Ansätze zur kollaborativen Softwareerstellung

Tobias Hildenbrand and Franz Rothlauf and Armin Heinzl

Working Paper 6/2006
February 2006

Working Papers in Information Systems

University of Mannheim
Department of Information Systems 1
D-68131 Mannheim/Germany
Phone +49 621 1811691, Fax +49 621 1811692
E-Mail: wifo1@uni-mannheim.de
Internet: `http://www.bwl.uni-mannheim.de/wifo1`

Ansätze zur kollaborativen Softwareerstellung

Tobias Hildenbrand, Franz Rothlauf, Armin Heinzl

Kernpunkte

Ziel dieses Beitrags ist es, einen umfassenden und systematischen Überblick über existierende Ansätze zur kollaborativen Softwareerstellung zu liefern. Zu diesem Zweck wird ein mehrdimensionaler Klassifikationsrahmen vorgestellt.

- Die unterschiedlichen Ansätze zur kollaborativen Softwareerstellung lassen sich sowohl auf Projekt- als auch Unternehmensebene anhand bestimmter Dimensionen einordnen.
- Zur zwischenbetrieblichen Zusammenarbeit in der Softwareindustrie bedarf es spezieller Vorgehensmodelle, Methoden und Werkzeuge aus dem Software Engineering.
- Ein theoretischer Bezugsrahmen kann die Integration der projektbezogenen Sichtweise und der wirtschaftlichen Rahmenbedingungen zwischen Unternehmen unterstützen.

Stichworte

Softwareentwicklung, Software Engineering, Industrialisierung, Softwareerstellungsprozess, zwischenbetriebliche Softwareerstellung, Zusammenarbeit, Kollaboration, Kooperation, Softwarewertschöpfungskette, Softwareökosystem

Kurzfassung

Die Erstellung von Software zur Unterstützung betrieblicher Abläufe wird in zunehmendem Maße komplexer. Da der Erstellungsprozess in der Softwareindustrie traditionell einer Werkstatt- bzw. Einzelfertigung entspricht, erfordert die stetig steigende Nachfrage nach betrieblicher Software und die fortschreitende Globalisierung die rationellere Gestaltung der Softwareentwicklung. In der Literatur werden daher immer häufiger die Industrialisierung der Softwareerstellung und neuartige Formen der Spezialisierung, Arbeitsteilung und Zusammenarbeit (engl. *Collaboration*) vorgestellt. Dabei kann im Wesentlichen unterschieden werden, ob die Zusammenarbeit einzelner Akteure und Arbeitsgruppen auf Projektebene oder die strategische Zusammenarbeit von Unternehmen innerhalb der Softwareindustrie behandelt wird. Über diese beiden grundlegenden Betrachtungsebenen hinweg lassen sich existierende Ansätze zur arbeitsteiligen Softwareerstellung entlang mehrerer Dimensionen, wie räumliche, zeitliche und organisatorische Verteilung der Aktivitäten im Prozess sowie Intensität und Richtung der Zusammenarbeitsbeziehungen klassifizieren. Ziel dieses Artikels ist es, einen umfassenden und systematischen Überblick über bestehende Ansätze zur kollaborativen Softwareerstellung zu geben, indem diese in einen generischen Klassifikationsrahmen eingeordnet werden. Des Weiteren soll eine etymologische und pragmatische Herleitung des Kollaborationsbegriffs die Etablierung eines eigenständigen Forschungsparadigmas im Rahmen der Wirtschaftsinformatik ermöglichen.[1]

[1] Diese Arbeit entstand im Rahmen des Projekts CollaBaWü des Forschungsverbunds PRIMIUM, der durch das Ministerium für Wissenschaft, Forschung und Kunst des Landes Baden-Württemberg gefördert wird. Des Weiteren wollen wir Michael Zapf für seine Vorarbeiten zu dieser Arbeit danken.

1. Einleitung

Die Erstellung von Software zur Abwicklung von Geschäftsprozessen wird in zunehmendem Maße komplexer. Die wachsende Komplexität der Anwendungen und die steigende Bedeutung von Software in unterschiedlichsten Anwendungsdomänen stellen erhöhte Anforderungen an den Erstellungsprozess. Das ist einer der Gründe dafür, dass derzeit ein Wandel hin zur Industrialisierung der Softwareerstellung stattfindet. Da in der Softwareindustrie traditionell im Sinne einer Werkstatt- bzw. Einzelfertigung entwickelt wird, erfordert die steigende Nachfrage nach Software eine rationellere Gestaltung des Softwareerstellungsprozesses. Dieser Handlungsbedarf wird zusätzlich durch die fortschreitende Standardisierung in der Softwaretechnik und die Globalisierung der Märkte verstärkt. Anstatt monolithischer Einzelfertigungen können Vorprodukte ausgewiesener Zulieferer, z.B. in Form von Komponenten, zu größeren Softwarebausteinen integriert werden. Über gemeinsame Standards, beispielsweise technologische Plattformen, können somit Synergieeffekte durch Spezialisierung, Automation und rationellere Beschaffungsmaßnahmen erzielt werden. In der Literatur werden daher immer häufiger neuartige Methoden und Techniken der Spezialisierung, Arbeitsteilung und Zusammenarbeit (engl. *Collaboration*) bei der Softwareerstellung behandelt. Allerdings hinkt die Einführung von kollaborativen Softwareerstellungsprozessen und -werkzeugen in der Praxis anderen Branchen, wie der Automobilindustrie, noch sehr weit hinterher. Jedoch erwägen einer IDC-Studie von 2003 zu Folge bereits über 50% der Softwareunternehmen kollaborative Methoden und Werkzeuge einzuführen, vor allem in den Bereichen *Requirements Engineering*, Changemanagement und Entwurfsspezifikation [Webs03].

Ziel dieses Beitrags ist es zum einen, einen generischen Klassifikationsrahmen für die teilweise sehr unterschiedlichen Ansätze zur kollaborativen Softwareerstellung zu konstruieren und somit ein konsolidiertes Verständnis des Kollaborationsbegriffs und der damit verbundenen Implikationen für die Softwaretechnik zu schaffen. Zum anderen sollen die gegenwärtig existierenden Ansätze und Forschungsarbeiten in diesem Bereich in übersichtlicher Form in diesen Klassifikationsrahmen eingeordnet und charakterisiert werden.

Im folgenden Abschnitt wird der Begriff der kollaborativen Softwareerstellung zunächst definiert, und die einzelnen Dimensionen des Klassifikationsrahmens vorgestellt. Abschnitt 3 klassifiziert zunächst Ansätze zur kollaborativen Softwareerstellung auf Projektebene, wohingegen Abschnitt 4 zwischenbetriebliche Zusammenarbeitsformen auf Unternehmensebene vorstellt. Der letzte Abschnitt fasst noch einmal die wesentlichen Ergebnisse zusammen und diskutiert zukünftige Herausforderungen und Forschungsfragen.

2. Kollaborative Softwareerstellung

In diesem Abschnitt sollen zunächst, ausgehend von der dargelegten Problemstellung in der Softwareindustrie, die begrifflichen Grundlagen für die Analyse der existierenden Ansätze geschaffen werden. Hierbei kommt es im Wesentlichen auf die Verwendung des Kollaborationsbegriffs bei der Softwareerstellung in Verbindung mit den speziellen Eigenschaften von Software als „Objekt“ der Kollaboration an. Um untersuchen zu können, wie Software kollaborativ erstellt werden kann, werden die relevanten Dimensionen zur Kategorisierung auf Basis existierender Ansätze aus Literatur und Praxis vorgestellt. Daraus abgeleitet wird der hier eingesetzte Klassifikationsrahmen als Ganzes beschrieben und vor dem Hintergrund unterschiedlicher Betrachtungsebenen von Softwareprojekten diskutiert.

2.1 Begriffsbestimmung

Die in der Einleitung dargelegte Problemstellung bei der Erstellung betrieblicher Software impliziert die Notwendigkeit einer stärkeren Industrialisierung des Softwareerstellungsprozes-

ses, d.h. einer stark arbeitsteiligen Vorgehensweise in Verbindung mit der Automation möglichst vieler Arbeitschritte [Gree04]. Eine stärkere Arbeitsteilung und Zusammenarbeit von Individuen, Arbeitsgruppen und rechtlich selbständigen Unternehmen impliziert wiederum eine stärkere Spezialisierung und Fokussierung auf deren jeweiligen Kernkompetenzen [Port80]. Die im Zusammenhang mit arbeitsteiliger Softwareerstellung verwendeten Begriffe, insbesondere die der Kollaboration und der Kooperation, werden im Folgenden sowohl aus etymologischer als auch pragmatischer Perspektive erläutert. Den Abschluss bildet eine Darstellung verwandter organisatorischer und technischer Aspekte der Zusammenarbeit in der Softwareentwicklung.

2.1.1 Etymologische Betrachtung

Als Grundlage für das Verständnis der in dieser Arbeit verwendeten Begriffe und Definitionen wird einer pragmatischen Betrachtung der Zusammenarbeitskonzepte in der Software Engineering-Literatur und Praxis eine etymologische Analyse vorangestellt.

Der deutsche Begriff der **Zusammenarbeit** umschreibt laut Duden „Tätigkeiten, die auf ein Ziel hin vereinigt sind“ [Dude04]. Sowohl in der softwaretechnisch als auch in der betriebswirtschaftlich geprägten Literatur werden in diesem Kontext immer häufiger auch die Begriffe „Kollaboration“ und „Kooperation“ verwendet, um Zusammenarbeit und arbeitsteilige Vorgänge zu beschreiben. Ein besonderes Gewicht kommt bei dieser Betrachtung der englischen Sprache zu, da ein Großteil der bisherigen Forschungsarbeiten auf Englisch publiziert wurde und somit den fachlichen Sprachgebrauch prägt.

Die Duden-Redaktion umschreibt den Begriff der **Kollaboration** in der 23. Auflage der deutschen Rechtschreibung noch sowohl durch „Mitarbeit“ als auch „Zusammenarbeit mit dem Feind“ [Dude04]. Laut dem Duden-Synonymwörterbuch kann das entsprechende Verb „kollaborieren“ im deutschen Sprachgebrauch mit den Ausdrücken „am gleichen/selben Strang ziehen“, „gemeinsame Sache machen“, „Hand in Hand arbeiten“, „kooperieren“ oder „zusammenarbeiten“ ersetzt werden [Dude04a].

Kollaborieren geht auf die lateinische Wurzel „cum laborare“ zurück, was im Lateinischen zu com- bzw con-laborare assimiliert wird. Das Nomen Kollaboration leitet sich von der Partizip Perfekt „*collaboratus*“ des spätlateinischen Verbs „*collaborare*“ ab, das ursprünglich „mitarbeiten“ bzw. „zusammen arbeiten“ bedeutet. „Labor“ bedeutete im Lateinischen „Arbeit“ sowie „Anstrengung“, „Mühe“ und „Last“ [Dude01, NODE01].

Aus „*collaborare*“ wurde im Französischen „*collaborer*“, was heute „*travailler en commun*“ (gemeinsam arbeiten) bzw. „*agir en tant que collaborateur*“ (in der Eigenschaft als Kollaborateur handeln) bedeutet. „*Collaborateur*“ wiederum wird im französischen Sprachgebrauch als „*personne qui collabore à une œuvre commun*“ (Person, die an einem gemeinsamen Werk mitarbeitet) bzw. im historischen Kontext der deutschen Besatzung in Frankreich von 1940 bis 1944 (Vichy-Regime) als „*Francais partisan de l’envahisseur allemand*“ (französicher Anhänger der deutschen Besatzer) definiert. Daher hat das französische Nomen „*Collaboration*“ vor allem im Französischen neben der neutralen Bedeutung „Zusammenarbeit“ die negative Konnotation des „*mouvement, attitude des collaborateurs*“ (Bewegung, Haltung der Kollaborateure) [Robe05]. Das Fremdwort wurde im Deutschen erstmals während des 2. Weltkriegs entlehnt [Dude01].

Im Englischen wird unter „*Collaboration*” im Wesentlichen „*the act of working together with another person or group to achieve something*“ verstanden. Allerdings hält sich auch im Englischen immer noch die historisch entstandene negative Konnotation der Unterstützung feind-

licher Truppen im Kriegsfall [LDCE03][2]. Im *New Oxford Dictionary of English* wird der Kollaborationsbegriff alternativ als „*the action of working with someone to produce or create something*“ bzw. auch mit der historische Konnotation als „*traitorous cooperation with an enemy*“definiert [NODE01]. Das *Collins English Dictionary* (CED) umschreibt „*Collaboration*“ zum einen als „*the act of working with another or others on a joint project* “ und zum anderen als „*something created by working jointly with another or others*“. Hier drückt „Kollaboration“ nicht nur den Prozess sondern auch das Produkt der Zusammenarbeit aus. Auch im *Oxford Advanced Learner's Dictionary* (OALD) kommt diese Bedeutung zum Tragen: „a piece of work produced by two or more people or groups of people working together“ [OALD05]. Im CED wird ebenfalls „*the act of cooperating as a traitor*“ als mögliche negative Bedeutung aufgeführt [Coll05].

Das *Merriam-Webster Dictionary*, das in erster Linie den US-amerikanischen Sprachgebrauch widerspiegelt, spricht dem Verb „*collaborate*” die drei grundlegende Bedeutungen zu: erstens „*to work jointly with others or together especially in an intellectual endeavor*“, zweitens „*to cooperate with or willingly assist an enemy of one's country and especially an occupying force*” und drittens „*to cooperate with an agency or instrumentality with which one is not immediately connected*“ zu [MeWe01]. Im Merriam-Webster online Thesaurus wird „*Collaboration*“ mit „*the state of having shared interests or efforts*“ bzw. „*the work and activity of a number of persons who individually contribute toward the efficiency of the whole*“ und wird hier synonym mit dem Begriff „*teamwork*“ verwendet.[3] Neu ist bei dieser Definition, dass Kollaboration nicht nur als Prozess bzw. Produkt der Zusammenarbeit sondern auch als Zustand, nämlich das Bestehen gemeinsamer Interessen und Bemühungen, verstanden wird.

Kollaboration bzw. Kollaborieren impliziert somit, historisch und über mehrere Sprachen betrachtet, die Zusammenarbeit mehrerer Personen oder auch Gruppen mit gemeinsamen Zielvorstellungen, beispielsweise in Form eines Projekts. Dabei kann Kollaboration einen Prozess, ein Produkt oder auch einen Zustand ausdrücken[4].

Kooperation hingegen wird im Deutschen häufig mit „Mitwirkung“ oder „Zusammenarbeit“ gleichgesetzt [Dude04]. Das Verb „kooperieren“ kann neben den oben genannten Synonymen des Verbs „kollaborieren“ im gehobenen Sprachgebrauch auch mit „zusammenwirken“ gleichgesetzt werden. Das Nomen „Kooperation“ gilt als synonym zu „Gemeinschaftsarbeit“, „Gemeinschaftsproduktion“ bzw. „Koproduktion“, „Teamarbeit“ und „Teamwork“. In einem bildungssprachlichen Kontext kann man es auch mit „Kollaboration“ bzw. im wirtschaftlichen Sinn mit „Verbund“ gleichsetzen [Dude04, Dude04a]. Im Kontext der Wirtschaftsinformatik bezeichnet Kooperation einen: „sozialen Prozess zwischen mehreren Aufgabenträgern zur Erreichung gemeinsamer Ziele“ [HeHR04, S. 379]). Der Kooperationsbegriff in der Organisationslehre bzw. allgemeinen Betriebswirtschaftslehre steht in der Regel für die Zusammenarbeit zweier oder mehrerer Unternehmen (als Gegensatz zum Wettbewerb).

Das Wort „Kooperation“ entstammt ursprünglich ebenfalls aus dem Lateinischen und kommt von dem passivischen Verb „*co-operari*“ (zusammen arbeiten, wirken, werktätig sein, beschäftigt sein, sich abmühen“ bzw. mitarbeiten/-wirken) [Dude01]. Aus dem Partizip Perfekt „cooperatus“ entstand das Nomen „*Cooperatio*“ (Mitwirkung, Mitarbeit), aus dem später im Französischen „coopération“ wurde. Das Nomen „Operatio“ steht im Lateinischen für das Arbeiten bzw. die Verrichtung [NODE01, Dude01, Robe05]. Im Englischen bezeichnet “Co-

[2] Kommen antagonistische Aspekte und/oder Konkurrenzbeziehungen im Spiel, sprechen manche Autoren auch bewusst von „Kollaboration“, da der Begriff aus der Zeit des zweiten Weltkriegs immer noch mit der negativen Konnotation der „Zusammenarbeit mit dem Feind“ belegt ist.

[3] http://www.m-w.com/cgi-bin/thesaurus?book=Thesaurus&va=collaboration&x=0&y=0 (Abruf am 16.11.05)

[4] Wenn Individuen mit Organisationen oder Organisationen untereinander zusammenarbeiten, kann Kollaboration implizieren, dass dies nicht notwendigerweise freiwillig geschieht bzw. normalerweise kein Kontakt zu dieser Organisation besteht.

operation" die „*assistance or willingness to assist*" oder „*joint operation or action*", wobei „*Operation*" als „*act, process, or manner of functioning or causing to function*" bzw. verstanden wird [Coll05]. Das *New Oxford Dictionary of English* definiert Kooperation als „*the process of working together to the same end*" [NODE01] und setzt damit ein gemeinsames Ziel der Zusammenarbeit voraus. Ähnlich wird die Kooperation auch im OALD und im *Longman Dictionary* dargestellt: „*the fact of doing something together or of working together towards a shared aim*" bzw. „*when you work with someone to achieve something that you both want*" und „*willingness to do what someone asks you to do*" [LDCE03, OALD05]. *Merriam Webster* definiert Kooperation für den nordamerikanischen Sprachraum als „*the action of acting or working with another or others*", „*common effort*" bzw.„*association of persons for common [mutual] benefit*" [MeWe01].

Speziell unter **zwischenbetrieblicher Kooperation** wird das „Zusammenwirken von Betriebswirtschaften, bei welchem durch einzelbetriebliche Ausgliederung und kollektive Ausübung von Aufgaben die wirtschaftliche Situation der kooperierenden Betriebswirtschaften verbessert werden soll" [Gehr78]. Nach Knoblich beruht diese Form der zwischenbetrieblichen Zusammenarbeit auf „freiwilligen, vertraglichen Vereinbarungen" und involviert „mindestens zwei rechtlich und wirtschaftlich selbständig bleibende Unternehmungen in bestimmten unternehmerischen Teilbereichen" [Knob69]. Laut Bidlingmaier liegt zwischenbetriebliche Kooperation immer dann vor, „wenn zwei oder mehrere Unternehmungen aufgrund freiwilliger vertraglicher Abmachungen gewisse Aufgaben gemeinschaftlich erfüllen", um dadurch einen vergleichsweise höheren „Grad der Zielerfüllung" zu erreichen [Bidl68]. Auf zwischenbetriebliche Formen der Zusammenarbeit und unterschiedliche Vertragsarten wird in Abschnitt 4 näher eingegangen.

Wie auch der Begriff „Kollaboration" drückt auch „Kooperation" im Wesentlichen die Zusammenarbeit mehrerer Personen oder Organisationseinheiten aus. Vor allem im englischen Sprachgebrauch wird dabei eine gemeinsame Zielsetzung vorausgesetzt. Im (betriebs-) wirtschaftlichen Sinn wird Kooperation als gegensätzlich zur Wettbewerbssituation betrachtet. Der Kollaborationsbegriff ist zwar auch ergebnisorientiert, identische Ziele werden aber nicht notwendigerweise unterstellt. Tabelle 1 fasst noch einmal die unterschiedlichen Bedeutungen der beiden Begriffe in den betrachteten Sprachen zusammen.

Sprache	Kollaboration	Kooperation
Latein	Zusammen + Arbeiten	Zusammen + Arbeiten/Wirken
Französisch	Gemeinsames Arbeiten	Gemeinsames Arbeiten
Englisch	Zusammenarbeit mehrerer Personen oder Gruppen (mit dem Ziel, gemeinsam etwas zu erreichen)	Zusammenarbeit mehrerer Personen oder Gruppen (mit einem gemeinsamen Ziel)
Deutsch	Zusammenarbeit, Ziehen am selben Strang, gemeinsame Sache machen, Hand in Hand arbeiten, Kooperation	Zusammenarbeit, Ziehen am selben Strang, gemeinsame Sache machen, Hand in Hand arbeiten, Verbund, Kollaboration

Tabelle 1: Etymologische Betrachtung des Kollaborationsbegriffs

Aufgrund der Betrachtung der kompletten Etymologie der beiden Begriffe „Kollaboration" und „Kooperation" und deren im Wesentlichen synonymen Verwendung im englischsprachigen Sprachgebrauch werden sie im weiteren Verlauf dieser Arbeit auch austauschbar verwendet. Ebenso wird der Begriff „Zusammenarbeit" gleichbedeutend eingesetzt. Kommen von den jeweiligen Autoren weitere Konnotationen hinzu, wird dies gesondert ausgewiesen.

2.1.2 Pragmatische Betrachtung

Im Folgenden wird die Verwendung der Begriffe „Kollaboration“ und „Kooperation“ in der anhand der in dieser Arbeit betrachteten Ansätze aus der SE-Literatur analysiert. In der englischsprachigen Literatur aus den Bereichen *Computer-Supported Cooperative Work* bzw. *Computer-Supported Collaborative Work* (CSCW) und *Groupware* werden bei der verteilten Bearbeitung eines gemeinsamen Objekts, bei dem es sich nicht notwendigerweise um Software handelt, *Cooperation* und *Collaboration* ebenfalls synonym verwendet.

Diese Ansicht wird auch in der Software Engineering (SE) *Community* geteilt, wo *Collaborative Software Development* (CSD) und *Collaborative Software Engineering* (CSE) in erster Linie die Beteiligung mehrerer Akteure oder Stakeholder am Softwareerstellungsprozess implizieren (siehe beispielsweise [DLPH98]). Tabelle 2 gibt einen Überblick über die in der Literatur verwendeten Begriffe und Ausdrücke.

Begriff	Quelle(n)
„Collaboration“ im SE-Kontext	[LoRo93], [ChSp00], [LaBF00], [Grue00], [AuBS02], [CDHP03], [CHRP03], [Micr04]
„Cooperation“ im SE-Kontext	[FrLa98], [GrBr03], [MaSo04]
„Coordination“ im SE-Kontext	[MaCr94], [ObWS94], [KrSt95], [AnZm01]
„Collaborative Software Development“	[KTCB92], [BoBo94], [DLPH98], [HaWo01], [HeRe01], [CaLM02], [BoBr03], [DRMP03], [Webs03], [WMMP03], [CoCl04]
„Collaborative Software Engineering“	[GHRC97], [Gold02], [BNRS03], [CoCh03], [Cook04]
„Collaborative Software Design“	[GSKR99]
„Cooperative Software Engineering“	[BKMS95], [GaRi02]
„Cooperative Software Development“/ „Kooperative Softwareentwicklung“	[AlWe98], [AlPo99], [Altm99]

Tabelle 2: Übersicht über die Begriffswahl in der Literatur

Vergleicht man die Häufigkeiten der Nennungen des Kollaborationsbegriffs mit der des Kooperationsbegriffs im SE, lässt sich auch über einen Zeitraum von über 20 Jahren seit der Einführung des „*Collaborative Software Development*“ [KTCB92], zumindest in englischsprachigen Publikationen, eine klare Tendenz hin zur Verwendung des Begriffs „Kollaboration“ im Allgemeinen und „Kollaborativer Softwareentwicklung“ (CSD) im Speziellen feststellen.

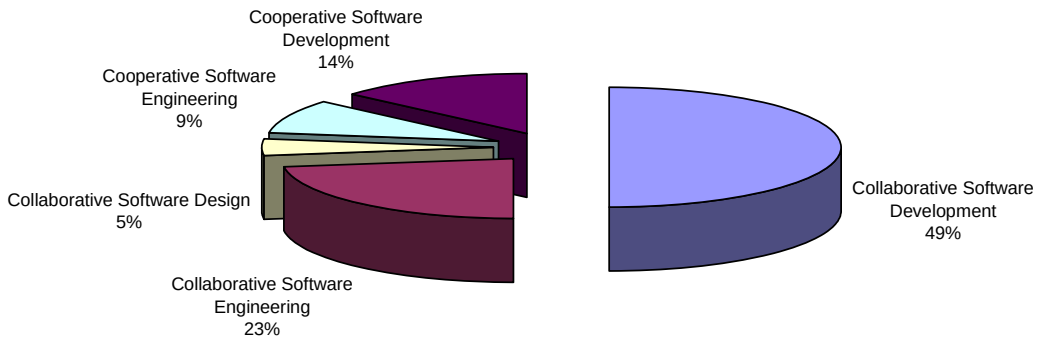

Abbildung 1: Verteilung der Begriffswahl für den arbeitsteiligen Prozess in der Literatur

Abbildung 1 soll dieses Mengengerüst anhand der Verteilung der Begriffswahl in den in Tabelle 2 vorgestellten Publikationen für den Prozess der arbeitsteiligen Softwareerstellung exemplarisch darstellen. Hierbei zeigt sich, dass die Begriffe „*Collaborative Software Development*“ mit 49% und „*Collaborative Software Engineering*“ mit 23% klar überwiegen und damit in nahezu drei Viertel der relevanten Forschungsarbeiten in diesem Bereich Verwendung finden. Zum erweiterten Kontext der kollaborativen Softwareentwicklung gehören auch die Arbeiten aus den Bereichen „*Distributed Software Development*“ [DoKG99, LlRA02], „*Global Software Development*” [HeMo01, HeMo03] und „*Open Source Software Development*“ [FeFi01, FoBa02, MoFH02, Henk03a, SpSz04] (zur Bedeutung von Ansätzen aus dem *Open Source*-Bereich siehe auch Abschnitt 3.3.3 sowie 4.2.4). Im weiteren Verlauf der Arbeit wird auf diese Ansätze gesondert eingegangen, sofern sie für kollaborative Erstellungsprozesse geeignet sind.

2.1.3 Organisatorische und technische Aspekte der Kollaboration

Altmann bezeichnet den Softwareentwicklungsprozess per se bereits als „zeitlich und räumlich verteilten, kooperativen Arbeitsprozess“ [Altm99, S. i]. Kollaborative Softwareentwicklung beinhaltet des Weiteren „die Abdeckung der **Kommunikations**- und **Koordinationsbedarfe** innerhalb eines Softwareentwicklungsprozesses, die für die Planung, Durchführung und Abstimmung aller aufgabenbezogenen, zeitlich und räumlich verteilten Aktivitäten erforderlich sind“ und „dementsprechend alle prozess- und produktbezogenen Aktivitäten aller Beteiligten, deren gemeinsames Ziel die Erstellung eines [gemeinsamen] Softwareprodukts ist“ [Altm99, S. 20, BoBr03, HeHR04].

Industrielle Arbeitsorganisation, d.h. aufgabenbezogene Verteilung und Zusammenarbeit, erfordert zunehmend auch komplexere **Koordinationsmechanismen** und Werkzeuge zur Abstimmung der verteilten Tätigkeiten im Sinne eines gemeinsamen Gesamtergebnisses [AlPo99, Altm99, MeSz03, StHa04]. Bei der Softwareentwicklung müssen verteilte Aufgaben über unterschiedliche Lokationen, Zeitzonen, Unternehmen und Kulturen sowie über einzelne Prozessstufen in der Wertschöpfungskette hinweg koordiniert werden. Diese Zusammenhänge versucht u.a. die *Coordination Theory* zu erklären. Hierbei wird, bezogen auf SE, Koordination als Prozess zur Bewältigung von Abhängigkeiten zwischen Aktivitäten und damit zur Erreichung übergeordneter Zielsetzungen gesehen [MaCr94, HeMo03]. Zur Koordination und Durchführung dieser verteilten, industriellen Leistungserstellungsprozesse kann auch ein erhöhter Kommunikationsaufwand entstehen [BKMS95]. **Kommunikation** beschreibt im SE den gezielten Austausch von Information zwischen den am Entwicklungsprozess beteiligten Akteuren, sowohl auf Projekt- als auch auf Unternehmensebene [LaBF00]. Zur Implementierung unterschiedlicher Kollaborationsformen sowie der dazu nötigen Koordinationsmechanismen und Kommunikationskanäle kommt der adäquaten **Werkzeugunterstützung** eine zentrale Stellung in der Softwaretechnik zu [BKMS95]. Dieser Zusammenhang spiegelt sich dann auch im Klassifikationsrahmen (Abbildung 2 in Abschnitt 2.3) wider. Überdies hat Software nicht nur in der Rolle eines Werkzeugs sondern auch als Objekt der Zusammenarbeit anhand ihrer Eigenschaften als immaterielles Gut einen sehr großen Einfluss auf die Gestaltungsmöglichkeiten der Arbeitsteilung, Koordination und Kommunikation [DRCM04].

2.2 Dimensionen der kollaborativen Softwareerstellung

Im Folgenden werden die in der Literatur häufig diskutierten Dimensionen der Verteilung sowie der Richtung als auch der Intensität des Softwareentwicklungsprozesses vorgestellt. Verteilung lässt sich hierbei in organisatorische, räumliche und zeitliche Unterdimensionen unterteilen. Dadurch können die unterschiedlichen Ansätze, unabhängig von der Betrachtungsebene, in einen gemeinsamen Klassifikationsrahmen der kollaborativen Softwareerstellung eingeordnet, charakterisiert und unterschieden werden.

2.2.1 Organisatorische Verteilung

Die organisatorische Verteilungsdimension wird in der Literatur häufig auch als „*unit of distribution*“ mit den Ausprägungen **inter-organisatorisch** (zwischenbetrieblich) und **intra-organisatorisch** bezeichnet [FrLa98, S. 148]. Bei intra-organisatorischen Formen kann des Weiteren gruppenübergreifende (*inter-team*) und gruppeninterne (*intra-team*) Zusammenarbeit unterschieden werden [CuKI88, S. 1269]. Ein hoher Grad der organisatorischen Verteilung, der die Zusammenarbeit von vielen organisatorisch getrennten Gruppen impliziert, hat zahlreiche organisatorische, ökonomische und juristische Implikationen für den Softwareerstellungsprozess. Auf der Projektebene wird in der Regel entlang identischer Zielsetzungen zusammengearbeitet. Zwischenbetriebliche Formen der Zusammenarbeit werden häufig unter dem Begriff „Kooperation“ subsumiert [HeHR04, S. 379], wobei, entgegen der ursprünglichen Definition von Kooperation, auch möglicherweise antagonistische Zielsetzungen einzelner Unternehmen zum Tragen kommen [Altm99, HeSi93]. Man findet die Unterscheidung der organisatorischen Zusammensetzung von Arbeitsgruppen und Softwareprojekten in der Literatur häufig im Kontext räumlicher und zeitlicher Verteilung bei der Kategorisierung von *Groupware* bzw. CSCW-Werkzeugen [StHa04, S. 422f.].

2.2.2 Räumliche Verteilung

Bei der räumlichen oder auch geografischen Verteilung im Softwareentwicklungsprozess unterscheidet man **räumlich nahe zusammenarbeitende** (*co-located*) und **räumlich verteilte** (*distributed*) Organisationseinheiten. Eine räumliche Verteilung kann u.a. durch das Zusammenarbeiten über mehrere Standorte eines oder mehrerer Unternehmen oder Firmenzusammenschlüsse bedingt sein. Nach Cain und Coplien wirkt sich der fehlende persönliche Kontakt bei der täglichen Zusammenarbeit selbst bei minimalen Distanzen und räumlichen Barrieren unmittelbar auf Kommunikations- und Zusammenarbeitsformen aus [CaCo96, Alle77]. Zahlreiche Arbeiten im Bereich *Global Software Development* (GSD) und *Distributed Software Development* (DSD) beschäftigen sich daher speziell mit den sozialen und softwaretechnischen Implikationen räumlich verteilter Entwicklungsprojekte [HeMo01, HeMo03]. Arbeiten in diesen Projekten überdies mehrere rechtlich selbständige Unternehmen zusammen (inter-organisatorische Verteilung, s.o.) und werden Aktivitäten über mehrere Zeitzonen und Kulturräume hinweg verteilt, kann sich dies insbesondere negativ auf Kommunikations- und Koordinationsaufwand auswirken [HeGr99, CaAg01, TCKO00].

2.2.3 Zeitliche Verteilung

Die bekannten Klassifikationen der Zusammenarbeit aus dem *Groupware*-Bereich unterscheiden neben den beiden bisher genannten Dimensionen immer auch, ob die Zusammenarbeit, dabei insbesondere die Kommunikation, zeitversetzt stattfindet. Informationen und Softwareartefakte können entweder IT-gepuffert (**asynchron**) oder gleichzeitig (**synchron)** ausgetauscht bzw. bearbeitet werden [ElGR91, StHa04, S. 422]. Teilweise wird unter „unmittelbarer Zusammenarbeit“ sowohl zeitlich als auch räumlich simultane Zusammenarbeit (räumliche Nähe/synchron) subsumiert [Altm99]. Herrscht kein ständig unmittelbarer Synchronismus der Zusammenarbeit, wie beispielsweise beim *Application Sharing*, wird häufig auch von nebenläufiger Softwareentwicklung gesprochen (*Concurrent Software Engineering*) [Balz98, S. 126ff., Altm99, S. 12]). Das Prinzip des *Concurrent Engineering* (CE) wurde aus den Gestaltungsprozessen anderer Ingenieursdisziplinen in die Softwareentwicklung übernommen [DeRi93, Kell98].

2.2.4 Richtung

Der Softwareerstellungsprozess lässt sich prinzipiell in einzelne Prozessstufen bzw. inhaltlichen Disziplinen unterteilen, wie z.B. „Analyse und Design“ und „Implementierung“ beim

Rational Unified Process [Kruc04]. Somit können bezüglich des Prozesses bzw. auf Unternehmensebene bezüglich der Wertschöpfungsstufe zwei orthogonale Richtungen der Zusammenarbeit unterschieden werden. Innerhalb einzelner Prozess- oder Wertschöpfungsstufen spricht man von **horizontaler** Kollaboration und dementsprechend bei stufenübergreifender Zusammenarbeit von **vertikaler** Kollaboration. Je nach Perspektive des betrachteten Unternehmens sprich man vorwärts oder rückwärts gelagerter vertikaler Kollaboration, da dies Ausrichtung hin zum Kunden bzw. zur Lieferantenseite ausdrückt [ThLo04]. Nach Porter kann die Softwareerstellung als zwischenbetrieblicher Wertschöpfungsprozess oder auch Wirtschaftskreislauf aufgefasst werden, wobei sich die „Richtung" dann an grob granulareren Wertschöpfungsstufen in der Softwareindustrie wie z.B. „Erstellung von Anwendungs- und Systemkomponenten" und „Anwendungsintegration" orientiert [Port80, HeSi93, MeSz03]. Im Zuge der aufkommenden Industrialisierung und Spezialisierung in der Softwareerstellung sprechen einzelne Autoren, bereits von zwischenbetrieblichen „Software-Wertschöpfungsketten" (*Software Value Chains*) [MeSz03]. Bei der unternehmensübergreifenden Zusammenarbeit innerhalb des so genannten „Softwareökosystems" (siehe Abschnitt 4.3) ergibt sich die Richtung aus dem Verhältnis der Kollaborationspartner in der Wertschöpfungskette [HeSi93, MeSz03]. Horizontale Kollaborationsbeziehungen finden sich zwischen Unternehmen derselben Branche bzw. derselben Wertschöpfungsstufe, wohingegen vertikale Beziehungen Unternehmen verbinden, die in einer Art „Lieferanten-Kunden-Verhältnis" stehen [Gehr78, HeSi93, S. 60f.]. Bei horizontalen Beziehungen sind zusätzlich konfligierende Zielsetzungen auf Grund überlappender Kompetenzen und Märkte zu erwarten [HeSi93, MeSz03, Henk03].

2.2.5 Intensität

Die Zusammenarbeit im Softwareerstellungsprozess kann unterschiedlich **niedrige** oder **hohe** Intensität aufweisen. Dies manifestiert sich u.a. daran, ob tatsächlich gemeinsam an einem Objekt gearbeitet wird, oder die Zusammenarbeit in erster Linie mittelbar über den systematischen Austausch von Wissen, Dienstleistungen und halbfertigen Softwareartefakten, wie z.B. Softwarekomponenten, erfolgt [MeSz01, MeSz03]. Hierbei besteht ein Zusammenhang zur zeitlichen Verteilung der Kollaboration, da synchrone Formen der Zusammenarbeit in der Regel intensiver sind als asynchrone. Bezogen auf die rein innerbetriebliche, projektorientierte Perspektive, lassen sich beim Austausch zwischen einzelnen Akteuren und Teilprojektteams wissens- und/oder zusammenarbeitsintensive Beziehungen differenzieren [DLPH98]. In Folge zwischenbetrieblicher Zusammenarbeit innerhalb der Softwarebranche entstehen unternehmensüberspannende „Wertschöpfungsnetze". Innerhalb dieser lassen sich neben der oben genannten Richtung der Kollaborationsbeziehungen auch deren unterschiedliche Intensitäten differenzieren (vgl. Abbildung 3). Neben der eigentlichen Zusammenarbeitsintensität unterscheiden Lawrence et al. des Weiteren Aspekte der Mitwirkung in einer Kollaborationsbeziehung (*Involvement*) und der Einbettung (*Embeddedness*) in weitere interorganisatorische Netzwerke [LaHP02, HaPL03]). Nach Heinzl und Sinß (1993) lässt sich die Intensität der zwischenbetrieblichen Zusammenarbeit überdies aus juristischer Sicht dadurch quantifizieren, inwieweit es sich um eine lose oder eine vertraglich bzw. kapitalmäßig geregelte Beziehung handelt [HeSi93, S. 61]. Die Intensität der Zusammenarbeit ist immer auch mit zeitlichen Aspekten verbunden. So spielen hierbei insbesondere die Häufigkeit bestimmter Kollaborationsbeziehungen (einmalig, sporadisch, regelmäßig oder dauerhaft), die zeitliche Befristung sowie der Zeithorizont (kurz-, mittel- oder langfristig) eine entscheidende Rolle [ThLo04].

Anhand der in der Literatur identifizierten Dimensionen der kollaborativen Softwareerstellung sollen die im Folgenden untersuchten Ansätze charakterisiert und eingeordnet werden. Tabelle 3 stellt noch einmal die hier relevanten Dimensionen mit den entsprechenden Ausprägungen und der relevanten Literatur im Überblick dar.

Dimension	Ausprägungen	Literaturquellen
Räumliche Verteilung	lokal, verteilt	[ScSU01], [LBSZ01], [StHa04], [Altm99]
Zeitliche Verteilung	synchron, asynchron	[ScSU01], [HeRe01], [LBSZ01], [StHa04], [Altm99]
Organisatorische Verteilung	intra-, interorganisatorisch	[FrLa98], [ScSU01], [LBSZ01], [StHa04], [HeSi93], [HeHR04]
Richtung	horizontal, vertikal	[Port85], [MeSz03], [HeSi93]
Intensität	niedrig ... hoch	[BeSc89], [HeSi93], [FrLa98], [MeSz01], [HeRe01], [LaHP02]

Tabelle 3: Dimensionen der kollaborativen Softwareerstellung

2.3 Klassifikationsrahmen

Der Sinnzusammenhang zwischen Kollaboration (Arbeitsteilung und Zusammenarbeit), Koordination und Kommunikation, soziotechnische Aspekte (Menschen) sowie dem Bedarf an unterstützenden Werkzeugen zusammen mit den Dimensionen sind in dem in Abbildung 2 dargestellten **Schalenmodell der kollaborativen Softwareerstellung** verdeutlicht. Basierend auf dem eingeführten Begriffssystem soll dieses Modell als Untersuchungsrahmen zur Klassifikation existierender Ansätze im Bereich der kollaborativen Softwareerstellung dienen. Das Modell spiegelt die Dimensionen wider, anhand derer man die unterschiedlichen Kollaborationsformen charakterisieren kann. Die einzelnen „Schalen" stellen die dimensionsübergreifenden Aspekte der kollaborativen Softwareerstellung dar. Die Abbildung soll verdeutlichen, dass neben organisatorischen Aspekten rund um die Kollaboration, nämlich betriebliche Aufgaben und Tätigkeiten, die Koordination und Kommunikation erfordern, auch soziotechnischen Aspekte eine bedeutende Rolle spielen. Ein soziotechnisches System beschreibt hierbei die Interaktion menschlicher Akteure mit den eingesetzten Werkzeugen. Dieses Zusammenwirken von Mensch, Aufgabe und Technik bildet gleichzeitig die Konstituenten eines Informationssystems und wird auch in Abschnitt 3 verwendet, um die analysierten Ansätze entsprechend der jeweiligen Schwerpunkte zu kategorisieren bevor die unterschiedlichen Dimensionen betrachtet werden.

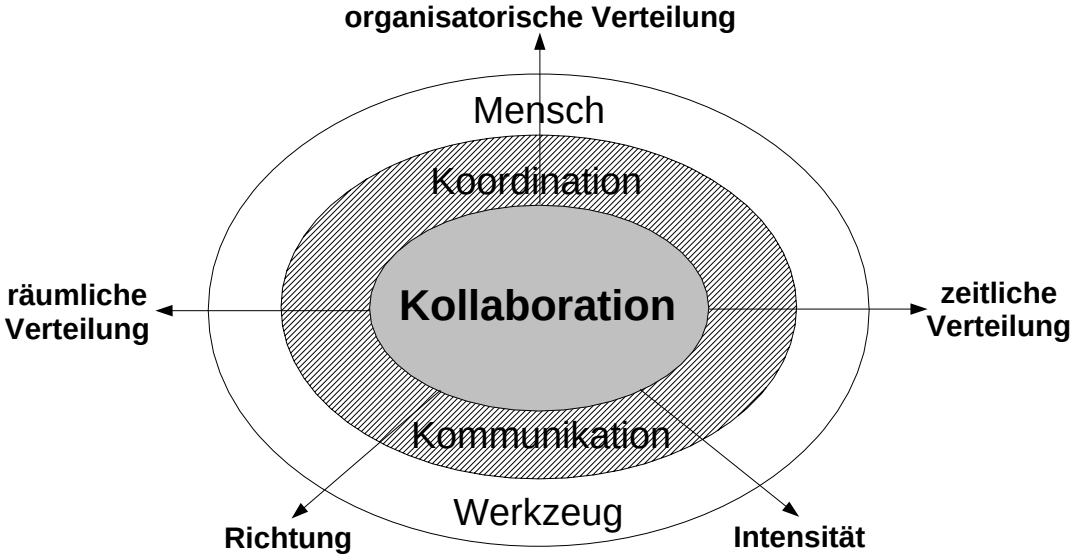

Abbildung 2: Klassifikationsrahmen für Ansätze zur kollaborativen Softwareerstellung

Bei der Begriffserklärung und der Vorstellung der einzelnen Dimensionen wurde bereits kurz auf die unterschiedliche Formen der Zusammenarbeit eingegangen. In Anlehnung an das *Layered Behavioral Model* von Curtis et al. [CuKI88], welches mehrere Stufen von Untersuchungsebenen mit unterschiedlichen Untersuchungsgegenständen für die Analyse von Softwareprojekten vorstellt (siehe Tabelle 4), werden in dieser Arbeit zwei grundsätzliche Untersuchungsebenen unterschieden, nämlich (a) die projektbezogene und (b) die unternehmensbezogene Ebene.

Untersuchungsebene	Untersuchungsgegenstand
Individuum	Kognition und Motivation
Arbeitsgruppe	Gruppendynamik
Projekt	Gruppendynamik
Unternehmen	Organisatorisches Verhalten
Geschäftsumfeld	Organisatorisches Verhalten

Tabelle 4: Layered Behavioral Model, [CuKI88]

Die **Kollaboration auf Projektebene** subsumiert sowohl die Zusammenarbeit von Individuen in Arbeitsgruppen mit mehr als zwei Mitarbeitern (Teams) als auch die Zusammenarbeit zwischen Projektteams. Hierbei liegt der Schwerpunkt auf der individuellen Kognition und Motivation bzw. der Gruppendynamik (siehe auch Faktor „Mensch“ in Abbildung 2). Sowohl in der Praxis als auch in der wissenschaftlichen Literatur werden unterschiedliche Aspekte der Zusammenarbeit, der Koordination und Kommunikation, beispielsweise bei agilen Entwicklungsmethoden, im Rahmen von *Groupware*-gestützten virtuellen Softwareentwicklungsgruppen und global verteilten Entwicklungstätigkeiten im Rahmen von *Off-* und *Nearshore*-Projekten hervorgehoben. Projektbezogene Ansätze zeichnen sich durch Erkenntnisse und Gestaltungsempfehlungen bezüglich der beteiligten Mitarbeiter, des Prozessmodells und der zu verwendenden Werkzeuge aus. Ausgehend von so genannten „Kollaborationsplattformen“, wie *SourceForge* im *Open Source*-Umfeld, und den zahlreichen Entwicklergemeinschaften (*Communities)* entstanden in einigen Branchen unternehmensübergreifende *Communities* mit kommerziellen Zielsetzungen [Henk03, Henk03a]. Diese bilden den Übergang von projektbezogenen hin zu unternehmensübergreifenden Formen der Zusammenarbeit.

Zwischenbetriebliche Kollaboration hingegen beschäftigt sich auf der Unternehmens- bzw. Industrieebene (Geschäftsumfeld bzw. Ökosystem) mit organisatorischem Verhalten und bildet somit den ökonomischen und juristischen Rahmen für kollaborative Softwareerstellungsprozesse auf Projektebene (vgl. Tabelle 4). Dabei werden Ansätze aus der Organisationstheorie, wie z.B. zwischenbetriebliche Wertschöpfungsketten, virtuelle Unternehmungen und Unternehmensnetzwerke, immer häufiger auch vor dem Hintergrund der Softwareerstellung betrachtet [HeSi93, MeSz01, MeSz03]. Exemplarische Fragestellungen sind hierbei gegenseitiges Vertrauen, konfligierende Zielsetzungen, gemeinsame Standards sowie Geschäftsmodelle.

In den beiden folgenden Abschnitten werden die existierenden Ansätze aus den beiden Kategorien „Kollaboration auf Projektebene“ und „Zwischenbetriebliche Kollaboration“ differenziert betrachtet und entlang der Dimensionen des Klassifikationsrahmens aus Abschnitt 2.2 analysiert.

3. Kollaboration auf Projektebene

Bei der Softwareerstellung umfasst die Projektebene sowohl einzelne Arbeitsgruppen im Projekt als auch die individuellen Entwickler (vgl. Tabelle 4). Softwareprojekte werden von Personen mit unterschiedlichen Fähigkeiten durchgeführt, sind Instanzen bestimmter Prozessmodelle und erfordern weitere Ressourcen, u.a. auch Entwicklungswerkzeuge zur Prozessunterstützung. Die projektbezogenen Ansätze, die sich in erster Linie in der SE-Literatur wieder finden, lassen sich daher nach den inhaltlichen Schwerpunkten

- vorhandene **Arbeitsgruppen** (soziotechnische Aspekte),
- zugrunde liegendes **Prozessmodell** (methodische Aspekte) und
- eingesetzte **Kollaborationswerkzeuge** (softwaretechnische Aspekte)

gliedern. Den drei Faktoren Mensch, betriebliche Aufgabe und unterstützende Technik kommt bei dieser Unterteilung jeweils eine unterschiedlich zentrale Stellung zu. Die grundlegende Problemstellung aller Ansätze ist die Zusammenarbeit, Koordination und Kommunikation (vgl. Abbildung 2) vieler unterschiedlicher Akteure im Entwicklungsprozess sowie die Nutzung der entsprechenden Werkzeuge mit dem Ziel der systematischen Verbesserung des Zusammenarbeitsprozesses [DLPH98, CoCh03]. Um den kollaborativen Charakter der analysierten Ansätze zu quantifizieren, werden diese, sofern möglich, entlang der fünf Dimensionen aus Tabelle 3 kategorisiert.

3.1 Gruppenorientierte Ansätze

Ausgehend von der Annahme „*in most organizations, software engineering is a team activity*“ [Somm04, S. 86] zeichnen sich die gruppenorientierten Ansätze vor allem durch die zentrale Betrachtung des Faktors „Mensch“ und soziotechnischer Fragestellungen bei der Erstellung von Software aus. Es stehen hierbei in erster Linie die innere Zusammensetzung der Arbeitsgruppen, die Koordinationsmechanismen und Kommunikationsprozesse, beobachtete Interaktionsmuster sowie die individuelle und kollektive Leistung im Mittelpunkt [Andr03].

Nachdem in der Literatur lange Zeit vornehmlich über immer neue Prozessmodelle und Entwicklungswerkzeuge diskutiert wurde, kann in der Praxis und zunehmend auch in der wissenschaftlichen Literatur eine Rückbesinnung auf die soziotechnischen Aspekte der Softwareentwicklung beobachtet werden [Hrus03]. In diesem Bereich sind vor allem personenzentrierte Ansätze für kleine bis mittelgroße Entwicklergruppen, so genannte agile Entwicklungsmethoden [High02]**,** wie beispielsweise ***Extreme Programming*** (XP) [Beck00, AgAl01], zu nennen. Charakteristische Zusammenarbeitsformen in solchen Projekten sind u.a. das paarweise Programmieren (*pair programming*), geteilte bzw. gemeinsame Verantwortlichkeiten und eine starke Einbindung des Kunden vor Ort. Im Gegensatz zu den meisten anderen Ansätzen berücksichtigt XP allerdings keine räumlich und zeitlich verteilte Zusammenarbeit. Auch eine mögliche organisatorische Verteilung der Entwicklungsaufgaben wird bei diesem Ansatz nicht diskutiert.

Des Weiteren existieren zahlreiche Arbeiten, welche sich mit dem Gruppenverhalten und der optimalen Zusammensetzung von Softwareentwicklungsgruppen auseinandersetzen (vgl. Modell von Curtis et al., Tabelle 4), beispielsweise im Hinblick auf Projektmanagement und *Team Performance* [HaWo01]. Im Zentrum stehen hierbei die Kommunikation und die Koordination der Arbeitsabläufe mehrerer, möglicherweise geografisch verteilter, Gruppenmitglieder während des Softwareerstellungsprozesses [BoBr03]. Sowohl die Arbeiten von Andres und Zmud (***Team Coordination Strategies***) als auch von Harrison et al. (***Work Team Diversity***) treffen Aussagen für ein nicht verteiltes Kollaborationsszenario mit hoher Intensität und synchroner sowie asynchroner Kommunikation [AnZm01, HPGF02].

Die Arbeiten von Layzell et al. hingegen untersuchen speziell das Verhalten von verteilt arbeitenden Gruppen (***Distributed Teams***) und geben Richtlinien für die aufgabenorientierte Organisation von verteilten Entwicklergruppen [FrLa98, LaBF00]. Dabei wird von einer synchronen und intensiven Form der Zusammenarbeit ausgegangen. Andere Ansätze, wie z.B. ***Virtual Teams***, betrachten neben den Effekten der geografischen auch die der organisatorischen Verteilung in Form von zwischenbetrieblicher Zusammenarbeit auf Gruppenebene [MRMK00]. Neben Kommunikations- und Koordinationsaspekten innerhalb der einzelnen Entwicklergruppen wird hierbei auch die Zusammenarbeit zwischen mehreren Gruppen innerhalb eines Projekts untersucht [SaLS00] (vgl. [CuKI88] bzw. Tabelle 4). Zusätzlich gehen Jarvenpaa und Leidner speziell auf den Faktor „Vertrauen“ in „*Global Virtual Teams*“ ein [JaKL98, JaLe99].

Tabelle 5 stellt noch einmal die hier dargestellten Ansätze entlang der Dimensionen des Vergleichsrahmens (siehe Tabelle 3) im Überblick dar.

Ansatz	Quelle	Räuml. Vert.	Zeitl. Vert.	Organ. Vert.	Intensität	Richtung
Extreme Programming	**[Beck00]**	lokal	synchron	intra/inter	hoch	horizontal
Team Coordination Strategies	**[AnZm01]**	lokal	synchron/ asynchron	n/a	hoch	vertikal/ horizontal
Work Team Diversity	**[HPGF02]**	lokal	synchron/ asynchron	n/a	hoch	horizontal
Virtual Teams	**[MRMK00], [SaLS00]**	verteilt	synchron/ asynchron	interorg.	niedrig	horizontal
Distributed Teams	**[LaBF00], [FrLa98]**	verteilt	synchron	intra/inter	hoch	vertikal/ horizontal

Tabelle 5: Teambasierte Ansätze zur kollaborativen Softwareerstellung

Neben den bisher vorgestellten Ansätzen mit Handlungsempfehlungen für die Gruppenorganisation in kollaborativen Szenarien existieren überdies Arbeiten, welche häufig auftretende Muster der Kommunikation und Interaktion identifizieren. Diese Muster wurden durch empirische Beobachtung zahlreicher Softwareprojekte identifiziert. Kraut et al. beispielsweise untersuchen in erster Linie Muster für kollaborative Beziehungen zwischen verteilt arbeitenden Gruppenmitgliedern und deren verwendeten Kommunikationsmittel (***Patterns of Contact and Communinaction***) [KrEG88]. In einer aktuellen Serie von Studien analysieren Martin und Sommerville Regelmäßigkeiten bei der Arbeitsorganisation, Aktivitäten und Interaktionen in Softwareprojekten (***Patterns of Cooperative Interaction***). Unter Verwendung einer „ethnomethodischen Perspektive“ werden dabei u.a. Muster für die Zusammenarbeit in Kleingruppen von Softwareentwicklern beschrieben [MaSo04].

Auch die archetypischen Ausprägungen von Entwicklergruppen (***Team Archetypes***) als Ganzes werden in der Literatur differenziert betrachtet. Nachdem Sawyer in früheren Arbeiten zu dem Schluss gekommen war, das weder das Vorgehensmodell noch die Entwicklungswerkzeuge einen Einfluss auf Software-Produktqualität und Gruppenleistung haben [SaGu98], beschäftigt er sich in jüngster Zeit mit den sozialen Prozessen in Softwareprojekten. Dabei wurden drei grundlegende Typen von Entwicklungsprojekten identifiziert. Der sequenzielle Typ orientiert sich an den traditionellen Prinzipien industrieller Produktionstechnik, wohingegen der Typ „Gruppe“ auf sozialpsychologischen Theorien basiert. Der Typ „Netzwerk“ bildet sich um das Softwareprodukt und Aufgaben werden entsprechend der vorhandenen Fähigkeiten an Individuen und Kleingruppen verteilt [Sawy04].

Soziotechnische Fragestellungen sind für den Erfolg von zusammenarbeitsintensiven Softwareprojekten in jeglicher Form entscheidend. Auch wenn bei den bisher vorgestellten Ansätzen zumeist Aspekte wie geografische Verteilung und entsprechende Zusammenarbeitsformen betrachtet werden, existiert ein allgemeiner Tenor, der besagt, dass soziale Interaktion und somit auch physische Treffen für den Gruppen- und Projekterfolg dennoch unerlässlich sind [FrLa98]. Auf entsprechende *Groupware* und andere softwaretechnische Ansätze zur Unterstützung von Entwicklergruppen wird in Abschnitt 3.3 eingegangen.

3.2 Prozessorientierte Ansätze

Prozessorientierte Ansätze betrachten weniger soziotechnische Fragestellungen, sondern legen mehr Wert auf einen formal strukturierten Softwareerstellungsprozess, beispielsweise in Form eines Prozess- oder Vorgehensmodells. Beispiele für umfassende Prozess-Frameworks sind der *Rational Unified Process* (RUP) und das V-Modell XT [Kruc04, BrRa05]. Zur Strukturierung dienen in erster Linie die einzelne Phasen bzw. Disziplinen des allgemeinen Softwarelebenszyklus, der sich entlang vertikaler Wertschöpfungsstufen von den Anforderungen hin zur fertigen Software erstreckt (vgl. Abschnitt 2.2). Entlang dieser Disziplinen werden im Folgenden die prozessorientierten Ansätze charakterisiert und in den Klassifikationsrahmen (siehe Tabelle 3) eingeordnet. Als gemeinsamer Nenner aus unterschiedlichen Vorgehensmodellen werden Arbeiten mit kollaborativem Charakter aus den Bereichen (1) Anforderungsanalyse, (2) Entwurf und Modellierung sowie (3) Implementierung, Test und Wartung unterschieden. In der SE-Literatur findet man auch erste Ansätze zur Unterstützung der horizontalen Zusammenarbeit innerhalb einzelner Phasen, wohingegen bisher keine der etablierten Entwicklungsmethodiken und Prozessmodelle die vertikal durchgängige Zusammenarbeit behandelt [HiKo04].

3.2.1 Kollaborative Anforderungsanalyse

Neue Formen der Zusammenarbeit wurden auf methodischer Ebene bisher vor allem im *Requirements Engineering*, d.h. bei der Anforderungserhebung, -spezifikation und –analyse, weiterentwickelt und erprobt. Da in dieser frühen Phase von Softwareprojekten die Interessen vieler Anspruchsgruppen (*Stakeholder*) koordiniert werden müssen [Somm04], existieren im Bereich „*Collaborative Requirements Engineering*“ einige Ansätze, die sich im Wesentlichen in den Verteilungsdimensionen sowie der Werkzeugunterstützung unterscheiden. Speziell für die verteilte **Anforderungserhebung** existieren zahlreiche empirische Studien, die die Vorteilhaftigkeit der geografischen Verteilung und dabei die Unterschiede zwischen synchroner und asynchroner Anforderungserhebung untersuchen [DaEb00, Dami01, LlRA02, DESG03].

Basierend auf dem Harvard-Konzept der Verhandlungstechnik wurde von Boehm und Ross die *Theory W* entwickelt [BoRo89]. Hierbei sollen in erster Linie asymmetrische Win-Lose-Situationen zwischen einzelnen *Stakeholdern* durch systematische Verhandlungs- und Moderationstechniken vermieden werden, um das Gesamtrisiko eines Softwareprojekts zu minimieren. Eine Anwendung dieser Theorie und gleichzeitig Weiterentwicklung des Spiralmodells [Boeh81] stellt das **WinWin-Spiralmodell** dar, welches zunächst als *Next Generation Process Model* bezeichnet wurde [BoBo94]. Dabei werden die sich verändernden Interessen unterschiedlicher Stakeholder sowie deren Koordination in jeder Phase über mehrere Iterationen hinweg berücksichtigt.

Die **EasyWinWin**-Methodik von Boehm, Briggs und Gruenbacher stellt die mittlerweile vierte Generation des WinWin-Ansatzes zur kollaborativen Anforderungserhebung dar [Grue00, BrGr02]. Die Methodik baut auf dem WinWin-Spiralmodell auf und integriert weitergehende Erkenntnisse aus den Bereichen „interpersonelle Beziehungen“ und „Erfolgsmanagement“. EasyWinWin zeichnet sich u.a. durch einen flexiblen, iterativen Erhebungsprozess aus, der es

ermöglicht, im laufenden Prozess neue *Stakeholder* zu integrieren. Außerdem wird der Aufbau von gegenseitigem Vertrauen durch physische Treffen und moderierte Diskussionen gefördert. Allerdings ist die Methodik komplex, teilweise nicht sehr intuitiv und trotz entsprechender *Groupware* primär nicht für ein verteiltes Umfeld konzipiert.

Groupware-basierte Werkzeugunterstützung für EasyWinWin ermöglicht mittlerweile Komplexitätsreduktion, verteiltes Arbeiten und asynchronen Informationsaustausch, beispielsweise durch die Protokollierung von Diskussionen [BoGB01, BoGB01a]. Die webbasierte Variante **ARENA** erlaubt ausschließlich asynchrones sowie verteiltes und mobiles Zusammenarbeiten bei der Anforderungserhebung. Allerdings ist ARENA nicht mit der ursprünglichen WinWin-*Groupware* kombinierbar [GrBr03, SGMT04].

Tabelle 6 stellt die unterschiedlichen WinWin-Umsetzungen im Überblick dar, wobei die Dimension „Richtung“ auf Grund des Fokus auf der kollaborativen Anforderungsanalyse nicht betrachtet wird.

Ansatz	Quelle	Organisat. Verteilung	Zeitliche Verteilung	Räumliche Verteilung	Intensität	Werkzeug-unterstütz.
WinWin-Spiralmodell	**[BoRo89], [BoBo94]**	inter/intra	synchron	lokal	hoch	keine
EasyWinWin	**[Grue00], [BrGr02]**	inter/intra	synchron/ asynchron	lokal/ verteilt	hoch	Groupware
EasyWinWin/ ARENA	**[GrBr03], [SGMT04]**	inter/intra	asynchron	verteilt	niedrig	Groupware

Tabelle 6: WinWin-basierte Ansätze für die kollaborative Anforderungsanalyse

3.2.2 Kollaborative Entwurfs- und Modellierungsprozesse

Kollaborative Ansätze innerhalb der Entwurfs- und Modellierungsphase wurden aus anderen Ingenieurdisziplinen übertragen (z.B. *Computer Aided* bzw. *Concurrent Design* Prozesse im Automobilbau). Bisher befinden sich die meisten dieser Ansätze im SE, u.a. aufgrund der speziellen Eigenschaften von Software, in einem prototypisch-experimentellen Stadium.

Joint Application Design (JAD) beschreibt eine Methodik für rechnergestützte Treffen und Workshops mit der Zielsetzung, aus den teilweise unterschiedlichen Anforderungen der *Stakeholder* zu einem gemeinsamen ersten Entwurf der Systemarchitektur zu kommen. Charakteristisch für JAD- Treffen sind u.a. ein fest definierter Zeitrahmen, eine strukturierte Umgebung mit visuellen Hilfsmitteln, ein Moderator und ein Protokollant [WoSi89, CaNu92, AvFi03]. Im Gegensatz zu JAD ist der ***Participatory Design***-Ansatz (PD) weniger ergebnisorientiert und fokussiert primär die intensive synchrone Zusammenarbeit zwischen Entwicklern und späteren Systemanwendern beim Entwurf der Software. Dabei wird nicht geografisch verteilt gearbeitet und es sind keine bestimmten Werkzeuge vorgegeben [CaWG93, Altm99].

Concurrent Software Engineering (CCSE) wurde ursprünglich durch den Ansatz von Dewan und Riedl geprägt [DeRi93]. Im Gegensatz zu JAD und PD sollen beim *Concurrent Software Engineering* möglichst viele Entwicklungsaktivitäten parallel bzw. überlappend ablaufen. Das bedeutet, dass dabei nicht notwendigerweise synchron zusammengearbeitet wird [Altm99, GSKR99, HiKo04]. Dean et al. hingegen betrachten mit ihrer ***Collaborative Software Engineering Methodology*** (CSEM) ebenfalls die nebenläufige, asynchrone Zusammenarbeit unterschiedlicher *Stakeholder* im Rahmen des Softwareentwurfsprozesses. Insbesondere Nutzer sollen in drei Stufen eingebunden werden: zunächst einige wenige Repräsentanten,

dann kleinere Benutzergruppen und schließlich die gesamte Nutzergemeinschaft. Im Gegensatz zum CCSE sieht CSEM eine hohe Intensität der Zusammenarbeit im Rahmen von regelmäßigen physischen Treffen zum Abgleich der Ergebnisse vor [DLPH98].

Nebenläufige Entwurfsprozesse werden auch immer stärker in UML-basierten Methoden zur modellgetriebenen Softwareentwicklung (*Model-Driven Development*, MDD) integriert. Wenn auch bisher nur wenige fundierte Methodiken existieren, ermöglichen moderne Entwicklungswerkzeuge, wie beispielsweise *Poseidon for UML Enterprise* [Gent05], Konesa [Cany01] und *Rational XDE* [IBM05, IBM05a], nebenläufige verteilte und modellgetriebene Softwareerstellungsprozesse (vgl. hierzu Abschnitt 3.3). Neben Vorgehensweisen zur Entwurfserstellung existieren des Weiteren werkzeuggestützte Ansätze zur Evaluation von Softwarearchitekturen. Hierbei sind in erster Linie die Arbeiten von Babar et al. zu nennen, die Architekturevaluationsverfahren für die verteilte Zusammenarbeit weiterentwickeln (***Distributed Software Architecture Evaluation Process***, [BKZJ04, BaGo04]). Tabelle 7 fasst noch einmal die beschriebenen Ansätze gemäß dem Klassifikationsrahmen zusammen.

Ansatz	Quelle	Organ. Verteilung	Zeitliche Verteilung	Räuml. Vert.	Intensität	Werkzeugunterstütz.
Joint Application Design	**[AvFi03]**	intra/inter	synchron	lokal	hoch	gegeben
Participatory Design	**[MuKu93]**	intra/inter	synchron	lokal	hoch	nicht explizit
Concurrent SE	**[DeRi93]**	intra/inter	asynchron	verteilt	niedrig	prototypisch
Collabor. SE Methodology	**[DLPH98]**	intra	asynchron	verteilt	hoch	prototypisch
Distributed Software Architecture Evaluation	**[BKZJ04]**	intra/inter	asynchron	verteilt	niedrig	nicht explizit

Tabelle 7: Kollaborative Ansätze für Entwurf und Modellierung

3.2.3 Kollaboration in Implementierung, Test und Wartung

Bei den Ansätzen zur kollaborativen Implementierung, Test und Wartung herrscht im Allgemeinen eine höhere Werkzeugorientierung vor als in den beiden bisher behandelten Disziplinen. Zusammenarbeit in der Implementierungsphase kann sowohl synchron, in Form von *Shared Workspaces* bzw. *Application Sharing* (z.B. *Collaborative Editing*), als auch asynchron, gepuffert über Versionsmanagementsysteme (*Code Repositories*), unterstützt werden [DeRi93, Koch95]. Da auch hier jedoch häufig methodische Aspekte fehlen, soll auf diese und weitere Werkzeuge detaillierter in den nachfolgenden Abschnitten eingegangen werden.

Mit der CAIS-Methode (***Collaborative Asynchronous Inspection of Software***, [MDTR93, MaFR94]) existiert ein verteilter Ansatz zur kollaborativen Qualitätssicherung als Teil der Postimplementierungsphase [SRHM97]. Ebenso in diese Kategorie fallen Ansätze zur verteilten und asynchronen Wartung bereits entwickelter Software (***Collaboration in Software Maintenance***) [LoRo93]. Andere Arbeiten versuchen, die Arbeitsabläufe in der Softwareentwicklung durch formale Modelle, z.B. Petrinetze, abzubilden und zu optimieren [Ober94, ObWS94]. Viele Autoren unterscheiden hierbei prozessorientierte und produktorientierte Ansätze zur kollaborativen Softwareerstellung [AlWe98, BCGM03], die im folgenden Abschnitt im Rahmen der werkzeugzentrierten Ansätze behandelt werden.

3.3 Werkzeugzentrierte Ansätze

Im Folgenden werden werkzeugzentrierte Ansätze zur kollaborativen Softwareerstellung aus der Literatur sowie einige kommerzielle Produkte vorgestellt. Bei den Werkzeugen zur Unter-

stützung zusammenarbeitsintensiver Softwareerstellungsprozesse lässt sich eine Evolution der Entwicklungsumgebungen bezüglich der Kommunikationsmöglichkeiten und der Integration verteilter softwareartefakte beobachten. Anfangs wurden in Softwareprojekten separate ***Groupware***-Systeme in Verbindung mit Editoren und evtl. Konfigurationsmanagement-Systeme (KMS) von den Entwicklern genutzt. **Integrierte Entwicklungsumgebungen** (*Integrated Development Environments*, IDE) hingegen versuchen dem einzelnen Entwickler eine einheitliche Oberfläche und Arbeitsumgebung für die Softwareerstellung zu präsentieren und mehr Produktivität zu gewährleisten, indem u.a. oben genannte Werkzeuge integriert und standardisiert werden. Erweiterte Koordinations- und Kommunikationsfunktionen für die synchrone und asynchrone Zusammenarbeit von räumlich verteilten Entwicklern und Entwicklungsgruppen bieten so genannte **Kollaborationsplattformen**, wie sie im Rahmen von *Open Source*-Projekten bereits häufig erfolgreich eingesetzt wurden.

3.3.1 Groupware für Softwareentwicklungsprojekte

Groupware unterstützt kollaboratives, verteiltes Arbeiten in Gruppen [HeHR04, S. 295] und wird zumeist nach zeitlicher und geografischer Verteilung untergliedert. Die computergestützte Zusammenarbeit wird häufig auch als „computerunterstütztes kollaboratives Arbeiten“ (CSCW) bezeichnet und beschreibt die „gemeinsame Bearbeitung eines Produkts oder einer Dienstleistung durch mehrere Aufgabenträger, unabhängig von der räumlichen und zeitlichen Situation der Zusammenarbeit“ [HeHR04, S. 157]. Bei der kollaborativen Softwareerstellung sind die Fälle „lokal und synchron“ (beispielsweise bei der Anforderungsanalyse), „verteilt und synchron“ (*Chat*, *Application Sharing* etc.) sowie „verteilt und asynchron“ (bei KMS und Projektverwaltungssystemen wie z.B. CVS und *Subversion*) von besonderer Bedeutung [ChSp00]. In der Literatur beschäftigen sich daher zahlreiche Arbeiten mit den Auswirkungen von *Groupware* und CSCW-Werkzeugen auf die SE-Prozesse [KTCB92, LaBF00, BKZJ04, DeDe04]. Auch bei den Ansätzen aus den Bereichen *Global Software Development* (GSD) und *Distributed Software Development* (DSD) wird die Bedeutung von Groupware und CSCW-Lösungen für die Softwareerstellung hervorgehoben [HeMo01, HeMo03].

3.3.2 Integrierte Entwicklungsumgebungen

IDEs vereinen neben dem Editor zur Quelltextbearbeitung sich gegenseitig ergänzende Entwicklungswerkzeuge wie Compiler, grafische *Frontends* für den Zugriff auf Konfigurationsmanagement-Systeme (KMS), *Groupware*, ebenso wie weitere Werkzeuge für Architekturentwurf und Modellierung (vgl. Abschnitt 3.2), *Unit Testing* und *Debugging*. Diese Zusammenstellungen von Entwicklungswerkzeugen werden häufig auch unter dem Begriff „***Computer-Aided Software Engineering***“ (CASE) subsumiert. Eine funktionale Klassifikation von CASE-Werkezeugen findet man bei [Somm04, S. 85ff.]. Als bekannte Beispiele für IDEs lassen sich die quelloffen erhältlichen Entwicklungsumgebungen *Eclipse* und *NetBeans* sowie kommerzielle Lösungen von IBM/Rational und Borland nennen, die teilweise auf den *Eclipse*-Quelltext aufbauen [IBM05, IBM05a]. IDEs sind auf die Bedürfnisse individueller Entwicklerarbeitsplätzen zugeschnitten und bieten nur teilweise Anbindung an zentrale Infrastrukturen wie KMS und *Groupware*-Server (vgl. [Somm04, S. 86])[5]. Immer häufiger findet man Ansätze, die sich bewusst mit Kommunikations- und Koordinationsaspekten und der Erweiterung von IDEs um Kollaborationswerkzeuge beschäftigen, um somit eine standardisierte Kollaborationsplattform für verteilte Softwareprojekte zu schaffen („*contextual collaboration*“ [CDHP03, CHRP03, CPRH05].

[5] Sommerville (2004) konstatiert in seinem Standardwerk „Software Engineering“ allerdings noch, dass sogar die aktuelle, ausgereifte CASE-Technologie nicht ausreichend Unterstützung für die Interaktion Entwicklungsteam-Mitglieder bietet.

3.3.3 Kollaborationsplattformen

Kollaborationsplattformen für die Softwareerstellung, wie beispielsweise *SourceForge* bei *Open Source*-Projekten, stellen eine Erweiterung von serverbasierten CSCW-Lösungen speziell für das SE dar [AuBS02]. Der Zugriff auf solche Kollaborationsplattformen erfolgt entweder über eine IDE als Client oder sogar mit einem herkömmlichen Browsern bzw. Mailprogrammen [Ster02, Kiss03].

Kollaborative Ansätze aus dem ***Open Source Software Development*** (OSSD) sind sehr werkzeugzentriert und primär für eine verteilte Umgebung im Internet ausgelegt. Augustin et al. sowie Szyperski und Spinellis argumentieren daher, dass kollaborative Praktiken, insbesondere Kollaborationswerkzeuge aus dem OSSD-Bereich, auch kommerzielle Entwicklungsprojekte beschleunigen und akute Probleme in solchen Projekten beheben können [AuBS02, SpSz04]. Ein Versionierungswerkzeug bzw. Quelltext-*Repository*, z.B. CVS, hat im *Open Source*-Umfeld eine Doppelfunktion als zentrales Koordinationsinstrument durch Rechtevergabe und Versionskontrolle sowie als Distributionssplattform durch einen integrierten Webserver [FoBa02]. Entwicklungsmethodiken sowie *Governance*-Prozesse zur Projektorganisation haben sich erst später entwickelt und institutionalisiert. Man kann bei OSSD-Projekten unterschiedliche Ausprägungen und Evolutionsstufen beobachten. Grundsätzlich arbeiten in OSSD-Projekten einzelne Programmierer verteilt, meist asynchron und parallel, zusammen. Wie im Fall von *Eclipse* oder auch *Mozilla*, existieren kommerziell inspirierte Entwicklergemeinschaften, welche in der Regel durch die Offenlegung von Quelltext und Gründung einer *Community* durch Unternehmen entstehen. Im *Embedded Linux*-Bereich findet man überdies auch primär kommerzielle Entwicklergemeinschaften, in denen Entwickler konkurrierender Unternehmen eine gemeinsame Quelltextbasis weiterentwickeln (vgl. Abschnitt 4.2.4).

Die Anbieter kommerzieller IDEs (IBM/Rational, Borland etc.) tendieren immer mehr zur Erweiterung ihrer Produktpaletten um Kollaborationsplattformen. Der Erfolg dieser **kommerziellen Entwicklungsplattformen** basiert u.a. auf dem erfolgreichen Einsatz der zugrunde liegenden Methoden und Werkzeuge in zahlreichen *Open Source*-Projekten [AuBS02]. Dies äußert sich u.a. darin, dass Firmen erfolgreich kommerzielle Versionen ehemals quelloffener Plattformen vertreiben, wie im Fall von *SourceForge Enterprise* [VASo05]. Laut einer Studie unter 151 Softwareentwicklungsfirmen beabsichtigen 65% der befragten Organisationen innerhalb der kommenden zwei Jahre eine Kollaborationsplattform einzusetzen [Webs03]. Viele der bereits im OSSD-Bereich entstandenen Kollaborationsformen kommen auch bereits bei globalen Softwareprojekten zum Einsatz. Neben asynchronem Informationsaustausch kommen hierbei häufig auch synchrone Formen der Zusammenarbeit, wie Chat oder IP-Telefonie, zum Einsatz, und es kann dadurch eine höhere Intensität der Zusammenarbeit erzielt werden [CaAg01, HeMo01, CaLM02, HeMo03]. Die unterschiedlichen Anforderungen an verteilte Softwareentwicklungsumgebungen kann man in die drei Bereiche (1) Wissensaustausch und -integration (Wissensmanagement), (2) Ermöglichung von Änderungen“ (Änderungs- oder *Change*-Management) sowie (3) breite Kommunikations- und Koordinationsunterstützung als Teil des Projektmanagements gliedern [CuKI88, S. 1283]. Nach Booch und Brown sollen kollaborative Entwicklungsumgebungen webbasiert, artefaktzentriert und multidimensional sein [BoBr03, S. 8].

Die grundlegenden Funktionalitäten einer Kollaborationsplattform lassen sich somit in die Kategorien „Koordination“, „Kollaboration“ und *Community Building* unterteilen, wobei der Aspekt „Kommunikation“ eine Querschnittsfunktion einnimmt [BoBr03]. Basierend auf unterschiedlichen Studien und eigenen Marktanalysen sind in Tabelle 8 einige der wichtigsten Funktionalitäten abgetragen, um den Funktionsumfang kommerziell erhältlicher Kollaborationsplattformen für die verteilte Softwareerstellung im Vergleich zur in der Praxis häufig verwendeten „Office-Alternative“ [HeRe01] in Verbindung mit der verteilten, gruppenorientier-

ten *SharePoint*-Technologie von Microsoft darzustellen [Micr04]. Ähnliche Betrachtungen zum Funktionsumfang von Kollaborationsplattformen finden sich u.a. bei [Kiss02] und [Webs03].

Plattform und Hersteller	Versionskontrolle	Issue Tracking	Mailinglisten	Diskussionsforum	Newsforum	Projektmgt.	Dokumentenmgt.	Wissenserfassung	Webbasierte Adm.	E-Mail-Integration	Echzeitkommun.	Quelltext Engin.	Umfragefunktion	Suchfunktion
CodeBeamer (Intland)	X	X		X		X	X	X	X	X		X		X
CollabNet Enterprise (CollabNet)	X	X	X	X	X	X	X	X	X					
GForge Enterprise (GForge Group)	X	X		X		X	X	X	X				X	X
Sourceforge Enterprise (VA Software)	X	X		X		X	X	X	X				X	X
Rational Suite (IBM)	X	X	X			X	X	X	X			X		X
Office/Sharepoint (Microsoft)	X		X		X	X	X	X	X	X				X

Tabelle 8: Funktionaler Vergleich kommerzieller Kollaborationsplattformen

Neben kommerziellen existiert auch eine Vielzahl **experimenteller Plattformkonzepte** und Prototypen aus dem universitären Umfeld und aus Forschungsprojekten. Der ***Cooperation Assistant*** unterstützt verteilte Teamarbeit durch eine integrierte Prozess- und Produktsicht sowie die Integration externer Entwicklungswerkzeuge über Adapter. Es können dabei während des gesamten SE-Prozesses gemeinsame Arbeitsgegenstände ausgetauscht werden [Altm99]. **CHIME** (*Columbia Hypermedia IMmersion Environment*) ist ein metadatenbasiertes Informationssystem und Navigationswerkzeug für die verteilte Bearbeitung von Softwareartefakten. Es erlaubt die zentrale Speicherung und Verwaltung von Wissen, beispielsweise in Form von Annotationen und Chat-Protokollen [DoKG99]. **GWSE** (*Global Working in Software Engineering*) stellt eine Entwicklungs- und Testumgebung für stark verteilte Arbeitsgruppen dar und basiert auf Lotus Notes erweitert um Konfigurations- und Projektmanagementfunktionen [GHRC97]. Die erweiterbare, kollaborative Softwarearchitektur von **CAISE** (*Collaborative Architecture for Iterative Software Engineering*) erlaubt die Integration von SE-Werkzeugen, die Archivierung und synchrone Nutzung von Artefakten und unterstützt dabei unterschiedliche Arten von Softwareartefakten und Programmiersprachen [CoCh03]. Das Projekt **COVEN** (COllaborative Versioning ENvironment) umfasst eine IDE samt eigenem KMS. Das hierarchische Quelltext-*Repository* soll Entwicklern die effiziente, asynchrone Kommunikation und Koordination ihrer Arbeit ermöglichen [ChSp00]. Eine Werkzeugsammlung zur Unterstützung der kollaborativen Entwicklung von komponentenbasierten Systemen bietet **OdysseyShare**. Es wird die Entwicklung, das Editieren und *Reverse-Engineering* von Quelltext und Modellen unterstützt. Außerdem enthält OdysseyShare eine *Workflow Engine*, Werkzeuge zur Unterstützung von Echtzeitkollaboration und unterstützt die Verwaltung einer verteilten „Komponentenbibliothek“ [WMMP03]. **MILOS** beinhaltet Werkzeuge zur Prozessmodellierung, Projektplanung/-umsetzung und bietet Rückverfolgbarkeit und Änderungsbenachrichtigung. Zusätzlich existiert eine Anbindung an MS-Project [MSHK99]. Bei **O-**

PHELIA handelt es sich um ein Framework zur Integration mehrerer heterogener SE-Werkzeuge zu einer Kollaborationsumgebung mit globaler Sicht auf die unterschiedlichen Softwareartefakte. Mit Hilfe des CORBA-Standards als Integrationsplattform werden ereignisgetriebene Metriken implementiert [WRSS03]. Das Open Source-Projekt **GENESIS** (*GEneralised eNvironment for procEsS management in cooperatIve Software engineering*) umfasst ein Workflow- und Artefaktmanagementsystem und bietet umfangreiche Koordinations- und Kommunikationsfunktionen. Es werden sowohl prozess- als auch produktorientierte Softwareprojekte unterstützt, wobei GENESIS mit einer eigenen Sprache zur Prozessdefinition arbeitet [GaRi02, BCGM03]. Ein Vergleich von OPHELIA und GENESIS findet sich bei [BNRS03]. Das **JAZZ**-Projekt, das von IBM unterstützt wird, stellt eine kollaborative Erweiterung zur quelloffenen Eclipse-IDE dar. Es vertritt dabei den Ansatz, die Nutzer in ihrer gewohnten Entwicklungsumgebung (vgl. Abschnitt 3.3.2) zu belassen und lediglich kontextbezogene Erweiterungen für die Kollaboration in Softwareprojekten zu bieten [ChHu03, CDHP03, CPRH05].

Plattform	Quelle	Organisat. Verteilung	Räumliche Verteilung	Zeitliche Verteilung	Richtung	Intensität
BeyondSniff	[BKMS95]	intra/inter	verteilt	asynchron	vert./hor.	niedrig
CAISE	[CoCh03], [CoCl04]	intra/inter	verteilt	synchron	vert./hor.	hoch
CHIME	[DoKG99]	intra/inter	verteilt	asynchron	vertikal	niedrig
Cooperation Assistant	[Altm99]	intra/inter	verteilt	syn./asyn.	vertikal	niedrig
COVEN	[ChSp00]	intra/inter	Verteilt	Asynchron	horizontal	niedrig
GENESIS	[GaRi02], [BCGM03]	intra/inter	verteilt	syn./asyn.	vertikal	sehr hoch
GWSE	[GHRC97]	intra/inter	verteilt	syn./asyn.	vert./hor.	hoch
JAZZ	[ChHu03], [CDHP03], [CPRH05]	intra/inter	verteilt	syn./asyn.	vertikal	hoch
MILOS	[MSHK99]	intra/inter	verteilt	asynchron	vertikal	sehr hoch
Odyssey-Share	[WMMP03]	intra/inter	verteilt	synchron	vert./hor.	niedrig
OPHELIA	[WRSS03]	intra/inter	verteilt	asynchron	vert./hor.	niedrig

Tabelle 9: Experimentelle Kollaborationsplattformen aus Forschungsvorhaben

Tabelle 9 stellt noch einmal die hier vorgestellten Ansätze gegenüber und ordnet sie in den Untersuchungsrahmen ein (vgl. Abbildung 2). Bei keiner der Plattformen werden intra- und interorganisatorische Szenarien der Zusammenarbeit systematisch unterschieden und im Wesentlichen unterscheiden sich die einzelnen Ansätze anhand der unterstützten Form (synchron/asynchron) und der Intensität der Kollaboration.

3.4 Fazit

Vergleicht man die bisher analysierten Ansätze, lässt sich zusammenfassend feststellen, dass vor allem im Bereich „*Collaborative Softare Development*“ die werkzeugzentrierten Ansätze

der Informatik (*Software Engineering*) deutlich in der Überzahl sind. Zwar existieren ebenfalls zahlreiche Vorgehensmodelle und Prozessrahmenwerke, die jedoch nicht dediziert und über alle Phasen des Softwarelebenszyklus hinweg auf die Aspekte und Dimensionen der kollaborativen Softwareerstellung eingehen. Es existieren bisher nur vereinzelte methodische Ansätze, die sich auf bestimmte Tätigkeiten fokussieren, z.B. auf die kollaborative Anforderungsanalyse. Des Weiteren finden sich selten Ansätze, die die menschliche Seite der Kollaboration mit deren sozialer Komplexität und soziotechnischen Problemen im Kontext von Softwareprojekten untersuchen [Andr03].

4. Zwischenbetriebliche Kollaboration

Neben den im vorangegangenen Kapitel angesprochenen gruppen-, prozess- und werkzeugorientierten Fragestellungen auf der Ebene einzelner Softwareprojekte kommen bei der Zusammenarbeit auf Unternehmensebene im zwischenbetrieblichen Kontext zusätzliche organisatorische, ökonomische sowie juristische Aspekte hinzu. Unterschiedliche Zusammenarbeitsformen und Kollaborationsbeziehungen geben somit den organisatorischen Rahmen für zwischenbetriebliche Entwicklungsprojekte vor. Hierbei kommen in erster Linie die Richtung, d.h. das Verhältnis der Kollaborationspartner zueinander, und die Intensität der Zusammenarbeit zum Tragen (vgl. Abschnitt 2.2 sowie [HeSi93]).

Die Motive für eine zwischenbetriebliche, kollaborative, Erstellung von Software lassen sich primär aus den Charakteristika industrieller Organisationen ableiten – „*to overcome size or resource limitations*“ und um mehr Möglichkeiten zur Spezialisierung innerhalb einzelner Branchen zu schaffen [LaHP02, S. 289]. So kann diese Form der Softwareerstellung zu einer Aufteilung und Reduktion von Kosten sowie zur Erweiterung fachlicher Kapazitäten bei den beteiligten Unternehmen führen. Des Weiteren können der gegenseitige Austausch von Erfahrungswissen und die Verminderung von Entwicklungsrisiken sowie die Entwicklung und Einführung von gemeinsamen Wettbewerbsstandards für die teilnehmenden Organisationen von Vorteil sein [HeSi93]. Die erfolgreiche Etablierung von Wettbewerbsstandards erfordert allerdings eine kritische Marktmacht des Konsortiums und stellt daher immer auch eine risikobelastete Unternehmung dar. Zwischenbetriebliche Zusammenarbeit bei der Anwendungsentwicklung erfordert überdies ein ausgewogenes Qualifikationsniveau zwischen den beteiligten Unternehmen sowie einen gesicherten Rückfluss von Know-how aus den gemeinsamen Aktivitäten, um Konflikten vorzubeugen [HeSi93]. Des Weiteren gilt es im zwischenbetrieblichen Kontext zunehmend, konfligierende Unternehmensziele sowie soziale Konflikte auf Mitarbeiterebene zu beherrschen [KuDi96] (siehe auch Abschnitt 3.3).

4.2 Formen zwischenbetrieblicher Kollaborationsbeziehungen

Neben den drei Verteilungsdimensionen des kollaborativen Arbeitens (räumlich, zeitlich und organisatorisch) lassen sich die etablierten Formen zwischenbetrieblicher Kollaborationsbeziehungen bei der Softwareerstellung in erster Linie anhand der beiden Dimensionen „Richtung“ und „Intensität“ (vgl. Tabelle 3) charakterisieren. Im zwischenbetrieblichen Kontext drückt die **Richtung** das Verhältnis der beteiligten Partnerunternehmen in der Wertschöpfungskette aus (vertikal oder horizontal). Bei [KuDi96] wird diese Dimension als *Interorganizational Interdependence* bezeichnet, wobei im Sinne einer Wertschöpfungskette gemeinsame sowie reziproke Interdependenzen einen horizontalen und sequenzielle Interdependenzen einen vertikalen Charakter aufweisen. Entlang der Vertikalen besteht zusätzlich ein grundlegender Unterschied zwischen kunden- und lieferantengerichteter Zusammenarbeit. Die **Intensität** drückt in diesem Fall aus, ob direkt, beispielsweise in Form gemeinsamer Entwicklungsaktivitäten, und/oder mittelbar durch den Austausch von Artefakten und Wissen zusammengearbeitet wird. Einen weiteren wichtigen Aspekt stellt dabei auch die juristische „Bindungsintensität“ dar, welche im zwischenbetrieblichen Kontext lose, vertragliche oder kapitalge-

bunden Zusammenarbeit unterscheidet. Wie bereits in Abschnitt 2.2.5 dargestellt subsumiert die Intensität immer auch zeitliche Aspekte der Kollaboration. Hierbei lässt sich beobachten, dass aus erfolgreichen einmaligen Projekten zwischen Unternehmen häufig sporadische oder gar regelmäßige Zusammenarbeit im Rahmen informeller Netzwerke resultieren kann [ThLo04]. Aus den beiden zentralen Formmerkmalen „Richtung“ und „Intensität“ ergeben sich im Wesentlichen die unterschiedlich organisatorisch strukturierte Formen von Kollaborationsbeziehungen (siehe Abbildung 3).

Neben den beiden zentralen Formmerkmalen kann auch die Anzahl der beteiligten Partnerunternehmen einen Einfluss auf die Struktur der Kollaborationsbeziehung ausüben. Man unterscheidet hierbei bilaterale und trilaterale Beziehungen sowie einfache und komplexe Netzwerkstrukturen der Unternehmen. Bei einfachen Netzwerken existiert ein fokales Unternehmen, das mit den jeweils anderen in Beziehung steht. Komplexe Netzwerke hingegen bilden zahlreiche multilaterale Beziehungen der einzelnen Unternehmen untereinander ab [ThLo04]. Ein weiterer Faktor, der eng mit der stark mit der räumlichen Verteilung (Abschnitt 2.2.2) zusammenhängt, ist die Herkunft der Partner. Theling und Loos unterscheiden hierbei die institutionelle und die geografische Partnerherkunft [ThLo04]. Die institutionelle Herkunft kann entweder direkt zwischenbetrieblich oder beispielsweise über Gremien überbetrieblich geartet sein. Im Rahmen der geografischen Herkunft unterscheidet man lokale, regionale (im Umkreis von 50 bis 100 km), nationale und internationale Partner.

4.2.1 Vertraglich geregelte Kollaborationsformen

Zwischenbetriebliche Kollaborationsbeziehungen –vertikal wie horizontal – werden häufig im Hinblick auf Ziele und Inhalte durch Rahmenverträge geregelt. Unternehmen stehen in **vertikaler** Richtung entlang der Wertschöpfungskette häufig in einer vertraglich geregelten Kunden/Lieferanten- bzw. Auftragnehmer/Unterauftragnehmer-Beziehung [HeSi93]. Werden Softwaresysteme oder aber einzelnen Komponenten auf Auftrag gefertigt, d.h. neu erstellt, regeln die Parteien diese Angelegenheit mit einem Vertrag, der in der Regel als **Werkvertrag** gemäß den §§ 631 ff. BGB zu qualifizieren ist. [Marl00, JuBe03]. Komponentenhersteller können ihrerseits wieder Unterauftragnehmer (*Subcontractors*) beauftragen, wodurch sich jedoch sowohl die vertikale Vertragsabstimmung als auch der Koordinationsaufwand im Softwareprojekt signifikant erhöhen kann [CuKI88]. Verfügt der Softwareanbieter bereits über vorgefertigte, standardisierte Anwendungssysteme oder –komponenten und veräußert deren Nutzungsrechte, handelt es sich um eine Lieferbeziehung zum abnehmenden Unternehmen. Diese kann dann über einen **Kaufvertrag** mit entsprechenden Anpassungsklauseln abgewickelt werden (*Supplier-Consumer Relationship*) [MeSz03, Marl00, JuBe03]. Insbesondere bei Standardsoftware wird eine zeitweise Überlassung von Software immer häufiger auch durch **Miet**- oder **Pachtverträge** (§§ 531 und 581 BGB) gesetzlich geregelt [Marl00, JuBe03]. Die Auslagerung (*Outsourcing*) der Anwendungsentwicklung kann ebenfalls als vertraglich geregelte Form der vertikalen Kollaboration betrachtet werden, da hierbei häufig auch auf Projektebene zusammengearbeitet werden muss [Dibb04]. Bei der Ausgliederung (*Spin-Off*) in eine Tochtergesellschaft liegt überdies nicht nur eine vertragliche sondern in der Regel auch eine kapitalmäßige Bindung vor [Hein93, HeSi93].

4.2.2 Gemeinschaftsunternehmen

Ebenfalls eine kapitalmäßige Bindung besteht, wenn mehrere Unternehmen derselben Wertschöpfungsstufe ein rechtlich selbständiges Gemeinschaftsunternehmen (*Joint Venture*) gründen bzw. mehrere Unternehmen gemeinsam vertikal ausgliedern, beispielsweise in Form eines gemeinsamen Softwarelieferanten. Hierbei handelt es sich um eine **horizontale** Form der Zusammenarbeit mit vertikalen Implikationen [HeSi93, Hein93]. Neben dem Kapitaleinlagen bringen die Gründungsgesellschaften meist einen wesentlichen Ressourcenanteil an Techno-

logie, Schutzrechten, technischem bzw. Marketing-Wissen und/oder Betriebsanlagen ein. Daher ist die **Intensität** solcher langfristiger Kollaborationsbeziehung für die beteiligten Unternehmen **sehr hoch** und impliziert gemeinsame Entwicklungstätigkeiten. Zusätzlich können über Werk- und Kaufverträge fehlende Kompetenzen und Kapazitäten von anderen Unternehmen zugekauft werden. Von ähnlich hoher Intensität sind lediglich noch das Erwerben von Anteilen an anderen Unternehmen und Fusionen [ThLo04].

4.2.3 Virtuelle Unternehmen

Ähnlich den Gemeinschaftsunternehmen stellen Virtuelle Unternehmen (*Virtual Enterprises*) „eine Form der Kooperation von rechtlich selbständigen Unternehmen, die gemeinsam Sachleistungen oder Dienstleistungen durch Zusammenwirken ihrer Kernkompetenzen erbringen und bei der Leistungserstellung gegenüber Dritten [Kunden, Anwenderunternehmen] wie ein [einziges] Unternehmen agieren.“ [HeHR04, S. 700]. Somit ist diese eher lose Form der zwischenbetrieblichen Zusammenarbeit besonders zur unternehmensübergreifenden Integration **vertikaler** Wertschöpfungsketten und effektiveren Leistungserbringung gegenüber Dritten geeignet. Als virtuelles Unternehmen bezeichnet man ein Unternehmen, das im Gegensatz zum traditionellen Unternehmen nicht an einen geografischen Ort gebunden ist und dessen Zweck meist in der Bearbeitung eines oder weniger Projekte besteht. Dies hat in der Regel eine relativ **geringe Intensität** der Zusammenarbeit zur Folge. Teilhaber eines virtuellen Unternehmens können Mitarbeiter vieler unterschiedlicher Unternehmen aus der ganzen Welt sein, die gemeinsam im virtuellen Unternehmen an einem Projekt zusammenarbeiten. Ermöglicht werden virtuelle Unternehmen in erster Linie durch die kollektive Bereitschaft zur vertrauensbasierten anstatt vertragsbasierter Organisation sowie durch moderne Informations- und Kommunikationstechnik.

4.2.4 Entwicklungsgemeinschaften

Eine unternehmensübergreifende Gemeinschaft (*Community*) von Entwicklern, wie man sie aus dem *Open Source*-Umfeld kennt, entsteht zumeist um eine zentrale, internetfähige Kollaborationsplattform, die mindestens aus einem gemeinsamen Quelltext-*Repository* und einer Mailingliste bzw. einem Forum besteht (*Community-Based Development*) [MeSz03, S. 78]. Das Repository übernimmt dabei die Doppelfunktion als Koordinationsinstrument (Versionskontrolle und Konfigurationsmanagement) und Distributionsplattform von Ergebnissen und Vorprodukten (vgl. werkzeugzentrierte projektorientierte Ansätze). Die Kommunikation erfolgt primär per E-Mail, also asynchron [FoBa02]. Die Zusammenarbeit findet über den asynchronen Austausch von Softwareartefakten und Wissen statt. Bei typischen *Open Source*-Projekten bestehen keine vertraglichen Bindungen zwischen den beteiligten Entwicklern und deren Unternehmen. Die **Intensität** der Zusammenarbeit in solchen *Communities* ist daher relativ **gering**. Auf diese Weise beobachtet man sogar **horizontale** Zusammenarbeit eigentlich konkurrierender Unternehmen, beispielsweise zwischen Anbietern von eingebetteten Systemen, welche gemeinsam den Kern eines auf Linux basierenden Betriebssystems weiterentwickeln [Henk03, Henk03a]. Neben diesen offenen Entwicklergemeinschaften können sich auch geschlossene Entwicklergemeinschaften (*Closed* bzw. *Professional Communities*) bilden, welche zwar mit denselben softwaretechnischen Methoden und Werkzeugen operieren (siehe Abschnitt 3.3.3), die Kollaborationsplattform und die Arbeitsergebnisse aber nur einem exklusiven Kreis von Entwicklern respektive Unternehmen zugänglich sind [AuBS02].

4.2.5 Standardisierungskonsortien

Unternehmensübergreifende Standards in der Softwareindustrie können sowohl den Erstellungsprozess (Methoden, Werkzeuge sowie Datenaustauschformate) als auch die zu erstellenden Softwareprodukte betreffen, beispielsweise in Form von Technologiestandards für Kom-

ponentenplattformen und Programmierschnittstellen [DRMP03, DRCM04]. Standardisierungskonsortien oder -gremien bilden die **loseste** Form der zwischenbetrieblichen Zusammenarbeit und weisen daher eine vergleichsweise geringe Intensität auf [HeSi93]. Ein Industriestandard kann hierbei sowohl als **horizontaler** als auch als **vertikaler** Koordinationsmechanismus dienen [MeSz03]. Der Entstehungsprozess solcher Standards in der Softwareindustrie wird u.a. von Messerschmitt und Szyperski analysiert. Dabei wird grundsätzlich die **De-facto**-Standardisierung durch die Marktmacht einzelner Unternehmen und **De-jure**-Standardisierung durch juristische Auflagen und regulierende Instanzen, beispielsweise durch Regierungen unterschieden [MeSz03, S. 232ff.]. Ein Beispiel für ein domänenübergreifendes Standardisierungskonsortium ist die *Object Management Group* (OMG), in deren Umfeld industrieweite Softwarestandards (u.a. CORBA, UML und MDA) entworfen und weiterentwickelt werden. Beteiligt sind dabei sowohl Anwender aus unterschiedlichen Branchen als auch reine Softwareunternehmen. Weitere Beispiele sind die *International Standards Organization* (ISO), die *Organization for the Advancement of Structured Information Standards* (OASIS) und das *Open GIS Consortium* (OGC).

4.3 Softwareökosysteme

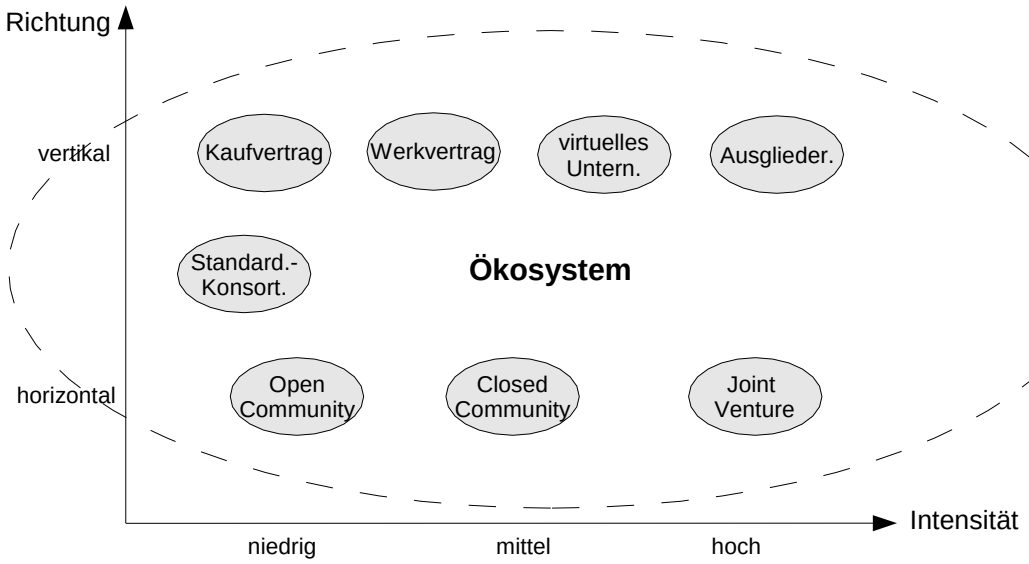

Abbildung 3: Einordnung der zwischenbetrieblichen Kollaborationsformen

Wie im vorhergehenden Abschnitt dargestellt, lassen sich unterschiedliche Formen der zwischenbetrieblichen Zusammenarbeit in erster Linie nach den Dimensionen Intensität und Richtung unterscheiden. Abbildung 3 soll diesen Zusammenhang noch einmal visuell verdeutlichen. Unternehmen, die bei der Softwareerstellung in einem Wertschöpfungsnetzwerk über unterschiedliche Formen von Kollaborationsbeziehungen im Sinne eines „Erreichbarkeitsgraphen“ in Verbindung stehen und an einer oder mehreren Wertschöpfungsketten innerhalb dieses Netzwerkes teilhaben, bilden ein so genanntes „**Softwareökosystem**“ [MeSz03].

Als konkrete Ausprägungen solcher Ökosysteme beschreiben Laubacher und Malone Netzwerke **homogener, kleinerer Unternehmen** im Gegensatz zu **monolithischen Konzernen** mit deutlich kleineren, spezialisierten Partnerfirmen (**heterogenes** Ökosystem) als zwei grundlegende Szenarien für die mögliche Organisationsform des 21. Jahrhunderts [LaMa97]. In der Softwareindustrie ist die letztere Form bereits bei traditionell monolithischen Unternehmen mit marktbeherrschender Stellung wie beispielsweise Microsoft, IBM und SAP erkennbar. Diese kultivieren zunehmend auch ein Netzwerk aus spezialisierten und zertifizierten Partnerunternehmen und bezeichnen diese Form der Zusammenarbeit als „Partnerökosystem“. Basierend auf der standardisierten Softwareinfrastruktur dieser „Big Player“, wie bei-

spielsweise Windows/Office (.NET) und NetWeaver, sollen die Partner für spezialisierte Erweiterungen dieser Plattformen sorgen. Es besteht aber auch ein Trend hin zum Zusammenschluss mehrerer kleiner Unternehmen, beispielsweise in Form virtueller Unternehmen und regionaler Cluster (vgl. Abschnitt 4.2.3).

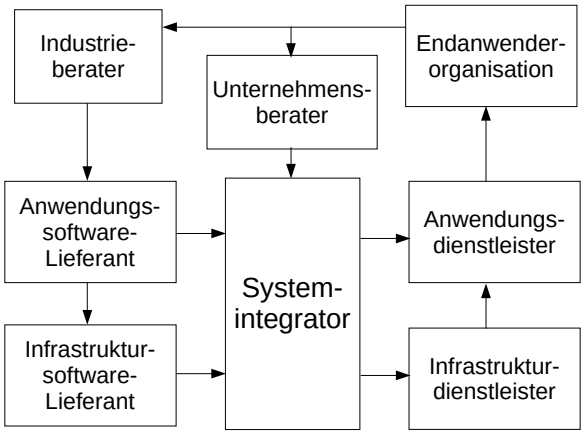

Abbildung 4: Grundlegendes Modell einer Softwarewertschöpfungskette [MeSz03]

Basierend auf den ursprünglichen Überlegungen von Porter identifizieren Messerschmitt und Szyperski innerhalb der Softwareindustrie unterschiedliche unternehmensübergreifende Wertschöpfungsketten und -netzwerke [Port80, MeSz03]. Diese reichen vom spezialisierten Komponentenanbieter bis hin zum Systemintegrator, der die Unternehmenssoftware für das Anwenderunternehmen bereitstellt und einführt. Zwischenstufen können durch unterschiedliche Anbieter von Anwendungs- und Infrastrukturkomponenten gebildet werden. In diesem Ökosystem spielen auch unterschiedliche Beratungshäuser als moderierende Instanzen sowie Anwendungs- und Infrastrukturdienstleister eine entscheidende Rolle. Abbildung 4 zeigt eine häufig beobachtbare Ausprägung einer unternehmensübergreifenden Softwarewertschöpfungskette [MeSz03, S. 174ff.].

4.4 Fazit

Wie in Abbildung 3 dargestellt, lassen sich existierende Ansätze zur zwischenbetrieblichen Kollaboration in erster Linie anhand deren Intensität bzw. Richtung unterscheiden. Wird die zwischenbetriebliche Zusammenarbeit schließlich auf Projektebene umgesetzt, kommen auch die drei Verteilungsdimensionen vermehrt zum Tragen (vgl. hierzu Tabelle 3). Vertraglich geregelte Formen der zwischenbetrieblichen Softwareerstellung sowie Gemeinschaftsunternehmen entsprechen der Definition einer „Kooperation“ aus der klassischen Organisationslehre (vgl. Abschnitt 2.1.1). Bei virtuellen Unternehmen und Entwicklergemeinschaften handelt es sich um modernere Formen der Zusammenarbeit, die sich vor allem mit der Ausbreitung des Internets als Kommunikationsmedium etabliert haben. Bei den Standardisierungskonsortien handelt es sich um die lockerste Form der zwischenbetrieblichen Kollaboration, jedoch bilden diese Institutionen insbesondere im Hinblick auf den Wissensaustausch und die Entwicklung gemeinsamer Standards eine wesentliche Grundlage zwischenbetrieblicher Softwareerstellung.

5. Offene Forschungsfelder

Im Zuge der zunehmenden Industrialisierung der Softwareindustrie auf Grund steigender Ansprüche an Produktivität und Qualität der Erstellungsprozesse muss die zwischenbetriebliche Kollaboration durch spezielle Vorgehensmodelle, Methoden und Werkzeuge aus dem SE operativ unterstützt werden, um unter anderem eine systematische **Spezialisierung** einzelner Unternehmen auch bei operativen Aufgaben zu fördern. Hierzu bedarf es einer weitergehenden

Integration der projektbezogenen softwaretechnischen Sichtweise und der wirtschaftlichen und juristischen Rahmenbedingungen sowie der möglichen zwischenbetrieblichen Kollaborationsformen auf Unternehmensebene. Im Folgenden werden zunächst die im Rahmen dieser Arbeit identifizierten Defizite und offenen Forschungsfragen dargestellt, um darauf weitere Forschungsbedarfe für die Zukunft abzuleiten.

5.1 Offene Forschungsfragen auf Projektebene

Auf Projektebene mangelt es den prozessorientierten Ansätzen häufig an der nötigen Werkzeugunterstützung und werkzeugzentrierten Ansätzen fehlt in der Regel eine fundierte methodische Grundlage in Form einer entsprechenden Entwicklungsmethodik. Ein Großteil der bisherigen Arbeiten im Bereich CSD ist noch zu plattform-/architekturzentrisch – unter Vernachlässigung methodischer sowie soziotechnischer Aspekte (vgl. [Andr03]).

Wie bei Jacobson, Booch und Rumbaugh dargestellt, bestehen allerdings bei der Entwicklung von Prozessen und Werkzeugen im SE reziproke Wirkungszusammenhänge: Entwicklungsprozesse stellen bestimmte Anforderungen an Werkzeuge („*Process drives Tools*“) und innovative Werkzeuge wiederum ermöglichen bestimmte Prozessschritte erst („*Tools impact Process*“) [JaBR99]. Zusammen mit der Tatsache, dass für die erfolgreiche Durchführung von Softwareprojekten soziale, insbesondere soziotechnische, Fragestellungen wichtig sind, scheint eine stärkere Integration bzw. eine gleichmäßige Betrachtung der drei hier dargestellten Kategorien von Ansätzen und Sichtweisen sinnvoll. Alle kollaborativen Ansätzen auf Projektebene weisen jeweils die Bestandteile der gruppenorientierten, der prozessorientierten und der werkzeugzentrierten Kategorie auf – lediglich mit unterschiedlicher Intonation der drei Faktor „Mensch“ (Individual- und Gruppenverhalten), „Aufgabe“ (Prozessschritte) und „Technik“ (Entwicklungswerkzeuge, Standards etc.).

Die in Abschnitt 3 vorgestellten Ansätze unterscheiden sich neben diesen unterschiedlichen Perspektiven im Wesentlichen in den Dimensionen der räumlichen und zeitliche Verteilung sowie der vertikalen Prozessabdeckung (vgl. Richtungsdimension) und der empfohlenen Intensität der Zusammenarbeit. Die organisatorische Verteilung bei der zwischenbetrieblichen Softwareerstellung sowie die dadurch zusätzlich möglichen Probleme, beispielsweise Zielkonflikte zwischen Unternehmen, werden in der SE-Literatur bisher eher indifferent behandelt, wohingegen nahezu alle Ansätze die Auswirkungen geografischer Verteilung diskutieren. Aus diesem Grund erscheint es sinnvoll, im Kontext kollaborativer Softwareerstellung auch existierende Ansätze und Zusammenarbeitsmodelle auf Unternehmensebene bzw. im Geschäftsumfeld der gesamten Softwareindustrie zu betrachten.

5.2 Offene Forschungsfragen auf Unternehmensebene

Außer im Fall der auf OSSD-Methoden basierenden Entwicklergemeinschaften sagen die existierenden zwischenbetrieblichen Ansätze jedoch bisher nichts darüber aus, welche konkreten SE-Methoden, Vorgehensmodelle und Werkzeuge diese unterschiedlichen Formen der zwischenbetrieblichen Kollaboration am besten unterstützen können. Explizite Ansätze, beispielsweise in Form von Vorgehensmodellen für die zwischenbetriebliche Softwareerstellung, existieren praktisch nicht, obwohl gerade der zwischenbetriebliche Aspekt mit seinen veränderten juristischen und ökonomischen Rahmenbedingungen den eigentlichen Softwareentwicklungsprozess, d.h. im Wesentlichen die Gestaltung der Aktivitäten, Rollen und Artefakten, stark beeinflussen kann. Zwischen konkurrierenden Firmen innerhalb derselben Wertschöpfungsstufen können zusätzlich Zielkonflikte zum Tragen kommen [HeSi93]. Es kommt daher in erster Linie auf die Erwartung eines längerfristigen reziproken Nutzens aus der Kollaborationsbeziehung und weniger auf exakt identische Zielsetzungen der beteiligten Unternehmen an. D.h., auch wenn ein antagonistisches Verhältnis zwischen den Unternehmen

herrscht, können Kollaborationsbeziehungen sowohl aus betriebwirtschaftlicher als auch aus gesamtökonomischer Sicht nützlich sein [Gold02, Henk03].

5.3 Das Paradigma der kollaborativen Softwareerstellung

Aus den offenen Forschungsfragen und Defiziten der existierenden Ansätze, insbesondere der mangelnden Integration der projektbezogenen und der unternehmensbezogenen Sichtweise, ergibt sich der Bedarf für ein neuartiges **Forschungsparadigma** im Rahmen der Wirtschaftsinformatik. Dieses soll im Kern, bezogen auf die Erstellung von Informationssystemen, die Konfluenz operativer Problemstellungen aus der Softwaretechnik (*Collaborative Software Engineering*) sowie betriebswirtschaftlich-organisatorischer Fragestellungen der zwischenbetrieblichen Kollaboration (unternehmensübergreifende Leistungserstellungsprozesse, *Interorganizational Collaboration*) behandeln. Hinzu kommen im zwischenbetrieblichen Kontext auch juristische und makroökonische Überlegungen, welche sich auf die unternehmensübergreifende Wertschöpfungsketten und Strukturen innerhalb der Softwareindustrie beziehen.

Ein wissenschaftliches Paradigma legt nach [Kuhn62] fest, was beobachtet und überprüft wird bzw. die Art der Fragen, die in Bezug auf ein Thema gestellt werden und die geprüft werden sollen (Bezugsobjekt). Des Weiteren definiert das Paradigma, wie diese Fragen gestellt und wie die Ergebnisse der wissenschaftlichen Untersuchung interpretiert werden sollen (Methodik). Das zentrale Bezugsobjekt der kollaborativen Softwareerstellung (KSE) ist der Erstellungsprozess auf Projektebene (vgl. Tabelle 4). Dies impliziert laut Curtis et al. auch eine Betrachtung der Individual- und Gruppenprozesse auf Mitarbeiter- bzw. Teamebene [CuKI88]. Die Unternehmens- und Geschäftsebene bildet mit der Betrachtung organisatorischen Verhaltens den betriebswirtschaftlichen Kontext der KSE. Hieraus ergeben sich einige grundlegende **Forschungsfragen**:

- Unter welchen Umständen ist es vorteilhaft, Software verteilt und/oder zwischenbetrieblich zu erstellen?
- Welches sind die Erfolgsfaktoren verteilter Softwareerstellungsprozesse, speziell im zwischenbetrieblichen Kontext?
- Wie müssen Vorgehensmodelle und Werkzeuge zur Unterstützung dieser Prozesse gestaltet werden?
- Wie werde die unterschiedlichen Aufgaben im Softwareerstellungsprozess am besten auf Mitarbeiter und Teams aus mehreren Unternehmen verteilt?
- Welche Rolle spielen Standards, beispielsweise in Form von gemeinsamen Architekturen und technologischen Plattformen, für den Erfolg von kollaborativen Softwareprojekten?

Zur Klärung dieser und weiterer Fragen werden entsprechende **Methodiken** benötigt. Neben qualitativen und quantitativen Methoden der empirischen Sozialforschung zum Verständnis existierender Ansätze erfordert die KSE auch zunehmend fundierte entwurfsorientierte Methoden um die vorhandene Wissensbasis zu erweitern (vgl. *Design Science*-Paradigma der Wirtschaftsinformatik [HMPR04]). Bei dieser konstruktivistischen Vorgehensweise sind vor allem die Methoden zur Evaluation der Artefakte aus den Entwurfprozessen von entscheidender Bedeutung. Experimente und Fallstudien bieten sich hierbei insbesondere an, um die Methoden und Werkzeuge der KSE einer systematischen Evaluation zu unterziehen [HeHa00]. Kontrollierte Experimente eignen sich insbesondere um die soziotechnischen Fragestellungen der KSE zu untersuchen, d.h. die Anwendung der Methoden und Werkzeuge durch die Prozessbeteiligten [Basi96, Tich98, MuTi01, SAAK02]. Eine aktuellen Überblick über experimentelle Arbeiten in der SE-Literatur der vergangenen zehn Jahre geben Sjoberg et al. [SHHK05]. Die Metastudie kommt zu dem Ergebnis, dass sich bisher nur knapp 2% der Publikationen des kontrollierten Experiments als Methodik bedienen.

Neben der weiteren Analyse existierender Ansätze im SE und der konzeptionellen Integration mit zwischenbetrieblichen Kollaborationsformen gilt es auch, die unterschiedlichen existierenden Softwareökosysteme in der Praxis empirisch zu untersuchen (vgl. Abschnitt 4.3). Beispiele für solche Ökosysteme sind die bereits genannten Partnernetzwerke von SAP und Microsoft, aber auch regionale Cluster von kleinen und mittleren Softwarefirmen, wie sie in Deutschland bereits zu beobachten sind. Ein Beispiel hierfür ist die Cluster-Initiative Unternehmenssoftware des Landes Baden-Württemberg, im Rahmen derer bereits zahlreiche Wirtschafts- und Wissenschaftsprojekte entstanden sind, beispielsweise ein „Transferprogramm zum Aufbau von Softwarelieferketten" (TASK)[6]. Neben neuen Ansätzen zur methodischen und softwaretechnischen Unterstützung zwischenbetrieblicher Softwareprojekte sind außerdem auch die unterschiedlichen Treiber und Barrieren der zwischenbetrieblichen Softwareerstellung auf unterschiedlichen Betrachtungsebenen weiter empirisch zu untersuchen (vgl. Tabelle 4).

Als theoretische Grundlage eines KSE-Paradigmas existiert bisher jedoch noch keine etablierte „*General Theory of Collaboration*", sondern lediglich vereinzelte Arbeiten, die sich seit Anfang der neunziger Jahre immer wieder sporadisch mit diesem Thema beschäftigen [Eggh91, WoGr91, SaLS00][7]. Noch weniger Theorie existiert bezüglich Kollaboration bei der Softwareerstellung. Sarker et al. beispielsweise, entwickeln einen methodischen Rahmen für die Kollaboration in virtuellen Teams über unterschiedliche Phasen im Softwareerstellungsprozess [SaLS00] (vgl. Abschnitt 3.1). Ein theoretischer Bezugsrahmen zur Einordnung von Ansätzen zur kollaborativen Softwareerstellung auf Projektebene und im zwischenbetrieblichen Unternehmenskontext kann den für die Ausbildung eines wissenschaftlichen Paradigmas nötigen Integrationsprozess über die unterschiedlichen Betrachtungsebenen konzeptionell unterstützen (siehe Abbildung 2).

6. Zusammenfassung und Ausblick

Ziel dieses Beitrags ist es, einen umfassenden und systematischen Überblick über existierende Ansätze zur kollaborativen Softwareerstellung zu liefern. Zu diesem Zweck wurde der Kollaborationsbegriff zunächst analysiert und erklärt, um auf dieser Basis dann die wesentlichen Dimensionen der kollaborativen Softwareerstellung zu identifizieren. Der hier verwendete Klassifikationsrahmen setzt sich zum einen aus den fünf Dimensionen räumliche, zeitliche und organisatorische Verteilung sowie Richtung und Intensität zusammen, zum anderen aus den wesentlichen Aspekten der Zusammenarbeit (Arbeitsteilung, Koordination und Kommunikation), ebenso wie der entsprechenden Werkzeugunterstützung (siehe Abbildung 2). Entsprechend der Literatur unterscheidet dieser Beitrag die Kategorien „Kollaboration auf Projektebene" und „zwischenbetriebliche Kollaboration" auf Unternehmensebene (vgl. Tabelle 4). Projektbezogene Ansätze fokussieren soziotechnische, methodische oder softwaretechnische Aspekte, wohingegen zwischenbetriebliche Kollaborationsformen die unterschiedlichen Beziehungen von Unternehmen im Softwareökosystem beschreiben und die Rahmenbedingungen für gemeinsame Softwareprojekte vorgeben. Die Dimensionen des Klassifikationsrahmens sind auf beide Kategorien gleichermaßen anwendbar. Aus der Konfluenz der projektbezogenen und der zwischenbetrieblichen Perspektive ergeben sich zahlreiche originäre Forschungsfragen, welche letztendlich zu einem eigenen Paradigma führen könnten.

Wie in Abschnitt 5 ausführlich dargelegt wurde besteht zukünftig Forschungsbedarf sowohl bei der Definition und der Etablierung eines Forschungsparadigmas der kollaborativen Soft-

[6] http://www.doit-task.de/

[7] Zur Weiterentwicklung dieser „General Theory of Collaboration" hat sich eine virtuelle Gemeinschaft gebildet, die sich mit den „*similarities and differences in the nature, methods and motivations of collaboration across any and every field of human endeavour*" beschäftigt:
http://collaboration.wikicities.com/wiki/Main_Page (2005-11-10)

wareerstellung als auch bei der Gestaltung der eigentlichen Lösungsansätze zur methodischen und softwaretechnischen Unterstützung der kollaborativen Softwareerstellung unter Berücksichtigung der soziotechnischen Perspektive und über mehrere Betrachtungsebenen (Individuum, Gruppe, Projekt, Unternehmen) hinweg.

Literaturverzeichnis

[AgAl01] *Agile Alliance*: Agile Manifesto. http://www.agilemanifesto.org/, 2001, Abruf am 2004-04-14.

[Alle77] *Allen, Thomas J.*: Managing the Flow of Technology. MIT Press, Cambridge, MA, USA, 1977.

[AlPo99] *Altmann, Josef;Pomberger, Gustav*: Kooperative Softwareentwicklung: Konzepte, Modell und Werkzeuge. In: Tagungsband der 4. Internationale Tagung Wirtschaftsinformatik (WI`99), Saarbrücken, 1999, S. 644-664.

[AlWe98] *Altmann, Josef; Weinreich, Rainer*: An Environment for Cooperative Software Development Realization and Implications. In: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences, Band 1, 1998, S. 27-37.

[Altm99] *Altmann, Josef*: Kooperative Softwareentwicklung – Rechnerunterstützte Koordination und Kooperation in Softwareprojekten. Dissertation, Universitätsverlag Rudolf Trauner, Linz, 1999.

[Andr03] Andriesen, J.H. Erik: Working with Groupware – Understanding and Evaluating Collaborative Technology. Springer, CSCW Series, London, 2003.

[AnZm01] *Andres, Hayward P.; Zmud, Robert W.*: A Contingency Approach to Software Project Coordination. In: Journal of Management Information Systems, 18 (2001) 3, S. 41-70.

[AuBS02] *Augustin, L.; Bressler, D.; Smith, G.*: Accelerating Software Development through Collaboration. In: Proceedings of the International Conference on Software Engineering 2002 (ICSE'02). Orlando 2002, S. 559-563.

[AvFi04] *Avison, David; Fitzgerald, Guy*: Information Systems Development – Methodologies, Techniques and Tools. 3. Aufl., McGraw-Hill, London 2004.

[BaGo04] *Babar, M.A.; Gorton, Ian*: Comparison of Scenario-Based Software Architecture Evaluation Methods. In: Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04), Busan, Korea, 2004, S. 600-607.

[Basi96] *Basili, Victor R.*: The Role of Experimentation in Software Engineering: Past, Current, and Future. In: Proceedings of the 18th International Conference on Software Engineering (ICSE '96), 1996, S. 442-449.

[BKZJ04] *Babar, M.A.; Kitchenham, B.; Zhu, L.; Jeffery, R.*: An Exploratory Study of Groupware Support for Distributed Software Architecture Evaluation Process. In: Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04), Busan, Korea, 2004.

[Balz00] *Balzert, Helmut*: Software-Entwicklung (Lehrbuch der Software-Technik, Band 1). 2. Auflage, Spektrum Akademischer Verlag, Heidelberg, 2000.

[BCGM03] *Ballarini, Daniele; Cadoli, Marco; Gaetta, Matteo; Mancini, Toni; Mecella, Massimo; Ritrovato, Pierluigi; Santucci, Giuseppe*: Modeling Real Requirements for Cooperative Software Development: A Case Study. Working Paper, 2003.

[BeSc89] *Bendifallah, Salah; Scacchi, Walt*: Work Structures and Shifts: An Empirical Analysis of Software Specification Teamwork. In: ICSE '89: Proceedings of the 11th International Conference on Software Engineering, 1989, S. 260-270.

[Bidl68] *Bidlingmaier, J.*: Begriff und Formen der Kooperation im Handel. In: Bidlingmaier, Jacobi, Uherek (Hrsg.): Absatzpolitik und Distribution. Wiesbaden, 1968, S. 354-359.

[BKMS95] *Bischofberger, Walter R.; Kofler, Thomas; Mätzel, Kai-Uwe; Schäffer, Bruno*: Computer Supported Cooperative Software Engineering with Beyond-Sniff. In: Proceedings of the 7th Conference on Software Engineering Environments (SEE'95), Noorwijkerhout, Niederlande, 1995.

[BoBo94] *Boehm, Barry W.; Bose, Prasanta*: A Collaborative Spiral Software Process Model Based on Theory W. In: Proceedings of the 3rd International Conference on the Software Process, Applying the Software Process, 1994.

[BoBr03] *Booch, Grady; Brown, Alan W.*: Collaborative Development Environments. In: Advances in Computers, 59 (2003) August, Academic Press.

[BoGB01] *Boehm, Barry W.; Gruenbacher, Paul; Briggs, Robert O.*: Developing Groupware for Requirements Negotiations: Lessons Learned. In: IEEE Software, 18 (2001) 3, S. 46-55.

[BoGB01a] *Boehm, Barry W.; Gruenbacher, Paul; Briggs, Robert O.*: EasyWinWin: A Groupware-Supported Methodology For Requirements Negotiation. In: 23rd International Conference on Software Engineering, IEEE Computer Society, 2001.

[BNRS03] *Boldyreff, Cornelia; Nutter, David; Rank, Stephen; Smith, Mike; Wilcox, Pauline; Dewar, Rick; Weiss, Dawid; Ritrovato, Pierluigi*: Environments to Support Collaborative Software Engineering. In: Proceedings of the 2nd Workshop on Cooperative Supports for Distributed Software Engineering Processes, Benevento, Italien, 2003.

[BoRo89] *Boehm, Barry W.; Ross, Rony*: Theory W Software Project Management: Principles and Examples. In: IEEE Transaction of Software Engineering, 1989, S. 902-916.

[Boeh81] *Boehm, Barry W.*: Software Engineering Economics. Prentice Hall, Englewood Cliffs, NJ, USA, 1981.

[BrRa05] *Broy, Manfred; Rausch, Andreas*: Das neue V-Modell XT - Ein anpassbares Modell für Software und System Engineering. In: Informatik Spektrum, 28 (2005) 3, S. 220-229.

[BrGr02] *Robert O. Briggs and Paul Gruenbacher*: EasyWinWin: Managing Complexity in Requirements Negotiation with GSS, In: Proceedings of the 35th Hawaii International Conference on System Sciences, 2002.

[CaCo96] *Cain, B.G.; Coplien, J.O.*: Social Patterns in Productive Software Development Organizations. In: Annals of Software Engineering, 1996.

[CaAg01] *Carmel, E. ; Agarwal, R.*: Tactical Approaches for Alleviating Distance in Global Software Development. In: IEEE Software, 18 (2001) 2, S. 22-29.

[CaLM02] *Canfora, Gerardo; Lanubile, Filippo; Mallardo, Teresa*: Can Collaborative Software Development Benefit from Synchronous Groupware Functions. In: Technical Report, University of Sannio, 2002.

[Cany01] *CanyonBlue*: Collaborative UML Development. http://www.canyonblue.com/CanyonBlue.pdf, Abruf am 2005-05-01.

[CaWG93] *Carmel, E.; Whitaker, R.D.; George, J.F.*: PD and Joint Application Design: A Transatlantic Comparison. In: Communcations of the ACM, 36 (1993) 6, S. 40-48.

[CDHP03] *Cheng, Li-Te; de Souza, Cleidson R.B.; Hupfer, Susanne; Patterson, John; Ross, Steven***:** Building Collaboration into IDEs. In: ACM Queue, 1 (2003) 9, S. 40-50.

[CHRP03] *Cheng, Li-Te; Hupfer, Susanne; Ross, Steven; Patterson, John*: Jazzing up Eclipse with Collaborative Tools. In: Proceedings of the 2003 OOPSLA Workshop on Eclipse Technology eXchange, ACM Press, 2003, S. 45-49.

[ChSp00] *Chu-Carrol, Marc C. and Sprenkle, Sara*: Coven: Brewing Better Collaboration through Software Configuration Management. In: Proceedings of the 8th ACM SIGSOFT International Symposium on Foundations of Software Engineering: Twenty-first Century Applications. San Diego, USA, 2000.

[CPRH05] *Cheng, Li-Te; Patterson, John;Rohall, Steven L.; Hupfer, Susanne; Ross, Steven*: Weaving a Social Fabric into Existing Software. In: Proceedings of the International Conference on Aspect-Oriented Software Development Chicago, Illinois USA, 2005.

[Cock03] *Cockburn, Alistair*: Agile Software-Entwicklung. Verlag Moderne Industrie, 2003.

[CoCh03] *Cook, Carl; Churcher, Neville*: An Extensible Framework for Collaborative Software Engineering. In: Proceedings of the Tenth Asia-Pacific Software Engineering Conference (APSEC'03), Chiang Mai, Thailand, 2003.

[CoCI04] *Cook, Carl; Churcher, Neville; Irwin, Warwick*: Towards Synchronous Collaborative Software Development. In: Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04), Busan, Korea, 2004.

[Coll05] Collins English Dictionary. 7. Aufl., HarperCollins Publishers, Glasgow, GB, 2005.

[Cook04] *Cook, Carl*: Collaborative Software Engineering: An Annotated Bibliography. Arbeitspapier TR-02/04. Christchurch, New Zealand, 2004. http://nz.cosc.canterbury.ac.nz/research/ reports/TechReps/2004/tr_0402.pdf, Abruf am 2005-03-13.

[CuKI88] *Curtis, Bill; Krasner, Herb; Iscoe, Neil*: A Field Study of the Software Design Process for Large Systems. In: Communications of the ACM, 31 (1988) 11, S. 1268-1287.

[DESG03] *Damian, Daniela E.; Eberlein, Armin; Shaw, Mildred L.G.; Gaines, Brian R.*: An Exploratory Study of Facilitation in Distributed Requirements Engineering. In: Requirements Engineering Journal: Special Issue on Selected Papers from RE'01, 8 (2003) 1, S. 23-41.

[DeDe04] *DeFranco-Tommarello, Joanna; Deek, Fadi P.*: Collaborative Problem Solving and Groupware for Software Development. In: Information Systems Management, 2 (2004), S. 67-80.

[DLPH98] *Dean, D.L.; Lee, J.D.; Pendergast, M.O.; Hickey, A.M.; Nunamaker, J.F.*: Enabling the Effective Involvement of Multiple Users: Methods and Tools for Collaborative Software Development. In: Journal of Mangement Information Systems, 14 (1998).

[DRMP03] *De Souza, Cleidson R.B.; Redmiles, David;Mark, Gloria; Penix, John; Sierhuis, Maarten*: Management of Interdependencies in Collaboratve Software Development. In: Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03), 2003.

[DRCM04] *De Souza, Cleidson R.B.; Redmiles, David; Cheng, Li-Te; Millen, David; Patterson, John*: How Good Software Practice Thwarts Collaboration – The Multiple Roles of APIs in Software Development.

[DeRi93] *Dewan, P.; Riedl, J.*: Toward Computer-Supported Concurrent Software Engineering. In: IEEE Computer, 26 (1993) 1, S. 17-27.

[Dibb04] *Dibbern, Jens*: Sourcing of Application Software Services: Empirical Evidence of Cultural, Industry and Functional Differences. Physica-Verlag, 2004.

[DoKG99] *Dossick, Stephen E.; Kaiser, Gail E.*: CHIME: A Metadata-Based Distributed Software Development Environment. In: Proceedings of the 7th European Software Engineering Conference held jointly with the 7th ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE-7), 1999, S. 464-475.

[Dude01] *Dudenredaktion* (Hrsg.): Duden Band 7:Herkunftswörterbuch – Etymologie der deutschen Sprache. 3. Aufl., Dudenverlag, Mannheim, 2001.

[Dude04] *Dudenredaktion* (Hrsg.): Duden Band 1: Die Deutsche Rechtschreibung. 23. Aufl., Dudenverlag, Mannheim, 2004.

[Dude04a] *Dudenredaktion* (Hrsg.): Duden Band 8: Synonymwörterbuch – Ein Wörterbuch sinnverwandter Wörter. 3. Aufl., Dudenverlag, Mannheim, 2004.

[EdSr03] *Edwards, H. Keith; Sridhar, Varadharajan*: Analysis of the Effectiveness of Global Virtual Teams in Software Engineering Projects. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03), 2003, S. 19-28.

[Eggh91] Egghe, L.: Theory of Collaboration and Collaborative Measures. In: Information Processing and Management: an International Journal, 27 (1991) 2/3, S. 177-202.

[ElGR91] *Ellis, C.A.; Gibbs, S.J.; Rein, G.*: Groupware: Some Issues and Experiences. In: Communications of the ACM, 34 (1991) 1, S. 39-58.

[FeFi01] *Feller, J. & Fitzgerald, B.*: Understanding Open Source Software Development. Addison-Wesley, 2001.

[FoBa02] *Fogel, K.; Bar, M.*: Open Source-Projekte mit CVS. MITP-Verlag, 2002.

[FrLa98] *French, Andy; Layzell, Paul*: A Study of Communication and Cooperation in Distributed Software Project Teams. In: Proceedings of the International Conference on Software Maintenance, Bethesda, Maryland, USA, 1998, S. 146-154.

[GaRi02] *Gaeta, Matteo; Ritrovato, Pierluigi*: Generalised Environment for Process Management in Cooperative Software Engineering. In: Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02), Oxford, England, 2002, S. 1049-1053.

[Gehr78] Gehrt, Ernst: Zwischenbetriebliche Kooperation. Poeschel, Stuttgart, 1978.

[Gent05] *Gentleware*: Poseidon for UML (Produktbeschreibung). http://www.gentleware.de/index.php?id=30, Abruf am 2005-04-28.

[GoKo99] *Goldmann, S.; Koetting, B.*: Coordinating Distributed Software Development Projects: Final Summary Report. In: Proceedings of the '99 Workshop on Coordinating Distributed Software Development Projects, IEEE, 1999, S. 9-14.

[Gold02] *Goldberg, Adele*: Collaborative Software Engineering. In: Journal of Object Technologie, 1 (2002) 1, S. 1-19.

[GHRC97] *Gorton, Ian; Hawryszkiewycz, Igor; Ragoonaden, Kenny; Chung, Charles; Lu, Shijian; Randhawa, Guneet*: Groupware Support Tools for Collaborative Software Engineering. In: Proceedings of the 30th Hawaii International Conference on System Sciences, USA, 1997, S. 157-166.

[GrBr01] *Gruenbacher, Paul; Briggs, Robert O.*: Surfacing Tacit Knowledge in Requirements Negotiation: Experiences Using Easy Win Win. In: Proceedings Hawaii International Conference on System Sciences, IEEE Computer Society, 2001.

[GrBr03] *Gruenbacher, Paul; Braunsberger, Patrick*: Tool Support For Distributed Requirements Negotiation. In: Cooperative Methods and Tools for Distributed Software Processes, Franco Angeli, Milano, Italy, 2003.

[Gree04] *Greenfield, Jack:* Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools. Wiley, Indianapolis, USA, 2004.

[Grot04] *Groth, Robert***:** Is the Software Industry's Productivity Declining? In: IEEE Software 21 (2004) 6, S. 92-94.

[GSKR99] *Graham, T.; Stewart, H.; Kopaee, A.; Ryman, A.; Rasouli, R.*: A World-Wide-Web Architecture for Collaborative Software Design. In: Proceedings of the Conference on Software Technology and Engineering Practice, Pittsburgh, Pennsylvania, USA, 1999, S. 22-29.

[Grue00] *Gruenbacher, Paul*: Collaborative Requirements Negotiation with EasyWinWin. In: Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA'00), IEEE, 2000.

[HaPL03] *Hardy, Cynthia; Phillips, Nelson; Lawrence, Thomas B.*: Resources, Knowledge and Influence: The Organizational Effects of Interorganizational Collaboration. In: Journal of Management Studies, 40 (2003) 2, S. 321-347.

[HPGF02] *Harrison, D.A.; Price, K.H.; Gavin, J.H.; Florey, A.T.*: Time, Teams, and Task Performance: Changing Effects of Surface- and Deep-Level Diversity on Group Functioning. In: Academy of Management Journal, 45 (2002) 5, S. 1029-1045.

[HaWo01] *Hause, Martha L.; Woodroffe, Mark R.*: Team Performance Factors in Distributed Collaborative Software Development. In: Proceedings of the 13th Workshop of the Psychology of Programming Interest Group (PPIG'01), Bournemouth, UK, 2001.

[HeGr99] *Herbsleb, J. D.; Grinter, Rebecca E.*: Splitting the Organization and Integrating the Code: Conway's Law Revisited. In: Proceedings of the 21st international conference on Software engineering (ICSE '99), Los Alamitos, CA, USA, 1999, S. 85-95.

[Hein93] *Heinzl, Armin*: Die Ausgliederung der betrieblichen Datenverarbeitung: eine empirische Analyse der Motive, Formen und Wirkungen. 2. Aufl., Schaeffer-Poeschel, 1993.

[HeHa00] Heinrich, Lutz J.; Häntschel, I.: Evaluation und Evaluationsforschung in der Wirtschaftsinformatik:Handbuch für Praxis, Lehre und Forschung. Oldenbourg Verlag, 2000.

[HeHR04] *Heinrich, Lutz J.; Heinzl, Armin; Roithmayr, F.*: Wirtschaftsinformatik-Lexikon. 7. Aufl., Oldenbourg, München 2004.

[HeMo01] *Herbsleb, J. D.; Moitra, D.*: Global Software Development. In: IEEE Software 18 (2001) 2, S. 16-20.

[HeMo03] *Herbsleb, James D.; Mockus, Audris*: An Empirical Study of Speed and Communication in Globally-Distributed Software Development. In: IEEE Transactions on Software Engineering 29 (2001) 6, S. 481-494.

[Henk03] Henkel, Joachim. *Software Development in Embedded Linux – Informal Collaboration of Competing Firms*. Working Paper, 2003.

[Henk03a] Henkel, Joachim. *Open Source Software from Commercial Firms -- Tools, Complements, and Collective Invention*. In: Zeitschrift für Betriebswirtschaft, 4/2004.

[HeRe01] *Herring, Charles; Rees, Michael*: Internet-based Collaborative Software Development using Microsoft Tools. In: Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (ISAS-SCI '01), Volume 1, 2001, S. 162-167.

[HeSi93] *Heinzl, Armin; Sinß, Michael*: Kooperationen zur zwischenbetrieblichen Entwicklung von Anwendungssystemen. In: Information Management, 2 (1993), 60-67.

[High02] *Highsmith, Jim*: Agile Software Development Ecosystems. Addison-Wesley, Boston, MA, USA, 2002.

[HiKo04] *Hildenbrand, Tobias; Korthaus, Axel*: A Model-Driven Approach to Business Software Engineering. In: Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, USA, 2004, S. 74-79.

[HMPR04] *Hevner, Alan R.; March, Salvatore T.; Park, Jinsoo; Ram, Sudha*: Design Science Information Systems Research. In: MIS Quarterly, 28 (2004) 1, S. 75-105.

[Hrus03] *Hruschka, P.*: Agility - (Rück-)Besinnung auf Grundwerte in der Softwareentwicklung. In: Informatik Spektrum (2003) 6, S. 397-401.

[IBM05] *IBM***:** Rational Software Development Tools. http://www.ibm.com/software/rational/, Abruf am 2005-04-28.

[IBM05a] *IBM Customer Success*: Florida Department of Health Relies on RUP and IBM Rational Tools for Long-Term Projects and Rapid Crisis Response. http://www3.software.ibm.com/ibmdl/pub/software/rational/web/success/fl_dept_health.pdf, Abruf am 2005-04-28.

[JaBR99] *Jacobson, Ivar; Booch, Grady; Rumbaugh, John*: The Unified Software Development Process. Addison-Wesley, Reading, USA, 1999.

[JaKL98] *Jarvenpaa, S.L.; Knoll, K. & Leidner, D.E.*: Is Anybody Out There? Antecedents of Trust in Global Virtual Teams. In: Journal of Management Information Systems, 14 (1998) 4, S. 29-64.

[JaLe99] *Jarvenpaa, S.L. & Leidner, D.E.*: Communication and Trust in Global Virtual Teams. In: Organization Science, 10 (1999) 6, S. 791-815.

[JuBe03] *Junker, Abbo; Benecke, Martina*: Computerrecht. 3. Aufl., Nomos-Verlagsgesellschagft, 2003.

[Kell98] *Kelley, D.S.*: Web Enabled Engineering Collaboration. In: Journal of Industrial Technology, 14 (1998) 2, S. 45-47.

[Kiss03] *Kiss, Ferenc*: Tools für Teams. In: JavaMagazin 10 (2003), S. 76-81.

[Knob69] *Knoblich, H.*: Zwischenbetriebliche Kooperation – Wesen, Formen und Ziele. In: Zeitschrift für Betriebswirtschaft, 39 (1969) 8, Wiesbaden, S. 497-514.

[KrEG88] *Kraut, Robert; Egido, Carmen; Galegher, Jolene*: Patterns of Contact and Communication in Scientific Research Collaboration. In: CSCW '88: Proceedings of the 1988 ACM Conference on Computer-Supported Cooperative Work, 1988, S. 1-12.

[KrSt95] *Kraut, Robert; Streeter, L.A.*: Coordination in Software Development. In: Communications of the ACM, 38 (1995) 3, S. 69-81.

[Kruc04] *Kruchten, P.*: The Rational Unified Process – An Introduction. 3. Aufl., Addison-Wesley Professional, 2004.

[KTCB92] *Kaplan, Simon M.; Tolone, William J.; Carrol, Alan M.; Bogia, Douglas P.; Bignoli, Celsina*: Supporting Collaborative Software Develpoment with ConversationBuilder. In: Proceedings of the Fifth ACM SIGSOFT Symposium on Software Development Environments (SDE 5), 1992, S. 11-20.

[KuDi96] *Kumar, K.; van Dissel, H.G.*: Sustainable Collaboration: Managing Conflict and Cooperation in Interorganizational Systems. In: MIS Quarterly, 20 (1996) 3, S. 279-330.

[Kuhn62] *Kuhn, Thomas Samuel*: The Structure of Scientific Revolutions. University of Chicago Press, Chicago, 1962.

[LaHP02] *Lawrence, Thomas B.; Hardy, Cynthia; Phillips, Nelson*: Institutional Effects of Interorganizational Collaboration: The Emergence of Proto-Institutions. Academy of Management Journal, 45 (2002) 1, S. 281-290.

[LaMa97] *Laubacher,R.J.; Malone, T.W.*: Two Scenarios for 21st Century Organizations: Shifting Networks of Small Firms or All-Encompassing "Virtual Countries"?. Arbeitspapier 21C WP \#001 (1997)

[LaBF00] *Layzell, P.; Brereton, O.; French, A.*: Supporting Collaboration in Distributed Software Engineering Teams. In: Proceedings of the Seventh Asia-Pacific Software Engineering Conference (APSEC'00), Singapur, 2000.

[LlRA02] *Lloyd, Wesley James; Rosson, Mary Beth; Arthur, James D.*: Effectiveness of Elicitation Techniques in Distributed Requirements Engineering. In: Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02), 2002.

[LoRo93] *Lougher, Robert; Rodden, Tom*: Supporting Long-term Collaboration in Software Maintenance. In: Proceedings of the Conference on Organizational Computing Systems (COCS '93), New York USA, 1993.

[LBSZ01] *Luczak, D.; Bullinger, H.-J.; Schlick, C.; Ziegler, J. (Hrsg.)*: Unterstützung flexibler Kooperation durch Software : Methoden, Systeme, Beispiele. Springer, Heidelberg 2001.

[LDCE03] *Longman Dictionary of Contemporary English*. 7. Aufl., Pearson-Longman 2003

[MaLa03] *Malone, Thomas W.; Laubacher,R.J.; Scott-Morton, M.S. (Hrsg.)*: Inventing the Organizations of the 21st Century. MIT Press, Cambridge 2003

[MaCr94] *Malone, Thomas W.; Crowston, Kevin*: The Interdisciplinary Study of Coordination. In: ACM Computing Surveys, 26 (1994) 1, S. 87-119.

[MaFR94] *Mashayekhi, Vahid; Feulner, Chris and Riedl. John*: CAIS: Collaborative Asynchronous Inspection of Software. In: Proceedings of the 2nd ACM SIGSOFT Symposium on Foundations of Software Engineering (SIGSOFT '94), New York, 1994, S. 21-34.

[Marl00] *Marly, Jochen*: Softwareüberlassungsverträge: Erscheinungsformen, Pflichtverletzungen, Vertragsgestaltung, Allgemeine Geschäftsbedingungen, Musterverträge. 3. Aufl., Beck Juristischer Verlag, München, 2000.

[MaSo04] *Martin, David; Sommerville, Ian*: Patterns of Cooperative Interaction: Linking Ethnomethodology and Design. In: ACM Transactions on Computer-Human Interaction, 11 (2004) 1, S. 59-89.

[MDTR93] *Mashayekhi, Vahid; Drake, Janet M.; Tsai, Wei-Tek; Riedl, John*: Distributed, Collaborative Software Inspection. In: IEEE Software, 10 (1993) 5, S. 66-75.

[MeSz01] *Messerschmitt, David G.; Szyperski, Clemens*: Industrial and Economic Properties of Software: Technology, Processes, and Value. Company Whitepaper 11, Microsoft, 2001.

[MeSz03] *Messerschmitt, David G.; Szyperski, Clemens*: Software Ecosystem: Understanding an Indispensable Technology and Industry. MIT Press, 2003.

[MeWe01] *Gove, Philip Babcock* (Hrsg.): Webster's Third New International Dictionary of the English Language. 3. Aufl., Springfield, USA, 2001.

[Micr04] *Microsoft Corporation*: Making Collaboration the Engine of Team Productivity: A Single Place for File Sharing and Collaboration Using Windows SharePoint Services. Company Whitepaper, März, 2004. http://download.microsoft.com/download/6/1/f/61f17de6-4a7e-4ba0-a9e4-c96d75e2cc42/WSSvision.doc. Abruf am 2005-05-10.

[MoFH02] *Mockus, A.; Fielding, R. & Herbsleb, James D.*: Two Case Studies of Open Source Software Development: Apache and Mozilla. In: ACM Transactions on Software Engineering and Methodology, 11 (2002) 3, S. 309-346.

[MSHK99] *Maurer, F.; Succi, G.; Holtz, H.; Kötting, B.*: Software Process Support over the Internet. In: Proceedings of the 1999 International Conference on Software Engineering (ICSE'99).

[MRMK00] *Majchrzak, A.; Rice, D.E.; Malhotra, A.; King, N.; Ba, Sulin***:** Technology Adaption: The Case of a Computer-Supported Inter-Organizational Virtual Team. In: MIS Quarterly, 24 (2000) 4, S. 569-600.

[MuTi01] *Müller, Matthias M.; Tichy, Walter F.*: Case Study: Extreme Programming in a University Environment. In: Proceedings of the 23rd International Conference on Software Engineering (ICSE'01), Washington, DC, USA, 2001, S. 537-544.

[NODE01] *Judy Pearsall* (Hrsg.): The New Oxford Dictionary of English. Oxford University Press, New York, USA, 2001.

[OALD05] *Hornby, A. S.* (Hrsg.): Oxford Advanced Learner's Dictionary. Oxford University Press, 2005.

[Ober94] *Oberweis, Andreas***:** Workflow Management in Software Engineering Projects. In: Proceedings of the 2nd International Conference on Concurrent Engineering and Electronic Design Automation, Bournemouth, UK, 1994.

[ObWS94] *Oberweis, Andreas; Wendel, Thomas; Stucky, Wolfried*: Teamwork Coordination in a Distributed Software Development Environment. In: Proceedings of the IFIP'94 Workshop FG 9: Communication and Coordination in Distributed Corporate Application Systems, Hamburg, Germany, 1994.

[Port80] *Porter, Michael E.*: Competitive Strategy: Techniques for Analyzing Industries and Competitors. Free Press, New York, USA, 1980.

[Robe05] Robert, Paul: PONS – Micro Robert Poche: Dictionnaire d'apprentissage de la langue francaise. Ernst Klett Verlag, 2005.

[SAAK02] *Sjoberg, D.I.; Anda, B.; Arisholm, E.; Karahasanovic, T.D.a.J.a.; Koren, E.F.; Vokác, M.*: Conducting Realistic Experiments in Software Engineering. In: Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE'02), 2002, S. 17-26.

[SaGu98] *Sawyer, Steve; Guinan, P.J.*: Software Development: Processes and Performance. In: IBM Systems Journal, 37 (1998) 4, 552-569.

[SaLS00] *Sarker, Suprateek; Lau, Francis; Sahay, Sundeep*: Building an Inductive Theory of Collaboration in Virtual Teams: An Adapted Grounded Theory Approach. In: Proceedings of the 33rd Hawaii International Conference on System Sciences, Volume 7, 2002.

[Sawy04] *Sawyer, Steve*: Software Development Teams. In: Communications of the ACM, 47 (2004) 12, S. 95-99.

[ScSU01] *Schwabe, G.; Streitz, N.; Unland, R.*: CSCW-Kompendium : Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten. Springer, Heidelberg 2001.

[SGMT04] *Seyff, Norbert; Grünbacher, Paul; Maiden, Neil; Tosar, Amit*: Mobile Werkzeuge im Requirements Engineering. In: Softwaretechnik-Trends, 24 (2004) 1.

[SHHK05] *Sjoberg, D.I.; Hannay, J.E.; Hansen, O.; Kampenes, V.B.; Karahasanovic, A.; Liborg, N.; Rekdal, A.C.*: A Survey of Controlled Experiments in Software Engineering. In: IEEE Transactions on Software Engineering, 31 (2005) 9, S. 733-753.

[Somm04] *Sommerville, Ian*: Software Engineering. 7. Aufl., Addison-Wesley, Boston, MA, USA, 2004.

[SRHM97] *Stein, Michael; Riedl, John; Harner, Soren J.; Mashayekhi, Vahid*: A Case Study of Distributed, Asynchronous Software Inspection. In: Proceedings of the 19th International Conference on Software Engineering (ICSE '97), New York, 1997, S. 107-117.

[Ster02] *Sterkmann, Frank*: Entwicklung global. . In: JavaMagazin 2 (2002), S. 36-40.

[StHa04] *Stahlknecht und Hasenkamp*: Einführung in die Wirtschaftsinformatik. 11. Aufl., Springer, 2004.

[SpSz04] *Spinellis***,** *Diomidis***;** *Szyperski, Clemens***:** How Is Open Source Affecting Software Development. In: IEEE Software, 21 (2004) 1, S. 28-33.

[TCKO00] *Teasley, Stephanie; Covi, Lisa; Krishnan, M.S.; Olson, Judith S.*: How Does Radical Collocation Help a Team Succeed? In: Proceedings of the 2000 ACM conference on Computer supported cooperative work (CSCW'00), Philadelphia, Pennsylvania, USA 2000, S. 339-346.

[ThLo04] *Theling, Thomas; Loos, Peter*: Determinanten und Formen von Unternehmenskooperationen. Arbeitspapier 18, Information Systems & Management, Johannes-Gutenberg-Universität Mainz, 2004.

[Tich98] *Walter F. Tichy*: Should Computer Scientists Experiment More? In: IEEE Computer, 31 (1998) 5, S. 32-40.

[VASo05] *VA Software*: SourceForge Enterprise Edition Product Introduction. http://www.vasoftware.com/sourceforge/, Abruf am 2005-04-29.

[Webs03] *Webster, Melissa*: An End-User View of the Collaborative Software Development Market. IDC, 2003.

[WMMP03] *Werner, Claudia; Mangan, Marco; Murta, Leonardo; Pinheiro, Robson; Mattoso, Marta; Braga, Regina; Borges, Marcos*: OdysseyShare: an Environment for Collaborative Component-Based Development. In: Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2003), Rio de Janeiro, Brazil, 2003, S. 61-68.

[WoGr91] *Wood, Donna J.; Gray, Barbara*: Toward a Comprehensive Theory of Collaboration. In: The Journal of Applied Behavioral Science, 27 (1991) 2, S. 139-162.

[WRSS03] *Wilcox, P. A.; Russell, C.R.; Smith, M.J.; Smith, A.D.; Pooley, R.J.; MacKinnon, L.M.; Dewar, R.G.*: A CORBA-Oriented Approach to Heterogeneous Tool Integration; OPHELIA. In: Proceedings of the ESEC/FSE Workshop on Tool-Integration in System Development, Helsinki, Finland, 2003, S. 1-5.