

Manifest¹

Strategische Bedeutung des Software Engineering in Deutschland

Software ist der fundamentale *Werkstoff* des *Informationszeitalters*. Innovative Produkte und Dienstleistungen sind ohne Software nicht mehr denkbar. Die Wettbewerbsfähigkeit der deutschen Wirtschaft hängt entscheidend von der Fähigkeit ab, Software-intensive Produkte und Dienstleistungen mit höchster Qualität zu erstellen. Software Engineering über dem Weltniveau ist die Voraussetzung dafür, dass Deutschland seine führende Stellung im Ingenieurbereich, etwa im Maschinenbau, halten und ausbauen und entsprechende Positionen in neuen Sparten, etwa im modernen Gesundheitswesen (e-Health), aufbauen kann.

Software wird in der Zukunft *integrierter* – in vielen Fällen sogar *dominierender* – *Teil großer komplexer Systeme* sein. Nicht nur in der Automobil- und Luftfahrtindustrie wird dieser Trend bereits heute deutlich sichtbar. Die erforderliche Integration von Mechanik, Elektronik und Software und die Vermeidung unerwünschter Wechselwirkungen kann nur durch die frühzeitige Integration der Modellierungskompetenzen des Software Engineering in den Entwicklungsprozess beherrscht werden.

Diese neue Positionierung von Software Engineering als systemische Disziplin erfordert eine *neue Ausrichtung und Stärkung* der Bereiche Forschung, Lehre und Technologietransfer. Es ist die auf sorgfältige Analysen abgestützte Überzeugung der Unterzeichner, dass verstärkte Anstrengungen in allen drei Bereichen notwendig sind, um die Herausforderungen des Informationszeitalters und der Globalisierung anzunehmen. Dieses Manifest stellt Forderungen inhaltlicher und organisatorischer Art für Forschung, Lehre und Technologietransfer auf, deren Erfüllung Voraussetzung für die Realisierung künftiger weltweiter wirtschaftlicher Potentiale ist.

Der Standort Deutschland wird sich in diesem Markt nur behaupten, wenn Software Engineering sowie Softwareingenieure auf höchstem Qualitäts- und Produktivitätsniveau verfügbar sind.

Aufruf

Ausbau des Software Engineering

An die Länder, aber auch an die Universitätsleitungen ergeht die dringende Aufforderung, die Anzahl der Gruppen im Software Engineering auszubauen, da diese im besonderen Maße für die Qualität der zukünftigen Softwareingenieure verantwortlich sind. Sie müssen die noch zu

¹ Dieses Manifest wurde im Informatik Spektrum Heft 3, 2006 veröffentlicht. Es entstand als Initiative des Dagstuhl Workshops 05402 "Challenges for Software Engineering Research", Oktober 2005

entwickelnden spezifischen Softwaretechniken ausgestalten, für eine Zusammenarbeit mit der Industrie zur Verfügung stehen, aber auch die Weiterbildung für Industrieangehörige in Zukunft koordinieren und größtenteils tragen.

Umstrukturierung

Ohne ihren wissenschaftlichen Charakter aufzugeben, muss sich die deutsche Software Engineering-Forschungslandschaft industrieller Kooperation stärker öffnen. Die oben angesprochenen Anwendungen, insbesondere in der Sekundärbranche, verlangen die Entwicklung/Weiterentwicklung spezifischer Softwaretechniken, die im benötigten Ausmaß derzeit nicht vorliegen.

Forschungsförderung

Angesichts der hohen gegenwärtigen Bedeutung und der sich noch verstärkenden Bedeutung in Zukunft ist von Seiten des BMBF eine wesentlich höhere Förderung für Software Engineering vorzusehen. Auch die Deutsche Forschungsgemeinschaft muss diese Thematik in besonderer Weise fördern, indem spezielle Programme aufgelegt werden und sich sowohl DFG als auch Gutachter erkennen, dass Grundlagenorientierung – wie in allen Ingenieurwissenschaften – nicht nur Theorie bedeutet.

Technologietransfer

Die Industrie wird aufgefordert, diese Zusammenarbeit verstärkt zu suchen bzw. anzunehmen. Innerhalb dieser Zusammenarbeit kann vorhandenes Wissen transferiert werden bzw. gemeinsam erarbeitetes für die Nutzung aufbereitet werden. Diese Zusammenarbeit besteht nicht aus kurzzeitiger und kurzfristiger Entwicklungsarbeit. Eine längerfristige Zusammenarbeit im beschriebenen engen Kooperationsmodus bezüglich des Horizonts der Aufgabenstellung und der Kooperationszeit ist eine ideale Basis für das Rekrutieren von Mitarbeitern, deren Qualifikation und Eignung der Industriepartner dann genau kennt.

Neuqualifizierung und Weiterbildung

Die Neuqualifizierung und Weiterbildung von im Beruf stehenden Praktikern ist aufgrund der Überlast der Informatik in der normalen Ausbildung noch unterentwickelt. Die Universitäten müssen diese Aufgaben aufgreifen. Die Aufgabe reicht von einzelnen Fortbildungsveranstaltungen zu spezifischen Themen oder zu einzelnen neuen aktuellen Ansätzen bis hin zu Ausbildungsgängen für Praktiker, insbesondere für Softwareentwickler mit „Fremdqualifikation“ bzw. in Zukunft auch für Bachelors. Die Wirtschaft muss sich aber bewusst sein, dass hier – wie in den USA – nicht unerhebliche finanzielle Aufwendungen auf sie zukommen. Diese sind jedoch als Investition in die Zukunft anzusehen und werden sich in erhöhter Produktivität niederschlagen.

Bestandsaufnahme

Software Engineering

Definition und Abgrenzung

Software Engineering ist die *Methodenlehre* der Softwarekonstruktion. Ihre Hauptbeiträge sind methodische Hilfsmittel wie Prozesse, Modelle, Werkzeuge und Prinzipien zur Konstruktion und Beherrschung des Einsatzes hochwertiger Software. „Hochwertig“ bedeutet hier

zuverlässig, sicher, leicht veränderbar, einfach benutzbar und mit den richtigen, erwünschten und vom Nutzer benötigten Funktionen.

Das Software Engineering grenzt sich zu seinen *Grundlagendisziplinen* (Kerninformatik, Ingenieurwissenschaften, Mathematik, Psychologie, Soziologie, Betriebswirtschaft) dadurch ab, dass es sich vor allem auf Methoden konzentriert und dabei immer den Anspruch praktischer Anwendbarkeit vor Augen hat. Die *Anwendungsfelder* liegen hauptsächlich im Bau von Konsum- und Investitionsgütern (Fächer: Elektrotechnik, Maschinenbau, Verfahrenstechnik) sowie der Unterstützung und Automatisierung von Unternehmensprozessen (Betriebswirtschaft, Gesundheitswesen, Verwaltung). Diese Anwendungsfelder liefern vor allem fachliche Anforderungen für Software. Das Software Engineering stellt die Methoden und Prozesse für die Erhebung der Anforderungen und deren technische Umsetzung bereit. Weder zu den Grundlagen- noch zu den Anwendungsfeldern hin ist eine scharfe Abgrenzung möglich; Software Engineering ist inhärent *transdisziplinär*.

Zur *Technik* für die Konstruktion von Software, Software Engineering in engerem Sinne, zählen Konzepte, Sprachen, Methoden, Werkzeuge und deren Grundlagen, Dienstkomponenten, Plattformen, Rahmenwerke aus wieder verwendbaren Bausteinen

Die Aufgabe des Software Engineering in Forschung und Lehre besteht darin, den Stand der Disziplin voranzutreiben und die Expertise der Softwareingenieure zu gewährleisten.

Software Engineering in der industriellen Praxis

Wachsende Bedeutung im Sekundärbereich

Software ist heute *wettbewerbsentscheidender* Faktor. Die Primärbranche zeichnet sich durch die Entwicklung eigenständiger Softwareprodukte aus. 80% der Softwareingenieure arbeiten in Sekundärbranchen (eingebettete Software in Produkten und Dienstleistungen, z.B. Maschinen- und Fahrzeugbau, Elektrotechnik, Telekommunikation, aber auch Unterhaltungsbranche, Medizin u.a.). Für die Wertschöpfung im Produktionsgüter- und Dienstleistungsbereich ist Software Engineering die „Produktionstechnik des 21. Jahrhunderts“, obwohl viele Unternehmen der Sekundärbranche die Entwicklung und Pflege von Software noch als reinen Kostenfaktor betrachten.

Die künftigen neuen Produkte werden ohne hochgradige Qualitätssicherungen, die nur durch *qualitätsorientierte Entwicklungs- und Produktionsprozesse* erreicht werden, nicht erfolgreich sein. Weitere Erfolgsfaktoren sind die Steigerung der Produktivität und Wertschöpfung sowie die Verkürzung von Markteinführungszeiten. In der Praxis werden die diesbezüglichen Potenziale innovativer Softwaretechniken in Deutschland bisher meist unzureichend genutzt. Die *Hauptursachen* für die *Defizite* sind hoher Entwicklungsdruck, Know-how- und Mitarbeitermangel. Dies – und nicht etwa nur zu hohe Arbeitskosten – sind auch Hauptgründe für das derzeit zu beobachtende Offshoring von Software Engineering Aufgaben.

Erschwerend wirkt sich auch der schnelle *Wandel* von hardware- zu softwareorientierten Fragestellungen aus, der viele *Unternehmen überfordert*. Ein Hauptproblem besteht darin, dass in den Sekundärbranchen überwiegend Software-Quereinsteiger zwar mit branchenspezifischer Ausbildung, aber ohne ausreichende softwaretechnische Expertise tätig sind. Das Fehlen akademisch ausgebildeter, branchenorientierter Softwareingenieure führt zur extensiven Auslagerung der Softwareentwicklung zu Fremdfirmen im Inland und verstärkt ins Ausland (Offshoring). Zudem können in Deutschland Unternehmen einige der höchstqualifizierten Software-

Engineering-Arbeitsplätze derzeit nur mit Experten aus anderen Ländern besetzen. Im Widerspruch dazu erfordert die strategische Bedeutung von Software dringend eine Umkehr zur Eigenentwicklung von Software im Kernkompetenzbereich, die auf einer entsprechenden Kompetenzweiterentwicklung in Hochschulen und außeruniversitären Forschungseinrichtungen aufsetzen muß.

Wandel des Gebiets

Die Abhängigkeit unseres gesamten öffentlichen und wirtschaftlichen Lebens von softwaregesteuerten Systemen und der breite Einsatz *komplexer* maschinenbaulicher, verfahrenstechnischer, automatisierungstechnischer oder elektrotechnischer *Erzeugnisse mit integrierter Software*, die zum Teil hochgradig sicherheitskritische Aufgaben übernehmen, hat über die letzten 30 Jahre zu einem radikalen Umdenken in der Konstruktion solch komplexer Systeme geführt. Das Software Engineering hat in Analogie zu anderen Ingenieurdisziplinen Methoden, Techniken, Verfahren und (Software-)Werkzeuge entwickelt, die Software als hochwertiges Industriegut realisieren, dessen Qualität messbar ist und das mit messbaren kosteneffizienten Verfahren entwickelt und produziert wird.

Dies bedeutet auch, dass *spezifische Techniken*, nämlich Methoden, Verfahren und Werkzeuge für *unterschiedliche Anwendungen* bzw. Branchen entwickelt und erprobt werden müssen sowie als Basis für solche Entwicklungen in einem weit höheren Maße als bisher umfangreiche Kooperationen zwischen Forschung und Industrie notwendig werden.

Software Engineering in der Forschung

Forschungsgruppen in Deutschland in Wissenschaft und Industrie

Es gibt etwa 50 *Forschungsgruppen* zum Thema Software Engineering an Universitäten in Deutschland, die nahezu alle als Unterzeichner dieses Manifests vertreten sind. Diese Zahl ist für ein Industrieland der Größe Deutschlands deutlich zu klein, da diese Software Engineering-Gruppen sowohl Hauptverantwortung für die Ausbildung der zukünftigen Softwareingenieure tragen als auch die einschlägigen Softwaretechniken entwickeln sollen. Eine Übersicht über die speziellen *Felder* und *Projekte* dieser Forschungsgruppen findet man unter <http://www-swe.informatik.uni-heidelberg.de/gi-fragebogen/index.php>.

Im Bereich der außeruniversitären Forschungseinrichtungen wird das Software Engineering derzeit vor allem in der IuK-Gruppe der Fraunhofer-Gesellschaft thematisiert. So befasst sich das Fraunhofer IESE schwerpunktmäßig mit experimentellem und empirischem Software Engineering, das Fraunhofer FIRST mit Fragen der Softwarearchitektur eingebetteter Systeme, das Fraunhofer ISST mit der Informationslogistik verteilter Systeme, das Fraunhofer FIT mit dem benutzerzentrierten Software Engineering und das Fraunhofer IITB mit Softwaretechnik im Umfeld bildverarbeitender Systeme. Daneben decken An-Institute wie das Hasso-Plattner-Institut in Potsdam, das FZI in Karlsruhe, die Abteilung Wirtschaftsinformatik des DFKI und das OFFIS in Oldenburg weitere Bereiche der Softwaretechnik ab. Auch die Max-Planck-Gesellschaft beginnt, Grundlagenaspekte des Software Engineering durch die Gründung eines entsprechenden neuen Instituts aufzugreifen.

In der *deutschen Industrieforschung* sind Forschungszentren, die sich mit allgemeinen Fragen des Software Engineering befassen, nur vereinzelt anzutreffen und in den letzten Jahren eher zurückgeführt worden. Entsprechend dem allgemeinen Industrietrend steht hingegen branchenspezifische Anwendungsforschung im Vordergrund. Genannt seien im Bereich der Telekommunikationssoftware die Forschungszentren von Ericsson und Microsoft in Aachen und

der Telekom in Berlin, im Bereich der Automobilsoftware die DaimlerChrysler-Forschungszentren in Berlin und Ulm sowie die BMW Car-IT GmbH in München, im Bereich der medizintechnischen Anwendungen Siemens MED in Erlangen, im Bereich Chip-nahe Software das Intel-Labor in Braunschweig, im Bereich der Heimanwendungen das Philips-Labor in Aachen.

Internationale Positionierung

Deutsche Wissenschaftlerinnen und Wissenschaftler im Bereich Software Engineering zeigen *international* eine hohe *Präsenz*. Insbesondere im Vergleich mit den europäischen Nachbarn lässt sich feststellen, dass in dem traditionell von den USA dominierten Feld neben den Briten gerade Deutsche in den Organisationskomitees und Herausgebergremien aller einschlägigen Konferenzen und Zeitschriften beteiligt sind. Dieses positive Bild zeigt sich auch in vielen Beiträgen zu Konferenzen und Zeitschriften, deren Anzahl allerdings noch steigerungsfähig ist.

Inhaltlich liegen international sichtbare *Schwerpunkte deutscher Forschung* bei formalen Methoden für die Konstruktion, Analyse und Verifikation von Modellen und Programmen, dem Requirements Engineering, dem Verstehen und Formalisieren des Software-Entwicklungsprozesses und der Unterstützung dieses Prozesses durch neue Werkzeuge sowie empirischen Untersuchungen (industriell) eingesetzter Methoden und Verfahren.

Gerade von deutschen Wissenschaftlern werden in letzter Zeit vermehrt anwendungs- oder branchenspezifische Schwerpunkte in Form von Themen wie etwa „Automotive Software Engineering“, „Embedded or Hybrid Systems“ aufgegriffen, d.h. Fragestellungen mechatronischer Systeme, und damit das *Zusammenspiel* zwischen *maschinenbaulichen* und *elektrotechnischen* sowie *informatischen* Erzeugnissen diskutiert.

Forschungsförderung

Die IT- und auch die Software-Engineering-Forschung in Deutschland sind geprägt durch eine gute *Balance* zwischen notwendiger Grundlagenforschung, einer starken anwendungsorientierten Ausprägung und einem beachtlichen Anteil direkter Industrieforschung.

Auf *nationaler Ebene* erweisen sich die Förderinstrumente der Deutschen Forschungsgemeinschaft (DFG) insbesondere für die Universitäten als effektiv. Aber auch die Fördermaßnahmen des Bundesministeriums für Bildung und Forschung (BMBF), die als eine Zielsetzung die anwendungsorientierte Forschung in den Vordergrund stellen und dies durch die notwendige Zusammenarbeit von Universitäten, Forschungseinrichtungen und Industrie erreichen, erweisen sich als positiv im Sinne der Wettbewerbsfähigkeit der IT-Industrie im internationalen Vergleich. Hervorzuheben ist dabei insbesondere der Fokus auf eine Stärkung der einschlägigen mittelständischen Wirtschaft, der dringend weiter geführt und konkret in den Projekten umgesetzt werden muss.

Kritisch ist allerdings anzumerken, dass die spezielle Förderung des Software Engineering durch die *Deutsche Forschungsgemeinschaft* innerhalb des Programms „Aktionsplan Informatik“ ausgelaufen ist und wegen der kurzen Laufzeit des Programms die erwünschten Ziele der Stärkung dieser Ausrichtung durch eine fokussierte Nachwuchsförderung nur ansatzweise erreicht wurden. Die Fördermaßnahmen der DFG zielen allgemein auf die Stärkung der Grundlagenforschung an den Universitäten, was häufig als Förderung theoretischer Aspekte interpretiert wird. Das Software Engineering, insbesondere die Erforschung von Grundlagen

von begründeten Konstruktionsverfahren muss auch in der DFG einen höheren Stellenwert erhalten und ebenso wissenschaftlich anerkannt werden wie die systematische Konstruktion eines mathematischen Beweises.

Anzumerken ist auch, dass die künftige *Förderung* der Softwaretechnik, die im *BMBF* bisher im Rahmen des Programms SE 2006 erfolgte, auf die Fortschreibung dieses Programms oder der Erarbeitung einer neuen Programmatik angewiesen ist. Hier besteht Handlungsbedarf. Ebenso ist kritisch zu vermerken, dass trotz der immer stärker werdenden Bedeutung von hochwertiger Software in technischen Prozessen und Produkten hardwarenahe Technologien einen beständig größeren Anteil der Forschungsmittel binden. Dem Argument, dass dies durch die kontinuierlich fortschreitende Miniaturisierung und dem dadurch entstehenden Innovations- und Wettbewerbsdruck bedingt ist, muss entgegengehalten werden, dass gerade durch die Miniaturisierung der Bedarf an Software in immer neuen Geräten immens steigt und der Qualitäts- und Kostendruck nur durch neue Verfahren zur Steigerung von Qualität und Produktivität abgefangen werden kann.

In der Förderpolitik anderer europäischer Länder ist Software Entwicklung von größerer Bedeutung. So hat etwa in Irland die Softwareforschung bezogen auf Landesgröße und Brutto-sozialprodukt einen weitaus höheren Stellenwert als in Deutschland. Dadurch wurden bereits mehrere herausragende deutsche Forscher abgeworben.

In den Programmen der *Europäischen Union* haben die Themen der ICT (Information and Communication Technology) noch immer einen hohen Stellenwert, wenn auch in den Vorschlägen zur Gestaltung des 7. Rahmenprogramms mit immer stärker werdendem Bezug zu Anwendungsfeldern und mit hohen Anforderungen an Interdisziplinarität. Die enorm zunehmende Größe vieler Projekte und die hohe Zahl von Partnern erschweren nicht nur die angemessene Beteiligung kleinerer universitärer Gruppen und Forschungseinrichtungen, sondern es entsteht auch die Gefahr, dass die Förderschwerpunkte vorwiegend die Interessenlagen allein der Großunternehmen berücksichtigen – zu Lasten der für Innovation und Arbeitsplätze so wichtigen KMU (Kleinen und mittleren Unternehmen) und ihrer Forschungsintensität. Alternativ befasst sich die EU mit der Ausgestaltung von Programmen zur expliziten Förderung neuer Technologien (FET = Future Emerging Technologies) und in diesem Rahmen auch mit der Thematik Software-intensiver Systeme. Auch hier wird ein starker Anwendungsbezug hergestellt und multidisziplinäre Themen werden hervorgehoben.

Im Rahmen von EUREKA ist die *ITEA* (Information Technology for European Advancement) Initiative auch für das Thema Softwaretechnik von hoher Bedeutung. Getragen wird die Initiative von der europäischen Industrie mit starkem IT-Bezug. Die Projekte sind zwar europäisch zusammengesetzt, werden aber nur auf jeweils nationaler Ebene teils im Rahmen spezieller Budgets, teils aber auch im Rahmen entsprechender nationaler Programme gefördert. Interessant bei ITEA ist die Themenfindung, die durch einen intensiven Roadmapping-Prozess gesteuert wird und entsprechend dokumentiert ist. Hier ist die Vorgehensweise auch am ehesten mit der Programmatik des US-amerikanischen PITAC-Reports² vergleichbar.

Der *PITAC*-Report (President's Information Technology Advisory Committee) ist für Forschung im Bereich der Informationstechnik noch immer eines der wichtigsten Dokumente (<http://www.nitrd.gov/pitac>). Beschrieben und auch ständig erneuert werden die Themenstellungen, die für die Weiterentwicklung der amerikanischen IT-Industrie als wichtig angesehen

² vgl. Detlef Schmid, Manfred Broy: . . . noch nicht zu spät! Das Walberberg-Memorandum zur Förderung der IT-Forschung. Informatik Spektrum 23(2): 109-117 (2000)

werden. Kontinuierlich sind in diesen Themen softwaretechnische Aufgaben und Problemstellungen enthalten (wie High Assurance Software and Systems Design). Der Bericht vom Februar 2005 widmet sich dem Thema „Cyber Security“ und analysiert u.a. die Verwundbarkeit unserer Systeme durch den vermehrten Einsatz von Software - eine sicher richtige und extrem wichtige Fragestellung.

Das „U.S. Center for National Software Studies“ veröffentlichte im April 2005 einen Bericht unter dem Titel „Software 2015: A National Software Strategy to ensure U.S. Security and Competitiveness“ (<http://www.cnsoftware.org/nsg/>). Als wichtigstes Thema für die zukünftige Forschung im Bereich Software wird die Verbesserung der Vertrauenswürdigkeit von Software identifiziert (improving software trustworthiness). Es ist somit zu erwarten, dass die USA zukünftig stark in diesen Bereich investieren werden. Europa und Deutschland sollten hier nicht den Anschluss verlieren.

Lehre

Software Engineering spielt in der *Aus- und Weiterbildung* an deutschen *Universitäten* eine wichtige und beständig wachsende Rolle. Mit dem Übergang auf die gestuften Bachelor- und Masterstudiengänge verstärkt sich diese Bedeutung.

Studium

Die *Grundlagen der Softwaretechnik* werden zurzeit meist in eigenständigen Vorlesungen präsentiert. Es gibt englisch- und deutschsprachige Lehrbücher, die eine weite Verbreitung erlangt haben und so zu einer Angleichung der Lehrinhalte führen. Außerdem lässt sich eine wachsende Tendenz zum projektorientierten Studium feststellen, bei dem jeder Studierende die Notwendigkeit und Vorzüge softwaretechnischer Grundlagen unmittelbar erlebt.

Durch die Umsetzung des Bolognaprozesses wird Software Engineering zu einem Pflichtfach in allen Informatik-Bachelorstudiengängen. Sowohl die fachspezifischen Empfehlungen der gängigen Akkreditierungsagenturen als auch die Empfehlungen der Gesellschaft für Informatik und des Fakultätentags Informatik legen *Softwaretechnik-Veranstaltungen verbindlich* fest. Die meisten Masterstudiengänge enthalten Angebote zur Verbreiterung oder zur Vertiefung innerhalb der Softwaretechnik.

Für die Betonung der Arbeitsmarktfähigkeit der Bachelor-Absolventen wird im Bolognaprozess ein obligatorisches *Praxisprojekt* gefordert, das den Studierenden praktische Erfahrungen im Einsatz von softwaretechnischen Prinzipien, Methoden und Werkzeugen vermittelt. Teamfähigkeit und Projektmanagement verstärken diesen Praxisbezug. Masterstudiengänge sehen Praktika vor, die inhaltlich und methodisch noch tiefer gehen. Zudem wird der *Umgang* mit ausgereiften *Werkzeugen* (Entwicklungsumgebungen, Dokumentations- und Versionsverwaltungssysteme) in der Informatikausbildung gestärkt.

Die Erfahrungen mit eigenen *Studiengängen* mit dem Schwerpunkt *Softwaretechnik* sind sehr gut. Versuche, wie etwa in Stuttgart, wurden nach erfolgreicher Evaluation inzwischen fest etabliert. In Akkreditierungsverfahren sind daher zunehmend Studiengänge zu finden, die ihren Schwerpunkt in der Softwaretechnik oder in deren Nähe sehen (wie z.B. Systems Engineering).

Neben Praxisnähe ist *Grundlagen- und Methodikorientierung* der *Kern* jeder universitären Softwaretechnik-Ausbildung. Zielsetzung ist dabei weniger die Erzeugung wissenschaftlichen

Nachwuchses. Stattdessen sollen bei Absolventen Selbstständigkeit, Problemlösungskompetenz, Denken in Konzepten sowie die Fähigkeit erzielt werden, sich in neue und schwierige Sachverhalte einzuarbeiten.

Diese Fähigkeiten werden für alle *anspruchsvollen* und komplexen *Entwicklungsaufgaben* in Industrie und Forschung benötigt. Insbesondere auf Masterniveau ist eine solche grundlagenorientierte Ausbildung für den Standort Deutschland nötig, damit Persönlichkeiten entstehen, die den Entwurf komplexer Systeme maßgeblich prägen bzw. Führungskompetenz für Projekte übernehmen können.

Verstärkte Nachfrage

Es gibt einen geschätzten Bedarf von 385.000 Softwareentwicklern in Deutschland³. Auch bei vorsichtiger Bewertung dieser Zahl ergibt sich daraus, dass ein weiterer *Ausbau* der *Software-technik* an deutschen Universitäten dringend erforderlich ist, denn nach dem aktuellen Stand der Forschung und Lehre wären das derzeit 7300 benötigte Softwareentwickler pro Forschungsgruppe. Vereinzelt begründen einschlägige Firmen bereits die Verlagerung ins Ausland mit dem Mangel an Softwareingenieuren.

Da auf dem Arbeitsmarkt Personen sehr unterschiedlicher Ausbildung tätig sind, ist die *Qualität und Kompetenz* des *Personals* oft verbesserungsfähig und verbesserungsbedürftig. Der hohe Anteil der Quereinsteiger, aber auch der bereits erwähnte Import hoch qualifizierter Softwareingenieure aus dem Ausland, ist eine Konsequenz nicht verfügbarer Absolventen mit erster Qualifikation..

Es gibt weiterhin eine große *Nachfrage* nach hoch qualifizierten *Softwareentwicklern*, insbesondere bei der Entwicklung kritischer Produkte. Hier werden erst seit etwa 10 Jahren verstärkt Informatiker herangezogen. In der Sekundärbranche werden in stärkerem Maße Abgänger mit interdisziplinären Vorbildungen gesucht. In diesen Firmen gibt es häufig erfahrene Ingenieure, die aufgrund der zunehmenden Softwareintensität der Produkte einen dringenden Bedarf an zusätzlichem Software-Engineering-Know-How haben. Die Hochschulen müssen hier gezielt neue Angebote entwickeln.

Durch den schnellen technologischen Wandel, gerade im Bereich des Software Engineering, besteht ein wachsender Bedarf an kontinuierlicher *Weiterbildung* und auch eine steigende Nachfrage nach der Zertifizierung von Software-Engineering-Kompetenzen.

Strategische Bedeutung

Die wachsende Bedeutung des Software Engineering wurde bereits betont. Wir beschreiben hier die vorhersehbare *künftige Situation* durch Extrapolation der Gegenwart.

Neue Funktionen

Moderne Produkte enthalten als wesentliche Komponente den Werkstoff Software in unterschiedlichen Funktionen und Rollen. Einerseits erlauben Software-basierte Lösungen neue

³ Zahlen aus: M. Broy, D. Rombach: Software Engineering. Wurzeln, Stand und Perspektiven. Informatik Spektrum 25:6 Dezember 2002, 438-451. Neuere umfassende Zahlen liegen nicht vor.

Funktionen in einzelnen Komponenten. Andererseits wird die überragende Bedeutung von Software aber gerade durch ihre integrierende Funktion deutlich: Software basierte Lösungen ermöglichen *neue Kombinationen* und *Interaktionen* von Komponenten, die bisher nicht gekannte Funktionen und Qualitäten realisieren.

Als *Beispiel* sei die so genannte *Context-Awareness-Qualität* genannt, die Funktionen verschiedener Komponenten so zusammenbringt, dass diese Umgebungsinformation zu erkennen und kooperativ und intelligent verarbeiten. Damit können neuartige Produkte und Services etwa im Automobil- und Flugzeugbau aber auch bei elektronischen Dienstleistungen (z.B. eHealth) in Wirtschaft und Verwaltung realisiert werden.

Neben der Konstruktion einzelner Komponenten aus Hard- und Softwarebestandteilen muss die *Integration* und *Adaption* dieser *Komponenten* beherrscht und verstanden werden. Während die Konstruktion einzelner Komponenten – beispielsweise eines Antiblockiersystems im Kraftfahrzeug – als gelöst gelten kann, stellt die Integration verschiedener solcher einzelner Komponenten, beispielsweise in ein Assistenz- und Managementsystem für energieeffizientes Fahren zusammen mit vielen anderen Komponenten weiterer Zulieferer eine *äußerst komplexe Aufgabe* dar. Diese Komplexität ist nicht nur auf Grund der gegebenen Problemstellung hoch, da viele verschiedene Funktionen beteiligt sind, sondern vor allem auch durch die *Schwierigkeiten der Kommunikation zwischen den beteiligten Disziplinen*.

Neue Qualität und Quantität

Insbesondere im Bereich eingebetteter Systeme und *eingebetteter Software* gibt es einen extrem wachsenden Forschungsbedarf, der durch den beschleunigt wachsenden Software-Anteil in eingebetteten Systemen verursacht wird. Zum Beispiel betrug in einem PKW gehobener Ausstattung im Jahr 2003 die Größe eingebetteter Software 70 MB, bis zum Jahr 2010 rechnet man mit 1 GB.

Diese Entwicklung kann nur dann erfolgreich beherrscht werden, wenn spezifische Methoden und Werkzeuge - allgemein *Techniken* - entwickelt und bereitgestellt werden, mit denen *eingebettete Anwendungen* korrekt, zuverlässig und effizient erstellt werden können. Dabei ist es wichtig, eng mit Ingenieuren und Wissenschaftlern der benachbarten Disziplinen zusammen zu arbeiten, um zu einem gemeinsamen einheitlichen Systemdenken zu gelangen.

Als Beispiel dafür kann man sich ein *elektronisches Auge* vorstellen. Wie im natürlichen Auge ist es wichtig, dass die „Hardware“ funktioniert, dass Sensoren das einfallende Licht messen und Aktoren darauf reagieren können, indem z.B. durch Vergrößern und Verkleinern der Pupille die Menge einfallenden Lichts reguliert wird. Darüber hinaus ist es genauso wichtig, dass die erhaltenen Informationen vom Gehirn bzw. einer geeigneten Software adäquat verarbeitet werden, weil nur dadurch das Sehen erst realisiert wird. So wie beim Menschen im Kindesalter sich die entsprechenden Gehirnnetzungen bilden müssen, damit Sehen im Gehirn möglich wird, müssen in eingebetteten Systemen Softwarefunktionen zur Verfügung gestellt werden, die die Funktionalität des Gesamtsystems zuverlässig und effizient steuern.

Software Engineering bei komplexen Aufgaben

Die Rolle und Bedeutung, des Software Engineering ist im Hinblick auf *drei* grundsätzlich verschiedene Problemebenen technischer Aufgaben zu differenzieren:

1. Grundlegende Innovationen in nahezu allen Bereichen des täglichen Lebens z.B. in Automobilbau, Bioinformatik, Luftfahrt: Hier liegt die Rolle darin, noch nie da gewesene Lösungen *überhaupt* erst zu *ermöglichen*.
2. Hochriskante Großprojekte z.B. in Unternehmens- und Verwaltungsautomatisierung: Die Rolle des Software Engineering besteht vor allem in der *Reduktion* des *Risikos*, dass ein solches Projekt scheitert.
3. „Normale“ Aufgaben wie der ganz alltägliche Bau von Software (vor allem in KMU): Die Rolle ist hier eine pädagogische. Längst verstandene Verfahrensweisen müssen in der vollen Breite der Wirtschaft so ausgerollt werden, dass gelungene Softwarekonstruktion eine *routinemäßige Kompetenz* allerorten wird.

Forderungen und Maßnahmen

Verstärkung der Kompetenz

Die obige Bestandserhebung hat gezeigt, dass trotz der inzwischen erzielten Erfolge die Software-Engineering-Kompetenz erheblich verstärkt werden muss. Insbesondere müssen folgende *Kompetenzfelder* gestärkt werden:

- Softwareingenieure in der Industrie: Sie müssen weitergebildet werden bzw. gezielt auf die komplexer werdenden Anforderungen hin ausgebildet werden.
- Management in der Industrie: Die Möglichkeiten der Gestaltung des Software Engineering zur Erzielung besserer Produkte, der Risikobeherrschung bei Großprojekten und der Eroberung neuer Geschäftsfelder müssen erkannt und genutzt werden.
- Unternehmensebene in Wirtschaft und Verwaltung: Der Beitrag des Software-Personals zum Unternehmenserfolg muss erkannt und gewürdigt werden.
- Forschung: Die deutsche Forschungslandschaft muss für die Besten weltweit attraktiv werden. Darüber hinaus müssen die Forschungsfragestellungen aus den oben beschriebenen wichtigen Anwendungen transdisziplinär aufgegriffen werden.
- Lehre: Die universitäre Lehre muss international wettbewerbs- und kooperationsfähig sein. Trotz Grundlagenorientierung muss sie auf Anwendung und Umsetzung vorbereiten.

Aus diesen Forderungen leiten sich die folgenden *organisatorischen Maßnahmen* ab.

Forschungsthemen und Förderprogramme

Zukunftsorientierte Forschungsthemen

In Deutschland muss insbesondere darauf geachtet werden, dass der Technologievorsprung im Bereich *eingebetteter softwarebestimmter Systeme* gehalten und ausgebaut werden kann. Gleichzeitig sollte der Rückstand bei der Systemsoftware verringert werden. Um Software von zunehmender Größe und Komplexität in kürzeren Zyklen mit höherer Qualität und für neue Ansprüche zu entwickeln, müssen die dazu verwendeten Prozesse zur Entwicklung und Wartung weiter verbessert und neue Prozesse entwickelt werden.

Damit Erfolg versprechende Forschungsprojekte realisiert werden können, muss eine *Zusammenarbeit* mit *Ingenieuren* erfolgen. Disziplinen wie etwa der Maschinenbau unterscheiden

sich allerdings in ihrem Wissenschaftsverständnis merklich von der Informatik. Während in der Informatik allgemein das Bestreben nach verallgemeinerbaren Methoden vorherrscht, besteht im Maschinenbau ein vorrangiges Ziel darin, funktionierende Systeme zu bauen. Disziplinübergreifende Forschungsprojekte müssen sowohl den methodischen Erkenntnisfortschritt als auch eine empirische Bestätigung beinhalten. Sie sind in Zukunft nur mit deutlich höheren Forschungsvolumina zu realisieren.

Neben diesen Fragestellungen, die sich im Umfeld eingebetteter Software ergeben, existieren *weitere Themenstellungen*, deren Erforschung unabdingbar ist, um die Zukunft zu gewinnen:

Prozessfragestellungen

- Frühe, umfassende Integration in den Systementwicklungsprozess (Systems Engineering) bei der Entwicklung von Software in Produkten,
- räumlich und organisatorisch verteilte, interdisziplinäre Entwicklungsprozesse,
- modellbasierte Software- und Systementwicklung (Modellierung, Simulation, Verifikation, Codegenerierung).

Qualitätsaspekte

- Erhöhung des Reifegrads von Softwareentwicklungsprozessen, einerseits durch organisatorische Maßnahmen (z.B. ISO 15504, CMMI), andererseits durch Einsatz neuer, spezifischer Techniken,
- Verbesserung der Planbarkeit und Vorhersagbarkeit von Softwareentwicklungsprozessen (z.B.: empirische Methodik, kontrollierte Experimente, industrielle Fallstudien, Methodenprototyping),
- Qualitätssicherung für komplexe Softwaresysteme (Korrektheit, Performanz, Verfügbarkeit, Sicherheit und Zuverlässigkeit).

Technische Aspekte

- Wartungs- und Weiterentwicklungsprozesse für Legacy-Systeme (z.B. Reverse Engineering, Reengineering und Migration auf zukunftsfähige Plattformen),
- Wiederverwendung im Entwicklungsprozess (Produktlinien, Varianten, domänenspezifische Architekturen),
- Requirements Engineering zur Ermittlung, Dokumentation, Validierung und Verfolgung komplexer Anforderungen,
- Usability Engineering und Entwicklung intuitiver und leistungsfähiger Mensch-Maschine-Schnittstellen,
- Betriebs- und Monitoringmodelle von (insbesondere verteilten) Softwaresystemen, Informationslogistik,

- Endbenutzer-Beherrschbarkeit Software-intensiver Produkte einschließlich der Möglichkeiten selbstbestimmter Ausgestaltung (z.B. Eigenprogrammierbarkeit von Mobiltelefonen und Autos),
- Werkzeugunterstützung zur Verifikation und Validierung (Theorembeweiser, Modellprüfer, Programmprüfer, Testmanagement und Testfallprüfung),
- Funktionsübergreifende, vernetzte Software in Anwendungssystemen (Ambient Intelligence, Automobil, Flugzeug, Gebäudeautomatisierung, Fabrikautomatisierung),
- neue intelligente Werkzeuge zur Unterstützung verbesserter Prozesse und zur Erzielung höherer Qualität,
- Software Engineering im Grid-Computing (Infrastruktur vs. Applikationen, Sicherheit),
- Aspektorientierte Softwareentwicklungsunterstützung als Integrationskonzept für übergreifende Belange und als natürliche Ergänzung zur Objektorientierung.

Erforderliche Forschungsschwerpunkte und Förderungen

Die *interdisziplinäre Zusammenarbeit* mit Ingenieuren und Forschern beim Schwerpunkt eingebettete Systeme ist zu fördern. Dies betrifft sowohl BMBF-Programme als auch die DFG. Die *BMBF-Förderprogramme* sollten großvolumiger als die bisherigen Software Engineering Initiativen ausfallen, da ähnlich wie in den rein ingenieurwissenschaftlichen Programmen auch Demonstrationsprototypen gebaut werden müssen. Dies ist arbeits- und kostenintensiver als reine Softwareentwicklung, da auch z.B. Hardware-Kosten anfallen. In der *DFG* sollte gezielt auf Grundlagenebene interdisziplinäre Forschung zwischen Ingenieuren und Informatikern gefördert werden. Oft fehlt derzeit das wechselseitige Bewusstsein für die Notwendigkeit der Zusammenarbeit: Informatiker können zur Softwareentwicklung in ingenieurwissenschaftlichen Anwendungen nicht nur beitragen, sondern darüber hinaus durch die Zusammenarbeit mit Ingenieuren neue Fragestellungen und Anforderungen an die Softwaretechnik erkennen und lösen.

Ein neuer Forschungsschwerpunkt sollte im *experimentellen Software Engineering* liegen, um die Effektivität der Softwareentwicklungsprozesse zu studieren. Hier ist insbesondere eine Zusammenarbeit zwischen Forschungseinrichtungen und der Industrie erforderlich. Deshalb ist hier eine Förderung durch das BMBF und die EU notwendig.

Aber auch die oben einzeln angesprochenen *längerfristigen Einzelfragestellungen*, die zu einem neuen Bewusstsein, zu qualitativ besseren Entwicklungsprozessen, zu spezifischen Techniken und zum gezielten Einsatz von Wiederverwendung führen, müssen von BMBF und DFG gefördert werden. Diese Förderung richtet sich in erster Linie an die Softwaretechnik-Forschungsgruppen selbst. Diese Gruppen erarbeiten erst die Inhalte und bereiten sie auf, so dass sie dann in Folge in interdisziplinärer Zusammenarbeit mit Ingenieuren genutzt werden können.

Lehre und Forschung

Die Bewältigung der obigen Herausforderungen erfordert auch eine *veränderte Schwerpunktsetzung* in der *Lehre*. Diese zielt sowohl auf die Entwicklung neuartiger Lehrangebote als auch darauf, bereits vorhandene Ansätze gezielt auszubauen.

Stärkung der Forschung durch veränderte Lehre

Um Softwaretechnik als *wissenschaftliche Disziplin* zu etablieren, die einen eigenständigen Kern an wissenschaftlichen Grundlagen vermittelt, ist es notwendig, dass sich längerfristig eine allgemein akzeptierte „Konstruktionslehre“ für Software etabliert. Neben der Vermittlung praxisrelevanter Inhalte muss die Lehre gezielter auf *Forschungsinhalte vorbereiten*. Beides erfordert das Verfassen von Rahmen-Curricula, die festlegen, welche der von nationalen und internationalen berufständigen Organisationen (Fakultätentag, GI, ACM, IEEE CS) empfohlenen Inhalte abgedeckt werden sollen, und die den internationalen Studenten- und Graduiertenaustausch vereinfachen.

Neben den klassischen Informatik-Studiengängen sind *spezifische Softwaretechnik-Studiengänge* in drei verschiedenen Stufen sinnvoll: Zum einen sollten konsekutive Bachelor/Master-Studiengänge in Software Engineering etabliert werden. Als Alternative sind auch forschungsorientierte Master-Studiengänge des Software Engineering sowie Systems Engineering denkbar. Schließlich sollte es auch Angebote einer disziplinübergreifenden Ausbildung vor allen Dingen für den Master geben, in denen neben Softwaretechnik als Schwerpunkt auch entsprechendes Kontext- bzw. Anwendungswissen (z.B. für eingebettete Systeme) vermittelt wird.

Grundsätzlich sollte der Master der Regelabschluss sein, da für die steigenden Anforderungen in der Praxis bei der Konstruktion komplexer Softwaresysteme eine Fülle von Fähigkeiten erforderlich ist, die allein in einer Bachelor-Ausbildung nicht zu bewältigen ist.

Eine zielgerichtet auf die Forschung hinführende *Lehre* erfordert verstärkte *Zusammenarbeit*. Vorstellbar ist der Austausch von speziellen, vertiefenden Lehrveranstaltungen, standortübergreifende Graduate Schools zu einschlägigen Softwaretechnik-Themen bis hin zu internationalen „Doktoranden-Netzwerken“.

Förderung einer praxisrelevanten Lehre

Um dem Anspruch der Praxisrelevanz gerecht zu werden, müssen *berufsfeldrelevante Inhalte* noch stärker in die Curricula einbezogen werden. Insbesondere sollten die Studierenden die in der industriellen Praxis eingesetzten Methoden, Werkzeuge und Techniken kennen und auch bewerten können. Projektarbeit und „Soft Skills“ sind zu verstärken. Eine stärkere Praxisorientierung erfordert die prinzipielle *Bereitschaft der Industrie*, sich an fall- und projektbezogenen Ausbildungsangeboten über Lehraufträge zu beteiligen. Auch das Angebot an Ringvorlesungen zu umfassenden Themen ist zu erweitern.

Um die Qualität der Ausbildung zu steigern, sollten *neue Organisationsformen* in der Lehre erprobt werden. Regionale „Lehrverbände“ ermöglichen eine ökonomischere Nutzung von Spezialvorlesungen. Durch den verstärkten Einsatz neuer Medien werden Originalbeiträge externer Spezialisten in konventionelle Lehrveranstaltungen vor Ort besser integriert. Ebenfalls denkbar sind Zusatzlehrangebote in der vorlesungsfreien Zeit.

Bei der Definition neuer Curricula muss die zentrale Frage gestellt werden, welche *Qualifikationen* der Absolvent haben sollte. Dies bedingt in der Regel auch Grundlagen aus einschlägigen Anwendungsbereichen. Informatik- und Softwaretechnik-Inhalte sind darauf abzustimmen.

Zukünftig müssen weiterqualifizierende Master-Programme für Absolventen anderer Fachrichtungen (*Retraining Master*) wie auch für bereits in der Industrie Tätige (*Professional Master*) definiert werden. Vertiefungen in zukunftssträchtigen Anwendungsbereichen (wie Automotive, Telekommunikation, Automatisierungstechnik, Gebäudetechnik) müssen Standard werden.

Industrie

Beim Software Engineering stellen wir einen gewaltigen Abstand zwischen Erfolg versprechenden Forschungsergebnissen und industriell eingesetzten Methoden fest. Dieser Abstand hat im Wesentlichen zwei Ursachen:

- Die für den Technologietransfer von Methoden und Prozessen notwendige enge Kooperation zwischen Forschung und Industrie in Deutschland nicht sehr ausgeprägt
- Für viele Methoden gibt es keine oder nur unzureichend belegbare Aussagen über deren Nutzen und Risiken, so dass die Industrie vor dem Risiko des Einsatzes zurückscheut

Für Innovation kommt dem Technologietransfer, der Umsetzung von angewandten Forschungsergebnissen in die industrielle Praxis, entscheidende Bedeutung zu. Insbesondere die sich gegenwärtig beschleunigenden Innovationszyklen in der Industrie erfordern einen unmittelbaren Technologietransfer. Dies erfordert verstärkte Förderung der angewandten Forschung und eine engere Kooperation zwischen Forschung und Industrie.

Verbesserung der Zusammenarbeit mit der Industrie

Die Unterzeichner dieses Manifests fordern die Industrie in Deutschland zur verstärkten Nutzung der universitären und außeruniversitären Forschungskompetenzen auf. Erfolgreiche Beispiele innovativer und erfolgreicher Kooperationen existieren zwischen der Industrie und Fraunhofer-Instituten, aber auch größeren universitären Forschungsgruppen. Allen diesen erfolgreichen Beispielen ist gemeinsam

- eine enge Zusammenarbeitsform (z.B.: Wissenschaftler und Industriemitarbeiter arbeiten in gemeinsamen Forschungslaboren, oder Mitarbeiter werden auf Zeit ausgetauscht),
- die Bereitschaft auf beiden Seiten zur partnerschaftlichen Zusammenarbeit (Wissenschaftler gehen auf praktische Probleme der Industrie ein, Industrie nutzt Wissenschaftler für Innovationen anstatt als verlängerte Werkbank).

Basierend auf Erfahrungen in anderen Ländern (z.B. USA) stellen gemeinsame Forschungslabore oder wechselseitiges Mitarbeiten (von Wissenschaftlern in der Industrie und Industriemitarbeitern an Hochschulen) durchaus Erfolg versprechende Modelle zur Transferbeschleunigung dar.

Transfer

Ein wesentlicher Aspekt des Technologietransfers ist der Nachweis des Nutzens einer neuen Methode in einem industriellen Kontext. Dies erfordert den (üblicherweise empirischen) Nachweis, dass eine neue Methode zu praxisrelevanten Projektzielen (etwa Reduktion der Entwicklungszeit) unter gegebenen realen Randbedingungen (etwa Erfahrung der Mitarbeiter, Entwicklungstechnologie, Art des Systems) beiträgt. In anderen Ingenieursdisziplinen gibt es Handbücher mit belegten Aussagen über die Eignung unterschiedlicher Methoden unter be-

stimmten Randbedingungen. Derartige in Handbüchern dokumentierten Erfahrungen fehlen im Software Engineering fast vollständig. Wenn aber solche Aussagen fehlen, werden neue Methoden unter Risikogesichtspunkten erst gar nicht erprobt.

Deshalb müssen – ähnlich wie in anderen Disziplinen – industrienaher Erprobungs- und Testmöglichkeiten für neue Methoden geschaffen werden. Erfahrungen aus der Vergangenheit legen nahe, dass dies aufgrund mangelnder Realitätsbedingungen weder von der Forschung, noch – bedingt durch Ressourcenmangel und Prioritäten für Entwicklungsaktivitäten – alleine von der Industrie geleistet werden kann. Gemeinsame Labore zwischen Forschung und Industrie sind ein Erfolg versprechender Ansatz. Erste Umsetzungen an Standorten wie Kaiserslautern, Berlin oder Potsdam haben zu nachhaltigen Transfererfolgen geführt. Derartige Modelle müssen weiter verbessert und verbreitet umgesetzt werden. Solche Laborangebote sind auch geeignet, die deutsche Industrie bei der Entwicklung zukünftiger Innovationen im eigenen Land zu unterstützen.

Organisatorische Maßnahmen

Auch staatliche Fördermaßnahmen sollten das Ziel erfolgreichen Technologietransfers unterstützen. Geeignete Maßnahmen sind

- Öffentliche Förderung deutscher Firmen nur bei Einbeziehung von universitären und/oder außeruniversitären Forschungseinrichtungen,
- Abstimmung staatlicher Fördermaßnahmen im Bereich der Grundlagenförderung (z.B.: DFG) und angewandter Forschung (z.B.: BMBF). Innovation wird nur beschleunigt, wenn parallel Grundlagenforschung und angewandte Forschung dieselben Themen adressieren und sich ergänzen.

Redaktion: Manfred Broy, TU München;
Matthias Jarke, RWTH Aachen und Fraunhofer FIT;
Manfred Nagl, RWTH Aachen;
Dieter Rombach, TU Kaiserslautern und Fraunhofer IESE

Mit-Autoren/ Unterzeichner:

Armin B. Cremers	Universität Bonn
Jürgen Ebert	Universität Koblenz-Landau
Sabine Glesner	TU Berlin
Martin Glinz	Universität Zürich
Michael Goedicke	Universität Duisburg-Essen
Gerhard Goos	Universität Karlsruhe
Volker Gruhn	Universität Leipzig
Wilhelm Hasselbring	Universität Oldenburg
Stefan Jähnichen	TU Berlin und Fraunhofer FIRST
Stefan Kowalewski	RWTH Aachen
Bernd Krämer	FernUniversität in Hagen
Stefan Leue	Universität Konstanz
Claus Lewerentz	BTU Cottbus
Peter Liggesmeyer	TU Kaiserslautern und Fraunhofer IESE
Christoph Lüth	DFKI und Universität Bremen

Barbara Paech	Universität Heidelberg
Helmut A. Partsch	Universität Ulm
Ilka Philippow	TU Ilmenau
Lutz Prechelt	FU Berlin
Andreas Rausch	TU Kaiserslautern
Willem-Paul de Roever	Christian-Albrechts-Universität zu Kiel
Bernhard Rumpe	TU Braunschweig
Gudula Rünger	TU Chemnitz
Wilhelm Schäfer	Universität Paderborn
Kurt Schneider	Universität Hannover
Andy Schürr	TU Darmstadt
Walter F. Tichy	Universität Karlsruhe
Bernhard Westfechtel	Universität Bayreuth
Wolf Zimmermann	Universität Halle-Wittenberg
Albert Zündorf	Universität Kassel