojekt „Softwareentwicklung in der Praxis im Kulturvergleich“

Im informatischen Teilprojekt wird die Arbeitsweise von EntwicklerInnen auf individueller und kollektiver Ebene im Kontext von Organisation und Markt untersucht. Damit bewegt sich das Projekt in einem Forschungsbereich, der von mehreren Disziplinen bearbeitet wird.

3.1.1 Forschungsstand

Einmal handelt es sich um das Software Engineering – ein Teilgebiet der Informatik, das Softwareentwicklung methodisch unterstützen will. Als anschlussfähig erweisen sich hier insbesondere jene Arbeiten, die ein konstruktivistisches Verständnis dieses Prozesses haben (vgl. zur Begründung dieser Perspektive Floyd et al. 1992). Das Software Engineering entstand als Disziplin Ende der 60er Jahre aufgrund der Schwierigkeiten mit der Produktqualität und Projektorganisation beim Erstellen großer Programme. Die in diesem Bereich durchgeführten Forschungsarbeiten hatten von Anfang an normativen Charakter. Es ging darum, geeignete Modelle und Methoden zu entwickeln, an denen sich Software-EntwicklerInnen und ProjektmanagerInnen in der Praxis orientieren sollten.

Auch in bestimmten Bereichen der Soziologie – insbesondere in der Technik-, Wissens-, Arbeits- und Ingenieursoziologie – spielt die Analyse der Arbeitsweise von EntwicklerInnen (Strübing 1993, Funken 1998), die qualifikatorischen Anforderungen an ihre Arbeit (Baukrowitz/Boes/Eckhardt 1994, Dostal 1997), die Organisation des Software-Entwicklungsprozesses (Weltz/Ortmann 1992), aber auch in einem allgemeineren Sinne die Arbeit von Ingenieuren (Glock 1997) und der technologische Innovationsprozess (Rammert 1994, Schulz-Schaeffer 1997) eine Rolle. In diesen soziologischen Arbeiten steht die Motivation im Vordergrund, Teilbereiche gesellschaftlichen Lebens zu verstehen und in größere gesellschaftliche Zusammenhänge einzuordnen.

In der Psychologie, und hier insbesondere in der Arbeitspsychologie, wird mit anderen Methoden versucht, die Arbeit von EntwicklerInnen zu analysieren und zu verbessern. Die Expertiseforschung (Sonntag 1996), Laborstudien mit Programmierern zu sehr spezifischen Aufgaben (Soloway/Ivengar 1986, Koenemann-Belliveau/Moher/Robertson 1991, Cook/Scholtz/Spohrer 1993, Gray/Boehm-Davis 1996) und Tätigkeitsanalysen am Arbeitsplatz (Brodbeck/Frese 1994) haben zum Ziel, beobachtete Phänomene zu erklären und – in dieser Hinsicht ähnlich wie das Software Engineering – Hinweise auf Verbesserungsmöglichkeiten zu geben.

Somit liegen zu einzelnen Aspekten der Arbeit von EntwicklerInnen in der Praxis – wenn auch aus unterschiedlichen Blickwinkeln und mit unterschiedlichen Zielen – bereits einige Ergebnisse vor, an die wir anknüpfen können. An ihnen wird deutlich, dass folgende Ebenen bei der Untersuchung von Softwareentwicklung berücksichtigt werden müssen:

- die individuelle Arbeitsweise der EntwicklerInnen,
- die kollektive Arbeitsweise in Entwicklungsteams und in Organisationen,
- die Rahmenbedingungen.

Bei den Rahmenbedingungen sind insbesondere die regionale Wirtschaftsstruktur und der Anwendungsbereich zu nennen.

3.1.2 Kulturen in der Softwareentwicklung

Wer Softwareentwicklung kulturvergleichend untersucht, muss bestimmen, welche Aspekte auf individueller oder kollektiver Ebene bzw. bei den Rahmenbedingungen vergleichbar sind und worin kulturelle Momente bestehen könnten.

Auf der individuellen Ebene spielen die Orientierungen, die EntwicklerInnen in ihrer (beruflichen) Sozialisation ausbilden, und ihr Vorgehen eine wichtige Rolle. Diese Orientierungen entstehen während der Beschäftigung der EntwicklerInnen mit dem Computer in der Freizeit oder in der formalen Ausbildung. Entlang bestimmter Orientierungen etablieren sich die erstrebenswerten Ziele beim Programmieren. Diese Orientierungen ändern sich nicht beliebig, sondern weisen eine gewisse Widerständigkeit gegenüber Änderungen der kollektiven Arbeitsweise oder der Rahmenbedingungen auf. Anhand dieser Orientierungen lassen sich verschiedene Arbeitsstile bei den EntwicklerInnen unterscheiden. (Sieber 2001)

Ob die EntwicklerInnen beim Programmieren systematisch oder intuitiv vorgehen, ob sie eine Lösung vorher theoretisch durchdenken oder sofort mit der Umsetzung beginnen, ob sie *top down* oder *bottom up* an eine Aufgabe herangehen – in diesem Spannungsfeld bewegt sich das Vorgehen der EntwicklerInnen beim Lösen ihrer Aufgaben. Weizenbaum (1978), Strübing (1993) und Baukrowitz/Boes/Eckhardt (1994) beschreiben diesen Dualismus ebenfalls. Auch in unserer Untersuchung hat sich dieses Ergebnis bestätigt, allerdings konnten wir feststellen, dass diese Orientierungen in der Programmierarbeit nicht nur auf individuelle Vorlieben zurückzuführen sind, wie es die bisherigen Untersuchungen nahe legen. Vielmehr haben die Art der Aufgabe und die Erfahrung der EntwicklerInnen einen Einfluss darauf, welches Vorgehen bevorzugt wird und wie flexibel es gehandhabt wird. (Sieber 2001)

Auf der kollektiven Ebene beeinflussen die im Software Engineering verbreiteten Prozessmodelle die Organisation des Entwicklungsprozesses. Die bisher existierenden Modelle lassen sich in einer Genealogie mit drei Hauptästen darstellen (vgl. Bremer 1998). Jeder Ast steht für eine Gruppe von Modellen, welche die Interessen einer bestimmten am Entwicklungsprozess beteiligten Gruppe in den Blick nehmen. Die Phasenmodelle und ihre Folgeversionen, bei denen einzelne Tätigkeiten im Entwicklungsprozess nacheinander ausgeführt werden, konzentrieren sich auf die Interessen des Managements an Kontrolle und Steuerung. Die Tätigkeitsmodelle und ihre Folgeversionen strukturieren den Entwicklungsprozess so, dass er programmiertechnischen Aspekten wie dem Vorgehen der EntwicklerInnen oder den Strukturierungsschemata der Programmiersprache entgegenkommt. Die Sichtenmodelle stellen den Aushandlungsprozess zwischen den AnwenderInnen und den EntwicklerInnen in den Vordergrund (vgl. Henninger/Sieber/Dilger in diesem Band).

Während der individuelle Arbeitsstil der Software-EntwicklerInnen das Repertoire möglicher Vorgehensweisen im Entwicklungsteam bestimmt, hängt es in der Praxis von der Machtkonstellation zwischen Management, Entwicklung und Anwendung ab, welches der oben beschriebenen Prozessmodelle favorisiert wird. Die Art und Weise der Kommunikation und Kooperation, aber auch der gewählte Geschäftsbereich und die Arbeitsorganisation auf Unternehmensebene beeinflussen dabei den kollektiven Entwicklungsstil. Auch an den

Von Technik zu Kultur und

Produkten selbst zeigt sich, auf welches Spektrum an individuellen Arbeitsstilen das Unternehmen zurückgreifen kann. (Sieber 2001)

Die regionale Wirtschaftsstruktur ist bedeutend für die Ansiedlung von Unternehmen, aber auch für die Schwerpunktsetzung bei der Wahl des Geschäftsfeldes. In unserer Untersuchung zeigte sich, dass Unternehmen, die sich auf den Verkauf technischer Dienstleistungen konzentrierten, ihre Kunden in erster Linie im näheren Umfeld suchten. Unternehmen, die auf die Herstellung und Vermarktung eines eigenen Software-Produktes spezialisiert waren, suchten ihre Kunden global. In diesen Unternehmen erwies sich die Suche von Kunden jedoch als langwieriger Prozess. Deshalb griffen auch die produktorientierten Unternehmen in dieser Phase auf Aufträge von regionalen Kunden zurück. (Henninger/Sieber 2000)

Inwieweit die Struktur der Anwendungsbranche relevant für die Softwareentwicklung werden kann, zeigen Konrad/Paul (1999) im Rahmen ihrer Untersuchung organisatorischer Innovationsprozesse in der Branche. Die bisher von uns untersuchten Unternehmen entwickelten Software für verschiedene Anwendungsbereiche. Um die Softwareentwicklung vergleichend analysieren zu können, wäre es deshalb für weitere empirische Untersuchungen sinnvoll, mehrere Unternehmen zu untersuchen, die für einen Anwendungsbereich entwickeln.

Zusammenfassend lässt sich feststellen, dass Softwareentwicklung verschiedene Momente enthält, die sich bei der Beschreibung als geeignete Vergleichskategorien identifizieren lassen. Diese Momente sind Pfeiler der Beschreibung von Kulturen der Softwareentwicklung auf individueller bzw. kollektiver Ebene, die im Handeln der Beteiligten deutlich werden und die durch ihr Handeln entstehen. Da es sich dabei nicht um Phänomene handelt, die sich nur auf eine Person beziehen, sondern um solche, die mehrere Personen oder Unternehmen teilen, ist es unserer Meinung nach gerechtfertigt, von Kulturen zu sprechen.

Aber welcher Kulturbegriff ist für die vergleichende Beschreibung in der Softwareentwicklung geeignet? Bisherige Untersuchungen thematisieren in der Mehrzahl der Fälle Unterschiede zwischen den USA, Deutschland und Skandinavien (vgl. Henninger/Sieber/Dilger in diesem Band). Vergleiche zwischen einzelnen Ländern greifen kulturverdächtige Phänomene wie die Arbeitsteilung, die Werkzeugnutzung, Vorstellungen vom Nutzer oder landesspezifische Vorgehensmodelle des Software-Entwicklungsprozesses auf und verdeutlichen nationale Unterschiede zwischen ihnen. Allerdings haben diese Vergleiche den Nachteil, dass die darin beschriebenen Phänomene durch die nationale Unterscheidung stereotyp wirken, was der Realität vermutlich nicht gerecht wird. Hier kann die Beschäftigung mit dem Kulturbegriff und seinen Facetten hilfreich sein, eine bessere Möglichkeit der Beschreibung zu finden.

3.2 Das Teilprojekt „Interaktion mit fiktionalen Hypertexten“

Das kulturwissenschaftliche Teilprojekt nimmt innerhalb der Chemnitzer Forschergruppe eine besondere Stellung ein. Es fügt den empirischen Herangehensweisen der anderen Projekte eine kulturhistorische und -theoretische Dimension hinzu. Das Interesse richtet sich auf digitale Spiele, die im Rahmen des Projekts als *hyperfictions* definiert sind, also als Fiktionen, die auf einem PC hergestellt und über ihn rezipiert werden. Im Vordergrund stehen multimediale Texte, mit

denen PC-NutzerInnen spielerisch interagieren. Diese Hyperfiktionen liefern ein ‚Mehr‘ an immersiver⁵ Fiktion, denn sie bieten den NutzerInnen die Möglichkeit, sie nicht nur zu rezipieren, sondern aktiv mit ihnen zu kommunizieren und in sie einzutauchen, indem sie selbst eine Rolle im Spiel übernehmen.

Im Vergleich zur Buch- und Filmrezeption gestalten sich die Prozesse der Textaneignung durch die PC-NutzerInnen aktiver, bedingt durch eine neue Form des Erzählens: Das netzartige System von *links* und *knots*, aus denen Hypertexte bestehen, muss jeweils individuell erschlossen werden. Dieser Prozess erfordert ein kreatives Mitgestalten bei der Rezeption.

Der Begriff ‚Hypertext‘ wird sowohl im englisch- wie im deutschsprachigen Raum vor allem auf Netzseiten bezogen, die auf HTML-Programmierung basieren, durch *links* und *knots* verbunden sind und über einen Browser zugänglich gemacht werden. Sie können grundsätzlich neben Schrift auch andere Medien enthalten. Von kulturwissenschaftlicher Seite gerieten hier bisher vornehmlich künstlerische Netztexte in den Blick.⁶ Bei künstlerischen Hypertexten stellt sich allerdings das Problem der technischen Machbarkeit auf besondere Weise: Wie schnell werden High-End-Hard- und -Software und technisches Know-how von KünstlerInnen aufgegriffen und verwertet? Und wollen sie das überhaupt?⁷

Die Forschergruppe „Neue Medien im Alltag“ untersucht den vernetzten Computer unter den Aspekten Alltag und soziokultureller Wandel. Für das kulturwissenschaftliche Teilprojekt ist deshalb die enorme Breitenwirkung von kommerziellen Computerspielen interessant. Aus kulturwissenschaftlicher Perspektive lässt sich an der Popularität von Computerspielen Alltag und soziokultureller Wandel besonders gut untersuchen.⁸ Aus Sicht des Teilprojektes gibt es demnach Anlass, auch kommerzielle PC-Spiele als Hypertexte zu bezeichnen. Diese sind meist sehr aufwendig programmiert, schwelgen in opulenter Grafik und setzen verstärkt auch Ton- und Musikeffekte ein.

Bei kommerziellen, technisch ausgefeilten Spielen werden die *links* und *knots* nicht in gleicher Weise offensichtlich wie in HTML-basierten Hypertexten. Ihre Programmierstruktur wird in der Wahrnehmung der NutzerInnen interessanterweise durch die ‚Realitätseffekte‘ der Visualisierungsstruktur überlagert. Deutlich wird dies in der wahrgenommenen Spielbewegung: Der Erzählraum eines Spiels erweckt den Eindruck eines zu erschließenden Geländes, innerhalb dessen sich die NutzerInnen zu bewegen scheinen. Tatsächlich bewegen sich nicht die SpielerInnen von Ort zu Ort, sondern die ‚Räume‘ bewegen sich wie die Seiten eines HTML-

⁵ Immersiv, von (lat.) Immersion = Eintauchen, bezieht sich auf die Eigenschaft von Computerspielen, SpielerInnen in das Geschehen quasi hineinzuziehen.

⁶ Zu künstlerischen Hypertexten s. etwa Suter/Böhler (1999).

⁷ Da viele Künstlerinnen und Künstler gerade in Europa und Nordamerika ihre Hypertexte möglichst vielen Menschen auf der ganzen Welt zugänglich machen wollen – auch denen, die nur eingeschränkte technische Möglichkeiten haben – verzichten die AutorInnen oft auf den Einsatz von aufwendigen Bild- und Tondaten. Außerdem haben viele ideologische Bedenken gegen den Einsatz von Technik, da sie diese als kommerziell ablehnen. Schließlich mangelt es einigen an den nötigen Programmierkenntnissen bzw. an Geräten.

⁸ Die Kulturwissenschaften haben sich, historisch betrachtet, aus der Notwendigkeit heraus entwickelt, daß sich Phänomene der Populär- und Massenkultur mit den Theorien und Methoden der klassischen Geisteswissenschaften nicht fassen lassen.

Von Technik zu Kultur und

Textes auf die *user* zu, da über *links* immer neue grafische Szenen auf den Bildschirm gerufen werden. Orientierungseffekte sollen also meist den Eindruck einer tatsächlichen Bewegung im Raum erzeugen (s. etwa Wenz 2001).

Das Projekt unterscheidet zwischen schriftbasierten und multimedialen Hypertexten und untersucht solche Texte, die sich explizit auf künstlerische Traditionen berufen, ebenso wie Spiele, die etwa die klassischen Philologien bisher nicht als Gegenstand akzeptieren, weil sie zur Populärkultur gerechnet werden: kommerzielle Computerspiele. Hypertexte gibt es auf *read-only*-Speichermedien wie CD-ROMs, aber auch als Einzelplatzversion und Multi-User-Modelle außerhalb und innerhalb des WWW.

Texte im Netz charakterisieren sich darüber, dass sich ihre Gestalt durch die Interaktionen einer potentiell unendlichen Anzahl von NutzerInnen manipulieren und modifizieren lässt.

Multimediale Hypertexte sind nach unserer Definition Erlebnis- und Spielwelten, in denen Schrift oft eine untergeordnete Rolle spielt. Sie machen den SpielerInnen – manchmal mit Hilfe von Avatar-Figuren – diverse Rollenangebote. Kommerzielle Spiele vermitteln sich heute über zwei- und dreidimensionale Bild- und Animationssequenzen, das gesprochene und geschriebene Wort und über Ton sowie Musik. Die neuen, digitalen Technologien können demnach traditionell getrennt in Erscheinung tretende und gesondert analysierte ästhetische Ausdrucksformen verbinden. Konventionelle Grenzziehungen zwischen gedrucktem und gesprochenem Text, Stand- und bewegtem Bild, Ton und Musik, Film und Video werden zunehmend problematisch. Mit ihrer Verlagerung in virtuelle Räume werden genrespezifische Grenzziehungen und Ordnungskriterien, welche die Wissenschaften für traditionelle Schrift-, Bild- oder Musiktexte und bisherige Medien entwickelt haben, teilweise unbrauchbar.

Unbrauchbar wird angesichts der Medienvielfalt und offensichtlichen Abhängigkeit von der Technik auch die Abgrenzung der Philologien und teilweise der Kulturwissenschaften zu technikorientierten Fragestellungen. Fiktionale Hypertexte sind aufgrund des Mediums, ohne das sie – ähnlich wie der Film – weder hergestellt noch rezipiert werden können, nicht denkbar. So sind sie – im Vergleich zum Buch – besonders offensichtlich nicht nur nach ästhetischen Prinzipien strukturiert, sondern auch von technischen Voraussetzungen abhängig. Es geht im Projekt darum, „Neuland im telematischen Raum zu vermessen, Textlandschaften anzulegen, Schreiben und Lesen eben auch als einen nomadischen Akt des Umherschweifens durch Text-Netzwerke zu begreifen.“ (Idensen 1996, S. 155). Zugleich aber wollen wir diese und ähnliche Metaphorisierungen des Digitalen *als einem Raum* kritisch hinterfragen. Die Kulturwissenschaften beschreiben das Digitale nämlich immer wieder als Erlebnisraum, der sich vor den NutzerInnen zum beliebigen Gebrauch auftut, als besäße er keine vorgegebene Erschließungsstruktur.

Von technischer Seite aus gesehen ist diese Wahrnehmung, die sich auf die Wirkung der räumlichen Effekte stützt, so nicht richtig. Die Struktur eines Hypertextes muss auch in bezug auf seine Erschließungsmöglichkeiten zu einem frühen Zeitpunkt der Programmierung festgelegt werden – im Vergleich zum Schreiben eines Buches oder dem Drehen eines Filmes sogar sehr früh. Die Bewegungs- und Manipulationsfreiheit, die den SpielerInnen suggeriert wird, erweist sich bei genauerem Hinsehen als – sehr erfolgreicher – technischer Realitäts- und

Immersionseffekt.⁹ Ein Spiel ist um so effektiver, je ‚offener‘ es wirkt, je stärker es also die programmierseitig vorgegebene Struktur vergessen lässt. Diese technische Struktur kann nur durch ein Untersuchen der Technik entdeckt werden, weil es eine der ästhetischen und genrespezifischen Leistungen eines Hypertextes ist, sie in der Spielbewegung vergessen zu lassen.

3.2.1 Ästhetik und Technik

In Auseinandersetzung mit den spielerischen Textpraktiken der Neuen Medien müssen die Philologien und die Kulturwissenschaften ihren Textbegriff überprüfen: Herkömmliche, auf Geschlossenheit drängende Definitionen von Text, Identität und Körper sehen sich angesichts der multimedialen, interaktiven, offenen Formen unter Druck gesetzt. Bereits in den 60er Jahren forderte Roland Barthes, den Werkbegriff durch das Konzept des offenen Textes zu ersetzen. Dieser Forderung kam die Kulturwissenschaft in vielerlei Hinsicht nach; sie operiert mit einem offenen Textbegriff, der neben Drucktexten auch Filme, Videos oder Musikstücke berücksichtigt und neben im engeren Sinne künstlerischen auch populäre und kommerziell erfolgreiche Texte einbezieht.

Hypertexte stellen heute die Wissenschaften, die sich vorwiegend mit narrativen und kulturellen Strukturen befassen, besonders offensichtlich vor Probleme. Wie die Kulturwissenschaftlerin Susana Pajares Tosca (2000) in ihrem Beitrag zu Abenteuerspielen auf dem Computer konstatiert, verändern Computerspiele das Erzählen ganz grundsätzlich, so dass traditionelle Fragen nach narrativen Strukturen, nach dem *plot* und seiner Rezeption durch Konzepte wie Interaktivität oder Immersion ergänzt werden müssen. Die Gegenüberstellung eines offenen, poststrukturalistischen Textbegriffes einerseits und eines (programmier-)technischen Strukturbegriffes andererseits ergibt interessante Widersprüche und führt zu neuen Erkenntnissen.

Von technischer Seite betrachtet, zeichnen sich gerade kommerzielle Spiele weniger durch Offenheit und Flexibilität ihrer programmierten Struktur aus, hingegen erweisen sie sich – wieder unter Berücksichtigung technischer Gegebenheiten – in bezug auf die Genreentwicklung als erstaunlich flexibel. Wichtige Genres bei Computerspielen auf dem PC sind etwa *adventure*, Rollenspiel, Strategiespiel, Simulation und *action*-Spiel. Die Grenzziehungen und Eigenschaften dieser Spielsorten bleiben nie stabil, woran nicht zuletzt die rasanten technischen Entwicklungen schuld sind.

So veränderte etwa das Aufkommen der 3D-Rendering-Grafik zwangsläufig die Gestalt einzelner Genres und löste Verschiebungen zwischen Genregrenzen aus. Die Feinheit der Navigation, die Geschwindigkeit des Bildaufbaus, der Einsatz von Ton hängen vom technischen Stand ab und ermöglichen Einzel- oder Mehrfachspielermodus, anspruchsvolle Grafik- oder Toneffekte, schnelle Bewegungen auf dem Bildschirm. All diese Aspekte sind zugleich ästhetisch und spieltechnisch relevant und für verschiedene Gruppen von SpielerInnen mehr oder weniger interessant.

⁹ Er geht so weit, dass die Spielerin zu ihrer Orientierung oft sogar eine Karte zeichnen muss, auf der die Verlinkungen räumlich erscheinen.

Von Technik zu Kultur und

Der konkurrenzbetonte Spielmarkt führt dazu, dass Firmen originell und innovativ auf technische Veränderungen reagieren müssen. Plagiate verzeihen die passionierten und informierten SpielerInnen nicht (s. Dees/Schmitz 2000). Dementsprechend locken Spielefirmen KäuferInnen mit neuen Spielideen und Varianten, aber auch mit immer größerer technischer Perfektion. So verändert sich die Gestalt von Spielen ständig, es werden neue Formen kreiert, bestehende abgewandelt. *Real-time*-Szenen sind immer häufiger anzutreffen, mehr Spiele bieten Mehrspielermodi, rundenorientierte Spiele können zunehmend durch *games* ersetzt werden, die simultan von mehreren Beteiligten gespielt werden. Nicht immer verlaufen diese Entwicklungen geradlinig. So werden spezifische, auf das jeweilige Genre zugeschnittene Eingabegeräte wie Joysticks am PC immer seltener, Tastatur und Maus gewinnen an Bedeutung.

All diese Veränderungen wirken auf die Vorlieben bzw. das Können von SpielerInnengruppen ein, während man auch davon ausgehen kann, dass spezifische Vorlieben sich auf die Verkaufszahlen niederschlagen. Das Geschlecht ist hier eine wichtige Größe: Nach wie vor werden die meisten Computerspiele explizit von Männern für Männer konstruiert, was sich nicht zuletzt genrespezifisch äußert – so genannte Ballerspiele laufen allen anderen Genres den Rang ab, und die Protagonisten sind meist nach wie vor männlichen Geschlechts. Doch es gibt auch einige Frauenfiguren unter ihnen, etwa die knallharte Amazone Lara Croft, der Star von *Tomb Raider*, dessen fünfter Teil Ende 2000 erschienen ist, oder April Ryan, die fehlbare Hauptfigur des *adventure*-Spiels *The Longest Journey*. Inwieweit sich diese beiden Spielehits allerdings in Bezug auf technische Aspekte der Identifikationsangebote und Körperkonzepte von anderen unterscheiden, ist bis jetzt noch kaum erforscht.¹⁰

Gerade bei Betrachten der technischen Aspekte weist einiges darauf hin, dass Spiele auch geschlechtsspezifisch rezipiert werden. Es spielen nämlich überwiegend entweder die Besitzer leistungsstarker, gut ausgestatteter PCs oder Spielende, deren Bastelwut wett macht, was auf Seite der Performance fehlt – in beiden Fällen handelt es sich häufiger um Männer als um Frauen. Auf stärker inhaltlicher Ebene lässt sich vermuten, dass der spektakuläre Einsatz der Tricktechnik, die gerade männliche Rezipienten von Science-fiction-Filmen fasziniert, bei Computerspielen einen ähnlichen Effekt hat. Das Zurschaustellen der technischen Höchstleistungen eines neuen Mediums findet traditionell beim männlichen Publikum Gefallen.¹¹

Angesichts konkreter Textstrukturen stellt sich die Frage nach der Offenheit dieser Texte ebenso neu wie die nach der *agency*, den Entscheidungs- und Handlungsspielräumen von SpielerInnen in einem Hypertext-System.

Denn neben den medienspezifischen ästhetischen Erzählstrategien kommen in Hypertexten auch programmiertechnische und -organisatorische Aspekte zum Tragen. Einerseits sind sie von den Verknüpfungs- und Kommentierungsmöglichkeiten der gewählten Hypertext-Programme

¹⁰ Interessanterweise abstrahieren KritikerInnen bei solchen Spielen notorisch von den technischen Aspekten, um sich ideologischen Gesichtspunkten zu widmen. Dabei ist gerade hier die Figurenführung und das Verhältnis von Körperanimation und grafischer Umgebung interessant, s. zu diesen Aspekten Gunzenhäuser (2000).

¹¹ Dieses Argument lässt sich durch ein gegenläufiges, ebenfalls geschlechtsspezifisches, ergänzen: In manchen (ebenfalls männlich dominierten) Insider-Kreisen ist gerade eine schlanke, wenig aufwendige Programmierung besonders angesehen, etwa wenn es darum geht, dass der PC kulturell als Demokratieggarant fungieren soll.

abhängig, mit deren Hilfe sie erstellt wurden. Andererseits haben individuelle Zielsetzungen und Arbeitsweisen im Team während der Entwicklung komplexer Hypertexte einen Einfluss auf die narrative und ästhetische Geschlossenheit, gerade auch im Zusammenspiel von Bild, Bewegung, Ton oder Schrift. Die Tatsachen technologischer Kultur führen Konzepte wie das der individuellen Autorschaft, der passiven Rezeption und der ‚richtigen‘ Interpretation ebenso offensichtlich ins Abseits wie das des geschlossenen Werkes.

3.2.2 Technologie-Kultur

Ein Vergleich der Aneignung verschiedener Medien zeigt, dass die technischen Bedingungen von entscheidender Bedeutung für spezifische Rezeptionsbedingungen sind. Diesen Punkt betont der Medienhistoriker Timothy Druckrey (1996) in der Einführung zu seiner Anthologie in bezug auf alle elektrischen und elektronischen visuellen Medien wie Photographie, Film und Video.

Grundsätzlich gilt diese Feststellung nicht nur für die Neuen Medien oder die Gegenwart, vielmehr lassen sich Beispiele der Konvergenz zwischen Kunst und Technik in allen Medien und zu jeder Zeit finden. Historisch betrachtet, zeigt die Arbeit von Leonardo da Vinci, dass einst technische, wissenschaftliche und künstlerische Entwicklungen in der bildenden Kunst ganz selbstverständlich zusammenliefen. Friedrich Kittler (1995) demonstriert, dass Technologien auch die Literatur beeinflusst haben, und es ist ein (in den Philologien oft vernachlässigter) Gemeinplatz, dass der jeweilige Stand der Drucktechnik maßgeblich für die Verbreitung von Schrifttexten war und ist.

Auch theoretische Formulierungen wie die der französischen Semiologie (Barthes 1988) stützten sich auf technische Modelle, etwa das der Kybernetik. In der Geschichte des Films finden sich viele Beispiele für die Wichtigkeit technischer Entwicklungen. *Tron* brachte 1982 nicht nur die virtuellen Welten der Computerspiele auf die Leinwand, sondern wäre ohne die neuesten Animations-Computertechnologien seiner Zeit gar nicht zu realisieren gewesen: Szenenweise waren Raumtiefe, Licht und Schatten computergeneriert; Form und Bewegung von Gegenständen und grafischer Kontext der Figuren im Cyberspace wurden durch die damaligen Möglichkeiten der MAGI-Synthavision-Programmierung bestimmt (s. Bukatman 1993, S. 217).

Bei Hypertexten auf dem PC kommen noch mehr technische Aspekte zum Tragen. So spielt etwa der Ton aufgrund von technologischen und kulturellen Wandlungsprozessen innerhalb kommerzieller Hypertexte eine immer größere Rolle. Durch das Entstehen neuer Hardwarekomponenten und Soundsoftware (z. B. 4-Kanal-Ton, *Environmental Audio* oder *Surroundsound*) bieten sich den EntwicklerInnen von Hypertexten neue Möglichkeiten der Integration von Audiodaten in die narrative oder emotional-appellative Struktur dieser Texte (s. Weske 2001¹²).

In einem Hypertext sind Fragen der Bildauflösung und der Übertragungsrates von Tondaten entscheidend für die Wahl, Zusammenstellung und Wichtigkeit einzelner Medien innerhalb des Monomediums Hypertext. Denn ein schriftzentrierter Hypertext kann mit bedeutend geringerem

¹² Die programmierhistorische Arbeit Weskes im Rahmen des Teilprojektes Keitel stellt ein Beispiel für die Zusammenarbeit zwischen Informatik (dem Teilbereich Akustik) und der Kulturwissenschaft dar.

technischem und finanziellem Aufwand gespeichert oder sogar ins Netz gestellt werden als ein multimedialer. Er kann auch auf kleinen PCs abgespielt werden und ist somit für mehr *user* zugänglich. Außerdem ist er leichter auf verschiedene Betriebssysteme portierbar und über die Zeit hinweg gesehen resistenter gegen die ständige Veränderung der Computerhard- und Software.

Sind die technischen Voraussetzungen und Entwicklungen etwa beim Buchdruck oder Film zumindest bis zum Einsatz des Computers noch relativ überschaubar, so erfordert die Analyse der visuellen Komponenten eines Computerspiels, das auf einer speziellen *graphic engine* aufsetzt, bereits sehr spezifische Computerkenntnisse; die zunehmend anspruchsvolle Einbindung von Tondaten setzt auch Kenntnisse auf dem Gebiet der *sound*-Generierung voraus. Informationen über die Arbeitsweise bei der Entwicklung all dieser Aspekte eines Spieles sind noch schwerer zu bekommen. Diese Lücke kann die Zusammenarbeit mit dem Fach Software Engineering schließen helfen.

3.3 Berührungspunkte

Aufbauend auf dem in den Projekten bisher entwickelten Verständnis für die jeweils anderen Begrifflichkeiten eröffnet sich ein neues Themenfeld, das sich einer innerdisziplinären Klärung jedoch noch mehr entzieht. Gemeint ist hier der Zusammenhang zwischen dem Entwicklungsprozess und den Rezeptionsmöglichkeiten, die sich an der Software, d. h. an den dem Erscheinungsbild fiktionaler Hypertexte zugrundeliegenden ‚Quelltexten‘ festmachen.

Der Entwicklungsprozess, das Vorgehen der EntwicklerInnen und ebenso ihre Konstruktion der Realität sind subjektive bzw. kollektive Leistungen, die in einem bestimmten Kontext stattfinden. Sie unterliegen kulturellen Einflüssen. Im informatischen Teilprojekt lieferte die empirische Analyse von Softwareentwicklung in Deutschland interessante Hinweise auf Zusammenhänge zwischen dem Vorgehen und der Prozessgestaltung, aber auch zwischen diesen beiden Bereichen und dem Produkt. In jenem dem Produkt zugrundeliegenden Quelltext sind diese Annahmen für die Maschine verständlich dokumentiert. In der konkreten Software äußern sich also auch kulturelle Einflüsse.

Trotz alledem bricht sich das Kulturverständnis der am Entwicklungsprozess Beteiligten erneut, wenn der Quelltext gelesen wird, – im konkreten Fall beispielsweise ein Spiel gespielt. Dabei fällt auf, dass die Interpretationsmöglichkeiten des Quelltextes nicht unendlich sind. Bestimmte Arbeitsstile, bestimmte Arbeitsformen im Kollektiv bedingen Möglichkeitsräume von unterschiedlichem Aussehen und von unterschiedlichem Ausmaß. Auch interaktive und offene Hypertexte sind nicht unbegrenzt interaktiv und offen. Im Gegenteil, sie offenbaren für SpielerInnen gelegentlich Grenzen, die ihnen ganz und gar nicht einleuchten. Hier kann eine gemeinsame Arbeit ansetzen, um herauszufinden, welche Vorgehensweise in der Entwicklung, welche Programmierziele, Entwicklungstools, Kooperationsformen und regionale bzw. nationale Settings zu welchen Begrenzungen der Interpretationsmöglichkeiten führen.

4 Fazit

Vor dem Hintergrund der organisatorischen Veränderungen in der Softwarebranche – der zunehmenden Internationalisierung der Softwareentwicklung – ist eine Analyse dieser Arbeit in kulturvergleichender Perspektive notwendig. Einige Arbeiten auf diesem Gebiet deuten Unterschiede zwischen verschiedenen Ländern an. Eine nationale Verortung bestimmter Entwicklungspraktiken erschiene im Hinblick auf die verschiedenen Ebenen, die für die Softwareentwicklung in der Praxis relevant sind, jedoch als verkürzt. Hier hilft der Blick in die Kulturwissenschaften. Die dort geleisteten theoretischen Vorarbeiten zur Geschichte des Kulturbegriffes und die Vielzahl seiner möglichen Definitionen sind ein hilfreicher Ausgangspunkt für das Software Engineering mit dem Ziel, eigene Konzepte zu entwickeln.

Die Andersartigkeit von Hypertexten im Vergleich zu konventionellen literarischen oder filmischen Texten erfordert ein hohes Maß an Aktivität, Kreativität und Flexibilität von den NutzerInnen, aber auch von denen, die solche Texte analysieren wollen. Es müssen neue Kompetenzen entwickelt werden, um Interaktionen mit *hyperfictions* zu verstehen. Eine Kulturwissenschaft, die herausarbeiten will, welche Strukturen die fiktionalen digitalen Texte typischerweise beinhalten und welche spezifischen Interaktionsangebote sie ihren NutzerInnen machen, ist auch auf die Kooperation mit der Informatik angewiesen. Nur so können spezifisch technische Entwicklungen kulturtheoretisch verwertbar gemacht werden.

Damit werden Kultur und Technik in einen für beide Fächer produktiven Zusammenhang gestellt, die Kulturen der Technik und die Technik der Kulturen untersucht und theoretisiert.

5 Literatur

- Barthes, Roland: Image, Music, Text. Trans. Steven Heath. New York : Hill & Wang, 1988.
- Baukowitz, Andrea; Boes, Andreas; Eckhardt, Bernd: Software als Arbeit gestalten. Konzeptionelle Neuorientierung der Aus- und Weiterbildung von Computerspezialisten. Westdeutscher Verlag : Opladen, 1994.
- Bender, Gretchen; Duckrey, Timothy (Eds.): Culture on the Brink: Ideologies of Technology. Seattle : Bay Press, 1994.
- Bremer, Georg: Genealogie von Entwicklungsschemata. In: Kneuper, Ralf; Müller-Luschnat, Günther; Oberweis, Andreas (Hrsg): Vorgehensmodelle für die betriebliche Anwendungsentwicklung. Stuttgart; Leipzig : Teubner, 1998, S. 32- 59
- Brodbeck, Felix C.; Frese, Michael (Hrsg.): Produktivität und Qualität in Software-Projekten.. Psychologische Analyse und Optimierung von Arbeitsprozessen in der Software-Entwicklung. München ; Wien : Oldenbourg Verlag, 1994.
- Bukatman, Scott: Terminal Identity: The Virtual Subject in Postmodern Science Fiction. Durham, London: Duke U. P., 1993.
- Cook, Curtis R.; Scholtz, Jean C.; Spohrer, James C. (Eds.): Empirical Studies of Programmers: Fifth Workshop. Proceedings. Norwood (NJ) : Ablex Publishing Corporation, 1993.
- Dees, Michael; Schmitz, Peter: Senkrechtstarter. Spielerische Höhepunkte des Jahres. In: c't, H. 25, 2000, S. 184-190
- Dostal, Werner: Informatik-Qualifikationen im Arbeitsmarkt. In: Informatik Spektrum Jg. 20, H. 2, 1997, S. 73-78

Von Technik zu Kultur und

- Druckrey, Timothy (Ed.): *Electronic Culture: Technology and Visual Representation*. Denville (NJ) : Aperture, 1996.
- Floyd, Christiane: *Software Development Process: Some Reflections on its Cultural, Political and Ethical Aspect from a Constructivist Epistemology Point of View*. In: *Cybernetics & Human Knowing*, Vol. 6, No. 2, 1999, pp. 5-18
- Funken, Christiane: *Perspektivenwechsel. Eine wissenschaftssoziologische Standortbestimmung der Aufgabenanalyse im Rahmen der Software-Entwicklung*. Universität Freiburg, Habilitation, 1998.
- Glock, Friedrich: *Zur Soziologie des Konstruierens*. Berlin : Wissenschaftszentrum Berlin für Sozialforschung, 1997.
- Gray, D. Wayne; Boehm-Davis, Deborah A. (Eds.): *Empirical Studies of Programmers: Sixth Workshop. Proceedings*. Norwood (NJ) : Ablex Publishing Corporation, 1996.
- Gunzenhäuser, Randi: *Darf ich mitspielen? Literaturwissenschaften und Computerspiele*. 2000, <http://computerphilologie.uni-muenchen.de/jg00/gunzenh/gunzenh.html> (Stand: 12.01.2001).
- Henninger, Annette; Sieber, Andrea: *Softwareentwicklung in kleinen und mittleren Unternehmen in Ost- und Westdeutschland*. Vortrag auf der Tagung „Neue Medien im Arbeitsalltag“ am 10./11.November 2000, <http://www.tu-chemnitz.de/NeueMedien/NeMeAA2000/> (Stand 12.01.2001)
- Idensen, Heiko: *Die Poesie soll von allen gemacht werden*. In: Matejovski, Dirk; Kittler, Friedrich A. (Hrsg.): *Literatur im Informationszeitalter*. Frankfurt ; New York : Campus, 1996.
- Kittler, Friedrich A.: *Aufschreibesysteme 1800 • 1900*. 3. Auflage. München : Fink, 1995.
- Koenemann-Belliveau, Jürgen; Moher, Thomas G.; Robertson, Scott P. (Eds.): *Empirical Studies of Programmers: Fourth Workshop. Proceedings*. Norwood (NJ) : Ablex Publishing Corporation, 1991.
- Konrad, Wilfried; Paul, Gerd: *Innovation in der Softwareindustrie: Organisation und Entwicklungsarbeit*. Frankfurt am Main ; New York : Campus, 1999.
- Markley, Robert (Ed.): *Virtual Realities and Their Discontents*. Baltimore : Johns Hopkins University Press, 1996.
- Moser, Mary Anne (Ed.): *Immersed in Technology: Art and Virtual Environments*. Cambridge (Mass.) : Massachusetts Institute of Technology, 1996.
- Rammert, Werner: *Prozesse der Technikgenese und der Stellenwert von Leitbildern in der Technikentwicklung*. In: Fricke, Else (Hrsg.): *Zur Zukunftsorientierung von Ingenieuren und Naturwissenschaftlern*. Bonn : Friedrich-Ebert-Stiftung, 1994, S. 9-18
- Scheffe, Peter: *Softwaretechnik und Erkenntnistheorie*. In : *Informatik-Spektrum* Jg. 22, H. 2, 1999, S. 122-135
- Schulz-Schaeffer, Ingo; Jonas, Michael; Malsch, Thomas: *Innovation reziprok: Intermediäre Kooperation zwischen akademischer Forschung und Industrie*. In: Rammert, Werner; Bechmann, Gotthard (Hrsg.): *Technik und Gesellschaft. Jahrbuch 9*. Frankfurt am Main ; New York : Campus, 1997, S. 91-124
- Shields, Rob (Ed.): *Cultures of the Internet: Virtual Spaces, Real Histories, Living Bodies*. London : Sage Publications, 1996.
- Sieber, Andrea: *Software entwickeln in der Praxis. Vier Fallstudien in kleinen Unternehmen Ostdeutschlands*. Chemnitz, 2001. (unveröffentlichtes Manuskript)
- Soloway, Elliot; Iyengar, Sitharama (Eds.): *Empirical Studies of Programmers. Papers presented at the First Workshop on Empirical Programmers, June 5-6, 1986, Washington, DC*. Norwood (NJ) : Ablex Publishing Corporation, 1986.
- Sonntag, Sabine: *Experten in der Software-Entwicklung. Untersuchung hervorragender Leistungen im Kontext intellektueller Teamarbeit*. Universität Gießen, Habilitation, 1996.
- Strübing, Jörg: *Subjektive Leistungen im Arbeitsprozess. Eine empirische Untersuchung von Arbeitsstilen in der Programmierarbeit*. Wiesbaden : Deutscher Universitätsverlag, 1993.

- Suter, Beat; Böhler, Michael (Hrsg.): Hyperfiction. Hyperliterarisches Lesebuch: Internet und Literatur. Basel ; Frankfurt a. M. : Stroemfeld, 1999.
- Toska, Susana Pajares: Playing for the Plot: Blade Runner as Paradigm of the Graphic Adventure Game. 2000, <http://www.dichtungdigital.de/2000/Tosca-31-Mai/> (Stand: 12.01.2001).
- Weizenbaum, Joseph: Die Macht der Computer und die Ohnmacht der Vernunft. Frankfurt a. M.: Suhrkamp, 1978.
- Weltz, Friedrich; Ortmann, Rolf G. (Hrsg): Das Softwareprojekt: Projektmanagement in der Praxis. Frankfurt am Main ; New York : Campus, 1992.
- Wenz, Karin: Narrativität in Computerspielen. 2001, <http://www.dichtung-digital.de/Wenz/30-Juni-99-N/intro.htm> (Stand: 12.01.2001).
- Weske, Jörg: Digital Sound and Music in Computer Games. 2001, <http://www.tu-chemnitz.de/~jwes/GAMESOUND/> (Stand: 12.01.2001).

Varying Cultures – A Challenge For Software Engineering?

Annette Henninger, Andrea Sieber, Werner Dilger

Abstract Software development is an interactive process that needs interpretations of real world situations. These interpretations, but also personal practices for work, co-operation and communication are influenced by cultural impacts. An examination of these culture-influenced areas of software engineering will elucidate different approaches on software development in practice. Based on a specification of our understanding of culture, we distinguish cultural frames on the organisational, professional and societal level. These frames influence the activities of developers. Vice versa, cultural frames are permanently changed through the individuals' activities. To describe these frames and activities more precisely, we refer to research on the process models of software development and on the activities of developers in empirical studies. In our conclusion, we suggest a research design to explore different individual and organisational approaches to software development.

1 INTRODUCTION

In present day software industry, large enterprises such as Microsoft, Hewlett Packard and others sub-contract sections of development projects to other countries. In addition to organisational co-operation, software developers often work together on their own initiative. Individuals might use newsgroups to discuss technical problems or to evaluate new technological products; or they might keep software on servers for others to experiment with or improve.

Well-developed international relationships among individuals as well as commercial organisations of software developers are widely believed to be due to the fact that software development is a formal activity, mostly based on mathematics and logic (Coy 1997). In other words, the very tools and programming languages that these individuals and companies use in their profession are also instrumental in enabling communication and cooperation with others worldwide. However, Schefe (1999) shows that software systems and development activities are embedded in real world situations with the consequence that the process of developing software has much more to do with ethnology and interpersonal understanding than with natural science. In his opinion a developer uses signs to describe real world situations, but the relation between the real world situations and these signs is socially constructed. This is the point where cultural factors can influence software development: each subject will produce more or less different interpretations or descriptions, depending on how he or she interprets a situation and which signs he or she chooses to describe it.

Another reason for assuming that software development is influenced by culture arises from Rammert's understanding of technogenesis in general. In his opinion technology development is

an open process, which emerges through the interaction of cultural orientation and social constellation on the micro-level, and from social institutions and the selective structures framing them on the macro-level. According to Rammert's view, cultural orientations involve visions of use, models of the users and ideals of construction, which are all more than individual. The social constellation describes the local relations between the participants in the construction process in a given context: who stays in contact with whom, who works with whom, and who exchanges ideas with whom? In his model, standardisation committees and traditions of construction, in the educational institutions as well as in the organisation, constitute the social institutions. Economy, politics and culture constitute the selective structures. (Rammert 1994, p.12)

Taking these results seriously, it seems reasonable to suppose that software engineering is influenced by cultural variations as so defined. What has not been discussed so far, however, are the areas affected and in which way they might be affected. How can they be described, and how can we learn more about cultural impacts and make them work positively for software development? With these questions in mind we started an empirical research project which analyses software development in small enterprises in Germany, the USA and Russia. This article is part of the theoretical framework which underlies our empirical analysis.

To answer the questions posed above we first need to explore existing understandings of culture and to clarify our own assumptions, which we will discuss in section 2. This section will lay out the foundation for our perspective in examining software engineering and any relevant influences of culture. In section 3 we analyse selected research findings on the software development process which have proven relevant for describing structural frames and individual activities which are sensitive to cultural influences. Finally, in section 4 we propose a research program in order to elucidate cultural impacts in software development.

2 The Understanding of Culture

Cultural influences are not often recognized or discussed within computer science. When we use this term we have to bear in mind its long history and tradition in anthropology and cultural studies. In the first paragraph of this section we try to categorise research findings related to this topic. In the second, we explicate our assumptions on the basis of a review of the relevant literature.

2.1 Distinctions

A cursory glance at the literature related to cultural issues shows the variety of approaches used. In anthropology alone there are 164 different definitions of the term 'culture' (Schneider/Barsoux 1997, p. 19). Also, the notion of 'culture' has changed a lot during the historical development of the concept, from its first application in anthropological studies of small and isolated tribes to modern cultural studies.

In current research, culture is seen as a multi-faceted, continuously changing and anything but uniform entity. For Bhabha, individuals interpret and change cultural messages as they may wish,

so that „Cultures are never unitary in themselves, nor simply dualistic in the relation of Self to Other.“ (Bhabha 1994, p. 35). Wicker goes even further. He also points out that culture cannot be understood as merely a complex unit which is the sum of unchangeable characteristics. Therefore the main interest of research dealing with culture cannot be a description of these ever-changing characteristics, but rather an identification of the (relatively stable) rules of possible transformations. For Wicker, not only population movement, but also the global information exchange beyond territorial borders are creating principles for cultures. For him culture is „ ... the ability to produce reciprocal symbolic relations and to form meaning through interaction.“ (Wicker 1997, p. 39)

The historical dimension, however, is only one way to categorise the different approaches in research on culture. Another possibility is to distinguish the research designs according to the preferred *perspective* on culture, the *objects* examined in the research, and the research *method* used (see Bolten 1995, Demorgon/Moltz 1996).

- Perspective

The perspective of cultural studies primarily focuses on culture as changeable or as constant. The studies which focus on cultural change emphasise the evolution and differentiation of human action and describe the dynamic aspects of culture. The perspective that arises from this approach stresses the individual's influence on cultural phenomena (see Demorgon/Moltz 1996, p.44). Other studies which explore cultural contingencies or the static aspects of culture consider culture to be a constant situational influence. These studies stress definite features of cultures, assuming that these features change only over a long period of time. This perspective focuses on cultural influences on individuals (Demorgon/Moltz, p. 68f.).

- Objects

Another possibility for researching culture is to differentiate between research objects, for example artefacts, shared knowledge or social practices (Bolten 1995, p. 28). These objects can be examined on different levels. Artefacts can be explored as independent objects but also with respect to their genesis and use. Shared knowledge and social practices can be explored in relation to individuals or groups. It is important to take into account how individuals and groups are therefore said to be related to each other and to cultural phenomena. For instance, Soraya claims that individuals have multiple social identities and belong to several groups (Soraya 1994). This means that possibly they also belong to several cultures. Hofstede (1993, p. 19ff.) distinguishes between groups and categories, for example profession, organisation, nationality and gender. In his view, a group consists of individuals in contact, whereas a category consists of individuals sharing a characteristic feature, who do not necessary need to be in contact. Both groups and categories may constitute cultures.

- Method

There are two idealised approaches for researching culture. The *emic* approach analyses individuals' descriptions and assumes that there is no objective reality separate from its subjective reconstruction. The *etic* approach assumes objectivity is outside of and independent from individuals. In the *etic* approach, pre-existing hypotheses are tested, whereas in the *emic* approach,

categories are generated from the material. That means in the *etic* approach proving a restricted hypothesis depends on a pre-existing structure while in the *emic* approach the structure in the data needs to be interpreted and worked out more or less from first principles. Studies on the *emic* approach thus often stop with a differentiated description of the analysed part of the real world, while the *etic* approach results in more or less restricted models. (Gudykunst/Nishida, p. 20)

The distinctions discussed above are adequate enough to develop an initial idea of what 'culture' can mean. The distinctions do not categorise all approaches completely, but are sufficient for our purpose. In spite of these different approaches, most publications agree on at least one point: There is no generally applicable definition of culture. The meaning of the term has to be redefined each time it is used and the implicit assumptions have to be made explicit to determine the range of the analysis (e.g., Jahoda 1996, Demorgon/Moltz 1996).

2.2 Assumptions

We will analyse specific organisations in certain countries, and a defined set of individuals. So our interpretations will be adequate for a specific, precisely described context. Our interpretations are influenced by personal experiences and our socialisation in East respectively West Germany. Thus, we assume they can never be absolutely objective. The validity of a generalisation of our research to other contexts, however, remains to be proved.

As Wicker (1997, p. 33f.) points out, fixed definitions of cultural characteristics are often used to create ethnic boundaries through the attribution of particular characteristics to the self and to members of other groups. This means that fixed characteristics can be used to stabilise ethnic divisions. In addition, culture is often used as an ideological category, to promote and strengthen specific political collectives. In spite of this criticism, we consider the concept of culture to be helpful for our research on software development. But to avoid the problems mentioned above, we want to reflect on our assumptions about culture and develop an approach which can be helpful for our research.

- For our study, we will use the term 'cultures' in the plural, in order to differentiate its multiple dimensions.

Cultures exist in several dimensions. Individuals have multiple social identities (see Soraya 1994), and they belong to several groups and categories (see Hofstede 1993). No matter which level of description we might choose, we can always find finer differentiations. We therefore speak of 'cultures' in plural, to show that we are trying to take this iteration into account.

- We consider cultures as something in continuous change.

Change is an important element of our society. It can be assumed that changing conditions lead to different or new strategies for living, whether people see themselves as creating these conditions or reacting to them. In such a situation cultures cannot be assumed to be static, they also change continuously along with everything else.

- We assume cultures to be something which individuals have in common.

‚Cultures‘ are the frames¹ of interaction with other individuals. These frames can be found in every dimension of cultures. Each frame is bound to the subject’s changing expressions, which result in change of the frames.

- We understand cultures as being created, changed and made visible by interaction.

Individuals do not exist on their own. They are always in some relation to other individuals: they are members of groups, organisations, or societies. Connections between individuals and abstract entities are produced through interaction. In this process, subjective expressions and objective structure constitute and change each other (see Wicker 1997, p. 39) and become visible. In this context we interpret individuals’ interactions as being in correspondence or contrast to other expressions.

These four assumptions specify what we are looking for and the way we need to analyse and describe cultures. On the basis of individuals’ interactions with others, with artefacts, or with organisations, we look for structural frames and the individual activities that are intertwined with these frames. Because of the ill-defined and unstable nature of the objects of our research we propose the following procedure: First, we will try to identify and categorise these frames and activities and their interrelations within a specific and explicitly described context. Then we focus on the transformations which these entities undergo. However, before conducting an empirical analysis, we need to examine the results of previous research.

3 Frames and Activities

Our overall goal of a description of cultural influences in software engineering leads us to distinguish individuals’ activities and the structural frames in which these take place. Both ‚activities‘ and ‚frames‘ seem to be clearly defined but not necessarily fixed, because changes evolve through the individuals’ activities with or against these frames. As the frames influence the individuals’ activities, the individuals’ activities also create new frames or change existing ones. To identify the rules of transformation of these activities and frames we need first to identify both elements and then identify how activities and frames change each other. In order to carry out our investigation we need to restrict our attention to a few of the frames and individuals that may be relevant to software development.

In the following we introduce research which discusses several theoretical models of the software development process, which we consider to be an important frame of software development. For the individuals involved in this process we concentrate on software developers and present empirical research on their activities as individuals, in groups, in organisations or in specific societies.

¹ The term ‚frame‘ refers to Goffman’s concept of interaction as discussed in Weiss 1993.

3.1 Frames in the Software Development Process

Relevant frames in the software development process can be found in the different models and their historical, organisational and societal background. Bremer delivers a summary description of theoretical process models. He presents a genealogical tree showing the genesis of several models and their relationships:

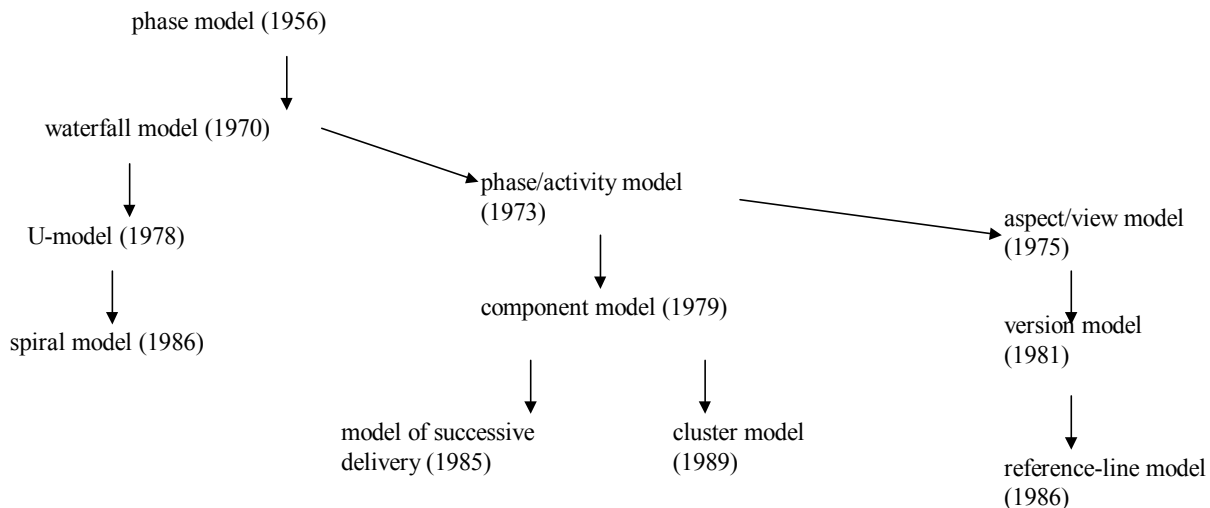


Fig. 1 Genealogical tree of software process models (Bremer 1998, p. 37)

The tree consists of three branches: a *left-hand* one, from the phase model down to the spiral model; a *right-hand* one, from the aspect/view model to the reference-line model; a *middle* branch containing the other models. For our purposes it is sufficient to have a general idea of how these models structure the development process (for further details see Bremer 1998).

The left-hand branch starts with the phase-model. It was developed by Benington in 1956 to support the production of large computer programs. The idea of structuring software development in this way came from management experiences in other industrial projects, such as construction, which use phase models for planning. Royce's waterfall model was oriented in the same way. It became very well known through Boehm's publication in 1976 and became a de facto standard in the field. In these models developing software is seen as the production of goods. The activities involved are grouped in phases: task analysis, specification, design, implementation, test, and maintenance, which are run through one after another. The intention of these models is to structure the whole process and to make results more reliable and controllable for others. (Bremer 1998, p. 39ff.)

The middle branch starts with the phase/activity model of Metzger (published in 1973), which evolved out of the waterfall model. It indicates a new understanding of software development as a process of development activities. Development activities, such as analysing, designing, programming and testing, were related to the phases defined in the waterfall model. The phases built a frame for reporting to the management only, while the project itself was conducted through defined activities (Bremer 1998, p. 44ff.). In addition to Bremer, we also consider the

evolutionary object-oriented model, designed by Bittner/Hesse/Schnath (1995), to be part of this middle branch. These models focus clearly on the developers' activities and structure the process mainly from this perspective.

The right-hand branch contains models which accentuate the different perspectives brought into the process by the participants, such as managers, developers and users. The first model which takes this approach is derived from the phase/activity model of Metzger. It was introduced in 1981 by Kerola and Freeman as the aspect/view model. The evolution of the process itself and the learning process for the different participants are central to the models on this branch. (Bremer 1998, p.49ff.)

On the basis of this genealogical tree, Bremer identifies two extremes in the history of process models. The models presented in the *left-hand branch* show the dominance of the developers. In this context users are seen as „trouble-makers“ and their only task is to deliver information. In the models on the *right-hand branch* the users are dominant. In this context developers are seen as partners in co-operation who deliver solutions to specific user problems.

In our opinion, the different branches also show structural differences. The models in the *left-hand branch* divide the development process into defined and controllable phases. Therefore, they help to structure the development process in a way that is appropriate for the management. The *middle branch* emphasizes the specific and context-dependent development activities. With their focus on development activities, these models present the process from the standpoint of the developers. The models in the *right-hand branch* focus on the process definition from the point of view of the participating groups who define, control and change the development process. These models present an application domain view of the software development process.

In dealing with variations of cultures in software engineering, two points are important in relation to Bremer's genealogy. First of all, these models do not only describe the software development process. They also build a frame that serves as a normative orientation for individuals. The models are used as formal rules for the process (see Scheffe 1999, p. 134). The best example of this phenomenon is the use of Royce's waterfall model. Bremer (1998), Grudin (1991) and Wertz/Ortmann (1992) all recognise the waterfall model's formal rules as a normative standard for the software development process. In practice, however, developers behave otherwise (see Strübing 1993, Wertz/Ortmann 1992).

Additionally, the different branches of Bremer's genealogy relate closely to the perspectives of specific groups of individuals participating in the software development process (management: *left-hand branch*, developers: *middle branch*, users: *right-hand branch*). Each group has its own preferred view of the process, an observation which supports the assumption that not only do frames affect individuals, but also individuals create and change frames, and in this case, models.

The perspective on software development is influenced by professional socialisation, and by organisational and societal conditions. In his historical analysis of the definition of computer science in West Germany, Coy (1997, p. 24) describes some important characteristics of the professional socialisation of developers. Mathematical and logical understanding of the discipline was dominant in the constituting process of computer science in Germany. Industrial demands

that might have required consideration of the application domain were disregarded. The reason for this omission lies in the scientific background of the discipline's founders, who were mathematicians and electronic engineers. In addition, the establishment of computer science was driven by university politics.

The dominant natural science and engineering background of computer scientists in West Germany and the USA had substantial consequences for the understanding of the discipline as well as for the perspective of the software development process. Software development was seen as the production of goods, which could best be structured into phases. This understanding in turn influenced the preferred programming paradigms and tools for the support of software development activities.

Whereas Coy's analysis focuses on the professional formation of computer scientists, Grudin (1991) develops further insights into organisational and societal conditions, mainly in the USA. He differentiates *in-house development*, *contract development*, and *product development* with respect to the point at which developers and users first make contact with one another during a project.

In-house development was the first context for software development. Originally, the developer and the user was one and the same person, or at least they worked in the same organisation. A similar situation occurred when a user organisation hired developers for specific projects. In-house development has been significant for Scandinavia, where both organisational conditions and political forces have contributed to its emergence. In the USA, this kind of development became obsolete because most of the development business was acquired by big computer system producers. In other countries the use of in-house development has mostly been limited to research projects. This kind of development has contributed to methods in process organisation and user participation to software engineering. The models on the *right-hand branch* of Bremer's genealogy have been applied in this context.

Contract development emerged first in the USA. The government had already started large-scale commercial software development projects in the 1960's. Beside producing software these projects also focused on exploring software development methods. Contract development plays a role in large user organisations that conduct long-term and large-scale projects within a restricted time-frame and budget. Phase models as described in Bremer's *left-hand branch* have been used to structure the process, to form the basis for the contract, to support the control activities of the management, and to assure quality products to the users. This kind of development has often been applied in public projects. The user organisation prepares a design specification and awards the contract to the developer organisation which makes the most suitable bid.

Product development became more important for commercial software systems in the USA in the sixties and seventies, with falling hardware costs for computer systems. The difference from contract development was that there was no precise requirement specification at the beginning, because the customers and end users existed only in the developers' imagination. Because users came in very late in the process, developers had to learn about their requirements indirectly, through market research, trade fairs and consultants. They concentrated on functionality and the human-computer interface. Because of the focus on development, the process could be

structured more flexibly, according to the developers' requirements. Although Grudin does not describe in detail the process models used, we assume that the models presented in Bremer's *middle branch* could be applied in such a context.

Also Güttler (1990) shows in his study of information processing in former East Germany that societal conditions such as the political system can have effects on the organisational conditions in the software development process and on the software product. The design of software for official or public use required governmental permissions because these departments were in charge of information processing. But developers preferred to deal with these departments as little as possible. Therefore they designed systems only for local use in one enterprise or even in one department. This resulted in very specific, local software solutions with limited value to other departments or enterprises. In addition, there was little co-operation between developers in different departments or enterprises.

Grudin's description of these three contexts and Güttler's study illuminate some of the organisational and societal conditions that allowed specific software development settings to emerge. We can now distinguish more precisely four frames relevant to the software development process:

1. theoretical descriptions, such as the process models
2. the situation inside the academic discipline
3. organisational settings, such as the size of the enterprise, or financing;
4. societal conditions, such as the political system.

Individuals are connected to all four frames: they work within these frames, their perspective is based on these frames, and they adopt routines imposed by the frames. These four frames will be part of our description of cultures in software development.

3.2 Developers' Activities in Practice

Although several other groups are involved in the software development process, we concentrate, as stated above, on developers. In this section of the paper we discuss previous research on their activities. In this research field, there are numerous studies on specific procedures in software development, known as „Empirical Studies of Programmers“ (see Gray/Boehm-Davis 1996 and volumes of these series, and Sonntag 1996 for a detailed analysis of this research field). Our interest, however, is to give a broader description of the work of software developers, in order to illuminate the range of individual activities and their relation to the frames mentioned above.

Schachtner (1993, p. 55ff.), for instance, considers the initial design activities of software developers in her empirical analysis of the relationship between developers and their computers. She describes in detail the procedures they perform at the beginning of a development process. In this first phase the developers she interviewed tried to understand the process which had to be programmed and the potential end users. She identified an interplay between the following activities: on the one hand, listening, watching, and asking the users, on the other, partitioning,

formalising, and finding rules for the perspective they acquired. According to Schachtner, the developers construct from their own perspective a model of the reality and of the users.

Rammert (1994, p. 15) states that such user models demonstrate the developers' beliefs about the behaviour and the knowledge of the users. He identifies different models in the USA and Germany: American technicians imagine an inexperienced person with only general knowledge and thus consequently they automate many sequences. German engineers count on someone who thinks rationally, works systematically and is accustomed to constant learning. Thus technicians usually provide a variety of options which make demands on users.

Funken (1994, p. 80ff.) describes more precisely the context in which such user models arise. She states that researchers must assume a hierarchical relation between developers and users as long as activities in the software development process are thought to be ruled by rational logic and technical systems, and are judged as unbiased. The communication between developers and users is affected by prejudices, experience and motivations. These orientations unconsciously control developers' understanding of the situation and influence their perception of the users. According to Rammert (1994) and Funken (1994), there are probably cultural influences on the developers' activities from the very beginning of the development process, depending on professional socialisation and nationality.

In the early 1990's, another analysis of programming work was conducted by Strübing (1993), who analysed developers' work from an individuals' perspective. Based on 21 interviews with West German developers the study describes developers' individual working style. This style is based on different aspects of the developers' work: *work organisation*, *use of tools*, *presentation of interim results* and *testing*. In his opinion each aspect is influenced by the constraints of the working environment, individual characteristics and organisational necessities. These variables do not exactly correspond to the frames mentioned in the previous section, but Strübing's description illuminates the typical practices of individuals and their relation to some of these frames.

- Work organisation

Strübing, like Bittner/Hesse/Schnath (1995), emphasises developers' cognitive orientation on the phase model. But Strübing finds that in practice, developers tend to mix the different phases of the development process. This mixing is enforced by the constraints of the work situation, because complicated tasks and external requests perturb the order of the phases. Professional experience, as well as a need for motivation or change, create individual characteristics which encourage (or discourage) such phase mixture. Already explicitly specified in the organisation as well as in the degree of division of labour, the process model presents organisational necessities that affect developers' individual activities (Strübing 1993, p. 119ff.).

Strübing (1993, p. 102) states that this phase mixture contradicts the findings of Kraft (1977), who found in an older study conducted in the US a stricter division of labour. Strübing explains his differing findings through changes in programming methods. Cross-cultural studies, however, suggest that the division of labour might also be influenced by national characteristics. An empirical study by Schroll-Machl (1996) concludes that there are differences in the division of labour between Germany and the USA. She describes German workgroups as preferring to work

on a problem in its entirety, while American groups work in an output-oriented way on the separated components of a problem. Taking Güttler's (1994) results seriously, software development in former East Germany was less co-operative than for example in West Germany because of the political interests in this industry. These findings show that the organisation of individual work in software development may also depend on societal or political factors.

- Use of tools

There is a large gap between the theoretically available tools and those tools actually used in practice. In computer science, the use of tools is rated highly, but empirical surveys show that their use in practice is very limited. For instance Deifel et al. (1999) analysed tool use in 13 organisations of different sizes in Germany during 1998. They questioned 20 developers and found no continuous application of software engineering tools, with the exception of programming environments. Although other tools were available, the developers used only graphical user interface builders, and occasionally tools for graphical language descriptions, in addition to the programming environments.

Nomura (1987) found a similar situation in Japan. He analysed the use of software engineering tools on the basis of statistics from the Japan Information Services Industry Association. He concluded that the available tools were not very usable in practice, that there were only a few suitable tools for the requirement and design phases, which were seldom used, and that programming and testing tools were often used but in not an up-to-date version. Nomura (1987, p. 266ff.) gave three reasons for this: First, programming is often conducted on the target platform, on which the tools do not work adequately. Secondly, there is a lack of adequate methodological education and training of developers. Thirdly, developers are more accustomed to the traditional 'paper and pencil' approach.

These findings show that tool use is concentrated in specific tasks in the development process. There is not much variety and most of the existing tools do not find their way into development practice. The authors cited above look for reasons in the programming situation, or in individual or organisational conditions in general. Strübing's findings (1993, p. 170ff.) pinpoint some factors influencing poor use of tools in practice. He finds that the variables influencing the use of tools are the time and money expense of tool use in respect to the budget of the project, as well as the range of tools actually installed. The developers are individually motivated to use them if their use is voluntary, if they have been educated in an environment where tool use is encouraged, and if their programming goals are maintainability and re-usability. Individually motivated voluntary use of tools contrasts with organisational conditions that support tool use, such as easy acquisition of tools or explicit instructions about tool use which are followed up at an organisational level.

Strübing (1993) also describes individual characteristics and practices, organisational needs and requirements of the work process for two other aspects of programming work: presentation of interim results and testing.

- Presentation of interim results

Developers draw diagrams and sketch designs when performing complex, difficult tasks with high quality requirements or when they need to talk to other developers about their work. How often they make such sketches also depends on their individual preferences for making such requirements or ideas visible for themselves or others and on their wish to preserve their own ideas in case of an unexpected disturbance. Organisational necessities arise from explicit documentation requests and from division of labour. (Strübing 1993, p. 134ff.)

- Testing

Strübing identified several different methods of testing. The specific needs of the application domain or innovative tasks required more intensive testing. The structure of the programming language and the availability of tool support influenced the ease of testing. Individual preferences for higher or lower risk lead to more or less systematic testing methods. Additional individual characteristics included the need to avoid uncertainty, experience and the ability for abstraction. On an organisation level, quality guidelines, time planning and the tools provided all influenced testing. (Strübing 1993, p.153ff.)

Strübing's study shows that there is a clear inconsistency between what developers think should be done and what they really do. Developers refer to the frames named in the previous section: the process models demanded in the organisation or learned at the university, organisational needs or market structures, which depend on societal conditions. Developers manifest an individual programming style which is either in harmony with or in contrast to these frames.

Strübing (1993, p. 189) also identified something what he named ‚collective habitus‘ and ‚shop-floor culture‘. The term ‚collective habitus‘ refers to the professional identity of developers as a group. They rather wanted to be identified as ‚software developers‘ and not as ‚programmers‘, whose work was seen as less creative and abstract. Their ‚shop-floor culture‘ also included a disapproval of careerism and a disrespect for authority. Expert knowledge and solidarity was rated highly within the developers' group. The ‚shop-floor culture‘ implied a collective working style, developed in interaction with the other members of the work group. This included programming styles, programming goals and the programming paradigms used. The ‚shop-floor‘ culture also determined which tools were be used and how programs were tested and documented. The individual working styles, arising from the developers' professional socialisation, was homogenised into a collective one in a continuous process, without individual differences entirely disappearing.

The studies mentioned above illuminate developers' individual and workgroup practices and their relationship to different frames. The unfolding relationships constitute what we call professional or organisational cultures. They are influenced by individuals, by groups and by organisational factors. Societal or national characteristics can also play a role.

These descriptions, however, are subject to change. Newly hired developers in a workgroup can force discussion of and changes to individual or collective working styles. Organisational changes through the restructuring of enterprises contribute to change mechanisms in frames and practices. Also, the increasing quest for interactive systems forces changes in the organisational

and societal contexts that have been described above (Grudin 1991). In the empirical studies already cited we find indicators of some of these changes. While nearly 80% of the software development projects analysed by Wertz/Ortmann (1992, p. 182) in 1989 were new developments, Deifel's (1999, p. 27f.) database from 1998 shows the converse: 14 out of 17 projects were customisations.

The following section summarises the findings of this section and suggests how variations of cultures and the changes in them can be described in the software engineering context.

3.3 Frames and Activities

In the studies of software engineering cited above we identified four frames: the process models as theoretical concepts, professional socialisation as an influencing background, organisational settings and societal settings, as working frames for individuals and organisations. In addition, the influence of national cultures may play a role. From empirical studies we have identified several activities in connection to software development which can be traced back to these frames.

Developers have been educated in a specific context within their discipline where priority is given to specific programming paradigms, tools, and software development process models. Developers' viewpoints are formed by this context and other experiences of technology acquired during their professional socialisation. At the same time, they are continually changing these frames by creating new technological artefacts and changing their activities by learning new programming paradigms and tools. Developers generate their own programming style, expressed in the priority they give to specific programming goals, and particular testing practices. Through their work activities developers construct individual and collective frames of work organisation. They construct a model of the user and of the development process through interaction with their colleagues.

Variations of cultures can be identified for individuals, for groups, and for the institutional environment in which developers were educated. Organisations contain entities which can be described as cultures. Regions, countries or geographically closed areas can contain several cultures.

Because Rammert (1994) takes account of cultural factors, his theoretical concept of technological genesis can serve as an orientation for what happens in software development. He regards technology development as an open process. On the micro-level, this process is influenced by cultural orientations and social constellations. On the macro-level, the process of technology development is framed by social institutions and selective structures. Through existing technology, individuals adapt in advance to the selective structures. This leads to a circular process: new technological development arises out of already existing technology. A similar circular interaction can take place between software engineering frames and individuals' activities.

Up to now our knowledge of the influence of the variations of cultures in software engineering has been drawn from theoretical analyses. It has become clear that research in software

engineering is only starting to relate theoretical concepts to the activities of individuals or workgroups in every day life. Empirical research findings in the social sciences concentrate more on individuals' activities but they make only implicit connections to software engineering models. The individual perspective, however, is important in describing cultures in software engineering and the changes these cultures undergo. In the last section we will discuss how we might learn more about individual perspectives.

4 Conclusion

The theoretical framework we have introduced is the basis for our empirical work. We attempted to define more precisely how variations of cultures can be described, how they influence software engineering and how they can be seen in relation to technological developments in the discipline. Our findings that ongoing technological changes both influence cultures and are influenced by them are also acknowledged in cultural studies (see Wicker 1997). On the basis of our assumptions we view cultures as something created, changed and made visible through individuals' activities and their interactions in workgroups, organisations and society. These activities and interactions are connected to the activities of other individuals and to structural frames, such as process models, professional socialisation, organisational conditions and societal conditions.

One way to clarify the interrelations between activities and frames and to describe these on-going changes is through conducting empirical studies and interpreting the data from the individuals' perspective. The perimeters we placed on our cultural approach allow a description of cultures only in those specific situations that we find in our data. Specific situations and contexts lead to specific practices and vice versa.

This determines our methodology. To explore variations of cultures in software engineering we are required to start on the individual level and ask the developers themselves how they do their work. Developers' responses give us indications of what practices, orientations and frames are relevant and how the developers create or change them through interaction. For our purposes, qualitative methods embedded in case studies seem to be the most suitable. We will concentrate on a specific organisational structure, small and medium-sized software development enterprises. These enterprises form the basis of software development in both West (Strübing 1993, Konrad/Paul 1999) and East Germany (Schwarz 2000). Bearing in mind that we are West Europeans, a cross-cultural analysis will be helpful to avoid a narrow national view. We therefore intend to examine enterprises not only in Germany, but also in the USA and Russia. This approach will allow us to say more about cultures in software engineering.

Cultures can impede or promote development activities. For software development, it is therefore necessary to know more about cultures, and how cultures are played out in the everyday practice of development. This insight is essential if we wish to make cultural differences work in favour of the development process.

Acknowledgements

This research is supported by a DFG (German Research Association) grant No. Bo 929/13-1. Special thanks to Naomi Hallan, Jörg Wellner and Dorian Woods. Their contributions are gratefully acknowledged.

REFERENCES

- Bhabha, Homi K.: The location of culture. London ; New York : Routledge 1994.
- Bittner, Udo; Hesse, Wolfgang; Schnath, Jürgen: Praxis der Softwareentwicklung: Methoden, Werkzeuge, Projektmanagement - eine Bestandsaufnahme. München ; Wien : Oldenbourg, 1995.
- Bremer, Georg: Genealogie von Entwicklungsschemata. In: Kneuper, Ralf; Müller-Luschnat, Günther; Oberweis, Andreas (Hrsg.): Vorgehensmodelle für die betriebliche Anwendungsentwicklung. Stuttgart ; Leipzig : Teubner 1998, S. 32-59
- Bolten, Jürgen: Ökonomische Transformation als interkulturelle Kommunikation: Eine Kritik ökonomischer Transformationsprozesse aus kommunikationstheoretischer Perspektive. In: Bolten, Jürgen; Dathe, Marion (Hrsg.): Aktuelle Probleme und Perspektiven west-/osteuropäischer Wirtschaftsbeziehungen. Berlin : Sternenfels, 1995, S. 46-59
- Coy, Wolfgang: Defining discipline. In: Freksa, Christian; Jantzen, Matthias; Valk, Rüdiger (Eds.): Foundations of computer science: Potential-Theory-Cognition. Berlin et al. : Springer 1997, pp. 21-35
- Deifel, Bernhard et al.: Die Praxis der Softwareentwicklung: Eine Erhebung. Informatik Spektrum Jg. 22, H. 1, 1999, S. 24-36
- Demorgon, Jacques; Moltz, Markus: Bedingungen und Auswirkungen der Analyse von Kultur(en) und interkulturellen Interaktionen. In Thomas, Alexander (Hrsg.): Psychologie des interkulturellen Handelns. Göttingen et al. : Hogrefe, 1996, S. 43-86
- Funken, Christiane: Das Bild des Entwicklers vom Benutzer: eine Problemskizze. In: Fricke, Else (Hrsg.): Zur Zukunftsorientierung von Ingenieuren und Naturwissenschaftlern. Bonn : Friedrich-Ebert-Stiftung, 1994, S. 75-92
- Gray, Wayne D.; Boehm-Davis, Deborah A. (Eds): Empirical studies of programmers: Sixth Workshop. Proceedings. Norwood (NJ) : Ablex Publishing, 1996.
- Grudin, Jonathan: The development of interactive systems: bridging the gaps between developers and users. In : IEEE Computer Vol. 24, No. 4, 1991, pp. 59-69
- Gudykunst, William B.; Nishida, Tsukasa: Theoretical perspectives for studying intercultural communication. In : Asante, Molefi Kete; Gudykunst, William B. (Eds.): Handbook of international and intercultural communication. Newbury Park et al. : Sage, 1989, pp. 17-46
- Güttler, Markus: Die Datenverarbeitung im statistischen Informationssystem der DDR: Ein Beitrag zur deutschen Vereinigung und zur Geschichte der Informatik. TU Berlin, 1990.
- Hofstede, Geert: Interkulturelle Zusammenarbeit: Kulturen-Organisationen-Management. Wiesbaden : Gabler, 1993.
- Konrad, Wilfried; Paul, Gerd: Innovation in der Softwareindustrie: Organisation und Entwicklungsarbeit. Frankfurt am Main ; New York : Campus, 1999.
- Kraft, Philip: Programmers and Managers: The Routinisation of Computer Programming in the United States. New York : Springer Verlag, 1977.
- Jahoda, Gustav: Ansichten über die Psychologie und die „Kultur“. In: Thomas, Alexander (Hrsg.): Psychologie des interkulturellen Handelns. Göttingen et al. : Hogrefe, 1996.

- Nomura, Toshitsuqu: Use of software engineering tools in Japan. In Proc. 9th ICSE (Monterey CA, April 1987), ACM Press, 1987, pp. 263-269
- Rammert, Werner: Prozesse der Technikgenese und der Stellenwert von Leitbildern in der Technikentwicklung. In: Fricke, Else (Hrsg.): Zur Zukunftsorientierung von Ingenieuren und Naturwissenschaftlern. Bonn : Friedrich-Ebert-Stiftung, 1994, S. 9-18
- Schachtner, Christel: Faszination und Provokation am Computer. Frankfurt/Main : Suhrkamp, 1993.
- Scheffe, Peter: Softwaretechnik und Erkenntnistheorie. In : Informatik-Spektrum Jg. 22, H. 2, 1999, S. 122-135
- Schneider, Susan; Barsoux, Jean-Louis: Managing across cultures. Hemel Hempstead ; Hertfordshire (UK): Prentice Hall Europe, 1997.
- Schroll-Machl, Sylvia: Kulturbedingte Unterschiede im Problemlösungsprozess bei deutsch-amerikanischen Arbeitsgruppen. In: Thomas, Alexander (Hrsg.): Psychologie des interkulturellen Handelns. Göttingen et al. : Hofgreffe, 1996, S. 383-409
- Schwarz, Anna: Diverging Patterns of Informalization between Endogenous and Exogenous Economic Actors in the East German Transformation Process: Results from a Case-Study in the Information-Technology Sector in Berlin-Brandenburg. In: Stöltung, Michael; Brie, Michael (Hrsg): The Transformation of the Informal. Chicago : University of Notre Dame, 2000. (in print)
- Sonntag, Sabine: Experten in der Software-Entwicklung. Untersuchung hervorragender Leistungen im Kontext intellektueller Teamarbeit. Universität Gießen, Habilitation, 1996.
- Soraya, Semira: Zum Kulturbegriff in der Multikultur. In: Thomas, Alexander (Hrsg.): Psychologie und multikulturelle Gesellschaft: Problemanalysen und Problemlösungen. Göttingen ; Stuttgart : Verlag für angewandte Psychologie, 1994, S. 15-22
- Strübing, Jörg: Subjektive Leistungen im Arbeitsprozess: Eine empirische Untersuchung von Arbeitsstilen in der Programmierarbeit. Wiesbaden : Deutscher Universitäts Verlag, 1993.
- Weiss, Hilde: Soziologische Theorien der Gegenwart: Darstellung der großen Paradigmen. Berlin ; Heidelberg ; New York et al. : Springer, 1993.
- Weltz, Friedrich; Ortmann, Rolf G.: Das Softwareprojekt: Projektmanagement in der Praxis. Frankfurt/Main ; New York: Campus, 1992.
- Wicker, Hans-Rudolf: From complex culture to cultural complexity. In: Werbner, Pnina; Modood, Tariq (Eds.): Debating cultural hybridity. London : Zed Books, 1997, pp. 29-45

Softwareentwicklung in St. Petersburg

Erste Eindrücke

Andrea Sieber

Abstract Software development in Russia has become important for the international software industry. But is software development in Russia the same as in Germany or in the USA? To get contact to Russian enterprises a pre-study was carried out for gaining first insights into St. Petersburg software industry. The dominance of international and midsize enterprises and self employed teams on the one hand and the low importance of small businesses on the other will not be understood without some insights into the former economic structures and the difficulties in changing the old socialistic system. Although the software industry produces some products of its own or adapts foreign software systems to the Russian market, businesses will more easily survive when fulfilling programming orders from outside the country. These „off shore development“ – orders stand for a relatively secure money flow. Although there are some differences between German and Russian developers in the understanding of deadlines, of programming itself, and of priorities for individual work.

The purpose of the pre-study was to test if an empirical research can be conducted in Russia and later related to the results of a similar empirical research in the USA and an already finished empirical research in Germany. That seems to be difficult, because of (1) the author's difficulties in understanding the everyday language, (2) the difficulties in getting secondary information on Russian software-market statistics and research concerning software development or regional economic networking in Russia, and (3) the relatively low importance of small software development businesses in Russia. A possible solution to these methodological and analyzing problems may be the investigation of an international enterprise with midsize agencies in St. Petersburg, Germany, and the USA.

1 Einleitung

Betriebliche Strukturen in der Softwarebranche in den USA und Europa sind einem zunehmenden organisatorischen Wandel unterworfen. Große Unternehmen gliedern Geschäftsbereiche aus und lassen sich in anderen Ländern nieder, um näher bei ihren (potentiellen) Kunden zu sein oder preisgünstiger zu entwickeln. In solchen Kontexten gehört die Entwicklung von Software in internationalen Teams bereits zum Alltag. Die Softwaretechnik, deren Ziel es ist, Softwareentwicklung durch den Einsatz geeigneter Methoden zu verbessern, kann diese Veränderungen nur nachträglich zur Kenntnis nehmen, da die Forschung im universitären Kontext stattfindet, der nur indirekt mit der Praxis verbunden ist. Das könnte auch einer der Gründe sein, weshalb es kaum theoretische Konzepte gibt, die derartige Veränderungen

verstehen helfen und darüber hinaus konstruktiv unterstützen. Um ein solches Konzept zu entwickeln, ist es aufgrund der fehlenden praktischen Erfahrungen an der Universität sinnvoll, zuerst die derzeitige Praxis von Softwareentwicklung auf individueller und organisatorischer Ebene in verschiedenen Ländern zu untersuchen. Auf diese Weise lässt sich der Einfluss kultureller Handlungsmuster auf diesen Ebenen näher bestimmen. In einem zweiten Schritt können dann Methoden entwickelt werden, die Softwareentwicklung in der Praxis bestmöglich unterstützen.

In dieser Weise geht auch das DFG-Projekt „Softwareentwicklung in der Praxis im Kulturvergleich“⁴¹ vor. Es will anhand empirischen Materials verdeutlichen, inwiefern kulturelle Handlungsmuster die Softwareentwicklung beeinflussen. Zuerst wird dabei Entwicklungsarbeit in kleinen Unternehmen in östlichen und westlichen Bundesländern in Deutschland untersucht, da ca. 60% der Unternehmen in der Softwarebranche in Deutschland kleine Firmen sind (vgl. Konrad/Paul 1999, Schwarz 2000). Danach ist eine ähnliche Analyse der Entwicklungsarbeit in Russland und den USA geplant. Das Ziel der Untersuchung ist es, das Vorgehen von EntwicklerInnen, Teams und Organisationen in den verschiedenen kulturellen Kontexten miteinander zu vergleichen.

Da der Zugang zu den Unternehmen in Russland aufgrund der fehlenden Kontakte für uns als Forscherinnen schwierig ist, nutzte ich einen von der Hans-Böckler-Stiftung organisierten und finanzierten Sprachaufenthalt in St. Petersburg im Sommer 1998, um einen Einblick in die Situation der Softwarebranche in Russland zu erhalten und Kontakte zu knüpfen. Mein allgemeines Wissen über Russland, seine wirtschaftliche und soziale Lage und das Leben in St. Petersburg verbesserte ich durch das Rahmenprogramm der Sprachschule (vgl. HBS 1996 und 1998), innerhalb dessen wir VertreterInnen von (Groß-) Unternehmen, Institutionen und Vereinen besuchten. Wie Alltag in St. Petersburg aussehen kann, lernte ich durch die private Unterbringung während der Sprachschule aus nächster Nähe kennen. Aber auch Gespräche mit St. Petersburger Unternehmern auf der C-Bit in Hannover im März 1999 und Gespräche während der Erhebung des empirischen Materials in den westlichen Bundesländern im Jahre 1999 mit russischen Entwicklern und Managern bzw. deutschen Managern, die mit russischen Unternehmen kooperierten, erweiterten meinen Horizont.

In diesem Artikel stelle ich meine Erfahrungen mit der St. Petersburger Softwarebranche zusammenfassend dar. Sie basieren auf den eben beschriebenen Erlebnissen, Expertengesprächen mit zehn Personen, die in sechs russischen Softwareunternehmen als Geschäftsführer, Verkäufer oder EntwicklerInnen tätig sind sowie zwei Interviews mit Geschäftsführern westdeutscher Softwareunternehmen, die in russischen Unternehmen entwickeln lassen. Im zweiten Abschnitt stelle ich den Zusammenhang zwischen der Wirtschaftsgeschichte des Landes und der Wirtschaftsstruktur der Softwarebranche dar. Im dritten Abschnitt beschreibe ich die Softwarebranche in St. Petersburg anhand der Organisationsformen von Softwareunternehmen, anhand des Arbeitsalltages von

EntwicklerInnen und anhand von Projekttypen. Ausführlicher schildere ich dabei die Besonderheiten des „off shore development“ – die Bearbeitung von ausländischen Entwicklungsaufträgen in Russland. Die Vorstudie offenbarte zugleich, an welchen Punkten es methodisch schwierig wird, bei der empirischen Untersuchung in Russland ähnlich wie in Deutschland vorzugehen. Im letzten Abschnitt stelle ich diese Schwierigkeiten dar und skizziere Lösungsmöglichkeiten.

2 Auswirkungen der sozialistischen Wirtschaftsstruktur auf die heutige Situation von Unternehmen

Die heutige Wirtschaftslage in St. Petersburg ist ohne einen Blick zurück nicht zu verstehen. In der UdSSR gab es in der Hauptsache volkseigene Großbetriebe, die zentralstaatlich geleitet und geplant wurden. Kleine Unternehmen oder Selbständige waren ebenso wenig anzutreffen wie in der ehemaligen DDR. Es gab einen regen Handel unter den einzelnen Sowjetrepubliken, da nicht jede Republik alle Güter produzierte. Ein Warenaustausch fand auch innerhalb der sozialistischen Staatengemeinschaft statt. Von geringster Bedeutung bezüglich des Umfangs war der Handel mit dem kapitalistischen Ausland. Gleichzeitig waren große Unternehmen Orte des privaten und kulturellen Lebens. Meine Gastgeberin beispielsweise war mit ihrem Mann jahrelang gemeinsam in einem solchen Unternehmen tätig. Sie berichtete, dass dort das gesamte Familienleben stattfand: Die Kinder wurden betreut, mit Kollegen wurde Sport getrieben, es ging gemeinsam in die werkseigene Sauna (die es heute auch noch gibt) oder es wurden Ausflüge und Feste für die Beschäftigten und ihre Angehörigen organisiert.

Mit der Perestroika wurden die volkseigenen Großbetriebe privatisiert. Allerdings gab es unter den Angestellten wie prinzipiell auch innerhalb der Bevölkerung viel Unsicherheit, ob die vorgenommenen Veränderungen tatsächlich von langer Dauer sein würden. Zudem hatte kaum jemand Erfahrungen mit der kapitalistischen Wirtschaftsweise. Unklarheit, undurchsichtige Entscheidungen und fehlendes Vertrauen trugen dazu bei, dass es schwierig war, die politisch gewollten Veränderungen wirklich umzusetzen. Meine Gastgeberin berichtete beispielsweise, dass ihr Betrieb zweimal privatisiert wurde. Obwohl bereits Anteilsscheine für die Belegschaft ausgegeben waren, wurde der Betrieb noch einmal verkauft. Jeden Arbeitstag begann sie seitdem in Erwartung neuer, ihr selbst unverständlicher Entscheidungen im Betrieb.

Gleichzeitig kamen die Handelsbeziehungen zwischen den ehemaligen Sowjetrepubliken und innerhalb der sich ebenso verändernden sozialistischen Staatengemeinschaft fast völlig zum Erliegen. Die privatisierten Großbetriebe bekamen keine Aufträge, denn so schnell ließ sich der Handel mit dem Ausland nicht vergrößern. Das Unternehmen meiner Gastgeberin reagierte auf diese Situation damit, dass es die Produktion über längere Zeiträume stilllegte. In diesen Phasen wurden die Produktionsarbeiter ohne Lohn nach Hause geschickt, um so die laufenden Kosten zu senken. Gab es erneut Aufträge, wurden sie weiterbeschäftigt und erhielten auch wieder ihren

¹ Die DFG fördert das Vorhaben im Rahmen der Chemnitzer Forschergruppe „Neue Medien im Alltag: Von individueller Nutzung zu soziokulturellem Wandel“ (Bo 929/13-1).

Lohn. Die Büroangestellten arbeiteten während der ganzen Zeit, um die Geschäfte des Unternehmens abzuwickeln. Sie bekamen ihren Lohn mit mehr oder weniger Verzögerungen.

Die komplizierte politische Situation an der Spitze des Landes, die Machtkämpfe zwischen Reformern und Kommunisten in der Regierung, die Unabhängigkeitsbestrebungen einzelner Sowjetrepubliken und die damit verbundenen politischen Turbulenzen verhinderten zudem eine schnelle Veränderung der Gesetzgebung, die das tägliche Leben und den Wandel des politischen und wirtschaftlichen Systems unterstützt hätten. Das betraf vor allem die Schaffung einer den Veränderungen angepassten und nachvollziehbaren Steuergesetzgebung und Gesetze zur Förderung und Sicherung der Existenz kleiner Unternehmen. In beiden Bereichen gab es große Handlungsspielräume, die von Behörden oder mafia-ähnlichen Vereinigungen zur Beseitigung unliebsamer Konkurrenten genutzt werden konnten. Ohne die schon zu Sowjetzeiten lebensnotwendigen „guten Beziehungen“, die auch mit finanziellen Zuwendungen verknüpft waren, schien es in Russland nach wie vor sehr schwierig, ein privates Unternehmen erfolgreich zu führen.

Der desolate wirtschaftliche Zustand im Land und die Schwäche der reformerischen Kräfte führte Ende des Sommers 1998 zu einer Abwertung des Rubel, der eine Verknappung aller Waren aus dem Ausland zur Folge hatte. Davon war auch die Softwarebranche betroffen, die für ihre Arbeit sowohl Hard- als auch Software aus dem Ausland benötigt. Wer in dieser Situation „gute Beziehungen“ und Dollar hatte, war gut beraten.

3 Die Softwarebranche in St. Petersburg

Wie der vorige Abschnitt zeigt, beförderte die wirtschaftliche und politische Lage in St. Petersburg, aber auch in Russland allgemein, die Weiterführung und Gründung von Unternehmen nur sehr begrenzt. Im Vergleich zu anderen Branchen (wie bspw. dem Maschinenbau), profitierte die Softwarebranche jedoch von der Internationalität von Rechnerstrukturen und Programmiersprachen. Dadurch konnte sie vergleichsweise unkompliziert Aufträge aus dem Ausland akquirieren. Auch die mathematisch fundierte Ausbildung trug dazu bei, dass russische Entwickler international konkurrieren konnten.

Die Softwarebranche in St. Petersburg lässt sich anhand der Organisationsformen, des Arbeitsalltags der EntwicklerInnen und der Projekttypen charakterisieren. Deren spezielle Eigenheiten sind aus der politischen und wirtschaftlichen Vergangenheit und Gegenwart verständlich. Eine besondere Bedeutung bekam in der aktuellen Situation die Ausführung ausländischer Entwicklungsaufträge, dem sogenannten „off shore development“. Die Besonderheiten dieser Entwicklungsart sind am Ende dieses Abschnittes ausführlicher beschrieben.

3.1 Organisationsformen

Die Softwareentwicklungslandschaft in St. Petersburg ist von den Organisationsstrukturen her gesehen auf den ersten Blick genauso vielfältig wie in Deutschland. Allerdings ist die Situation, in

Westdeutschland haben kleine Unternehmen bereits eine lange Tradition und in den östlichen Bundesländern wurden eigens Förderprogramme aufgelegt, um die Gründung von technologieorientierten kleinen Unternehmen zu fördern. In Russland ist die Gesetzeslage wesentlich unklarer. Das Land ist mehr mit seiner politischen Stabilisierung beschäftigt als mit einer Klärung der rechtlichen Grundlagen für private Wirtschaftsunternehmen, geschweige denn deren Förderung. Das bleibt nicht ohne Folgen für die Situation der Softwarebranche. *Repräsentanzen* ausländischer Unternehmen, *mittlere Unternehmen* mit einem russlandweiten Filialnetz oder *selbständige EntwicklerInnen bzw. Entwicklungsteams* („wild teams“) scheinen in dieser Situation die besten ökonomischen Überlebenschancen zu haben. Neben diesen verschiedenen organisatorischen Zusammenhängen, in denen Software entwickelt wurde, stellten auch florierende *Großunternehmen*, wie beispielsweise Banken oder staatliche Versorgungsbetriebe, EntwicklerInnen an, welche die Technik im Unternehmen pflegen, anpassen oder weiterentwickeln.² *Kleine Unternehmen* leiden demgegenüber besonders an den weggebrochenen Kontakten, der unklaren Gesetzeslage, der Abwanderung der Entwickler und dem Geldmangel im Land und haben deshalb keine guten Überlebenschancen.

Repräsentanzen hatten in der Stadt fast alle namhaften amerikanischen Softwarekonzerne wie beispielsweise HP, Sun Microsystems, IBM. Der wichtigste Geschäftsbereich für derartige Unternehmen war das „off shore development“. Sie hatten den Vorteil, dass sie konzernintern Aufträge aus dem Ausland, d.h. den anderen Filialen des Konzerns, bekamen. Gleichzeitig konnten diese Repräsentanzen die in den anderen Konzernteilen hergestellten Systeme nutzen und für den russischen Markt anpassen, wenn sie einen entsprechenden Kunden gefunden hatten. Sie mussten also nicht erst ein eigenes Produkt herstellen, sondern konnten sofort mit der Akquisition von zahlungskräftigen Kunden beginnen. Dazu pflegten sie insbesondere Kontakte zu großen, evtl. noch staatseigenen Unternehmen, weil dort die Wahrscheinlichkeit am größten war, dass Aufträge von entsprechendem finanziellen Umfang, wie es im Mutterunternehmen üblich war, zustande kamen.

Mittlere Unternehmen konzentrierten sich entweder auf den russischen Markt oder auf das „off shore development“. Ein Unternehmen, das ich in St. Petersburg kennen lernte, hatte nach der Perestroika mit der Entwicklung eines Warenwirtschaftssystems begonnen. Der Unternehmensbereich, in dem die Produktentwicklungen stattfanden, befand sich in Moskau. In anderen russischen Städten, so auch in St. Petersburg, gab es Filialen, die dieses Produkt in der jeweiligen Region vertrieben und an die Bedürfnisse der Kunden vor Ort anpassten. Ein Drittel der Beschäftigten dieses Unternehmens waren Entwickler. Die Filiale in St. Petersburg finanzierte sich nicht ausschließlich über den Vertrieb und die Anpassung dieses Produktes, sondern dort wurde zusätzlich ausländische Software verkauft. Die Auftragslage schwankte jahreszeitlich. Während im Sommer kaum Aufträge zu bearbeiten waren, nahmen die Verkäufe zum Jahreswechsel hin zu. Der Erfolg des Unternehmens beruhte darauf, dass es ein breit einsetzbares

² Sie stehen in unsere Untersuchung jedoch nicht im Mittelpunkt des Interesses, deshalb sollen sie hier nicht näher betrachtet werden.

russisches Produkt an die Bedürfnisse der Kunden anpasste, und dass das Unternehmen es geschafft hatte, innerrussische Handelsbeziehungen aufzubauen bzw. zu erhalten.

Zwei andere mittlere russische Unternehmen, zu denen ich von Deutschland aus den Kontakt herstellte, finanzierten sich nahezu ausschließlich über die Bearbeitung von Entwicklungsaufträgen aus dem Ausland. Die Geschäftsführer waren in beiden Unternehmen längere Zeit in Deutschland gewesen, hatten dort studiert und gearbeitet. Die dabei entstandenen Kontakte bildeten den Ausgangspunkt für die Entwicklungsaktivitäten. Sie pflegten diese Kontakte regelmäßig und besuchten ihre Geschäftspartner in deren Unternehmen. Auf diese Art und Weise konnten Probleme angesprochen und geklärt werden, und es kamen neue Aufträge zustande.

Aus den technischen Abteilungen der Großbetriebe waren nach der Perestroika zum Teil *kleine Unternehmen* hervorgegangen, die ihre bis dahin bestehenden Kontakte nutzten, um auf eigenen Füßen zu stehen. Sie entwickelten für den Großbetrieb, aus dem sie hervorgegangen waren, und versuchten, darüber weitere Kundenkontakte zu bekommen. Ihr Kundenkreis war auf die Region beschränkt und kämpfte mit den gleichen wirtschaftlichen Problemen wie sie. Im Vergleich zu den Repräsentanzen ausländischer Unternehmen oder zu den mittleren Unternehmen konnten kleine Software-Firmen preiswerter und schneller entwickeln. Kleine Unternehmen konnten jedoch nicht so preiswert arbeiten wie selbständige EntwicklerInnen bzw. Entwicklungsteams, da sie auch Büroangestellte finanzieren mussten. Die fehlende Tradition kleiner Unternehmen in der Sowjetunion und damit einhergehend das geringe Vertrauen in ihre Überlebensfähigkeit und ein geringer Bekanntheitsgrad machten es ihnen schwierig, Aufträge zu bekommen. Aufträge mit großem Volumen wurden eher von den Repräsentanzen ausländischer Unternehmen oder den mittleren Unternehmen bearbeitet, Aufträge von geringerem Umfang von selbständigen EntwicklerInnen bzw. Entwicklungsteams. Gleichzeitig hatten kleine Unternehmen Schwierigkeiten, ihr Entwicklungspersonal zu halten. Qualifizierte EntwicklerInnen wanderten aus oder entwickelten auf eigene Faust. Hinzu kamen bei diesen Unternehmen die erwähnten Schwierigkeiten mit der Besteuerung.

Durch die undurchsichtige und sich auf die Unternehmen nachteilig auswirkende Steuergesetzgebung verbunden mit der schwierigen Wirtschaftslage im Land gab es einen großen Bereich an Entwicklungsaktivitäten, die von *selbständigen EntwicklerInnen bzw. Entwicklungsteams* (sogenannten „wild teams“, einem losen Zusammenschluss mehrerer selbständiger EntwicklerInnen) bearbeitet wurden. Auf diesem Weg umgingen sie die Schwierigkeiten, welche kleine Unternehmen hatten. Sie hatten zudem den Vorteil, dass sie sehr flexibel bezüglich der verwendeten Technologien waren und besonders schnell und billig arbeiten konnten, auch vor Ort beim Kunden. Flexibel bezüglich der verwendeten Technologien zu sein war eine wichtige Eigenschaft dieser EntwicklerInnen, da sich nicht jedes russische Unternehmen die Umstellung auf den amerikanischen Stand der Technik leisten konnte. Selbständige EntwicklerInnen kannten sich mit der alten russischen Technik aus oder waren zumindest bereit, damit zu arbeiten und dafür Software zu schreiben. Durch den fehlenden organisatorischen Überbau war es

ausreichend, wenn die Bezahlung einen individuell zufriedenstellenden Lebensunterhalt gewährleistete.

Die Software, die in diesem Zusammenhang entstand, war sehr gut an die Anwendungssituation angepasst. Die Kürze der Entwicklungszeit führte jedoch zu einer geringeren programmtechnischen Qualität dieser Software, was die Wiederverwendung des Quelltextes stark einschränkte. Schafften sich die Kunden neue Technik an, konnten sie die alten Programme nicht mehr verwenden. Allerdings spielten solche Überlegungen bei den russischen Kunden kaum eine Rolle, da sie selbst nicht wussten, wie lange sie ihr Unternehmen am Leben erhalten konnten.

3.2 Arbeitsalltag der EntwicklerInnen

Bei den EntwicklerInnen stand die Sicherung eines relativen Lebensstandards mit ihrer Arbeit im Vordergrund ihrer Überlegungen. Auch in diesem Bereich waren gute Beziehungen nützlich, welche die Entwicklerin als Männerbünde darstellte.

Die Entwicklerin arbeitete in einem kleinen Unternehmen. Sie programmierte - verglichen mit dem amerikanischen Stand der Technik - hauptsächlich in alten Programmiersprachen. Es handelte sich dabei um ältere Versionen von Programmiersprachen, die auch heute noch verwendet werden. Sie arbeitete direkt mit Kunden zusammen, machte dort Schulungen oder half ihnen bei der Analyse ihres Problems. Vor ihrer Tätigkeit in diesem Unternehmen hatte sie bereits in drei anderen kleinen Unternehmen gearbeitet. Auch zum Zeitpunkt des Interviews war sie auf der Suche nach einer neuen Beschäftigung, da sie die Überlebenschancen des Unternehmens als sehr gering einschätzte: Zum Unternehmen gehörten sechs Personen, drei davon waren EntwicklerInnen. Ein Entwickler hatte das Land aufgrund der schlechten finanziellen Lage des Unternehmens bereits verlassen. Eine Besserung der Auftragslage war nicht in Sicht, da sich die wirtschaftliche Lage des Landes nach wie vor verschlechterte.

Die befragte Entwicklerin schilderte die Suche nach einem Arbeitsplatz in St. Petersburg als sehr schwierig. Um eine interessante und gut bezahlte Arbeit zu erhalten, bedurfte es guter Beziehungen. Da sie selbst nicht über solche verfügte, bat sie ihre männlichen Verwandten, nach Angeboten Ausschau zu halten. In ihren Augen waren gute Beziehungen Männerbünde, die das Leben in St. Petersburg erleichterten. Diese Männerbünde waren ihrer Meinung nach auch der Grund dafür, weshalb sie ihre Beschäftigungschancen am ehesten in kleinen Unternehmen sah, obwohl sie durch deren prekäre ökonomische Lage zu einem häufigen Arbeitsplatzwechsel gezwungen war. Die lukrativen Jobangebote würden zuerst an Entwickler gehen. Dieser Situation hätte sie durch Auswanderung entgehen können, worin sie jedoch für sich persönlich keine Alternative sah. Sie hatte kaum Auslandserfahrung und wollte ihr soziales Netz in Russland nicht verlieren.

Die Entwickler, mit denen ich sprach, arbeiteten in mittleren Unternehmen oder Repräsentanzen ausländischer Unternehmen. Sie waren auf Empfehlung bereits dort arbeitender Entwickler angestellt worden oder weil sie Personen aus dem Management des Unternehmens aus dem Studium oder der Armee kannten. Sie hatten sich in den Unternehmen auf bestimmte Aufgaben

spezialisiert. Da die Bezahlung vergleichsweise gut war und diese Unternehmen eine ökonomische Perspektive hatten, waren die Entwickler sehr zufrieden mit ihrem Arbeitsplatz.

Alle EntwicklerInnen arbeiteten zu Hause. Es fehlte in den Unternehmen der Platz, um jedem Beschäftigten einen eigenen Schreibtisch zur Verfügung zu stellen. Eine Ausnahme stellte die Repräsentanz eines ausländischen Unternehmens dar, die einen großen Rechnerraum zur Verfügung hatte. Alle berichteten, dass sie oft in den Abendstunden und am Wochenende arbeiteten. Im Unternehmen prüften sie lediglich die Ergebnisse ihrer zu Hause geleisteten Entwicklungsarbeit. Ihrer Meinung nach hatten sie zu Hause mehr Zeit und Ruhe, sich in neue Programme einzuarbeiten.

3.3 Projekttypen

Aus der Beschreibung der Organisationsformen von Unternehmen in St. Petersburg wird bereits deutlich, dass die Firmen Projekte unterschiedlichen Charakters bearbeiteten. Es lassen sich drei verschiedene Projekttypen unterscheiden, die je nach Auftraggeber die ökonomische Entwicklung des Unternehmens förderten oder hemmten.

Entwicklungsaufträge aus dem Ausland wurden vergleichsweise zuverlässig und gut bezahlt. Diese Art der Entwicklung wurde als „off shore development“ bezeichnet. Zudem wurden *Produkte von ausländischen Firmen für den russischen Markt angepasst*. Beschriftungen, Hilfetexte und die Dokumentation wurden übersetzt und an die beim Kunden in Russland vorhandene Technik und an die üblichen Arbeitsabläufe angepasst. Es gab auch *Entwicklungen eigens für den russischen Markt*. Allerdings bestand dabei das Problem, dass die Anzahl der Aufträge durch den Zusammenbruch vieler Handelsbeziehungen stark zurückgegangen war, und dass ein großer Teil der Auftraggeber die Software gar nicht oder nicht angemessen bezahlte.

Aufgrund der großen Bedeutung von Entwicklungsaufträgen aus dem Ausland für Softwareunternehmen in St. Petersburg soll im nächsten Abschnitt genauer auf die Besonderheiten dieser Projekte eingegangen werden.

3.3.1 Die Besonderheiten des „off shore development“

Für die Unternehmen in St. Petersburg waren Entwicklungsaufträge aus dem Ausland eine zuverlässigere Einnahmequelle als inländische Aufträge. Auch für ausländische Softwareunternehmen war es attraktiv, in St. Petersburg entwickeln zu lassen. Die Lohnkosten waren vergleichsweise gering, die Ausbildung der russischen EntwicklerInnen entsprach dem europäischen und amerikanischen Niveau. Sie stand sogar in dem Ruf, noch besser zu sein. Trotzdem ist die Entwicklung ausländischer Software in St. Petersburg nicht ohne Voraussetzungen möglich. Bei den Gesprächen mit russischen und deutschen Managern während der Materialerhebung in Deutschland zeigte sich, dass es zwei Möglichkeiten gibt, wie Aufträge in St. Petersburg erfolgreich bearbeitet werden können: Entweder die Aufgaben sind klar zu spezifizieren (1) oder die Entwickler sind in die Projekte im auftraggebenden Unternehmen zu integrieren (2).

(1) Klare Spezifikation

Selbst kleine Unternehmen in Westdeutschland machten von der Möglichkeit Gebrauch, klar zu spezifizierende Aufgaben in St. Petersburg bearbeiten zu lassen. Für Entwicklungsaufgaben bedeutet eine klare Spezifikation, dass die alltagssprachlichen und interpretationsoffenen Anforderungen der KundInnen in eindeutige und logische Strukturmerkmale des Programms übersetzt wurden. Durch die Arbeitsteilung zwischen Geschäftsführung und Entwicklung müssen Geschäftsführer auch in Deutschland die Anforderungen der KundInnen an die Entwicklungsabteilung weitervermitteln. Durch die technische Kompetenz der Geschäftsführer in kleinen Unternehmen fiel es ihnen nicht schwer, Anforderungen als technische Parameter für das zu entwickelnde Programm zu formulieren. In Deutschland gaben sie die Anforderungen oft mündlich weiter, was im weiteren Prozessverlauf häufig zu Missverständnissen führte. Die Weitergabe von Entwicklungsaufträgen nach Russland motivierte sie dagegen, diese Anforderungen schriftlich in einem Pflichtenheft darzulegen. Die russischen Manager traten dann als Mittelsmänner auf. Sie beherrschten Deutsch oder zumindest Englisch und besuchten die deutschen Unternehmen gelegentlich. Anforderungen, Verträge und Arbeitsergebnisse wurden per Email ausgetauscht. Die westdeutschen Geschäftsführer kleiner Unternehmen erlebten bei der Vergabe von Aufträgen nach St. Petersburg mehr Unsicherheit bezüglich des Projekterfolges als bei einer Entwicklung in der eigenen Firma. Sie befürchteten, dass irgendwann der Email-Kontakt abrechen könnte und sich unter der Telefon-Nummer niemand mehr melden würde. Die russischen Unternehmer lebten ihrerseits in der Angst, dass der westdeutsche Auftraggeber doch nicht wie versprochen zahlen würde. Sie hatten dann kaum Möglichkeiten, zu ihrem Geld zu kommen, es sei denn, sie würden in Deutschland klagen.

(2) Integration russischer Entwickler in Projekte

Diese Form der Bearbeitung von ausländischen Entwicklungsaufträgen findet sich hauptsächlich in Repräsentanzen ausländischer Unternehmen. Die EntwicklerInnen aus St. Petersburg sind dann in die Projekte im auftraggebenden Unternehmen integriert. Die Weitergabe von Entwicklungsaufgaben nach Russland war Teil der Arbeitsteilung in den international operierenden Unternehmen. Allerdings bedeutete diese Praxis für die Entwickler, dass sie sich anfangs längere Zeit und später regelmäßig über kürzere Zeiträume im Ausland aufhalten mussten, um das Unternehmen, ihre KollegInnen, die KundInnen, die verwendeten Tools und das Softwaresystem, für das sie programmierten, kennen zu lernen. Die russischen Entwickler gehörte dann quasi zum Projektteam im ausländischen Unternehmen dazu, war jedoch in St. Petersburg angestellt. Sie hielten sich einige Zeit im Ausland auf und gingen dann mit einem Aufgabenpaket zurück nach St. Petersburg. Die weitere Kommunikation erledigten sie per Telefon oder Email. Für die befragten russischen Entwickler hatte diese Arbeitsweise zur Folge, dass sie in der Regel zwei bis drei Jahre brauchten, bis sie die ausländische Software durchschauten und die unternehmensübliche Art zu programmieren verinnerlicht hatten.

Bei der Bearbeitung von Entwicklungsaufträgen thematisierten sowohl die russische als auch die deutsche Seite länderspezifische Unterschiede. Sie betrafen den Umgang mit Terminen, das Selbstverständnis der Entwickler und ihre Prioritätensetzungen innerhalb der Arbeit.

- Umgang mit Terminen

Die Geschäftsführer der befragten westdeutschen Unternehmen mit Kooperationserfahrungen in St. Petersburg waren der Meinung, dass die russischen Entwickler nicht in der Lage seien, Termine als verbindlich anzusehen und einzuhalten. Selbst wenn der Termin überschritten sei, sahen sie keine Notwendigkeit, die Programmieraufgabe zu Ende zu bringen. Für den westdeutschen Geschäftsführer war es dagegen wichtig, dass die in Auftrag gegebene Software zum festgelegten Termin fertig wurde. Um derartige Termine einzuhalten, arbeiteten die EntwicklerInnen in Deutschland auch am Wochenende, was die russische Seite ihrer Meinung nach nicht machte. Der russische Geschäftsführer sah als Ursache dieser Differenzen ein anderes Selbstverständnis der russischen Entwickler und die damit verbundene Prioritätensetzungen in der Arbeit: Inhalte genießen seiner Meinung nach für russische Entwickler eine höhere Priorität als Termine.

- Selbstverständnis

Der russische Geschäftsführer mit Entwicklungserfahrung in Deutschland erlebte deutsche EntwicklerInnen als „Handwerker“. Sie realisierten in der vorgegebenen Zeit, was von KundInnen gefordert wurde. Den russischen Entwicklern war seiner Meinung nach demgegenüber die Leistungsfähigkeit der Software wichtiger als die Einhaltung von Terminen. Sie sahen Programmieren als „Kunst“. Ein Gesprächspartner drückte es so aus, dass er für seine eigene Ehre entwickelte. Er verfolgte auch dann Ideen weiter, wenn er sie in der vorgegebenen Zeit nicht realisieren konnte.

- Prioritäten in der Arbeit

Den russischen Entwicklern war die technisch optimale Erfüllung aller gestellten inhaltlichen Anforderungen wichtiger als der vereinbarte Termin oder auch ein systematisches Testen und das Schreiben einer Dokumentation. Sie lösten die Aufgabe lieber perfekt, als dass sie sich überlegten, wie sie ihre Arbeitszeit einteilen mussten, um innerhalb des vorgegebenen Zeitrahmens zu einem zufriedenstellenden Ergebnis zu gelangen. Ihrer Meinung nach rechneten sie ihre Leistungen bei KundInnen auch nicht mit der Genauigkeit ab, wie das deutsche EntwicklerInnen tun.

Im Gegensatz zu diesen wechselseitigen Zuschreibungen deuten unsere bisherigen Untersuchungsergebnisse darauf hin, dass diese Stereotype nicht mit der Realität übereinstimmen. Der Widerspruch zwischen den auf Inhalte konzentriert arbeitenden EntwicklerInnen ohne Blick für das Umfeld und den rational alle Randbedingungen bedenkenden BerufsprogrammiererInnen wird bereits seit Weizenbaum (1978) und noch heute (vgl. Baukrowitz/Boes/Eckhardt 1994) strapaziert. Er wird den vielfältigen Arbeitsstilen in der Realität nicht gerecht (vgl. Strübing 1993). Unsere Untersuchungen zeigen, dass es auch in Deutschland eine Orientierung auf die maschinelle Effizienz der Software gibt, die eng damit verknüpft ist, dass Programmieren als „Kunst“ empfunden wird, die nicht rational zu erklären ist. Gleichzeitig war das aber nur eine von vier möglichen Orientierungen (vgl. Sieber 2001). Bei der Ausprägung der Orientierung spielte in dieser Untersuchung die berufliche Sozialisation und die bisherige Berufserfahrungen eine besondere Rolle. Die Arbeitsweise in den Unternehmen und die Eigenschaften der entwickelten Software hingen demgegenüber besonders mit der

Entwicklungsgeschichte des Unternehmens, seiner regionalen Einbindung und den internen und externen Kooperationsformen zusammen.(vgl. Henninger/Sieber 2000)

4 Schwierigkeiten und Lösungsansätze

Die Vorstudie in St. Petersburg erbrachte erste Erkenntnisse über die Softwarelandschaft in Russland sowie methodische und analytische Schwierigkeiten einer ländervergleichenden empirischen Untersuchung in Russland, den USA und einer bereits abgeschlossenen empirischen Untersuchung in Deutschland. Die Erfahrungen zeigen, dass wir bei einer ähnlich wie in Deutschland ausgerichteten empirischen Untersuchung in Russland mit Schwierigkeiten in dreierlei Hinsicht rechnen müssen: Sprachbarrieren, Vergleichbarkeit des Samples, Zugang zu Sekundärliteratur.

- Sprachbarrieren

Trotz langjährigen Russischunterrichtes und trotz des vierwöchigen Sprachaufenthaltes stellt das Verstehen der InterviewpartnerInnen ein methodisch nicht vernachlässigbares Problem dar. Die Gespräche waren durch zahlreiche Verständnisfragen meinerseits gekennzeichnet, was sich ungünstig auf die Gesprächssituation auswirkte: Die Befragten erzählten weniger detailliert und wählten ihre Worte so, dass ich sie verstehen würde. Es liegt nahe, dass dadurch die Darstellung der Themen des Leitfadens aus Sicht der Befragten verfälscht würde.

Diese Schwierigkeiten ließen sich auf dreierlei Weise umgehen: (1) wenn ein Dolmetscher am Gespräch teilnehmen würde, (2) wenn ein bilingualer Interviewer eingesetzt würde, der mit den Zielen des Projekts und den Methoden vertraut wäre oder (3) wenn die Gesprächspartner Englisch sprächen. Jede dieser Möglichkeiten birgt wiederum Einschränkungen bestimmter Art. Ein Dolmetscher würde die Gesprächssituation verändern und seine Interpretationen des Gesagten einbringen. Die Ausbildung eines bilingualen Interviewers wäre innerhalb des Projektes nicht zu bewältigen und EntwicklerInnen, die sehr gut Englisch sprechen, haben vermutlich Arbeitserfahrungen im Ausland und transportieren ein bestimmtes Bild russischer und ausländischer Entwicklungsweisen. Diese Aussagen könnten jedoch mit den bereits vorliegenden Ergebnissen zur Arbeitsweise von EntwicklerInnen in Deutschland und den in der nächsten Projektphase zu gewinnenden Erkenntnissen über Entwicklungsarbeit in den USA kontrastiert werden.

- Vergleichbarkeit des Samples

Bereits in der Vorstudie deutete sich an, dass die Organisationsformen von Unternehmen der Softwarebranche in Deutschland und in Russland nicht ohne weiteres miteinander vergleichbar sind. Während in Deutschland kleine Unternehmen die Branche maßgeblich mitbestimmen, scheinen kleine Unternehmen in Russland eine eher untergeordnete Rolle zu spielen. In St. Petersburg scheint die Softwarebranche geprägt zu sein von ausländischen Repräsentanzen, mittleren Unternehmen und Selbständigen. Die Interviews mit EntwicklerInnen in den kleinen deutschen Unternehmen lassen jedoch vermuten, dass der organisatorische Kontext, in dem sie arbeiten, einen erheblichen Einfluss auf ihre individuelle und die kollektive Arbeitsweise hat (vgl.

Sieber 2001). Das bedeutet, dass es ohne die Analyse eines mittleren Unternehmens in Deutschland sehr schwierig wird, einen Vergleich mit der Arbeitsweise von EntwicklerInnen in Russland zu ziehen.

- Zugang zu Sekundärliteratur

Die Geschäftsbereiche der Softwareentwicklungsunternehmen, aber auch die Merkmale der Software und die bevorzugten Projekttypen hängen nach den Erkenntnissen aus der Erhebung in Deutschland (vgl. Henninger/Sieber 2000) mit der Wirtschaftsstruktur in der Region zusammen. Bisher gestaltete es sich außerordentlich schwierig, Zugang zu deutsch- oder englischsprachigen statistischen Daten oder gar wissenschaftlichen Untersuchungen zu erhalten, die regionale Zusammenhänge in Russland thematisieren, oder welche die Softwareentwicklungsbranche bzw. Softwareentwicklung untersuchen.

Es wird also nur schwer möglich sein, eine methodisch und analytisch ähnlich wie die Untersuchung in Deutschland ausgerichtete Forschung in Russland durchzuführen. Dazu ist die Branche nach den Erkenntnissen der Vorstudie zu verschieden von Deutschland strukturiert und dazu sind die Schwierigkeiten mit der Sprache und beim Zugang zu Sekundärliteratur zu groß. Eine Möglichkeit, die Entwicklungsarbeit in Deutschland, Russland und perspektivisch den USA trotz allem zu vergleichen, bietet sich an durch die empirische Untersuchung eines Unternehmens, das Niederlassungen in diesen Ländern hat. Auf dieser Basis ließe sich die individuelle und kollektive Arbeitsweise der EntwicklerInnen innerhalb eines gleichartigen organisatorischen Settings vergleichen. Eine derartige Untersuchung könnte zudem Gemeinsamkeiten und Unterschiede zwischen den Selbstaussagen der EntwicklerInnen und den Ergebnissen der teilnehmenden Beobachtungen zutage fördern.

Literatur

- Baukrowitz, Andrea; Boes, Andreas; Eckhardt, Bernd: Software als Arbeit gestalten: Konzeptionelle Neuorientierung der Aus- und Weiterbildung von Computerspezialisten. Opladen : Westdeutscher Verlag, 1994.
- Henninger, Annette; Sieber, Andrea: Softwareentwicklung in kleinen und mittleren Unternehmen in Ost- und Westdeutschland, Vortrag auf der Tagung „Neue Medien im Alltag“ am 10./11. November 2000, <http://www.tu-chemnitz.de/NeueMedien/NeMeAA2000/>, (Stand 12.01.01).
- Hans-Böckler-Stiftung (Hrsg): St. Petersburg - im Kleinen ganz groß. Ein Bericht über die 3. Sommersprachschule der HBS an der Universität St. Petersburg 1996. Düsseldorf : HBS, 1996.
- Hans-Böckler-Stiftung (Hrsg): Von russischer Seele, starken Frauen und ungewisser Zukunft. Ein Bericht über die 5. Sommersprachschule der HBS an der Universität St. Petersburg 1998. Düsseldorf : HBS 1998.
- Konrad, Wilfried; Paul, Gerd: Innovation in der Softwareindustrie: Organisation und Entwicklungsarbeit. Frankfurt am Main ; New York : Campus, 1999.
- Schwarz, Anna: Diverging Patterns of Informalization between Endogenous and Exogenous Economic Actors in the East German Transformation Process: Results from a Case-Study in the Information-Technology Sector in Berlin-Brandenburg. In: Stöltung, Michael; Brie, Michael (Eds.): The Transformation of the Informal. Chicago : University of Notre Dame, 2000. (in print)
- Strübing, Jörg: Subjektive Leistungen im Arbeitsprozess: Eine empirische Untersuchung von Arbeitsstilen in der Programmierarbeit. Wiesbaden : Deutscher Universitätsverlag, 1993.

Sieber, Andrea: Software entwickeln in der Praxis. Vier Fallstudien in kleinen Unternehmen Ostdeutschlands. Chemnitz 2001. (unveröffentlichtes Manuskript)

Weizenbaum, Joseph: Die Macht der Computer und die Ohnmacht der Vernunft. Frankfurt a. M. : Suhrkamp, 1978.

Vorgehensmodelle, Projekttypen und Arbeitspraktiken als Bausteine zur realitätsnahen Beschreibung von Softwareentwicklung

Andrea Sieber, Annette Henninger

Abstract In a case study it is shown that different models of the development process and different project types are not enough to illuminate software development in everyday life. A better understanding is established by describing working practices as they are documented in an empirical study in small software development businesses. The results show why some project models do not succeed in practice and why some tools will not be used for development.

1 Einleitung

Kleine Software-Unternehmen sind in besonderem Maße auf zufriedene KundInnen und MitarbeiterInnen angewiesen, da sie im Vergleich zu großen Firmen nicht über ausreichend Kapital verfügen, um eine schlechte Auftragslage länger abzapuffern und um ihre MitarbeiterInnen dem Durchschnitt entsprechend zu bezahlen. Gleichzeitig wird ihnen eine große Flexibilität und Schnelligkeit bei der Reaktion auf Wünsche der KundInnen, Softwareentwicklung in guter Qualität und ein motivierendes Arbeitsklima nachgesagt. Auch große Software-Unternehmen nutzen diese Vorteile, indem sie Aufträge, die schnell erledigt werden müssen, an kleine Firmen vergeben. Nach Strübing (1993, S.70) hatten 1992 von 4500 Softwarefirmen in Deutschland nur 24 mehr als 200 Beschäftigte. Im Osten Deutschlands wurde nach der Wende die Gründung von kleinen Technologieunternehmen gefördert, so dass zahlreiche kleine Firmen entstanden, die Dienstleistungen in diesem Bereich anbieten (vgl. Hinz 1998, S. 94). Diese Tendenzen waren für uns Anlass, die in den kleinen Softwareunternehmen praktizierten Entwicklungsmethoden genauer zu untersuchen.¹ Die Analyse kleiner Unternehmen bietet zudem den Vorteil, dass man sowohl das Management als auch die Entwickler einbeziehen kann und so Aufschluss über individuelle und kollektive Vorgehensweisen in einer bestimmten Organisation gewinnt.

Bisher wird Softwareentwicklung hauptsächlich anhand von Vorgehensmodellen beschrieben. Weltz/Ortmann haben in ihrer empirischen Untersuchung von Managementpraktiken in 46

¹ Diesem Artikel liegen qualitativ empirische Untersuchungen in 10 kleinen und mittleren Software-Unternehmen zugrunde. Unser Dank gilt der Hans-Böckler-Stiftung und der DFG (Bo 929/13-1) für die finanzielle Unterstützung.

Softwareprojekten festgestellt, dass sich Unternehmen unterschiedlichster Größe bei der Organisation des Projektablaufs zwar an Phasenmodellen orientieren, diese aber immer wieder situativ anpassen (1992, S. 81f.). Bittner/Hesse/Schnath (1995) wenden sich den Projekten aus einem Teil dieses Samples zu. Auf der Grundlage der Anforderungen aus der Entwicklungspraxis entwerfen sie ein Vorgehensmodell für ein dynamisches Projektmanagement (ebd., S. 177ff.). Bei beiden Studien bilden Projekte die Untersuchungseinheit. Analysiert werden Projektverläufe in Unternehmen unterschiedlicher Größe. Wetz/Ortmann unterscheiden dabei zwischen Neu- und Weiterentwicklungen. Bittner/Hesse/Schnath differenzieren verschiedene Projekttypen nach dem Anwendungsbereich. Auch Deifel et al. (1999) beziehen sich bei ihrer Analyse auf Projekte und identifizieren fünf verschiedene Projektinhalte, die zu vier verschiedenen Prozesstypen führen. Der Einfluss des organisatorischen Kontexts auf den Entwicklungsprozess wird bei diesem Zugang nicht berücksichtigt.

Einen anderen Zugang wählt Strübing (1993) bei seiner Untersuchung zu Arbeitsstilen in der Programmierarbeit. Seiner Studie liegen Interviews mit 30 EntwicklerInnen zugrunde, die in unterschiedlichen organisatorischen Kontexten arbeiten. Strübing unterscheidet zwischen intuitiv-situativen und systematisch-methodischen Arbeitsstilen. Elemente dieser Arbeitsstile sind die Organisation der Arbeitsschritte, die Darstellung von Zwischenergebnissen, das Testen und der Umgang mit Tools. Strübing beschreibt ebenso wie Wetz/Ortmann die situative Anpassung der normativ vorgegebenen Phasenmodelle. EntwicklerInnen, die einen intuitiv-situativen Arbeitsstil verfolgen, neigen nach Strübing dazu, verschiedene Phasen zu durchmischen, während EntwicklerInnen des anderen Typs sich eher den Phasen anpassen (ebd., S. 119ff.). Auch Strübing verfolgt nicht das Ziel, den Zusammenhang zwischen individuellem Arbeitsstil und kollektivem Vorgehen in einer Firma herauszuarbeiten. Unter „kollektivem Habitus“ versteht er das Selbstverständnis der EntwicklerInnen als Gruppe und ihre Abgrenzung von anderen betrieblichen Gruppen.

Diese verschiedenen empirischen Zugänge zum Verlauf von Entwicklungsprojekten erklären jeweils einen bestimmten Ausschnitt des Prozesses. Die genannten Untersuchungen stellen jedoch keinen Bezug her zwischen Vorgehensmodellen, Projekttypen und Arbeitsstilen und auch nicht zwischen diesen Elementen und dem organisatorischen Kontext, in den die Softwareentwicklung eingebettet ist. In einem Unternehmen spielen jedoch alle genannten Elemente gleichzeitig eine Rolle. Unser Anliegen ist es deshalb, Softwareentwicklung innerhalb eines bestimmten organisatorischen Kontexts (kleine Unternehmen) mithilfe dieser Elemente (Vorgehensmodell, Projekttyp, Arbeitsstil) zu analysieren und dabei ihre Wechselwirkungen zu berücksichtigen.

Mit diesem Ziel haben wir unter Nutzung von Methoden der empirischen Sozialforschung EntwicklerInnen in kleinen Softwareunternehmen bei ihrer Arbeit beobachtet und in leitfadengestützten Interviews nach ihrer Vorgehensweise befragt. Bei der Auswertung des Datenmaterials haben wir festgestellt, dass nur der Geschäftsführer das Projektmanagement ausführlich beschreibt. Die EntwicklerInnen sprechen hauptsächlich über Tätigkeiten, die für die Umsetzung der Projekte wesentlich sind. Diese unterschiedlichen Perspektiven spiegeln sich

implizit in den Vorgehensmodellen der Softwaretechnik wider (Abschnitt 3.1). Während die Vorgehensmodelle unterschiedliche Perspektiven auf den Projektverlauf verdeutlichen, können sie nur unzureichend den Ablauf unterschiedlicher Projekttypen erklären. Der Projektverlauf und die Durchführung einzelner Arbeitsschritte sind genauso abhängig von einer projektspezifischen Interaktionsstruktur mit den KundInnen (Abschnitt 3.2). Diese Interaktionsstruktur lässt jedoch offen, weshalb manche Projekte mit mehr, andere mit weniger Erfolg durchgeführt werden. Dazu bedarf es der Analyse von individuellen und kollektiven Vorgehensweisen der EntwicklerInnen (Abschnitt 3.3). Abschließend arbeiten wir die Wechselwirkungen zwischen diesen Elementen heraus und formulieren empirisch begründete Hypothesen über den Verlauf von Entwicklungsprojekten in kleinen Softwareunternehmen (Abschnitt 4). Um unseren Erkenntnisweg nachvollziehbarer zu machen, wird im folgenden Abschnitt zunächst die Firma SuperTech² vorgestellt. An ihrem Beispiel lassen sich die typischen Probleme und Konflikte bei der Arbeitsorganisation gut aufzeigen, mit denen eine kleine ostdeutsche Software-Firma beim Markteintritt konfrontiert ist.

2 Die Firma SuperTech – ein exemplarischer Fall

Die Firma SuperTech wird 1991 von Bruno Haubold gegründet, um einen an der Universität entwickelten Prototypen in ein Produkt zu verwandeln. Als erstes bemühte Haubold sich um Fördermittel. Mit diesem Geld konnte er nach und nach vier Mitarbeiter einstellen, zuerst seine Frau, die die Büroarbeiten übernimmt, dann Bernhard Wolf, einen ehemaligen Kollegen von der Universität, und später zwei weitere Hochschulabsolventen, Albert Fricke und Sylvia Krone. Nach einem Jahr wird eine erste Version des Produkts als Betaversion ins Internet gestellt. Aus einer aufwendigen Werbeaktion per Telefon und E-Mail ergibt sich lediglich ein Kundenkontakt, der nach zwei weiteren Jahren tatsächlich zu einem ersten großen Anwendungsprojekt führt. Es folgen zwei weitere Projekte, dann lässt das Interesse des Kunden nach. Glücklicherweise kann SuperTech einen 6-monatigen Auftrag von einem neuen Kunden akquirieren. Bei allen Projekten sind die Entscheidungszeiten aufgrund der Größe der auftraggebenden Unternehmen sehr lang. Die finanzielle Ausstattung reicht allerdings zur Finanzierung aller MitarbeiterInnen von SuperTech aus. Haubolds Schilderungen zum Verlauf der Anwendungsprojekte umfassen die Überzeugungstaktiken von SuperTech bei den ersten Präsentationsterminen, die Datenerhebung, das Verfassen eines Angebotes bis zur Realisierung und Ergebnispräsentation.

Danach folgen keine weiteren Anwendungsprojekte. Marketingaktivitäten in den bisherigen Anwendungsbranchen lohnen sich nicht, da sich alle Unternehmen untereinander kennen und erst bei erwiesenen wirtschaftlichen Vorteilen ebenfalls Projekte starten würden. Eher zufällig wird SuperTech aufgefordert, ein Angebot zur Lösung einer Programmieraufgabe bei einer öffentlichen Einrichtung zu entwickeln. Die Firma erhält den Zuschlag und kann sich so ein weiteres halbes Jahr finanzieren. Der Verlauf des Auftragsprojekts unterscheidet sich von den

² Bezeichnungen, die auf tatsächliche Personen und Organisationen schließen lassen, wurden aus Gründen der Anonymität geändert.

Anwendungsprojekten, da die Firma erst mit dem Kunden in Kontakt tritt, als er das Angebot bereits akzeptiert hat. Der Auftraggeber testet die Entwürfe, diskutiert mit SuperTech darüber und formuliert mehrfach neue Anforderungen. Es kommt zum Streit über den Abschluss des Projekts. Zudem war die interne Arbeitsteilung bei SuperTech ungünstig gelöst, und es kommt auch zu internen Konflikten. Am Ende dieses Projektes entlässt Haubold zwei MitarbeiterInnen. Bruno Haubold studierte Informationstechnik und arbeitete danach in einem universitären Forschungsprojekt. Er ist Geschäftsführer der Firma SuperTech und gleichzeitig Entwickler. Haubold diskutiert konzeptionelle Ideen gerne mit allen EntwicklerInnen beim gemeinsamen Frühstück, was durchaus bis mittags dauern kann. Bei der Umsetzung merkt Haubold, wie weit seine Ideen reichen. Kann er Programmierprobleme nicht alleine lösen, holt er Bernhard Wolf an den Rechner. Meist findet er schon dadurch den Fehler, dass er Wolf das Problem erklärt, oder dessen Fragen helfen ihm dabei. Wolf hat Automatisierungstechnik studiert. Er war ebenfalls in einem Forschungsprojekt an der Universität beschäftigt und kooperierte dabei mit Haubold. Danach wurde er Entwickler in seiner Firma. Bei der Programmierung achtet Wolf darauf, dass in den Basisklassen nur die Grundfunktionalität realisiert wird. Alle spezifischen Anforderungen und komplizierten Algorithmen löst er in den Ebenen darüber. So bleibt sein Konzept einfach und überschaubar.

Albert Fricke hat wie Bruno Haubold Informationstechnik studiert. Ihm sind Plattformunabhängigkeit und eine verteilte Konzeption der Software wichtig. Er überlegt sich schon zu Beginn von Projekten, welche Funktionalität noch kommen könnte, und realisiert auch komplizierte Algorithmen in den Basisklassen. Die Arbeit bei SuperTech war für Fricke anfangs sehr interessant, aber als sich nach 5 Jahren immer noch nicht der Einsatz des Produktes abzeichnet, lässt seine Motivation stark nach. Er überlegt, sich selbständig zu machen. Bevor es dazu kommt, wird er von Haubold entlassen. Sylvia Krone, eine Informatikerin mit Universitätsabschluss, wurde für die Oberflächengestaltung eingestellt. Zudem wurden ihr alle gestalterischen Aufgaben und das Schreiben des Produkt-Handbuches übertragen. Dabei testet sie das Produkt aus Anwenderperspektive, wobei sie jeweils prüft, ob sich die Software verhält wie vorgesehen. Auf diese Art entdeckt sie viele Fehler. Krone ist es gewohnt, Entwurfsentscheidungen zu dokumentieren. Dafür wird sie in der Firma gehänselt. Als Haubold noch mehr Überstunden von ihr verlangt, beschließt sie, einen anderen Job zu suchen. Ihrem Weggang kommt Haubold durch die Kündigung zuvor.

Arbeitsbeginn bei SuperTech ist zwischen 8 und 9 Uhr morgens. Die Sekretärin geht als erste am Nachmittag, Sylvia Krone gegen 18 Uhr, Wolf und Fricke verlassen die Firma gegen 20 Uhr, Haubold zwischen 20.30 und 21.30 Uhr. Steht ein Termin ins Haus, wird auch am Wochenende und an Feiertagen gearbeitet. Die SuperTech-EntwicklerInnen nutzen eine objektorientierte Programmiersprache. Jede Person ist selbst dafür verantwortlich, sich über Organisatorisches, wie z. B. Termine, zu informieren. Die Rechner sind vernetzt, aber es gibt keine Software, welche die gemeinsame Verwaltung des Quellcodes, von Adressen oder Terminen erleichtert. Werden global verfügbare Quelltextdateien durch die jeweiligen EntwicklerInnen erneuert, dann geschieht dies auf Zuruf.

3 Softwareentwicklung in kleinen Unternehmen

Bei SuperTech gibt es kein explizites, vom Geschäftsführer festgelegtes Vorgehensmodell, nach dem sich alle EntwicklerInnen zu richten haben. Implizit hat sich trotzdem ein wiederkehrender Ablauf der Softwareentwicklung herausgebildet. Die Software entsteht in mehreren Zyklen, wobei die in den Zyklen entstehenden Entwürfe aufeinander aufbauen.

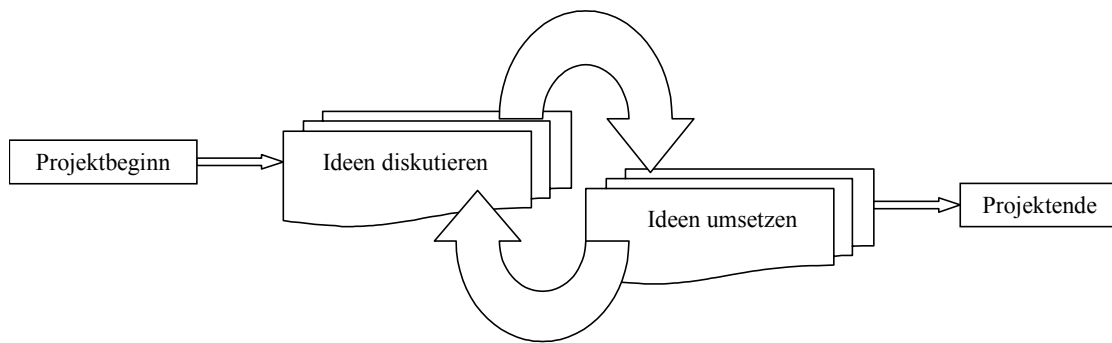


Abb. 1 Vorgehen bei der Realisierung

Abb. 1 zeigt die zwei Phasen des Zyklus, welche die EntwicklerInnen bei der Realisierung benennen: *Ideen diskutieren* und *Ideen umsetzen*. Das Projekt beginnt mit einer Diskussion, wenn ein Termin beim Kunden ansteht. Der Wechsel von der Diskussion zur Umsetzung erfolgt, wenn ein gemeinsames Konzept erstellt wurde und die Arbeit aufgeteilt ist. Die Umsetzung besteht im Schreiben und Ausführen von Quelltext und der gemeinsamen Klärung kleinerer Probleme. Der Wechsel zur erneuten gemeinsamen konzeptionellen Diskussion erfolgt, wenn die EntwicklerInnen der Meinung sind, dass die festgestellten konzeptionellen Schwächen so groß sind, dass das Konzept nicht mehr haltbar ist und im Projekt noch Zeit für einen neuen Zyklus ist. Die Abläufe unterscheiden sich von Projekt zu Projekt. Die genauen Zeitpunkte des Projektbeginns und der Wechsel zwischen den Phasen sowie die Ausführung der Tätigkeiten variieren zwischen den EntwicklerInnen.

Drei sich in den untersuchten Unternehmen systematisch wiederholende Auffälligkeiten im Projektverlauf wollen wir im folgenden einer genaueren Analyse unterziehen:

1. Nur der Geschäftsführer nimmt auf die anderen Schritte neben der Realisierung Bezug (Abschnitt 3.1).
2. Der Verlauf der Projekte innerhalb der Firma ist unterschiedlich (Abschnitt 3.2).
3. Der Wechsel zwischen den in Abb. 1 beschriebenen Elementen der Realisierung variierten subjektiv (Abschnitt 3.3).

3.1 Vorgehensmodelle und Sichtweisen

Während der Geschäftsführer von SuperTech den gesamten Projektverlauf vom ersten Kontakt mit KundInnen bis zur Projektübergabe ausführlich beschreibt, beziehen sich die EntwicklerInnen vornehmlich auf einzelne Etappen im Realisierungszyklus sowie die Ziele und Schwierigkeiten bei der Aufgabenbewältigung. Aus ihren Schilderungen geht hervor, dass AnwenderInnen mit wenig technischem Verständnis wieder andere Ansprüche an den Projektverlauf stellen. Unserer Meinung nach lässt sich ein impliziter Einfluss dieser unterschiedlichen Sichtweisen auf die als zentral erachteten Tätigkeiten im Projektverlauf gut anhand der Genealogie der Entwicklungsschemata aufzeigen, die Bremer (1998, Abb. S. 37) zusammengestellt hat (vgl. auch Henninger/Sieber/Dilger in diesem Band, Abb. S. 26). Der dargestellte Stammbaum hat drei Äste: Einen linken vom Phasenmodell bis zum Spiralmodell, einen rechten vom Aspekte-/Sichtenmodell bis zum Referenzlinienmodell und einen mittleren vom Phasen-/Tätigkeitsmodell abwärts. Schaut man sich an, was die Modelle in den einzelnen Ästen verbindet, so lässt sich eine Management-Perspektive, eine Entwicklungs-Perspektive und eine Anwendungs-Perspektive identifizieren.

Das erste Modell im *linken Ast* entstand, um die zuverlässige Herstellung umfangreicher Softwareprodukte zu ermöglichen. Hierzu wurden Managementenerfahrungen aus anderen Branchen auf die Softwareentwicklung übertragen. Auf diese Weise entstand das wohl bekannteste Software-Entwicklungsmodell: Royces Wasserfall-Modell. Die stufenweise Abarbeitung der einzelnen Tätigkeiten (Aufgabenanalyse, Spezifikation, Design, Implementation, Test, Wartung) ermöglichte die Strukturierung und Kontrolle des Entwicklungsprozesses aus der Perspektive des Managements. Die Modelle im *mittleren Ast* nehmen demgegenüber wesentlich stärker Bezug auf die Tätigkeiten in der Entwicklung. Dabei liegt in jedem dieser Modelle der Fokus auf anderen Tätigkeiten – eben jenen, welche die Autoren für ihren Entstehungskontext als zentral ansahen. Während sich im linken Ast die Entwicklungstätigkeiten den definierten Phasen unterzuordnen hatten und nicht näher benannt wurden, ist es in diesen Modellen genau umgekehrt: Hier bestimmen die konkreten Entwicklungstätigkeiten die einzelnen Projektphasen. Die Management-Anforderung nach Kontrollierbarkeit tritt dabei in den Hintergrund. Die Modelle im *rechten Ast* wurden in einem Kontext entwickelt, in dem die Einbeziehung der zukünftigen AnwenderInnen in den Entwicklungsprozess ein zentraler Bestandteil war. Ziel dieser Modelle war es, die Tätigkeiten in den Blick zu rücken, die nötig waren, um einen Konsens zwischen den unterschiedlichen Interessen der AnwenderInnen, der EntwicklerInnen und der ManagerInnen herbeizuführen.

Die Modelle legen je nach ihrem Entstehungskontext bzw. je nach den Intentionen derjenigen, die sie aufgestellt haben, den Schwerpunkt auf unterschiedliche Tätigkeiten und Abläufe im Entwicklungsprozess. Dabei wird deutlich, dass EntwicklerInnen andere Tätigkeiten als wichtig für einen erfolgreichen Projektverlauf ansehen als ManagerInnen oder AnwenderInnen. Die Modelle im linken Ast strukturieren den Prozess so, dass er für das Management steuer- und kontrollierbar ist. Diese Sichtweise entspricht in unserem Beispiel den Interessen Haubolds als Geschäftsführer. Die Modelle des mittleren Astes sind geprägt von dem Wunsch, den Prozess

entlang der Entwicklungstätigkeiten zu strukturieren. Dieser Ast entspricht der Sichtweise Haubolds (in seiner Funktion als Entwickler) sowie der Entwickler Wolf und Fricke. Die Modelle im rechten Ast haben zum Ziel, die Perspektive der AnwenderInnen in den Entwicklungsprozess einzubringen. Dieses Ziel wird bei SuperTech in der Hauptsache von Sylvia Krone verfolgt. Diese unterschiedlichen Sichtweisen auf den Prozessverlauf erklären zwar, weshalb der Geschäftsführer andere Tätigkeiten beschreibt als die EntwicklerInnen, aber nicht den prinzipiell unterschiedlichen Verlauf der Projekte. Dazu ist es nötig, sich genauer mit dem Projektkontext zu beschäftigen.

3.2 Projekttypen

Bei der Analyse des Fallbeispiels zeigt sich, dass sich die Projektverläufe im Förderprojekt zur Produktentwicklung, in den Erst- und Folgeprojekten zur Anwendung des Produkts von SuperTech und bei der Auftragssoftware für einen öffentlichen Auftraggeber unterscheiden. Beim Förderprojekt wird nach der Bewilligung des Antrages der Realisierungszyklus nur durch die selbstgesteuerte Definition von Zwischenständen (Prototyp, Beta-Version, Version 1.0) unterbrochen, an die sich verschiedene Marketingaktivitäten (Bereitstellen im Internet, Mails versenden, Telefonate mit potentiellen KundInnen) anschließen, bevor der Realisierungszyklus wieder aufgenommen wird.

Erstprojekte zur Anwendung des Produktes beginnen mit der Phase der Projektinitiierung. Dabei wird in einem ersten Realisierungszyklus eine vorläufige Version der Anwendungssoftware für den Präsentationstermin beim Kunden entwickelt. Dieser Phase folgt die Datenerhebung, eine Angebotserstellung und die Realisierung, die schon vor Bestätigung des Angebotes beginnt und durch die Präsentation des Ergebnisses abgeschlossen wird. Bei Folgeprojekten reduziert sich die Projektinitiierung auf einen Anruf des Kunden, die Datenerhebung reduziert sich auf die Frage, ob es Änderungen gab, danach erfolgt die Angebotserstellung und die Realisierung, die bei dieser Projektart erst nach der Bestätigung des Angebotes beginnt.

Im Projekt für den öffentlichen Auftraggeber schreibt die Firma SuperTech zuerst ein Angebot. Nachdem der Zuschlag erteilt wurde, erstellt SuperTech gemeinsam mit den AnwenderInnen ein Pflichtenheft. Danach beginnt der Realisierungszyklus, der um den Test des Entwurfs durch die AnwenderInnen und die Diskussion über noch bestehende Anforderungen mit ihnen erweitert ist. Dieser erweiterte Realisierungszyklus wird solange durchlaufen, bis die AnwenderInnen mit dem Ergebnis zufrieden sind bzw. die EntwicklerInnen der Meinung sind, dass das Pflichtenheft erfüllt ist.

Eine Möglichkeit der Systematisierung dieser verschiedenen Abläufe bietet Grudin (1991). Er differenziert verschiedene Projektkontexte anhand des Zeitpunkts, an dem AnwenderInnen und EntwicklerInnen miteinander bekannt werden, und unterscheidet dabei drei Projekttypen: *Vertragsentwicklung*, *Produktentwicklung* und *In-Haus-Entwicklung*. Bei der *Vertragsentwicklung* sind die AnwenderInnen vor den EntwicklerInnen bekannt, da erstere ein Projekt ausschreiben mit dem Ziel, ein geeignetes Software-Unternehmen für die Lösung der Aufgabe zu finden. Diese Form der Entwicklung findet bei öffentlichen Ausschreibungen statt, wie sie für öffentliche Ein-

richtungen vorgeschrieben sind. Während einer *Produktentwicklung* sind die AnwenderInnen erst bekannt, wenn das Produkt tatsächlich verkauft wird. Bis dahin haben die EntwicklerInnen zwar eine gewisse Vorstellung über die zukünftigen AnwenderInnen, aber es kann passieren, dass am Ende ganz andere Personen das Produkt nutzen. Bei der *In-Haus-Entwicklung* kennen sich beide Seiten von Anfang an. Die EntwicklerInnen gehören zur Anwendungsorganisation oder werden eigens für die Entwicklung von Software angestellt. Grudin merkt allerdings an, dass sich nicht alle Projekte eindeutig diesen Kontexten zuordnen lassen. Es gibt Mischformen, wie z. B. eine Vertragentwicklung, bei welcher die AuftragnehmerInnen vorher bekannt sind und die Ausschreibung auf sie zugeschnitten wurde.

Das Förderprojekt bei SuperTech wäre nach Grudins Systematik eine typische Produktentwicklung. Es gibt noch keine konkreten KundInnen und verschiedene Anwendungsbereiche sind vorstellbar. Die Erst- und Folgeprojekte zur Anwendung des Produkts lassen sich als In-Haus-Entwicklungen charakterisieren, auch wenn die EntwicklerInnen nicht bei der Anwendungsorganisation angestellt sind. Doch sind EntwicklerInnen und AnwenderInnen in diesen Projekten von Anfang an bekannt. Das Projekt für den öffentlichen Auftraggeber ist eine typische Vertragentwicklung.

Auffällig ist bei allen von SuperTech bearbeiteten Projekten der distanzierte Umgang der EntwicklerInnen mit dem Kunden. Lediglich Sylvia Krone macht sich während der Produktentwicklung Gedanken über die Bedürfnisse der AnwenderInnen. Außer von Fricke wird das von keinem anderen Entwickler als wichtig erachtet. Während der In-Haus-Entwicklung hat SuperTech nur am Anfang und am Ende des Projekts direkten Kontakt mit dem Kunden. Ansonsten beschränkt sich die Verbindung auf Telefonate, falls die EntwicklerInnen Informationen benötigen oder der Kunde eine Frage hat. Bei der Vertragentwicklung möchte der Kunde in den Realisierungszyklus integriert werden und bei der Definition des Projektendes mitreden. Dies wird von den SuperTech-Mitarbeitern negativ bewertet. Aus ihrer Sicht behindert die intensive Kundenbeteiligung die effektive Durchführung des Projekts und zögert seinen Abschluss hinaus. Das Projekt endet im Streit über die vertragsgemäße Erfüllung.

Die von Grudin identifizierten drei Projektkontexte machen zwar die projektspezifische Interaktionsstruktur zwischen EntwicklerInnen und AnwenderInnen vorhersagbar, aber nicht den Umgang mit den KundInnen und den konflikträchtigen Verlauf der Vertragentwicklung bei SuperTech. Dazu ist eine genauere Betrachtung der Arbeitspraktiken der EntwicklerInnen nötig.

3.3 Arbeitspraktiken

Unter Arbeitspraktiken verstehen wir das Vorgehen der EntwicklerInnen in bestimmten Situationen. Dabei handelt es sich um soziale Handlungen, die ohne Berücksichtigung der Interpretationen der Akteure nicht erklärt werden können. Diese subjektiven Interpretationsprozesse liefern den Schlüssel dafür, welche Einstellung eine Person z. B. zu zeitlicher Planung hat oder wann bestimmte Entwicklungsmethoden eingesetzt werden. Wird eine Situation in einer bestimmten Art und Weise interpretiert, dann wird die dazugehörige

Handlung ausgeführt. Je mehr diese Handlungen routinisiert sind, desto unbewusster erfolgt der jeweilige Interpretationsprozess (Kelle 1994, S.15).

Die Arbeitspraktiken der EntwicklerInnen sowie ihre Situationsdeutungen sind für die in Abb.1 beschriebenen Phasen des Entwicklungszyklus und die Wechsel von einer Phase zur nächsten bedeutsam. Die EntwicklerInnen verbinden unterschiedliche Absichten mit zeitlicher Planung. Sie äußern unterschiedliche Einschätzungen, wann Ideen als ausreichend diskutiert gelten, um zur Umsetzung überzugehen, wann die konzeptionellen Schwächen der aktuellen Implementierung so weitreichend sind, dass neu konzipiert werden muss. Ebenso unterscheidet sich ihr Vorgehen bei der Diskussion und der Umsetzung der Konzepte.

- Knappe Zeitplanung vs. Einplanung von Pufferzeiten

Haubold beginnt bei neuen Kunden aus strategischen Gründen mit der Arbeit, noch ehe er die Bestätigung des Auftrages in der Hand hält. Ein Grund hierfür ist, dass er die notwendige Zeit für die Realisierung des Projektes im Auftrag bewusst zu knapp bemisst, um einen Vorteil gegenüber eventuellen Konkurrenten zu haben. Bei Folgeprojekten oder beim Auftragsprojekt beginnt er erst nach der Auftragsbestätigung mit der Arbeit. Vor einer Ergebnispräsentation beginnt er mit dem Ziel, sich durch eine technisch perfekte Lösung Vorteile zu verschaffen, auch noch kurz vor dem Termin mit einer kompletten Revision der Implementierung. Dafür nimmt er auch Feiertagsarbeit in Kauf. Krone dagegen würde anders vorgehen. Mit der Erfahrung, dass in der Firma „Murphies Gesetze“ gelten, kalkuliert sie „lieber zwei Tage mehr als zwei zu wenig“, um genügend Zeit zu haben, die Software zu testen und auftretende Fehler zu beseitigen. Sie will sicher gehen, dass die Software läuft, und nimmt an, dass beim Kunden in der Flut neuer Informationen die technischen „Highlights“, die in der letzten Änderung hinzugekommen sind, untergehen.

- Mündliche Diskussion vs. Dokumentation von Entwicklungsentscheidungen

Der Wechsel zwischen Diskussions- und Umsetzungsphasen erfolgt, wenn die Konzepte diskutiert wurden und die Arbeit aufgeteilt ist. Während für Haubold und Wolf die mündliche Besprechung der Konzepte ausreicht, erscheinen Fricke und Krone diese Diskussionen als zu oberflächlich. Sie vermissen einen ausführlichen, dokumentierten Entwurf und das Festhalten von Entwurfsentscheidungen und etwaiger Änderungen. Krone wird mit ihren Bemühungen um schriftliche Dokumentation der Entwurfsentscheidungen nicht ernst genommen. Haubold hält ihr entgegen, dass Quelltext schon Prosa sei. Zudem gehe es darum, schnell einen Prototyp zu haben, den man auch wieder verwerfen könne. Fricke versucht, seine Kollegen durch selbstgeschriebene Tools zu mehr Schriftlichkeit zu bewegen, hat aber dabei auch keinen Erfolg.

- Zyklische Realisierung vs. Antizipation zukünftiger Anforderungen

Der Wechsel von der Umsetzungsphase zur erneuten Diskussion des Konzepts erfolgt, wenn die Implementierung so viele konzeptionelle Schwächen enthält, dass diese ein bestimmtes Maß übersteigen. Wolf beschreibt diesen Punkt als ein Gefühl, das er „aus dem Bauch heraus hat“. Haubold würde sich als Ausgleich zur Organisationsarbeit, die er erledigt, gern intensiver mit den Komponenten der Software beschäftigen und fühlt sich noch eher als Wolf zu Neukonzeptionen gedrängt. Zudem wird er durch Fehler, die er ignoriert oder umständlich umgeht, zusätzlich an

die längst fällige konzeptionelle Überarbeitung erinnert. Während für Haubold und Wolf nach der ersten Implementierung ein zweimaliges Re-Design normal ist, denken Fricke und Krone bei den Komponenten, die sie gemeinsam entwickeln, von vornherein über perspektivisch notwendige Funktionalität nach. Sie tendieren weniger zu weitreichenden konzeptionellen Änderungen im Verlauf eines Projekts. Sie sind allerdings von den konzeptionellen Änderungen der beiden anderen Entwickler an den Schnittstellen betroffen und dadurch teilweise gezwungen, ihre Implementierung zu überarbeiten.

- Allgemein verständliche vs. komplexe Konzepte

Vor gemeinsamen Diskussionen macht sich Wolf Gedanken darüber, warum er mit dem bestehenden Konzept unzufrieden ist. Seine Verbesserungsvorschläge trägt er in Diskussionen allgemeinverständlich vor. Fricke erarbeitet seine Lösungen eher im Alleingang. Er kann seine Programmierkonzepte nur auf einer abstrakten Ebene erklären, da sie sehr komplex sind. Dadurch kann er sich nicht verständlich machen und zieht so in Diskussionen den Kürzeren. Haubold versucht bei Diskussionen, alle von seiner Meinung zu überzeugen. Krone hält sich meist zurück. Sie stellt Verständnisfragen und versucht, mit ihren Beiträgen einen Ausgleich zwischen konfligierenden Positionen zu erreichen.

- Effizienz und Übersichtlichkeit vs. Kompaktheit und Komplexität

Unterschiedliche Vorgehensweisen mit großen Differenzen zwischen Haubold und Wolf einerseits und Fricke und Krone andererseits zeigen sich auch bei der Umsetzung der gemeinsam besprochenen Konzepte. Wolf strebt bei seiner Software eine einfache, übersichtliche und effiziente Konzeption an. Er legt Wert auf speicher- und rechenzeitsparende Implementationen, die änderungsfreundlich sind. Er strukturiert seinen Quelltext so, dass in den Basisklassen nur die Grundfunktionalität realisiert ist. Wenn diese perfekt funktioniert, wendet er sich komplizierteren Problemen zu, achtet aber darauf, dass diese in der obersten Schicht realisiert sind und nicht vererbt werden. Er realisiert Anforderungen erst dann, wenn sie in der Anwendung tatsächlich gebraucht werden. Hierdurch hat er zu einem relativ frühen Zeitpunkt eine stabil laufende Version. Wolf arbeitet am liebsten mit Haubold zusammen, da dieser ähnlich vorgeht. Haubolds Ziel ist es, in möglichst kurzer Zeit eine sinnfällige Lösung zu realisieren. Diese Lösung findet er nach zwei bis drei Überarbeitungen des ursprünglichen Konzepts. Stellt der Kunde weitere Anforderungen, hält Haubold das für nicht rechtmäßig. Haubold und Wolf arbeiten am liebsten ungestört an einer Aufgabe. Sie bevorzugen einen Wechsel zwischen Diskussionen und Stillarbeit nach aktuellem Bedarf.

Fricke dagegen ist Plattformunabhängigkeit und Fehlerrobustheit durch verteilte Realisierung der Softwarekomponenten wichtig. Er setzt gezielt die Vererbungsmechanismen der objektorientierten Sprache ein und implementiert bereits in den Basisklassen komplizierte Algorithmen und spezifische Funktionen. Dadurch läuft seine Version erst relativ spät stabil. Wolf erscheint diese Vorgehensweise riskant. Die komplizierten Vererbungsmechanismen sind für ihn zu undurchsichtig. Sylvia Krone sieht ihr Vorgehen als gegensätzlich insbesondere zu dem von Haubold und Wolf. Beim Testen und beim Erstellen der Dokumentation versetzt sie sich in

die Gedankenwelt der AnwenderInnen und bemängelt das mangelnde Problembewusstsein von Haubold in dieser Hinsicht.

Aus der Analyse der individuellen Arbeitspraktiken wird ersichtlich, dass das Entwicklerteam bei SuperTech in zwei Gruppen zerfällt. Die eine Gruppe bilden Haubold und Wolf. Ihr *Arbeitsstil* lässt sich als *erfahrungsgeleitet* beschreiben: Sie diskutieren Probleme mit Vorliebe mündlich und bevorzugen einen effizienten, übersichtlichen und wartungsfreundlichen Quelltext, dessen Basisklassen ausschließlich die Grundfunktionalität enthalten. Die erste Implementation überarbeiten sie in der Regel zwei Mal. In der anderen Gruppe befinden sich Fricke und Krone. Ihr *Arbeitsstil* lässt sich als *theoriegeleitet* beschreiben. Bevor sie anfangen zu programmieren, denken sie über die Anforderungen nach, die jetzt oder später zu realisieren sind, und versuchen dadurch den Aufwand für spätere Überarbeitungen zu reduzieren. Sie nutzen auch komplizierte Mechanismen der Programmiersprache und streben einen fehlerrobusten, kompakten Quelltext an. Beide halten Entwurf und Test der Software aus Anwendungssicht für notwendig.

Bei der Produktentwicklung waren diese unterschiedliche Arbeitsstile in der Firma kein Problem, da jede EntwicklerIn ihre eigenen Komponenten zu realisieren hatte und sie sich nur über die Schnittstellen verständigen mussten. Wenn ein einheitliches Vorgehen erforderlich ist, gibt es jedoch Schwierigkeiten. So kann im Auftragsprojekt die Aufgabe nicht vollständig in Komponenten zergliedert werden. Dies führt zu Konflikten unter den EntwicklerInnen über gemeinsam verwendete Dateien, die jede nach ihren Vorstellungen ändert, was zu Fehlern in anderen Programmteilen führt.

Im vorgestellten Fallbeispiel löst der Geschäftsführer solche Konflikte durch seine Autorität oder sie werden durch die strikte Trennung der Arbeitsbereiche vermieden. Zudem dominiert er mit seiner Vorgehensweise den kollektiven Arbeitsstil. Die Machtposition, die Haubold aufgrund seiner Funktion und aufgrund seiner Unterstützung durch Wolf hat, macht es einsichtig, warum sich der theoriegeleitete Arbeitsstil von Fricke und Krone in der Firma nicht durchsetzen kann. Beide finden weder mit ihren Bemühungen um mehr Schriftlichkeit und den Einsatz von Tools noch mit ihrem Engagement für eine partizipative Softwareentwicklung Gehör. Der Konflikt zwischen beiden konkurrierenden Arbeitsstilen endet mit der Kündigung von Fricke und Krone

4 Zusammenfassung

Die Vorgehensmodelle für das Projektmanagement beschreiben in ihrer Perspektivität, welche Handlungen planungsleitend sind. Daran wird deutlich, wie management-, entwicklungs- oder anwendungsnah im Unternehmen geplant wird. Im beschriebenen Fall handelt es sich um ein kleines Softwareunternehmen, das den Entwicklungsprozess nach einem impliziten entwicklungsorientierten Modell strukturiert. Dieses Vorgehensmodell hat jedoch kaum Bezug zum tatsächlichen Projektverlauf und zum Vorgehen der EntwicklerInnen. Der jeweilige Projektkontext lässt sich nach der Interaktionsstruktur zwischen AnwenderInnen und EntwicklerInnen typisieren. Mit der Firmengründung begann bei SuperTech eine Produktentwicklung, die zunehmend von Kundenentwicklungen abgelöst wurde. Dabei waren

auch die Kundenentwicklungen von einer Konzentration auf technische Aspekte gekennzeichnet. Die Vertragsentwicklung forderte eine stärkere Orientierung auf die Bedürfnisse der AnwenderInnen. Dabei kam es zu Konflikten mit den Kunden und im Entwicklungsteam. Der spezifische Verlauf der einzelnen Projekte und die dabei auftretenden Konflikte lassen sich durch die Analyse der individuellen Arbeitspraktiken erhellen. Die Konflikte im Team entstanden bei SuperTech entlang von zwei konkurrierenden Vorgehensweisen. Aufgrund der internen Machtstrukturen in der Firma können sich EntwicklerInnen mit einem theoriegeleiteten Arbeitsstil nicht gegen den erfahrungsgeliteten Arbeitsstil des Geschäftsführers durchsetzen.

Aus diesen drei Bausteinen - Vorgehensmodelle, Projekttypen, Arbeitspraktiken - ergibt sich das in Abb. 2 dargestellte Modell für eine realitätsnahe Beschreibung von Softwareentwicklung in kleinen Unternehmen.

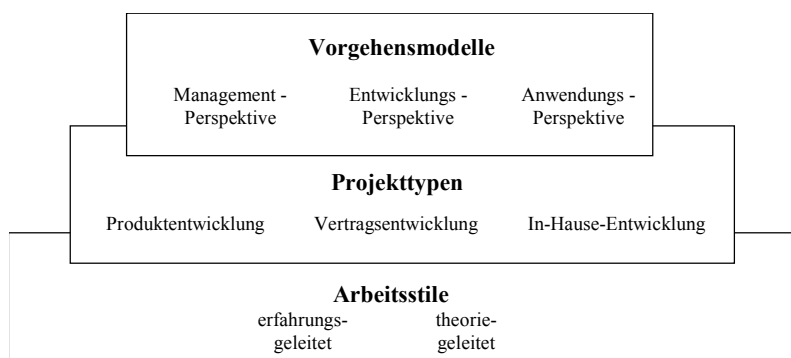


Abb. 2 Bausteine einer realitätsnahen Beschreibung von Softwareentwicklung

Bezüglich der Wechselwirkungen zwischen den drei Bausteinen (Vorgehensmodelle, Projekttypen, Arbeitspraktiken) leiten wir folgende empirisch begründete Hypothesen ab:

1. Die Sichtweisen auf den Entwicklungsprozess hängen von den Funktionen und von den Interessen der Personen ab.
2. Entwicklungsnah geplante Projekte führen zu einer Dominanz der entwicklungstechnischen Aspekte des Produktes.
3. Die Projekttypen bestimmen die Interaktionsstruktur zwischen Auftraggeber und -nehmer, die Arbeitspraktiken bestimmen den spezifischen Projektverlauf.
4. Unterschiede in den individuellen Arbeitspraktiken weisen auf konfliktrträgliche Situationen bei einem notwendig kollektiven Vorgehen im Entwicklungsteam hin.
5. Die kollektiven Arbeitspraktiken verdeutlichen die Machtverhältnisse im Entwicklungsteam.

Mit diesen drei Bausteinen konnte der konkrete Projektverlauf im Fallbeispiel erklärt werden. Bei der weiteren Untersuchung ist an anderen Fallbeispielen zu prüfen, ob sich weitere Typisierungen finden. Hinsichtlich der bisher identifizierten Arbeitsstile kommen wir zu folgenden Hypothesen:

- Erfahrungsgelitete Arbeitsstile bei der Programmierung führen zu einer einfacheren Programmstruktur als theoriegeleitete. Sie sind mit einer Vorliebe für Mündlichkeit und

starken Bezug auf momentan bekannte Anforderungen bei der Implementierung verbunden.

- Theoriegeleitete Programmierung führt zu kompakten Basisklassen und einer komplizierten Vererbungshierarchie. Sie sind mit einem zukunftsantizipierenden Entwurf verbunden, der die Perspektive der AnwenderInnen berücksichtigt und bis zur Lauffähigkeit mehr Zeit benötigt.

Unser Modell erlaubt zudem die Identifizierung von für den Projekterfolg günstigen bzw. ungünstigen Kombinationen zwischen den einzelnen Bausteinen. So erwies sich im untersuchten Fallbeispiel die Kombination aus Entwicklungsperspektive bei der Projektplanung, Vertragsentwicklung als Projekttyp und der Dominanz eines erfahrungsgeliteten Arbeitsstils im Entwicklungsteam als ausgesprochen konfliktträchtig. Die auf der Basis des Modells in Abb. 2 entwickelten Hypothesen ermöglichen damit Prognosen über den Projekterfolg, deren Übertragbarkeit in weiteren, auch quantitativ ausgerichteten Untersuchungen überprüft werden könnte.

Deutlich wird an diesem Fallbeispiel, dass die Flexibilität und Schnelligkeit kleiner Unternehmen bei der Softwareentwicklung wesentlich mit ihrer Größe und der dadurch möglichen Arbeitsweise im Zusammenhang steht. Zeitaufwendige Konsensbildungsprozesse werden durch die Machtverhältnisse im Entwicklungsteam vermieden. Zudem stellt sich im Laufe der Zeit eine Homogenisierung des Arbeitsstils ein, die eine effektive und konstruktive Zusammenarbeit ermöglicht. Fraglich bleibt, ob das Unternehmen damit auch perspektivisch überlebensfähig bleibt.

5 Literatur

- Bittner, Udo; Hesse, Wolfgang; Schnath, Johannes: Praxis der Softwareentwicklung: Methoden, Werkzeuge, Projektmanagement. Eine Bestandsaufnahme. München ; Wien : Oldenbourg 1995.
- Bremer, Georg: Genealogie von Entwicklungsschemata. In : Kneuper, Ralf; Müller – Luschnat, Günther; Oberweis, Andreas : Vorgehensmodelle für die betriebliche Anwendungsentwicklung. Stuttgart ; Leipzig : Teubner 1998, S. 32-59
- Deifel, Bernhard u. a.: Die Praxis der Softwareentwicklung: Eine Erhebung. Informatik-Spektrum, Jg. 22, H. 1, 1999, S. 24-36
- Grudin, Jonathan: The development of interactive systems: Bridging the gaps between developers and users. In : IEEE Computer, Jg. 24, H. 4, 1991, S. 59-69
- Hinz, Thomas: Betriebsgründungen in Ostdeutschland. Berlin : edition sigma, 1998.
- Kelle, Udo: Empirisch begründete Theoriebildung. Zur Logik und Methodologie interpretativer Sozialforschung. Weinheim : Deutscher Studienverlag, 1994.
- Strübing, Jörg: Subjektive Leistungen im Arbeitsprozeß: Eine empirische Untersuchung von Arbeitsstilen in der Programmierarbeit. Wiesbaden : Deutscher Universitäts-Verlag, 1993.
- Weltz, Friedrich; Ortman, Rolf G.: Das Softwareprojekt: Projektmanagement in der Praxis. Frankfurt/Main ; New York : Campus, 1992.

Stil als Konzept zur Analyse von Kommunikationsarbeit

Zur Konvergenz soziologischer und linguistischer Perspektiven

Jana Haase, Frank Kleemann, Ingo Matuschek,
Carsten Schulz-Nötzold

Abstract Communication is gaining a growing importance in the workplace. In many jobs communication is developing from a supporting factor of work to the central object of work activities. In this sense, ‚communication work‘ can be regarded as a distinct area of work. Call centers may be the most prominent example of communication work. In order to study this complex new area of work, a multidisciplinary approach is useful. In this paper we suggest analysing communication work on the basis of the ‚style‘ concept, which will allow us to draw on a diversity of methodological resources from the humanities. We will explore a whole range of different aspects of ‚style‘ and connect them to our research topic. This will lead to the concepts of ‚work style‘ and ‚communication style‘, which we will illustrate and develop further. The final question will be whether and how the two concepts derived from different disciplines can be productively related to each other.

1 Einleitung

Eine wichtige Tendenz in der Entwicklung der Arbeitswelt ist die Zunahme der Bedeutung von Kommunikation für den Arbeitsprozess. In immer höherem Maße wird Kommunikation von einem rein arbeitsbegleitenden Faktor zum eigentlichen Gegenstand der Tätigkeit. In diesem Sinne ist es gerechtfertigt, von ‚Kommunikationsarbeit‘ als eigenständigem Tätigkeitsfeld zu sprechen (Höflich 1996; Knoblauch 1996; Knoblauch/Heath 1999; Mikl-Horke 1994; Rammert 1992). Das wohl prominenteste Beispiel dafür ist der in den letzten Jahren entstandene Bereich der Call-Center-Arbeit (Bittner u.a. 2000; Riede/Matuschek/Anderson 2000; TIB 2000; die Beiträge in Matuschek/Henninger/Kleemann 2001). Zur Untersuchung dieser komplexen neuen Arbeitsform sind unterschiedliche disziplinäre Zugänge erforderlich. Hierzu stellt dieses Papier erste konzeptionelle Überlegungen an und besitzt deswegen noch einen stark explorativen Charakter.

Als eine geeignete Beschreibungskategorie wollen wir das Konzept ‚Stil‘ vorschlagen, da es mehrere Analysebereiche und -ebenen gleichzeitig erfassen kann (Mehrebenenphänomen). Es erlaubt den Rückgriff auf eine Vielzahl von methodischen Ressourcen aus Kultur- und Sozialwissenschaften. Wir werden im folgenden zunächst einen kurzen systematisierenden

Überblick über die relevanten Forschungsstränge zum ‚Stil‘-Begriff geben (Kap. 2) und diese anschließend auf unseren Forschungsgegenstand engführen (Kap. 3). Die Konzepte des ‚Arbeitsstils‘ und ‚Kommunikationsstils‘ werden anhand eines Fallbeispiels entwickelt und vorgestellt (Kap. 4). Damit soll gezeigt werden, dass die beiden Konzepte komplementär aufeinander bezogen werden können. Dadurch wird eine ganzheitlichere Erfassung des Untersuchungsgegenstands, ‚Kommunikationsarbeit‘ ermöglicht (Kap. 5).

2 ‚Stil‘ als Analysekategorie

2.1 Konzeptionen von ‚Stil‘

Im Laufe seiner 2500 Jahre währenden historischen Entwicklung hat der Begriff ‚Stil‘ einen enormen Bedeutungsumfang gewonnen und ist in entsprechend vielfältige Verwendungsbereiche vorgedrungen. Die unterschiedlichsten wissenschaftlichen Traditionen – von der antiken Rhetorik angefangen – haben darauf Einfluss genommen (vgl. im Überblick Heinz 1986; Sowinski 1999). Daraus resultieren uneinheitliche Gebrauchsweisen des Begriffes und eine große Reichhaltigkeit an Methoden und Perspektiven der Stilanalyse. Für die Kultur- und Sozialwissenschaften stellt der Begriff ‚Stil‘ mittlerweile eine äußerst komplexe Analysekategorie dar. Man geht davon aus, dass es sich bei ‚Stil‘ um ein sog. Mehrebenenphänomen handelt. D.h. ‚Stile‘ können auf unterschiedlichste analytische Gegenstände und Analyseebenen bezogen werden: z. B. Baustile, Denkstile, Epochenstile, Fahrstile, Musikstile, Sprachstile, Verhaltensstile usw. – die Aufzählung ließe sich noch lange fortsetzen. Im folgenden ist nur ein kurzer Abriss über ‚Stil‘-Konzepte ausgewählter Disziplinen möglich, die in bezug auf unseren Gegenstandsbereich der ‚Kommunikationsarbeit‘ besondere Relevanz erlangen: Linguistik [A], Wirtschafts- und Arbeitswissenschaft [B] und Soziologie [C].¹

[A] Zunächst soll es um den Bereich *Linguistik* gehen, wobei allerdings nur drei Ansätze aus der pragmatischen, d.h. handlungstheoretisch orientierten Text- und Gesprächslinguistik referiert werden – womit zugleich angedeutet ist, von welcher Art die weiter unten zu entwickelnde Konzeption ‚Kommunikationsstil‘ sein wird.²

Der *Variationsansatz* definiert ‚Stil‘ i.w.S. als Art und Weise der Handlungsdurchführung und sinnhaften Wahl aus dem kommunikativen Repertoire (Hymes 1979, S. 167; Sandig 1986, S. 23).³ ‚Stil‘ ist somit die Gestalt (Eigenschaft) von Kommunikation, die das (nicht kontrollierbare) Ergebnis von angemessenen und erfolgreichen sprachlichen (Gestaltungs-)Handlungen ist. Gestaltungsleitende Faktoren können sein: Zeit, Gruppe, Normen, Person, Rollen, Textsorte,

¹ Für die Bereiche Rhetorik und Literaturwissenschaft, Kunst-, Musik- und Architekturwissenschaft sei aus Platzgründen nur auf wenige Überblicksdarstellungen verwiesen (vgl. Heinz 1986; Sowinski 1999; Panofsky 1996; 1999).

² Zu anderen linguistischen Konzepten wie ‚Stilstatistik‘, ‚Funktionalstilistik‘, ‚Strukturalistische Stilistik‘, ‚Generative Stilistik‘ u.a. vgl. den Überblick von Sowinski (1999, S. 31-45) und Püschel (2000).

³ Kommunikatives Repertoire bezeichnet die Menge aller kommunikativen Formen und Verfahren, die einer sozialen Gruppe während der Interaktion zur Verfügung steht (Keppler 1994, S. 23; Gal 1987, S. 286f.).

Medium etc. (Püschel 1995). ‚Stil‘ wird danach als ein Mittel der Situationsanpassung von Handlungen und Beziehungsgestaltung (z. B. Anpassung an Rollen) verstanden, das auf intersubjektiv geltende und erwartbare Typen ‚stilistischen Sinns‘ gegründet ist (Sandig 1986, S. 239; 1995, S. 32; Dittmar 1997, S. 223).

Der *ethnographische Ansatz* der Soziolinguistik versteht ‚Stil‘ als die sprachliche Symbolisierung sozialer Identität (Kallmeyer 1995; Dittmar 1997, S. 222). ‚Sozialer kommunikativer Stil‘ umfasst die kulturspezifischen Kommunikationsformen einer Gruppe und die ihnen zugrunde liegenden Normen und Konventionen. ‚Stil‘ beschreibt hier die distinktive Form des Ausdrucksverhaltens und erlangt damit soziale Bedeutung für die Selbst- und Fremdwahrnehmung von Gruppen. ‚Sozialer Stil‘ wird von den Mitgliedern der Gruppe aktiv hergestellt (kulturelle Arbeit) und prägt wiederum die Mitglieder der Gruppe (Milieu), so dass von einem dynamischen Prozess der Stilbildung gesprochen werden kann.

Die sog. *Interaktionale Stilistik* betont die aktive und sinnherstellende Bedeutung und Funktion von ‚Stil‘⁴ und ‚Stilisierung‘⁵ im Rahmen sozialer Handlungszusammenhänge (Auer/di Luzio 1984; Franck 1984; Tannen 1984; Selting 1989; Selting/Hinnenkamp 1989; Selting/Sandig 1997) und entwickelte sich aus der Kritik an statischen Stilkonzepten, wie sie u.a. von Labov (1976) entwickelt wurden. Sie fragt nach dem für die Beteiligten sinnvollen, d.h. bedeutsamen Gebrauch konkurrierender sprachlicher Gestaltungs- und Ausdrucksmittel im Vergleich zu paradigmatischen Alternativen in der sich entwickelnden Interaktionssituation. ‚Stil‘ gilt hier als ein interpretiertes und an Rezipientenkategorien orientiertes holistisches kommunikatives Zeichen, das Ressourcen und Interpretationsrahmen für Handlungen bereit stellt. Von ‚Stil‘ wird bei diesem Ansatz erst gesprochen, wenn Strukturmerkmale nicht nur zu einem gewissen Grad harmonieren, sondern auch zusammen oder jeweils individuell von den Mitgliedern einer Kultur- bzw. Sprechgemeinschaft konsistent interpretiert werden (Auer 1989).

[B] Außerhalb dieser auf Kommunikation/Sprache bezogenen Kernbereiche wird der ‚Stil‘-Begriff auch in den *Wirtschafts- und Arbeitswissenschaften* seit längerem verwendet, vor allem im Kontext verhaltenstheoretischer Untersuchungen zum ‚Führungsstil‘. Diese Studien tendierten allerdings dazu, Situationsfaktoren nur selektiv zu berücksichtigen und die Verhaltensflexibilität der Beteiligten zu überschätzen. Neuere Ansätze dagegen fußen auf einer Metatheorie sozialer Konstruktionen: die Beteiligten werden als bewusst handelnde Akteure innerhalb eines Interaktionsprozesses gesehen, in dem Führung beiderseitig konstruiert wird. Diese Situation ist prinzipiell für die Beteiligten mehrdeutig und insofern komplex und dynamisch angelegt. Der ‚Führungsstil‘ wird entlang der subjektiven Wirklichkeitskonstruktionen mit den daraus folgenden

⁴ Der ‚Stil‘-Begriff wird dabei vom Begriff ‚Variation‘ als Eigenschaft sprachlicher Daten unterschieden, die ohne Rekurs auf die Verstehensleistungen der Interaktionsteilnehmer festgestellt werden kann. ‚Variation‘ wird dann nicht zu ‚Stil‘, wenn die beobachteten Variationsphänomene, nicht nur einzeln, sondern auch in ihrem Zusammenwirken, von den Teilnehmern unbemerkt und/oder von ihnen uninterpretiert bleiben; d.h. also, wenn sie nicht dazu verwendet werden, dem Sprecher bestimmte soziale Prädikate zuzuordnen oder um ihn als Individuum zu charakterisieren. Es gibt linguistische ‚Variation‘ als solche, aber ‚Stil‘ immer nur in Beziehung zu einem interpretierenden Teilnehmer der Kultur und in Beziehung zu einem Anderen.

⁵ ‚Stilisierung‘ meint hier die Repräsentation, Inszenierung etc. sozial typisierter und interpretierter Sinnfiguren in der Interaktion.

Erwartungen ausgehandelt (von Lang 2000). Aus arbeits- und organisationspsychologischer Perspektive liegen ebenfalls Beiträge zu ‚Denk- und Führungsstilen‘ in der Arbeitswelt vor (Schultz-Gambard/Altschuh 1993). Danach korrespondieren die ‚Stile‘ mit dem Ausbildungsstand bzw. dem innerbetrieblichen Status.

[C] In der gegenwärtigen (deutschsprachigen) *Soziologie* findet der Begriff ‚Stil‘ trotz einer langen Tradition des Konzepts nur in Teilbereichen, dort aber sehr intensive Verwendung. Allgemein fasst man ‚Stil‘ hier als einen symbolvermittelten Verweisungszusammenhang, d.h. als „Ausdruck, Instrument und Ergebnis sozialer Orientierung“ (Soeffner 1986). Die damit angesprochene doppelte Form – einerseits Instrument, andererseits Ergebnis – verweist auf ein komplexes Wechselverhältnis von *Stilbildung* und *Stilbindung*.

Diese komplementäre Konzeption gründet in dem bereits durch Simmel (1989) und Weber (1984) thematisierten Spannungsbogen zwischen einem Leben gemäß ständischer oder klassenbezogener Konvention und einem individuell ausgebildeten ‚Stil‘ des eigenen Lebens (Luger 1992). Zwischen diesen Polen bewegen sich auch die Forschungsansätze, die seit den achtziger Jahren in unterschiedlicher Form den Begriff ‚Lebensstil‘ aufnehmen.⁶ Je nach theoretischem Bezug stehen sozialstrukturelle Restriktionen, milieuspezifische Unterschiede oder individuelle Alltagspraktiken im Mittelpunkt. Mehr oder weniger erscheint ‚Lebensstil‘ dann entweder als Ausdruck objektiver (sozioökonomischer) Gegebenheiten oder als kulturelle Überformung des Alltäglichen (Berger/Hradil 1990; Hradil 1992; Müller 1992; Drieseberg 1995; Lüdtke 1989; 1994; 1995; Schwenk 1996). Eine dritte Position betont dagegen die Eigenwirksamkeit lebensstilspezifischer Prozesse (Hörning/Ahrens/Gerhard 1996). Sozialstrukturelle bzw. kulturelle Faktoren spielen in diesem Konzept durchaus eine Rolle, sind aber (zugunsten einer Dynamisierung des ‚Stil‘-Konzeptes selbst) nur mittelbar wirksam. Angesichts der zunehmenden Komplexität moderner Gesellschaften und damit erhöhter Kontingenz geht man von der Notwendigkeit aus, diesen Kontingenzen durch eine permanente Prozessierung des ‚Stils‘ zu begegnen. Dies ermöglicht erst die unerlässliche Integration und Abgrenzung des Subjekts in bzw. von der Gesellschaft. Der ‚Lebensstil‘ ist in diesem Sinne nicht substantialistisch im Wesen des Menschen zu verorten, sondern muss mit dem Ziel der Vergesellschaftung immer wieder neu hergestellt werden.

Aus- und Abgrenzungen sind auch das Thema der ‚Stil‘-Diskussionen in der soziologischen *Jugendforschung*. Dort werden ‚Stile‘ im Sinne von ‚Stilisierungen‘ diskutiert, mit denen Jugendliche im Zuge ihrer sozialen Persönlichkeitsentwicklung die Zugehörigkeit zu bestimmten Gruppen verwirklichen. Oftmals an besonders auffälligen ‚Stilisierungen‘ orientiert, ist man sich einig, dass Jugendliche sich permanent Elemente der etablierten Kultur selektiv aneignen und sie individuell variieren. Betont wird dabei der an unmittelbaren Erfahrungen ausgerichtete Bastelcharakter solcher ‚Stilbildungen‘. Jugendliche bringen ‚Stilmodifikationen‘ hervor, indem sie permanent neue Elemente in ihren ‚Stil‘ aufnehmen und ihn dadurch verändern. Das geschieht weniger geplant als im Modus des Erprobens und Nutzens von Gelegenheiten. Ob und wenn ja, wie diese

Gelegenheiten allerdings wahrgenommen werden, ist milieuspezifisch geprägt (Soeffner 1986; Schäffer 1996; Ferchhoff/Neubauer 1996).

Kulturosoziologische Konzeptionen gehen davon aus, dass ein ‚Stil‘ ein Bündel charakteristischer Merkmale des Handelns sei, die nicht auf die manifesten Ziele von Aktivitäten oder auf ausdrückliche Verhaltensregeln reduzierbar sind (Hahn 1986). ‚Stile‘ sind demnach mehr als funktional oder instrumentell auf ein Ergebnis ausgerichtetes Handeln. Sie sind vielmehr Ausdruck für die Disposition, Handlungen in einer bestimmten Art und Weise auszuführen, also vorhandene Ressourcen verschiedenen Situationen angepasst immer wieder zu verwenden. Eine Trennung nach unterschiedlichen Lebenssphären – hier Arbeit, dort Leben – erscheint daher wenig sinnvoll.

Im Bereich der *Arbeitssoziologie* ist ‚Stil‘ bisher mit Ausnahme der Arbeit von Strübing (1993) zu ‚Arbeitsstilen‘ in der Programmierarbeit kein Thema. Hier wird ein ‚Stil‘-Begriff angewandt, der den ‚Programmierstil‘ als von subjektiven Dispositionen geprägt sieht. In den Blick geraten aber auch die materiellen und organisatorischen Bedingungen der Programmierarbeit. Dem Arbeitshandeln liegt eine je individuelle Interpretation der Organisationsziele zugrunde. Aufgrund des einseitigen Bezugs auf die Programmiertätigkeit konnten distinkte Tätigkeitsmerkmale allerdings nur als expressive ‚Stilisierungen‘ identifiziert werden.

2.2 Systematisierung der Stilkonzeptionen

Die angeführten ‚Stil‘-Konzeptionen unterscheiden sich nicht nur in ihrem Gegenstandsbezug, sondern auch in der Auffassung davon, wie ‚Stil‘ empirisch zu fassen sei. Entlang zweier Stränge – einerseits der Rezeption vorhandener Konzeptionen und andererseits unseres empirischen Materials – haben wir grundlegende Anforderungen an unsere Konzepte herausgearbeitet. Um konstitutive Merkmale von ‚Stil‘-Begriffen aufzuzeigen, die dem Gegenstand der ‚Kommunikationsarbeit‘ angemessen sind, formulieren wir im folgenden *formale* Kriterien anhand paradigmatischer Bestimmungen der oben angeführten Wissenschaftsdisziplinen:

- Die meisten Ansätze sind als analytische Konzepte empirisch-deskriptiv ausgelegt. In der Wirtschafts- und Arbeitswissenschaft dagegen werden Begriffe des ‚Führungs- bzw. Managementstils‘ auch präskriptiv verwendet und erhalten somit eine normative Ausrichtung.
- ‚Stile‘ werden unterschiedlichen Trägern zugeordnet: Während insbesondere die Literatur-, Kunst- und Musikwissenschaften primär gegenständliche Stilkonzeptionen verwenden, bei denen der ‚Stil‘ im Werk verortet wird, rechnen die Sprach- und Sozialwissenschaften den ‚Stil‘ der Person (bzw. sozialen Gruppen) zu. Damit lassen sich formbezogene von performativen Konzeptionen unterscheiden. In letzteren ist die Möglichkeit angelegt, dass der ‚Stil‘ einer Person sich im Zeitverlauf wandelt, also variabel ist.
- Dem ‚Stil‘ wird in unterschiedlichen Konzeptionen entweder überformender oder holistischer Objektstatus beigemessen. Überformend wird ‚Stil‘ als zusätzliche Komponente im Sinne der

⁶ Auf eine Auseinandersetzung mit dem im Rahmen der klassentheoretisch orientierten Kulturosoziologie

symbolischen Überhöhung eines ansonsten funktionalen Zusammenhangs betrachtet. Demgegenüber beachten holistische Konzeptionen von ‚Stil‘ auch die Art und Weise der Verbindung von Elementen zu einem kohärenten Ganzen.⁷

- Zu unterscheiden sind ferner intentionale von präreflexiven Konzeptionen der ‚Stilgenerierung‘: Während sprachliche Hervorbringungen, ‚Führungsstile‘ und expressive Handlungen mit der Zielsetzung sozialer Distinktion vor allem unter dem Aspekt ihrer zielgerichteten Hervorbringung analysiert werden, betrachten die übrigen Ansätze den ‚Stil‘ als überwiegend präreflexiv zustande gekommene Gestaltung der eigenen Praxis.

3 Personaler Arbeitsstil und Kommunikationsstil: Verbindungslinien

Mit den genannten Unterscheidungen lassen sich für unseren Gegenstand, Arbeits- und Kommunikationshandeln in ‚subjektivierten Arbeitsformen‘, bereits einige wesentliche und zur Analyse erforderliche Eigenschaften der Konzepte ‚personaler Arbeitsstil‘ [A] und ‚Kommunikationsstil‘ [B] bestimmen. Zwischen beiden Konzeptionen bestehen enge Verbindungslinien [C].

[A] ‚Kommunikationsarbeit‘ unterliegt dem Prozess der ‚Subjektivierung‘ von Arbeit (vgl. Kleemann/Matuschek/Voß 1999, 2001). Damit werden individuelle Herstellungsleistungen für Arbeitshandlungen wichtiger.⁸ Zu deren Erfassung bietet sich ein (1) empirisch-deskriptiv und (2) personen- sowie performanzbezogen vorgehendes Konzept an. Der ‚Arbeitsstil‘ beschreibt dann (3) holistische Muster bzw. Strukturierungen des Arbeitsalltags sowie den dahinter stehenden *modus operandi*, nicht aber eine individuell-stilistische Überhöhung funktionaler Handlungen. Soweit er überwiegend auf der (4) präreflexiven Ebene von Praktiken zu verorten ist, sind ‚Stil‘-Spezifika nur in Relation zu (berufs-)biographischen Hintergründen, subjektiven Sinndeutungsmustern und situativen Rahmenbedingungen des Arbeitens und Lebens, die eine Veränderung der Handlungsweise erlauben oder erzwingen, erkennbar.

Der Begriff des ‚Stils‘ scheint uns deshalb sinnvoll auf den Gegenstand ‚subjektivierter Arbeit‘ anzuwenden zu sein, weil ‚Stil‘-Konzepte verschiedener Provenienz stets Muster bzw. Ähnlichkeiten erfassen, die für eine Mehrzahl (und nur dafür) von Handlungen bzw. Gegenständen des selben Typus verwendbar sind. Ohne dass damit deterministische Annahmen verbunden sind, wird der Aspekt der Wiederholbarkeit und Regelmäßigkeit betont. Das ‚Stil‘-Konzept erfasst dann über die Einzelhandlungen hinweg zunächst die dahinter erkennbaren

entwickelten Habitus-Konzept von Bourdieu (1987) soll hier verzichtet werden (vgl. zur Debatte Eder 1989).

⁷ Ein methodischer Hinweis: Auch bei holistischen Konzepten werden singuläre ‚Stile‘ nicht immer anhand gleicher allgemeiner Kriterien analysiert, sondern es sind jeweils charakteristische Merkmale in der Art ihres Zusammenwirkens, die einen bestimmten ‚Stil‘ kennzeichnen.

⁸ Diese Herstellungsleistungen gehen nicht allein aus der Situation hervor, sondern die Arbeitenden handeln aufgrund individueller, in früheren Situationen angeeigneter und eingeübter (routinierter) Verfahrensweisen. Nur in unbekanntem Situationen müssen neue Verfahren entwickelt werden (Giddens 1988; Joas 1992).

durchgängigen Muster, die von der arbeitenden Person hervorgebracht werden. Im weiteren richtet sich der Blick auf die dem ‚Arbeitsstil‘ zugrundeliegenden Methoden seiner Herstellung. Zugleich impliziert der ‚Stil‘-Begriff, dass die zentralen Merkmale von ‚Stilen‘ nicht immer die gleichen sind (nicht für jeden ‚Arbeitsstil‘ von Personen sind z. B. Milieuprägungen ein bestimmendes Moment). Theoretisch sind eine Vielzahl je unterschiedlich zu gewichtender Einflussfaktoren vorstellbar, die wechselseitig wirksam sind. Im Einzelfall wird es aber nie die Gesamtheit dieser Einzelfaktoren sein, die den ‚Stil‘ bedingen. Daraus ergibt sich die Frage, wie ein ‚Stil‘ erkennbar wird. In ‚subjektivierten Arbeitsformen‘ wird dies im Kontrast zwischen verschiedenen Arbeitenden deutlich, die gleiche Aufgaben auf unter Umständen deutlich unterschiedliche Art und Weise strukturieren. Insoweit sollte das Konzept des ‚personalen Arbeitsstils‘ (Kleemann 2000; Kleemann/Matuschek 2001; Matuschek/Kleemann/Voß 2001) ein formales Analysegerüst sein, dessen Ebenen empirisch dafür offen sind, unterschiedliche individuelle Methoden und Verfahren zu erfassen, Arbeitshandeln zu koordinieren. Auch insofern erweist sich der ‚Stil‘-Begriff als geeignet, da es sich bei ‚Stil‘ stets um ein relationales Konzept handelt: Erst im Kontrast zu anderen ‚Stilen‘ wird ein ‚Stil‘ beschreibbar. Damit verbietet sich methodisch eine den zu untersuchenden ‚Stilen‘ äußerlich bleibende Dimensionierung, wie sie zum Beispiel der sozialstrukturell orientierten ‚Lebensstil‘-Forschung zu eigen ist. In methodologischer Hinsicht erfordert das ‚Stil‘-Konzept sowohl die fallimmanente wie die fallübergreifende Kontrastierung. Erst in den solchermaßen rekonstruierten⁹ ‚Stil‘-Komponenten und ihrer Zusammensetzung lässt sich erkennen, was den bereichsspezifischen ‚Stil‘ einer Person ausmacht und was ihn von anderen bereichsspezifischen ‚Stilen‘ derselben Person unterscheidet. Im Zuge der fallübergreifenden Kontrastierung wird zudem die individualistisch-reduzierte Variante kulturalistischer ‚Stil‘-Konzeptionen überwunden.

[B] Das hier vertretene Konzept von ‚Kommunikationsstil‘ wurde im DFG-Projekt „Die sprachliche Aneignung von Computermedien“¹⁰ im Rahmen der DFG-Forschergruppe „Neue Medien im Alltag“ an der Technischen Universität Chemnitz entwickelt. Über die Analyse der kommunikativen Praktiken während der Nutzung von Computermedien, die mit Audio- und Videogeräten dokumentiert wurden, wollen wir rekonstruieren, wie der Umgang mit dieser Technik erlebt, gedeutet und bewertet wird. Uns interessiert dabei unter anderem, wie die sozialen Herausforderungen durch den Computer, vor allem in der Arbeitswelt, sprachlich-kommunikativ bewältigt werden. Wir fragen danach, welches Repertoire an kommunikativen Mustern den Computernutzern zur Verfügung stehen, wenn sie z. B. Aufgaben bearbeiten und Probleme lösen, Wissen vermitteln oder Arbeitsabläufe koordinieren. Weiterhin fragen wir danach, welche

⁹ Vgl. zur Rekonstruktiven Sozialforschung allgemein Bohnsack (1999); speziell für Leitfadeninterviews Matuschek (1999).

¹⁰ Vgl. zu Konzeption und Ergebnissen dieses Projektes die vorliegenden Publikationen (Haase/Stöckl 1998; Habscheid 1998; 2000; 2001; Habscheid/Holly 2001; Holly 1996; 1997; 2000; Holly/Biere 1998; Holly/Habscheid 1999; 2000; Klemm/Graner 2000) und Arbeitspapiere (Haase 1999; 2000a-c; Habscheid/Haase/Schulz-Nötzold 2000; Schulz-Nötzold 2000a,b).

Konzeptualisierungen des Rechnersystems, der Benutzeroberfläche etc. hierbei im Sprachgebrauch der Nutzer sedimentiert sind (Technikbilder).

Der Begriff ‚Stil‘ gilt uns in diesem Projekt als ein heuristisches Instrumentarium zur empirischen Erforschung und Beschreibung der Vielfalt und Differenz empirischen, d.h. arbeitsbezogenen Sprechens (Karl Bühler) in computerunterstützten Arbeitssettings. Dieses kann als ein Teilbereich von ‚Kommunikationsarbeit‘ verstanden werden. Unter Rückgriff auf den ‚Muster‘-Begriff sollen individuelle Handlungen zu Handlungstypen zugeordnet werden (Repertoire). Hierbei ist die situationsspezifische Auswahl an Mustern aus diesem Repertoire als erster Aspekt des ‚Stilistischen‘ zu berücksichtigen. Des Weiteren gilt es, die Variationsbreite, d.h. die unterschiedliche situative Ausprägung vergleichbarer ‚Muster‘, zu erfassen. ‚Stil‘ bezeichnet hier also die Art und Weise der ‚Musterrealisierung‘, deren kommunikative Funktionalität in der Anpassungsfähigkeit an unterschiedliche Sprechanlässe liegt. Nach unserem Verständnis können wir so mit dem Begriff ‚Stil‘ sowohl die Gestalt (Eigenschaft) von Kommunikation als auch den dynamischen Prozess seiner Hervorbringung durch die Interaktanten beschreiben. Ziel der ‚stilistischen‘ Analyse im genannten Projekt ist eine Verdichtung der ‚Stilmerkmale‘ des empirischen Sprechens von Computernutzern zu ‚sozialen Stiltypen‘ des kommunikativen Nutzerverhaltens (Gruppenspezifika).

[C] Verbindungslinien zwischen den hier herausgearbeiteten Konzeptionen des ‚personalen Arbeitsstils‘ und des ‚Kommunikationsstils‘ betreffen zum einen das wissenschaftliche Erkenntnisinteresse und zum anderen die methodische Vorgehensweise bei der Analyse.

- Erkenntnisinteresse:

Beide Entwürfe haben zum Ziel, die Differenz von (Sprach-)Handlungs- und Orientierungsmustern der Arbeitsgestaltung im Hinblick auf ihre die Komplexität reduzierende Funktion zu beschreiben. Im Mittelpunkt des Interesses stehen vor allem die Ethnomethoden (Garfinkel 1967) der Problembewältigung in Arbeitskontexten, insbesondere im Spannungsfeld von Routinisierung und Kreativität in „Krisen habituellen Handelns“ (vgl. Joas 1992). Ein Unterschied besteht lediglich im jeweils fokussierten Untersuchungsbereich (Muster des Arbeitshandelns vs. Muster arbeitsbezogener Kommunikation) und in der jeweiligen Datengrundlage (Leitfadeninterview vs. Audioaufnahmen authentischer Gespräche). Trotzdem beschäftigen sich beide Konzeptionen mit dem Prozess der aktiven Hervorbringung von Handlungen (Reproduktion und situationsbezogene Aktualisierung), aber natürlich – aufgrund der unterschiedlichen Datenbasis – aus unterschiedlichen Perspektiven (typische Beschreibungen der Beteiligten vs. Vollzugswirklichkeit). Gerade das wird von uns als ein analytischer Gewinn betrachtet, der uns dem Ziel einer ganzheitlicheren Betrachtung des Phänomens ‚Kommunikationsarbeit‘ ein Stück näher bringt.

- Methode:

Beide Entwürfe legen einen ‚Stil‘-Begriff zugrunde, der sowohl empirisch-deskriptiv als auch holistisch ausgerichtet ist. Analytisch gesehen wird ‚Stil‘ als ein relationales Phänomen betrachtet

(immanenter und fallübergreifender Vergleich), dass sich auf unterschiedlichen Ebenen der sozialen Wirklichkeit entfaltet (Mehrebenenphänomen). Ein ‚Stil‘ ist eine individuelle bzw. gruppenspezifische Handlungsweise, die als solche das Resultat einer sozialen Praxis darstellt. Deshalb sind ‚Stile‘ nach unserer Auffassung wesentlich durch die analytischen Dimensionen Situativität, Sozialität, Sinn und Biographie geprägt.

Die Zusammenschau der eben aufgezeigten Verbindungslinien führte uns zu der forschungsleitenden Vermutung, dass wir bei der Analyse von ‚Kommunikationsarbeit‘ mit dem Konzept ‚Stil‘ auf charakteristische Ähnlichkeiten zwischen den Bereichen ‚Arbeitsstil‘ und ‚Kommunikationsstil‘ stoßen könnten. Geht man davon aus, dass alle (Sprach-)Handlungen und handlungsleitenden Orientierungen mittels ‚indem-Relationen‘ (vgl. Heringer 1974) hierarchisch strukturiert sind, dann ließe sich die Hypothese formulieren, dass sich einerseits Merkmale von ‚Arbeitsstil‘ im ‚Kommunikationsstil‘ manifestieren und andererseits Merkmale von ‚Kommunikationsstil‘ den ‚Arbeitsstil‘ prägen. Die Frage lautet also: Gibt es spezifische Verknüpfungen zwischen den relevanten konstituierenden Merkmalen von ‚Arbeitsstil‘ und von ‚Kommunikationsstil‘ bzw. umgekehrt? Zu klären wäre dann allerdings ebenfalls, ob sich so etwas wie ein übergeordnetes und sinngebendes ‚stilgenerierendes Drittes‘ erkennen und aufzeigen ließe. Allerdings sind unsere gemeinsamen Analysen noch nicht soweit vorangeschritten, dass sich hierüber gesicherte Aussagen machen lassen. Das wollen wir aber künftig weiter verfolgen und präzisieren. Das folgende Fallbeispiel soll einen kurzen Einblick in diese Überlegungen vermitteln.

4 Ein Fallbeispiel aus dem Bereich Call Center

4.1 Feldbeschreibung

Die hier zugrunde liegenden Daten wurden in den Jahren 1999 und 2000 in einem Call Center einer Direktbank erhoben, und zwar im Rahmen des DFG-Projektes „Neue Arbeits- und Alltagspraktiken bei medienvermittelten autonomisierten Arbeitsformen“ innerhalb der DFG-Forschergruppe „Neue Medien im Alltag“ an der Technischen Universität Chemnitz (vgl. Voß 1999). Sie umfassen neun Interviews mit Call Center-Agenten und fünf Managern der unteren bzw. mittleren Ebene. Hinzu kommen 89 Gesprächsaufzeichnungen von authentischen Beratungsgesprächen einiger der interviewten Call Center-Agenten. Diese Gesprächsaufzeichnungen werden gewöhnlich im Rahmen der Schulungen der Direktbank genutzt und uns seitens der Institution dankenswerterweise zur Verfügung gestellt. Zusätzlich hatten die Projektmitglieder die Gelegenheit, umfangreiche Beobachtungen, auch während laufender Beratungsgespräche, durchzuführen. Als Kontrast zum untersuchten Call Center konnten bisher Beobachtungen in einem weiteren Call Center in der Finanzdienstleistungsbranche sowie in einem selbständigen Call Center-Dienstleister herangezogen werden. In die vorliegende Abhandlung fließen aus methodologischen Gründen allerdings nur die Datensätze aus dem erstgenannten Call Center ein.

4.2 Falldarstellung und Analyse

4.2.1 Zum personalen Arbeitsstil von Herrn K.

- Allgemeine Angaben

Herr K. ist 28 Jahre alt, verheiratet und hat ein halbjähriges Kind. Er ist als Call Center-Agent in der schon erwähnten Direktbank angestellt. Dort arbeitet er im Beschwerdemanagement.

- Situativität

Herr K. ist als vollzeitbeschäftigter Angestellter der Direktbank eingebunden in ein Gleitschichtsystem, das es der Bank erlaubt, ihren Service an sieben Tagen der Woche von 7 bis 22 Uhr anzubieten. Damit ist er bei einer ca. 40-stündigen Arbeitswoche in regelmäßigen Abständen auch zur Arbeit am Wochenende verpflichtet. Call Center-Agenten dieser Bank arbeiten in Teams von bis zu zehn Personen, die von einem Teamleiter geführt werden. Diese Teams arbeiten relativ autonom, d.h. im Rahmen personalwirtschaftlicher Vorgaben sind individuelle Gestaltungen der Arbeitszeit möglich. Die Arbeitsleistung wird prinzipiell teambezogen erfasst, es besteht allerdings die Möglichkeit, personenbezogene Leistungsdaten zu erheben. Die Teamleiter unterstützen und kontrollieren das Kommunikationsverhalten ihrer Teammitglieder; sie fungieren als unteres Management.

Die Aufgabe von Herrn K. ist es, kundenseitige Beschwerden und Anfragen entgegenzunehmen und, soweit es ihm möglich ist, zu lösen oder entsprechende Schritte einzuleiten. Dazu erhält er zu Arbeitsbeginn eingegangene Fälle übertragen, die er selbst sichten, priorisieren und anschließend im Laufe des Tages (gegebenenfalls auch längerfristig) bearbeiten muss. Dazu stehen ihm verschiedene Wege technisch vermittelter Kommunikation zur Verfügung. Quantitativ gesehen steht die Telefonie an erster Stelle, gefolgt von der E-mail-Bearbeitung. Die Hälfte seiner Arbeitszeit hat Herr K. Telefonbereitschaft, d.h. er muss eingehende mündliche Beschwerden ad hoc bearbeiten. Dazu gehört auch die Aufgabe, den Verlauf und das Ergebnis dieser Gespräche zu dokumentieren und wenn nötig weitere Schritte zu veranlassen. Die zur Verfügung stehende Technik wird als handlungsentlastend begrüßt, etwaige Mängel durch Improvisation umgangen.

Bei seinem umfangreichen und eigenverantwortlich zu gestaltenden Aufgabenbereich legt Herr K. großen Wert auf eine gute Planung. Dafür nimmt er es auch in Kauf, regelmäßig mindestens eine halbe Stunde vor dem eigentlichen Schichtbeginn im Hause zu sein, um genügend Zeit für die Planung zu haben. Die Telefonbereitschaft bzw. die dann eingehenden Gespräche führen zu Unterbrechungen seiner schriftlichen Sachbearbeitungen. Im Eigentlichen sind es diese Unterbrechungen, die aus der Perspektive Herrn K.'s die vorgängigen Planungen so wichtig machen. Sie ermöglichen ihm trotz seiner vielfältigen und sich zeitlich überlagernden Aufgaben ein effizientes Arbeiten auf Grund der auf seinen Fachkenntnissen aufbauenden routinisierten Systematik. Neben den unternehmensseitig bereitgestellten Neuen Medien greift Herr K. dazu auch auf persönliche Notizen zurück, die ihm helfen, seine Arbeit zu organisieren. Emotional aufgeladene Situationen behindern in seinen Augen eine rationale Bearbeitung und stören

dadurch die Effizienz. Aus diesem Grund verhält er sich am Telefon betont sachlich und emotional zurückhaltend. Das ist zugleich der einzige „Verstoß“ gegen die Leitlinien seines Arbeitgebers, der die Telefonate zu emotionalen Events für die Kunden gestalten will. Herr K. gesteht allenfalls Freundlichkeit zu und will sich nicht zu einer nicht-authentischen Emotionalität drängen lassen, die nicht seinem Naturell entspricht. Alle anderen arbeitsorganisatorischen Vorgaben seitens der Bank (z. B. Parallelität von Sachbearbeitung und Telefonie, Schichtsystem etc.) werden demgegenüber beinahe fatalistisch mit dem Hinweis auf das bei Vertragsunterzeichnung bestehende Wissen um diese Strukturen akzeptiert.

- **Sozialität**

Herr K. ist ein familienzentrierter Mensch. Private soziale Beziehungen werden weitgehend von der Frau K.s gepflegt. Kontakte zu Kollegen sind zwar durchaus freundschaftlich, aber eher selten. Sie ergeben sich beiläufig während der Arbeitszeit und sind zwischen den Mitgliedern seines Teams enger als gegenüber anderen Personen. Im Hinblick auf seine Arbeit sind ihm die Kollegen als Informations- und Ratgeber auf der Basis gegenseitiger Hilfe wichtig. Damit lassen sich Situationen, in denen es z. B. um schnelle Informationen für einen Kunden geht, entschärfen. Über diesen funktionalen Aspekt hinaus hat die Einbindung in eine Gruppe zusätzliche kompensatorische Wirkungen, wenn dadurch emotionale aufgeladene Situationen (z. B. Streit mit Kunden) bearbeitet werden können.

Herr K. ist sich der Bedeutung seiner individuell erbrachten Leistung bewusst, stellt sich zugleich aber in den Gesamtrahmen der Bank als eine nach außen wirkende Einheit. Diese Stellung versucht er auch den Kunden zu vermitteln: Er fungiert nicht als eigenständige Person, sondern als Vertreter der Institution. Das ist für ihn wichtig, um die Basis der Kommunikation zwischen ihm und dem Kunden zu klären. Dazu gehört für ihn auch, sich dem Sprachniveau des Kunden anzupassen. Für Herrn K. sind dies von ihm zu erbringende Vorleistungen, die eine erfolgreiche Arbeit zwar nicht garantieren, aber wahrscheinlicher machen.

- **Sinn**

Arbeit ist für Herrn K. der unhinterfragte Mittelpunkt seines Lebens: Sie sichert das Leben seiner Familie ab. Diese zentrale Stellung bringt es mit sich, andere Bereiche des Lebens nach den Anforderungen der Arbeit auszurichten. Bestandteil dieser Orientierung und der daraus hervorgehenden Arbeitsmotivation ist ein auf die Person bezogenes Leistungsstreben: durch persönlichen Einsatz sind viele Aufgaben zu bewältigen und Arbeitszufriedenheit erst möglich. Als Arbeitsaufgabe wird dabei nicht nur die konkrete Einzelhandlung angesehen, sondern die antizipierte Erwartung des Arbeitgebers an die Mitarbeiter, zum Erfolg des gesamten Unternehmens beizutragen. Es lässt sich für Herrn K. von einer Übernahme dieser Perspektive ausgehen: er sieht sich als Rädchen im Getriebe einer Organisation, die nur im effizienten Zusammenwirken aller funktionieren kann. Zugleich ist die eigene Verantwortlichkeit eng abgesteckt: verantwortlich ist Herr K. nur für seine eigene Arbeitsleistung, nicht aber für den Erfolg des Unternehmens an sich. Daraus folgt für Herrn K., die Kontrolle seiner Arbeitsleistung auch selbst vorzunehmen und an den durch den Arbeitsvertrag vereinbarten Pflichten zu messen. Dazu gehört auch die Bereitschaft zur Fortbildung. Insgesamt hat Herr K. damit die

Erwartungen seines Arbeitgebers an ein eigenverantwortliches Arbeiten verinnerlicht. Dazu ist vorausschauende Planung unabdingbar. Der hohen Bedeutung, die Herr K. gewissenhafter Vorbereitung zumisst, steht die Unwägbarkeit des Verhaltens der Kunden gegenüber: ihre Ziele müssen mit denen der Bank nicht übereinstimmen, weshalb der Kontakt prinzipiell kontingenter Natur ist. Diese Kontingenzen beherrschen bzw. die darin liegenden Risiken des Misserfolgs minimieren zu können, bedarf es organisationsseitig, aber auch von Seiten des Mitarbeiters sachlich-rationaler Arbeitsplanung. Dieser grundlegenden Orientierung an planbaren arbeitsorganisatorischen und technischen Voraussetzungen steht die Bereitschaft Herrn K.'s zur gelegentlichen Improvisation ergänzend zu Seite.

- Biographie

Vor seiner jetzigen Tätigkeit hat Herr K. verschiedene Ausbildungen durchlaufen: nach einer 1988 begonnenen Berufsausbildung mit Abitur¹¹ (Schwerpunkt: Maschinenbau), die er nach der Wende erfolgreich abschloss, wurde er in seiner Heimatstadt Rostock zum Bankkaufmann ausgebildet. Kurz nach Abschluss dieser Ausbildung leistete er seinen Wehrdienst. Weil ihm ein anvisiertes BWL/VWL-Studium als zu formalistisch angelegt erschien, begann er danach ein Magisterstudium der Politologie mit den Nebenfächern Soziologie und englische Sprachwissenschaft. Dieses Studium beendete er trotz eines studienbedingten Auslandsaufenthaltes innerhalb der Regelstudienzeit. Daraufhin entschloss er sich zur Rückkehr in seinen Ausbildungsberuf im Finanzdienstleistungsbereich und reagierte auf eine Stellenanzeige seiner derzeitigen Arbeitgeberin.

Planung und die darauf hin folgende Umsetzung in die Praxis als Mittel der Beherrschung einer unsicheren Umwelt kennzeichnen auch die biographischen Muster Herrn K.'s. Seine Statuspassagen in die schulische und berufliche Ausbildung hat er unter Hinzuziehung von Ratgebern sehr genau festgelegt und dann Schritt für Schritt verfolgt. Überraschende Wendungen hat es gegeben, sie wurden aber jeweils in Neuorientierungen umgemünzt. Möglicherweise zeigen sich in dieser Fähigkeit, der grundlegenden Erwartung an planbare Zukunft zuwiderlaufende Ereignisse produktiv zu bearbeiten, die spezifischen Wendeerfahrungen Herrn K.'s. Er wurde aus einer relativ absehbaren Zukunft im Industriebereich in die ökonomischen Unwägbarkeiten des Niedergangs der ostdeutschen Industrie entlassen, und hat sich in dieser Situation in einer Gemengelage aus Strategie und Zufall erfolgreich zu dem entwickelt, der er heute ist.

- Zusammenfassung

Der ‚Arbeitsstil‘ von Herrn K. lässt sich als kontroll- und planungsorientiertes Arbeitshandeln angesichts der antizipierten Kontingenzen darstellen. Der sachlich-rationale Modus schließt auch die Bereitschaft dazu ein, strukturelle Vorgaben der Organisation durch informelle Kooperationen, sei es mit Kollegen oder Kunden, und durch eigene individuelle Leistungen zu ergänzen, um ein nach seinen eigenen Maßstäben effizientes Arbeiten zu ermöglichen.

¹¹ Die Berufsausbildung mit Abitur war eine Einrichtung des DDR-Bildungswesens. Im Wechsel von berufsfachlichen Praxistagen und curricularen Schultagen konnten sowohl ein Facharbeiterabschluss wie das Abitur zeitgleich erworben werden. Herr K. hat diese Variante dem ihm ebenfalls möglich Erwerb des Abiturs auf der Erweiterten Oberschule (EOS; vergleichbar mit der gymnasialen Oberstufe in der BRD) vorgezogen.

4.2.2 Zum Kommunikationsstil von Herrn K.

Aus sprachwissenschaftlicher Sicht ist zunächst festzuhalten, dass Mitarbeiter von Call Centern im wesentlichen mit zwei Sorten von Telefongesprächen zu tun haben: ‚Reklamations- und Service-Gespräche‘. Bevor wir uns dem Gesprächsverhalten von Herrn K. zuwenden, sollen vorweg zuerst einmal die wesentlichen Charakteristika dieser Gesprächssorten kurz dargelegt werden.

- ‚Reklamationsgespräch‘

Der kommunikative Zweck von ‚Reklamationsgesprächen‘ liegt in BEANSTANDUNGEN/ BESCHWERDEN seitens des Kunden, deren Berechtigung zugleich zu prüfen ist, und der Befriedigung der Ansprüche des Kunden. Hierbei handelt es sich um eine potentiell konflikt- und emotionsträchtige Gesprächsform, die einer organisatorischen Vorstrukturierung und Formalisierung unterliegt (Brünner 2000, S. 101f.). Handlungslogisch liegt dabei eine Verletzung der teilweise rechtlich geregelten Prozedur ‚JEMANDEM ETWAS VERKAUFEN DURCH DEN VERKÄUFER (ORGANISATION)‘ zugrunde (Antos 1988, S. 12). Einerseits kann man ‚Reklamationsgespräche‘ als Subtyp des Musters VORWURF-ENTSCULDIGUNG-RECHTFERTIGUNG (Ohama 1987) beschreiben, oder als Spezialfall von Problemlösungen, bei dem die Urheberschaft an dem Problem einer der Parteien (Organisation oder Agent oder Kunde) zugeschrieben wird (Fiehler/Kindt/Schnieders 1999). Aufgrund dessen ist ein Call Center-Agent im Gespräch verschiedenen Maximenkonflikten ausgesetzt:

- Sprich die Sprache des Kunden (Laie) vs. Zeige dich als Fachmann (Experte);
- Handelnder der Organisation (Agent) vs. handelndes Individuum (Person);
- Image der Organisation positiv darstellen vs. Eingestehen von Fehlern/Kundenperspektive übernehmen;
- Emotions-/Beziehungsbearbeitung vs. Bearbeitung des Sachproblems;
- kundenorientierte Kommunikation vs. organisationsorientierte Kommunikation.

Auf dieser Basis entwickeln sich zumeist asymmetrische Beziehungskonstellationen in den Gesprächen, die mit Perspektivendivergenzen zwischen Klient vs. Agent verbunden sind (vgl. Brünner 2000, S. 111):

Klient	Agent
Unkenntnis der institutionellen Abläufe	Kenntnis der institutionellen Abläufe
Sicht auf die Reklamation als Einzelfall	Sicht auf die Reklamation als Standardfall
persönliche Betroffenheit	keine persönliche Betroffenheit
Abwicklung bedeutet eine Störung des Alltags	Abwicklung ist Bestandteil alltäglicher Routine
erwartet individuelle Behandlung	wendet allgemeine Regeln an
(negative) Emotionen	emotionale Neutralität

Eine idealtypische Handlungsstruktur von ‚Reklamationsgesprächen‘ sieht etwa wie folgt aus (vgl. Antos 1988, S. 12; Brünner 2000, S. 107; Fiehler/Kindt 1994; Fiehler/Kindt/Schnieders 1999):

1. Aufgabenfeld: Emotions- und Beziehungsbearbeitung

- Ausdruck von Emotionen durch VORWÜRFE (Enttäuschung, Ärger, etc.) oder Prosodie (Lautstärke etc.); Reaktion auf Emotionen durch VORWURFSBEARBEITUNG (ANTEILNAHME, ENTSCHULDIGEN, RECHTFERTIGEN); Wiedergutmachung durch ANBIETEN VON BENEFITS zur Renormalisierung der Situation (Kallmeyer 1979)
- Imagepflege (Organisationsimage-Agentenimage-Klientenimage)

2. Aufgabenfeld: Bearbeitung des Sachproblems

- Problemexplikation: MELDEN, ZUSTÄNDIGKEIT KLÄREN, DARSTELLUNG (BERICHTEN, BEWERTEN), NACHFRAGEN, PRÜFEN, URSACHE/SCHULD ZUSCHREIBEN, HERSTELLEN/RATIFIZIEREN EINER GEMEINSAMEN/GETEILTEN SICHTWEISE AUF DAS PROBLEM,
- Problemlösung: VORSCHLAGEN, ANNAHME/ABLEHNUNG, BEWERTEN/RATIFIZIEREN DER LÖSUNG

• ‚Servicegespräch‘

Der kommunikative Zweck von ‚Servicegesprächen‘ besteht in Dienstleistungen im kaufmännischen Kundendienst (BERATEN, ERKLÄREN, INFORMIEREN etc.), daneben aber auch immer in der Pflege eines positiven Organisationsimages, mit dem Ziel, durch die Kompetenzdarstellung die Kundenzufriedenheit und Kundenbindung zu erhalten (Brünner 2000, S. 119). Bisher wurden von linguistischer Seite lediglich sog. ‚Diagnose-Gespräche‘ im EDV-Service, die vorwiegend der Fehlerbehebung dienen, untersucht. Bei dieser Gesprächsform können natürlich ebenso Maximenkonflikte und Perspektivendivergenzen auftreten, wie sie bereits oben dargestellt wurden. Auf die idealtypische Handlungsstruktur kann hier aus Platzgründen nicht ausführlicher eingegangen werden; ausserdem lagen der folgenden Analyse hauptsächlich ‚Reklamationsgespräche‘ zugrunde.

• Analyse der Gesprächsdaten

Empirisch lässt sich feststellen, dass die konstitutiven sprachlichen Handlungsmuster in allen aufgezeichneten Telefongesprächen von Herrn K. sowie anderer Call Center-Agenten kaum variieren und sich im Rahmen der oben erwähnten Muster bewegen. Dies hat seine Ursache einerseits in den aufgeführten Charakteristika der Gesprächssorten und andererseits in der organisationalen Prägung des gesamten Sprachhandelns während der Arbeit. Es ist zudem bemerkenswert, dass sich Herr K. im Interview ebenfalls als fest eingebunden in die Organisation betrachtet (siehe Kap. 4.2.1, Abschnitt Sinn), was sich nicht zuletzt in den Vorgaben zum Gesprächsverhalten am Telefon manifestiert (Existenz von Gesprächsleitfäden). Es existiert hier also eine Ähnlichkeit im analytischen Befund zwischen realen Gesprächsdaten und reflexiven Daten (Interview). Deshalb wollen wir die Analyse auf den einzig variablen Aspekt im

Gesprächsverhalten, die Art und Weise des Beziehungsmanagements (vgl. Überblick bei Holly 2001), eingeführt. Hier kommen die individuellen Gestaltungsleistungen der Call Center-Agenten am deutlichsten zum Vorschein. Zunächst sollen kurz noch einmal die Interview-Daten herangezogen werden, bevor wir zur Analyse der Gesprächsdaten übergehen.

Bei der Beschreibung seiner biographischen Entwicklung (Schule, Ausbildung etc.) im Interview ist bemerkenswert, dass Herr K. die positiven Dinge in seinem Leben als selbst erarbeitet darstellt, die negativen Dinge dagegen als von der Situation verursacht. Man könnte dies als seine persönliche Grundhaltung bezeichnen (siehe auch Kap. 4.2.1, Abschnitt Biographie). Als Herr K. im Interview über den Inhalt seiner Arbeit im Call Center spricht, fällt auf, dass die persönlichen Einstellungen dazu zwar einerseits so inszeniert werden, wie sie den Vorgaben und dem Image der Arbeitsorganisation entsprechen (*interessant, spannend, abwechslungsreich* etc.), aber andererseits die Arbeit trotzdem eher als *belastend* erlebt und bewertet wird (z. B. in Schilderungen zum Tagesablauf und Auswirkungen auf die Familie). Dem Vorgesagten entspricht, dass Herr K. es positiv dargestellt, dass er seinen Beruf aus eigenem Interesse angestrebt hat. Sofern er seine Arbeit in Teilbereichen als negativ erlebt (Schichtbetrieb, Belastungen), wird diese Situation als nicht zu ändernde Vorgabe der Organisation zugeschrieben und fatalistisch hingenommen.

Komplementär dazu soll im folgenden die Kommunikation im realen Vollzug (Audio-Dokumentation authentischer Telefongespräche) mit einem gesprächsanalytischen Instrumentarium besprochen werden. Tatsächlich deckt sich Herr K.'s Gesprächsverhalten mit seinen Selbstaussagen, was aus unserer Erfahrung höchst selten ist, da Sprachhandeln größtenteils als unbewusstes und routinisiertes Handeln vollzogen wird. Gegenüber seinen Klienten stellt er die eigene Leistung als „von mir *für Sie* erarbeitet“ dar. Sprachlich zeigt sich dies vor allem durch das Ausfüllen der fakultativen (syntaktischen) Ergänzungen der Verben (vgl. zur Verbvalenz Holly, 1980). Mit dieser im Vergleich zu anderen Mitarbeitern hervorstechenden Besonderheit stellt Herr K. speziell seine persönlichen Verdienste in bezug auf seine Problemlösungskompetenz in den Vordergrund. Auffällig ist in diesem Zusammenhang der häufige Gebrauch der Personalpronomina *ich* und *mein* zur Darstellung des eigenen Engagements. Der folgende Gesprächsausschnitt¹² macht das am Beispiel einer Image-bestätigenden Sequenz in Form eines HÖFLICHEN ANGEBOTES deutlich (vgl. Holly 1979, S. 51f.). Herr K. reagiert auf eine ‚Reklamation‘ folgendermaßen:

„Ähm, mein Angebot sieht wie folgt aus, also das is ne Sache, die kann ich ohne mit der Wimper zu zucken, sofort *für Sie* machen. Ich kann Sie verbinden für die Order, für die nachträgliche Order, und ähm, also als Zeichen des Entgegenkommens schreib ich Ihnen hundert Mark gut.“

Wenn dagegen etwas außerhalb seines unmittelbaren Kontrollbereiches liegt oder er noch keine Problemlösung für den Klienten finden konnte, dann zieht sich Herr K. auf eine defensive Strategie zurück, er verschiebt die Verantwortlichkeit dafür auf die Organisation der Bank. Probleme werden damit so perspektiviert (vgl. Kallmeyer 1996), dass sie als quasi „von

außerhalb“ verursacht erscheinen. Das trägt erheblich zur Wahrung seines persönlichen Image am Telefon bei. Das wird vor allem deutlich am Wandel des Gebrauchs der Personalpronomina: die Verwendung von *ich* - so wie bei der Darstellung der eigenen Leistung - wechselt sofort zur Verwendung von *wir*, *uns* etc. Dies ist meist verbunden mit Image-korrektiven Sequenzen wie **ENTSCULDIGUNGEN** und **RECHTFERTIGUNGEN**. Man kann sehen, dass insgesamt die spezifische Verwendung von Personalpronomina ein wichtiges Mittel zur Beziehungsgestaltung ist (vgl. Holly 1979; 2000). Dazu ein Beispiel, wie Herr K. in Sachen einer ‚Reklamation‘ wieder selbst aktiv wird, aber die bisherige Erfolglosigkeit als organisational verursacht darstellt (*wir*-Perspektive):

„Genau, also *wir* haben die Empfangsprotokolle durchgesehen, und dieses Fax ist also mit der Absendernummer nicht eingetroffen bei *uns*.“

Es ist besonders hervorzuheben, dass dieser spezifische Gebrauch der Personalpronomina nicht (!) in den Schulungsvorgaben und Gesprächsleitfäden der Bank verankert ist, sondern als eine eigenständige Gestaltungsleistung von Herrn K. betrachtet werden kann.

- Zusammenfassung

Der ‚Kommunikationsstil‘ von Herrn K. zeichnet sich im Vergleich zu anderen Call Center-Agenten durch eine in erhöhtem Maße bewusste und stark kontrollierte (!) Form des Beziehungsmanagements in bezug auf die Kunden aus. Dieses ist vor allem durch einen hohen Anteil positiver persönlicher Selbstdarstellung geprägt, die darauf zielt, vor allem das eigene positive Image gegenüber dem Anrufenden zu wahren. Das wird von Herrn K. auch im Interview explizit hervorgehoben. Vorrangiges kommunikatives Ziel dieser offensiven Selbstdarstellung ist es, so die Interview-Aussage von Herrn K., sich als effektiver Problemlöser zu verkaufen. Vor allem letzteres ist nicht bei allen Kollegen von Herrn K. in dieser Deutlichkeit festzustellen. Das Gesprächsverhalten ist weiterhin sehr sachlich und durch einen geringen Grad an emotionaler Beteiligung geprägt, was sich ebenfalls mit Herrn K.’s Darstellungen im Interview deckt. Er ist gleichfalls darauf bedacht, auch die Emotionalität (meist Verärgerung) der Kunden im Gespräch zu kanalisieren und damit abzuschwächen (z. B. durch ausdrückliches Zustimmung). Dies ist nicht bei allen seiner Kollegen der Fall.

5 Resümee

Unsere Überlegungen sollten gezeigt haben, dass eine Analyse des Gegenstandes ‚Kommunikationsarbeit‘ auf Basis unterschiedlicher Datensätze (Interview, Audioaufnahmen), jeweils aus Perspektive zweier unterschiedlicher Wissenschaftsdisziplinen (Arbeitssoziologie, Angewandte Linguistik) und mit Hilfe der Konzepte des ‚personalen Arbeitsstils‘ und des pragmatisch orientierten ‚Kommunikationsstils‘ einen sinnvollen Versuch darstellt, die Erkenntnismöglichkeiten über moderne Formen von Arbeit und Arbeitskommunikation zu

¹² Hinweis: Aus Platzgründen können nur ganz wenige Beispiele in einer stark vereinfachten Transkription aufgeführt werden.

erweitern. Der Vorteil liegt darin begründet, mehrere Aspekte des Gegenstandes ‚Kommunikationsarbeit‘ gleichzeitig in den Blick nehmen zu können und so eine relativ ‚dichte Beschreibung‘ (Clifford Geertz) zu erreichen. Das hat nicht zuletzt auch den positiven Effekt, dass auf diesem Wege die Forschungsergebnisse über die einzelnen Disziplinen hinweg, die sich häufig isoliert voneinander mit dem gleichen Gegenstand beschäftigen, überprüft und dadurch valider gemacht werden können.

Arbeits- und Kommunikationsstil sind auf vielfältige Weise miteinander verflochten. Der Arbeitsstil von Herrn K. ist, wie seine Selbstaussagen im Interview belegen, in hohem Maße von Kontrollbedürfnissen gegenüber einer kontingent erscheinenden Umwelt geprägt. Dies konstituiert sich im Gesprächsverhalten unter anderem durch die bewusst reflektierte Art des Beziehungsmanagements. Insofern könnte man von einer ‚indem-Relation‘ (s. Kap. 3) ausgehen, die die Bereiche des Arbeitshandelns und der arbeitsbegleitenden Sprachhandlungen miteinander verbindet. Diese holistische Beschreibung erlaubt es, weitere Typen von Kommunikationsarbeitern zu erfassen und über eine Typologisierung zu weiterführenden theoretischen Erkenntnissen über diese Form moderner Arbeit zu gelangen.

Wenn sich darüber hinaus zeigen ließe, dass Kohärenzen zwischen Arbeits- und Kommunikationsstil auf der einen Seite und Lebensstil auf der anderen Seite bestehen, ist davon auszugehen, dass diesen ein gemeinsamer, übergreifender ‚Handlungsstil‘ des Individuums zugrunde liegt.

6 Literatur

- Altschuh, E.; Schultz-Gambard, J.: Denk- und Führungsstile von Managern in Ost- und Westdeutschland. Pfaffenweiler ; Centaurus, 1993.
- Antos, G.: Zwischen Kunde und Computer: Interaktionsprobleme bei telefonischen Reklamationsgesprächen. In: Gutenberg, N. (Hrsg.): Kann man Kommunikation lehren? Konzepte mündlicher Kommunikation und ihrer Vermittlung. Frankfurt am Main : Scriptor, 1988. S. 9-17
- Auer, P.: Natürlichkeit und Stil. In: Hinnenkamp, V.; Selting, M. (Hrsg.): Stil und Stilisierung. Tübingen : Niemeyer, 1989. S. 27-59
- Auer, P.; di Luzio, A.: Interpretive Sociolinguistics. Tübingen : Narr, 1984.
- Berger, P.A.; Hradil, S. (Hrsg.): Lebenslagen, Lebensläufe, Lebensstile. In: Soziale Welt, Sonderband 7, Göttingen : Otto Schwartz, 1990.
- Bittner, S.; Schietinger, M.; Schroth, J.; Weinkopf, C.: Call Center - Entwicklungsstand und Perspektiven. Eine Literaturanalyse. Gelsenkirchen : Institut Arbeit und Technik, 2000. (IAT-Projektbericht 2000-01)
- Bohnsack, R.: Rekonstruktive Sozialforschung: Einführung in Methodologie und Praxis qualitativer Forschung. Opladen : Leske & Budrich, 1999.
- Bourdieu, P.: Die feinen Unterschiede. Kritik der gesellschaftlichen Urteilskraft. Frankfurt am Main : Suhrkamp, 1987.
- Brünner, G.: Wirtschaftskommunikation: Linguistische Analyse ihrer mündlichen Formen. Tübingen : Niemeyer, 2000.
- Dittmar, N.: Grundlagen der Soziolinguistik – Ein Arbeitsbuch mit Aufgaben. Tübingen : Niemeyer, 1997.
- Drieseberg, T. J.: Lebensstil-Forschung. Heidelberg : Physica, 1995.

- Ferchhoff, W.; Neubauer, G.: Jugendkulturelle Stile zwischen Selbstinszenierung und Vereinnahmung. In: Mansel, J.; Klocke, A.: Die Jugend von Heute. Selbstanspruch, Stigma, Wirklichkeit. Weinheim ; München : Juventa, 1996, S. 32-52
- Fiehler, R.; Kindt, W.: Reklamationsgespräche. Schulungsperspektiven auf der Basis von Ergebnissen diskursanalytischer Untersuchungen. In: Bartsch, E. (Hrsg.): Sprechen, Führen, Kooperieren in Betrieb und Verwaltung. Kommuniaktion in Unternehmen, München : Reinhardt, 1994. S. 255-269
- Fiehler, R.; Kindt, W.; Schnieders, G.: Kommunikationsprobleme in Reklamationsgesprächen. In: Brüner, G.; Fiehler, R.; Kindt, W. (Hrsg.): Angewandte Diskursforschung. Bd.1: Grundlagen und Beispielanalysen, Opladen : Westdeutscher Verlag, 1999, S. 120-154
- Franck, D.: Stil und Interaktion. In: Spillner, B. (Hrsg.): Methoden der Stilanalyse. Tübingen : Narr, 1984, S. 121-135
- Gal, S.: Art. Linguistic Repertoire. In: Ammon, U.; Dittmar, N.; Mattheier, K.J. (Hrsg.): Sociolinguistics: Soziolinguistik. Berlin ; New York : de Gruyter, 1987, S. 286-292
- Garfinkel, H.: Studies in Ethnomethodology. Englewood Cliffs (NJ) : Prentice Hall, 1967.
- Giddens, A.: Die Konstitution der Gesellschaft. Frankfurt am Main ; New York : Campus, 1988.
- Haase, J.: Verfahren der sprachlichen Aneignung von Hypertextstrukturen. Problemlösungen – Konzeptualisierungen – Raumdeixis. TU Chemnitz, Germanistische Sprachwissenschaft, Magisterarbeit, 1999.
- Haase, J.: Aufgabenbearbeiten und Problemlösen beim sprachlichen Umgang mit dem Computer. TU Chemnitz, Germanistische Sprachwissenschaft, Arbeitspapiere „Die sprachliche Aneignung von Computermedien“, 2000a.
- Haase, J.: Lokale Deixis in Gesprächen zum Problemlösen beim sprachlichen Umgang mit dem Computer. TU Chemnitz, Germanistische Sprachwissenschaft, Arbeitspapiere „Die sprachliche Aneignung von Computermedien“, 2000b.
- Haase, J.: Computerbilder. Konzeptualisierungen bei der Nutzung komplexer Technik. TU Chemnitz, Germanistische Sprachwissenschaft, Arbeitspapiere „Die sprachliche Aneignung von Computermedien“, 2000c.
- Haase, J.; Stöckl, H.: Im Dialog mit dem Wortschatz. Handlungstheoretische Konzeptionen für Wörterbücher im Internet. In: Sprache und Datenverarbeitung: International Journal for Language Data Processing. Jg. 22, H. 2, 1998. S: 5-37
- Habscheid, S.: Die sprachliche Aneignung von Computermedien. In: Hielscher, M.; Sichelschmidt, L.; Strohner, H. (Hrsg.): Medium Sprache. Frankfurt am Main : Lang, 1998, S. 123-135
- Habscheid, S.: Medium in der Pragmatik: Eine kritische Bestandsaufnahme. In: Deutsche Sprache H. 28, 2000, S. 126-143
- Habscheid, S.: Empraktisches Sprechen in computergestützten Arbeitssettings. In: Matuschek, I.; Henninger, A.; Kleemann, F. (Hrsg.): Neue Medien im Arbeitsalltag. Wiesbaden : Westdeutscher Verlag, 2001. (i. Dr.)
- Habscheid, S.; Haase, J.; Schulz-Nötzold, C.: Neue Medien und soziokultureller Wandel. TU Chemnitz, Germanistische Sprachwissenschaft, Arbeitspapiere „Die sprachliche Aneignung von Computermedien“, 2000.
- Habscheid, S./Holly, W.: Sprachliche Aneignung von Computermedien. In: Boehnke, K.; Doering, N. (Hrsg.): Neue Medien im Alltag. Die Vielfalt ihrer Nutzungsformen. Lengerich u. a. : Pabst Science, 2001. (i. Dr.)
- Hahn, A.: Soziologische Relevanzen des Stilbegriffs. In: Gumbrecht, H. U.; Pfeiffer, K. L. (Hrsg.): Stil. Geschichten und Funktionen eines kulturwissenschaftlichen Diskurselements. Frankfurt am Main : Suhrkamp, 1986, S. 603-311
- Heinz, R.: Stil als geisteswissenschaftliche Kategorie. Würzburg, 1986.
- Heringer, H.-J.: Praktische Semantik. Stuttgart : Klett, 1974.

- Hinnenkamp, V.; Selting, M. (Hrsg.): Stil und Stilisierung. Tübingen : Niemeyer, 1989.
- Höflich, J. R.: Technisch vermittelte interpersonale Kommunikation: Grundlagen, organisatorische Medienverwendung, Konstitution „elektronischer Gemeinschaften“. Opladen : Westdeutscher Verlag, 1996.
- Holly, W.: Imagearbeit in Gesprächen. Zur linguistischen Beschreibung des Beziehungsaspekts. Tübingen: Niemeyer, 1979.
- Holly, W.: Syntaktische Analyse. Eine verkürzte, vereinfachte und veränderte Version von Heringers Konstitutionssystem. 2. Aufl., Trier (Typoskript), 1980.
- Holly, W.: Alte und neue Medien. Zur inneren Logik der Mediengeschichte. In: Rüschoff, B.; Schmitz, U. (Hrsg.): Kommunikation und Lernen mit alten und neuen Medien. Frankfurt am Main u.a. : Lang, 1996, S. 9-16
- Holly, W.: Zur Rolle von Sprache in Medien: Semiotische und kommunikationsstrukturelle Grundlagen. In: Muttersprache H. 107, 1997, S. 64-75
- Holly, W.: Was sind ‚Neue Medien‘ – Was sollen ‚Neue Medien‘ sein? In: Voß, G. G.; Holly, W.; Boehnke, K. (Hrsg.): Neue Medien im Alltag: Begriffsbestimmung eines Forschungsfeldes. Opladen : Leske & Budrich, 2000.
- Holly, W.: Art. Beziehungsmanagement und Imagearbeit. In: Antos, G.; Brinker, K.; Heinemann, W.; Sager, S.F. (Hrsg.): Text- und Gesprächslinguistik. Ein internationales Handbuch zeitgenössischer Forschung. Berlin/New York : de Gruyter, 2001.
- Holly, W.; Biere, B. U.: Zur Einführung: Medien im Wandel. Neues in alten, Altes in neuen Medien. In: dies. (Hrsg.): Medien im Wandel. Opladen : Westdeutscher Verlag, 1998, S. 7-11
- Holly, W.; Habscheid, S.: Die sprachliche Aneignung von Computermedien. In: Boehnke, K. et al.: Neue Medien im Alltag: Von individueller Nutzung zu soziokulturellem Wandel. Lengerich u.a. : Pabst Science, 1999, S. 120-158
- Holly, W.; Habscheid, S.: Die sprachliche Aneignung von Computermedien – Vorstellung eines Projekts. In: Kallmeyer, W. (Hrsg.): Sprache und neue Medien. Jahrbuch des Instituts für Deutsche Sprache 1999. Berlin/New York : de Gruyter, 2000, S. 127-141
- Hörning, K. H.; Ahrens, D. ; Gerhard, A.: Die Autonomie des Lebensstils. Wege zu einer Neuorientierung der Lebensstilforschung. In: Schwenk, O. (Hrsg.): Lebensstil zwischen Sozialstrukturanalyse und Kulturwissenschaft. Opladen : Leske & Budrich, 1996, S: 33 - 52
- Hradil, S. (Hrsg.): Zwischen Bewusstsein und Sein. Opladen : Leske & Budrich, 1992.
- Hymes, D.: Über Sprechweisen. In: ders., Soziolinguistik. Zur Ethnographie der Kommunikation. Frankfurt am Main : Suhrkamp, 1979, S. 166-192
- Joas, H.: Die Kreativität des Handelns. Frankfurt am Main : Campus, 1992.
- Kallmeyer, W.: Kritische Momente. Zur Konversationsanalyse von Interaktionsstörungen. In: Frier, W.; Labrousse, H. (Hrsg.): Grundfragen der Textwissenschaft. Amsterdam, 1979, S: 59-110
- Kallmeyer, W.; Keim, I.: Kommunikation in der Stadt. Teil 3, Berlin ; New York : de Gruyter, 1995.
- Kallmeyer, W. (Hrsg.): Gesprächsrhetorik. Tübingen : Narr, 1996.
- Keppeler, A.: Tischgespräche. Frankfurt am Main : Suhrkamp, 1994.
- Kleemann, F.: Informatisierung der Arbeit: Folgen für Arbeitsverhältnisse und subjektive Leistungen der Arbeitenden. In: kommunikation@gesellschaft, Jg.1, 2000, <http://www.kommunikation-gesellschaft.de/> bzw. <http://www.rz.uni-frankfurt.de/fb03/K.G/> (Stand 12.01.2001).
- Kleemann, F.; Matuschek, I.: Arbeits- und Technikstile in medienvermittelter Arbeit. In: Matuschek, I.; Henninger, A.; Kleemann, F. (Hrsg.): Neue Medien im Arbeitsalltag. Wiesbaden : Westdeutscher Verlag, 2001. (i. Dr.)
- Kleemann, F.; Matuschek, I.; Voß, G. G.: Zur Subjektivierung von Arbeit. Beitrag im Projektverbund „Arbeit und Ökologie“ von DIW – WI – WZB. Wissenschaftszentrum Berlin, 1999. (discussion paper P 99 – 512)

- Kleemann, F.; Matuschek, I.; Voß, G. G.: Subjektivierung von Arbeit. Ein Überblick zum Stand der Diskussion. In: Moldaschl, M.; Voß, G. G. (Hrsg.): Subjektivierung von Arbeit. München ; Mering : Hampp, 2001. (i. Dr.)
- Klemm, M.; Graner, L.: Chatten vor dem Bildschirm. Nutzerkommunikation als Fenster zur alltäglichen Computerkultur. In: Thimm, C. (Hrsg.): Soziales im Netz. Sprache, soziale Beziehungen und Identität im Internet. Opladen : Westdeutscher Verlag, 2000, S. 156-179
- Knoblauch, H.: Arbeit als Interaktion. Informationsgesellschaft, Post-Fordismus und Kommunikationsarbeit. In: Soziale Welt 47, 1996, S. 344-362
- Knoblauch, H.; Heath, C.: Technologie, Interaktion und Organisation: Die Workplace Studies. In: Schweizerische Zeitschrift für Soziologie Jg. 25, H. 2, 1999, S. 163-181
- Labov, W.: Die Isolierung von Kontextstilen. In: ders.: Sprache im sozialen Kontext. Beschreibung und Erklärung struktureller und sozialer Bedeutung von Sprachvariation, Bd.1, Kronberg/Ts. : Scriptor, 1976, S. 29-66
- Lang, R.: Lehrbrief Führung, 2000, <http://www.tu-chemnitz.de/wirtschaft/bwl5/lehrbrief> (Stand 12.01.2001)
- Luckmann, T.: Grundformen der gesellschaftlichen Vermittlung des Wissens: Kommunikative Gattungen. In: Kölner Zeitschrift für Soziologie und Sozialpsychologie, Sonderheft 27, 1986, S. 191-211
- Lüdtke, H.: Expressive Ungleichheit. Zur Soziologie der Lebensstile. Opladen : Westdeutscher Verlag, 1989.
- Lüdtke, H.: Zeitverwendung und Lebensstile. Empirische Analysen zu Freizeitverhalten, expressiver Ungleichheit und Lebensqualität in Westdeutschland. Marburg : MBSF 5, 1995.
- Lüdtke, H.; Matthäi, I.; Ulbrich-Hartmann, M.: Technik im Alltagsstil: Eine empirische Studie zum Zusammenhang von technischem Verhalten, Lebensstilen und Lebensqualität privater Haushalte. Marburg : MBSF 4, 1994.
- Matuschek, I.: Zeit und Devianz: Zeitorientierung, Langeweile und abweichendes Verhalten bei Jugendlichen. TU Berlin, Dissertation, 1999, <http://dohost.rz.hu-berlin.de/dissertationen/matuschek-ingo-1999-07-19/PDF/Matuschek.pdf>. (Stand 12.01.2001)
- Matuschek, I.; Henninger, A.; Kleemann, F. (Hrsg.): Neue Medien im Arbeitsalltag. Wiesbaden : Westdeutscher Verlag, 2001. (i. Dr.)
- Matuschek, I.; Kleemann, F.; Voß, G. G.: Personaler Arbeitsstil: Ein Konzept zur Untersuchung ‚subjektiverter‘ Arbeit. In: Moldaschl, M.; Voß, G. G. (Hrsg.): Subjektivierung von Arbeit. München; Mering : Hampp, 2001. (i. Dr.)
- Mikl-Horke, G.: Arbeits- und Industriesoziologie. München : Oldenburg Verlag, 1994.
- Moldaschl, M.; Voß, G. G. (Hrsg.): Subjektivierung von Arbeit. München ; Mering : Hampp, 2001. (i. Dr.)
- Müller, H.P.: Sozialstruktur und Lebensstile. Der neuere theoretische Diskurs über soziale Ungleichheit. Frankfurt am Main ; New York : Campus, 1992.
- Ohama, R.: Eine Reklamation. In: Osnabrücker Beiträge zur Sprachtheorie H. 38, 1987, S. 27-52
- Panofsky, E.: Stil und Medium im Film. Mit Beiträgen von Irving Lavin und William S. Heckscher. Frankfurt am Main : Fischer TB, 1999.
- Panofsky, E.: Aufsätze zu Grundfragen der Kunstwissenschaft. Frankfurt am Main : Suhrkamp, 1996.
- Pongratz, H.; Voß, G. G.: Fremdorganisierte Selbstorganisation. In: Zeitschrift für Personalforschung Jg.7, H. 1, 1997, S.30-53
- Püschel, U.: Stilpragmatik – Vom praktischen Umgang mit Stil. In: Stickel, Gerhard (Hrsg.): Stilfragen, Berlin ; New York : de Gruyter, 1995, S. 303-328
- Püschel, U.: Art: Stil. In: Antos, G.; Brinker, K.; Heinemann, W.; Sager, S.F. (Hrsg.): Text- und Gesprächslinguistik. Ein internationales Handbuch zeitgenössischer Forschung. Bd.1, Berlin ; New York : de Gruyter, 2000, S. 473ff.

- Rammert, W.: Neue Technologien – neue Begriffe? Lassen sich die Technologien der Informatik mit den traditionellen Konzepten der Arbeits- und Industriesoziologie noch angemessen erfassen? In: Malsch, T.; Mill, U. (Hrsg.): ArByte. Modernisierung der Industriesoziologie? Berlin : sigma, 1992, S. 29-51
- Rieder, K.; Matuschek, I.; Anderson, P.: The workers' and customers' contribution to successful interaction in call centre work. Paper presented at the workshop: Are Regimented Forms of Work Organisation Inevitable? Call Centres and the Changes for an Innovative Organisation of Service Work in Europe, Duisburg, Gerhard-Mercator-Universität, December 2 and 3, 2000. (i. Dr.)
- Sandig, B.: Stilistik der deutschen Sprache. Berlin ; New York : de Gruyter, 1986.
- Sandig, B.: Tendenzen der linguistischen Stilforschung. In: Stickel, G. (Hrsg.): Stilfragen. Berlin ; New York : de Gruyter, 1995, S. 27-61
- Schäffer, B.: Die Band. Stil und ästhetische Praxis im Jugendalter. Opladen : Leske und Budrich, 1996.
- Schulz-Nötzold, C.: Kooperation und Koordination beim Sprechen vor dem Computer. TU Chemnitz, Germanistische Sprachwissenschaft, Arbeitspapiere „Die sprachliche Aneignung von Computermedien“, 2000a.
- Schulz-Nötzold, C.: Interaktionsprofile: Formen der funktionalen Rollendifferenzierung beim Umgang mit dem Computer. TU Chemnitz, Germanistische Sprachwissenschaft, Arbeitspapiere „Die sprachliche Aneignung von Computermedien“, 2000b.
- Schwenk, O. (Hrsg.): Lebensstil zwischen Sozialstrukturanalyse und Kulturwissenschaft. Opladen : Leske & Budrich, 1996.
- Selting, M.: Speech styles in conversation as an interactive achievement. In: Hickey, L. (Ed.): The pragmatics of style. London : Routledge, 1989.
- Selting, M.; Sandig, B. (Hrsg.): Sprech- und Gesprächsstile. Berlin ; New York : de Gruyter, 1997.
- Simmel, G.: Philosophie des Geldes. Frankfurt a.M.: Suhrkamp, 1989 [1900].
- Soeffner, H.-G.: Stil und Stilisierung. Punk oder die Überhöhung des Alltags. In: Gumbrecht, H. U.; Pfeiffer, K. L. (Hrsg.): Stil: Geschichten und Funktionen eines kulturwissenschaftlichen Diskurselements. Frankfurt am Main : Suhrkamp, 1986, S. 317-342
- Sowinski, B.: Stilistik. 2.Auflage, Stuttgart : Metzler, 1999.
- Strübing, J.: Subjektive Leistungen im Arbeitsprozeß: Eine empirische Untersuchung von Arbeitsstilen in der Programmierarbeit. Wiesbaden : Deutscher Universitäts Verlag, 1993.
- Tannen, D.: Conversational style. Analyzing talk among friends, Norwood (NJ) : Ablex Publishing, 1984.
- TIB e.V. (Hrsg.): Vom Call Center zum Communication Center. Zukunftsbranche oder Eintagsfliege? Tagungsband zur Fachtagung am 6. und 7. Juli 2000 in Hamburg
- Voß, G. G.: Das Ende der Teilung von „Arbeit und Leben“? An der Schwelle zu einem neuen gesellschaftlichen Verhältnis von Betrieb- und Lebensführung. In: Beckenbach, N.; van Treeck, W. (Hrsg.): Umbrüche gesellschaftlicher Arbeit. In: Soziale Welt, Sonderband 9, Göttingen, 1994, S. 269-294
- Voß, G. G.: Die Entgrenzung von Arbeit und Arbeitskraft. Eine subjektorientierte Interpretation des Wandels der Arbeit. In: MitAB Jg. 31, H. 3, 1998, S. 473-487
- Voß, G. G.: Neue Arbeits- und Alltagspraktiken in medienvermittelten autonomisierten Arbeitsformen. In: Boehnke, K. et al: Neue Medien im Alltag: Von individueller Nutzung zu soziokulturellem Wandel. Lengerich u.a. : Pabst Science, 1999, S. 245-284
- Weber, M.: Die Protestantische Ethik I. Eine Aufsatzsammlung. Herausgegeben von Johannes Winckelmann. Gütersloh : GTB Siebenstern, 1984.

Personenverzeichnis

Die Herausgeberin und der Herausgeber dieses Berichtes leiten Teilprojekte innerhalb der Chemnitzer DFG-Forschergruppe „Neue Medien im Alltag. Von individueller Nutzung zu soziokulturellem Wandel“ (Bo 929/13-1).

Werner Dilger, Prof. Dr. rer. nat., leitet seit 1993 die Professur für Künstliche Intelligenz an der TU Chemnitz. Innerhalb der Forschergruppe ist er für die informatischen Teilprojekte „Softwareentwicklung in der Praxis im Kulturvergleich“ und „Modellierung und Simulation der Rezeption textuell repräsentierter Inhalte im Internet“ verantwortlich. Seine Forschungsschwerpunkte sind Maschinelles Lernen, Data Mining, Multi-Agentensysteme und Robotik.

dilger@informatik.tu-chemnitz.de

(<http://www.tu-chemnitz.de/informatik/homepages/ki>)

Randi Gunzenhäuser, Dr. phil., hat seit 2000 eine Stelle als wissenschaftliche Mitarbeiterin im kulturwissenschaftlichen Teilprojekt „Interaktion mit fiktionalen Hypertexten“. Ihre Interessen sind Literaturtheorie, *Gender Studies*, Puritanismus, Moderne, Medien- und Kulturtheorien. Ihre Habilitationsschrift trägt den Titel „Menschmaschinen – Maschinenmenschen: Körperkonzepte im Wandel“.

randi.gunzenhaeuser@phil.tu-chemnitz.de

(<http://www.tu-chemnitz.de/phil/hypertexte/gunzenhaeuser.htm>)

Annette Henninger, Dr. phil., Politikwiss., hat seit 1998 eine Stelle als wissenschaftliche Mitarbeiterin im informatischen Teilprojekt „Softwareentwicklung in der Praxis im Kulturvergleich“. Sie arbeitet schwerpunktmäßig zu den Themen Arbeit, Arbeitsmarktpolitik, Geschlechterforschung, Technik und Regionalpolitik.

annette.henninger@informatik.tu-chemnitz.de

(<http://www.tu-chemnitz.de/informatik/homepages/ki>)

Jana Haase, M.A., ist seit Januar 2000 Wissenschaftliche Mitarbeiterin im Teilprojekt der Germanistischen Sprachwissenschaft „Die sprachliche Aneignung von Computermedien“. Ihre Arbeitsschwerpunkte sind Sprechen und Problemlösen, Neue Medien und Gesprächsanalyse.

jana.haase@phil.tu-chemnitz.de

(<http://www.tu-chemnitz.de/phil/germanistik/sprachwissenschaft/jana.haase/>)

Personenverzeichnis

Evelyne Keitel, Prof. Dr. phil., ist seit 1993 Professorin für Amerikanistik an der TU Chemnitz. Sie leitet das Teilprojekt „Interaktion mit fiktionalen Hypertexten“. Ihre Forschungsschwerpunkte sind Postmoderne, Literaturtheorie, *reader-response criticism*, feministische Literaturwissenschaft, *Early Republic*, Detektivliteratur, *popular culture* und Neue Medien.

evelyne.keitel@phil.tu-chemnitz.de

(<http://www.tu-chemnitz.de/phil/hypertexte/keitel.htm>)

Frank Kleemann, Dipl.-Soz., M.A., ist seit 1998 wissenschaftlicher Mitarbeiter an der Professur Industrie- und Techniksoziologie und teilzeitlicher Mitarbeiter im industriesoziologischen Teilprojekt „Neue Arbeits- und Alltagspraktiken in medienvermittelten autonomisierten Arbeitsformen“. Seine Arbeitsschwerpunkte sind Arbeit und Informationstechnologien, insbesondere Telearbeit, Wandel der Arbeitsgesellschaft und Soziologie des Alltags.

frank.kleemann@phil.tu-chemnitz.de

(<http://www.tu-chemnitz.de/phil/soziologie/voss/>)

Ingo Matuschek, Dr. rer. soc., arbeitet seit 1999 als wissenschaftlicher Mitarbeiter im industriesoziologischen Teilprojekt „Neue Arbeits- und Alltagspraktiken in medienvermittelten autonomisierten Arbeitsformen. Seine Schwerpunkte in der Forschung sind Arbeitssoziologie, Qualitative Methoden, Zeitsoziologie und Soziologie abweichenden Verhaltens.

ingo.matuschek@phil.tu-chemnitz.de

(<http://www.tu-chemnitz.de/phil/soziologie/voss/>)

Andrea Sieber, Dipl.-Inf., arbeitet seit 1998 als wissenschaftliche Mitarbeiterin im informatischen Teilprojekt „Softwareentwicklung in der Praxis“. Sie forscht zum Schwerpunkt Informatik in der Arbeitswelt und beschäftigt sich insbesondere mit Arbeitspraktiken in der Softwareentwicklung und Prozessmodellen.

andrea.sieber@informatik.tu-chemnitz.de

(<http://www.tu-chemnitz.de/Informatik/HomePages/KI>)

Gunter Süß, M.A., ist seit 2000 wissenschaftlicher Mitarbeiter an der Professur für Amerikanistik und teilzeitlicher Mitarbeiter im kulturwissenschaftlichen Teilprojekt „Interaktion mit fiktionalen Hypertexten“. Seine Hauptinteressengebiete sind *popular music*, *digital media*, Film und Kulturtheorien.

gunter.suess@phil.tu-chemnitz.de

(<http://www.guntersuess.de>)

Carsten Schulz-Nötzold, M.A., ist seit 2000 wissenschaftlicher Mitarbeiter im DFG-Projekt „Die sprachliche Aneignung von Computermedien“ im Fachgebiet Germanistische Sprachwissenschaft an der TU Chemnitz. Seine Forschungsschwerpunkte sind Linguistische Gesprächsforschung und Medienkommunikation.

carsten.schulz@phil.tu-chemnitz.de

(<http://www.tu-chemnitz.de/phil/germanistik/sprachwissenschaft/>)