

Dagstuhl-Manifest zur strategischen Bedeutung des Software Engineering in Deutschland

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:

Die Teilnehmer des Dagstuhl Workshop 05402 "Challenges for Software Engineering Research"

Armin B. Cremers, Jürgen Ebert, Sabine Glesner, Martin Glinz, Michael Goedicke, Gerhard Goos, Volker Gruhn, Wilhelm Hasselbring, Stefan Jähnichen, Stefan Kowalewski, Bernd Krämer, Stefan Leue, Claus Lewerentz, Peter Liggesmeyer, Christoph Lüth, Barbara Paech, Helmut A. Partsch, Ilka Philippow, Lutz Prechelt, Andreas Rausch, Willem-Paul de Roever, Bernhard Rumpe, Gudula Rünger, Wilhelm Schäfer, Kurt Schneider, Andy Schürr, Walter F. Tichy, Bernhard Westfechtel, Wolf Zimmermann, Albert Zündorf.

Zur Bedeutung von Software

Software hat sich zu einem zentralen *Werkstoff* des *Informationszeitalters* entwickelt. Innovative Produkte und Dienstleistungen sind ohne zuverlässige Software nicht mehr denkbar. Dementsprechend hängt die Wettbewerbsfähigkeit der deutschen Wirtschaft entscheidend von der Fähigkeit ab, Software-intensive Produkte und Dienstleistungen mit höchster Qualität schnell und kostengünstig zu erstellen. Software Engineering (abgekürzt SE, deutsch Softwaretechnik) auf Weltniveau ist die Voraussetzung dafür, dass Deutschland seine führende Stellung im Ingenieurbereich (z.B. Export-Weltmeister im Maschinenbau) halten und ausbauen bzw. eine entsprechende Position in neuen Sparten (z.B. e-Health) aufbauen kann.

Die Bedeutung von Software Engineering hat sich verstärkt und wird weiter wachsen. Während Software in der Vergangenheit eine weitgehend isolierte Komponente großer komplexer Systeme war, so wird sie in der Zukunft *integrierter* – in vielen Fällen sogar dominierender – *Teil großer komplexer Systeme* sein. In der Automobilindustrie 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* in den drei Bereichen Forschung, Lehre und Technologietransfer. Dieses Manifest stellt Forderungen inhaltlicher und organisatorischer Art für Forschung, Lehre und Technologietransfer auf, deren Erfüllung der wirtschaftlichen Bedeutung von Software und Software Engineering Rechnung tragen soll.

Es ist die auf sorgfältige Untersuchungen abgestützte Überzeugung der Unterzeichner, dass verstärkte Anstrengungen in allen drei Bereichen notwendig sind, um den Herausforderungen



des Informationszeitalters und der Globalisierung in Deutschland gerecht zu werden. *Arbeitsplätze der Zukunft* können bevorzugt mit Software-intensiven Produkten und Dienstleistungen geschaffen werden. Der Standort Deutschland wird sich in diesem Markt nur behaupten, wenn Software Engineering (d.h. die Produktionsprozesse für den Werkstoff Software) sowie Softwareingenieure (d.h. die Arbeitskräfte zur Konstruktion von Software) auf höchstem Qualitäts- und Produktivitätsniveau verfügbar sind.

Teil A: Bestandsaufnahmen

In diesem Teil wird eine umfassende *Bestandsaufnahme* des Gebiets Software Engineering vorgenommen. Aus der Diskrepanz zwischen der stark gewachsenen wirtschaftlichen Bedeutung und dem Stand der Praxis in der Industrie, dem Stand von Forschung und Lehre sowie dem Fördervolumen wird im Teil B der notwendige Handlungsbedarf abgeleitet.

1. Software Engineering – Begriffsbestimmung und Entwicklung

Definition und Abgrenzung

Software Engineering (SE) (oder deutsch: Softwaretechnik) ist die *Methodenlehre* der Softwarekonstruktion. Ihre Hauptbeiträge sind methodische Hilfsmittel wie Prozesse, Modelle, Werkzeuge und Prinzipien zur Konstruktion und Beherrschung des Einsatzes hochwertiger Software zu akzeptablen Kosten. „Hochwertig“ bedeutet hier: Zuverlässig, sicher, leicht veränderbar, einfach benutzbar und vor allem 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*.

Die *Technik* für die Konstruktion von Software ist vielgestaltig. Hierzu zählen Konzepte, Sprachen, Methoden, Werkzeuge und deren Grundlagen, Dienstkomponenten, Plattformen, Rahmenwerke aus wieder verwendbaren Bausteinen. Die Bereitstellung dieser Technik definiert das Software Engineering in engerem Sinne.

Die Aufgabe der Forschung und Lehre des Software Engineering besteht darin, den Stand der engeren Disziplin voranzutreiben und zu einem breiten Kenntnisstand der Softwareingenieure beizutragen.

2. Zum Stand des Software Engineering in der industriellen Praxis

Entwicklung

Das Software Engineering hat in den letzten Jahren eine deutliche *Entwicklung* nach vorne genommen: Die Anzahl der Softwareingenieure in der Industrie ist gestiegen, Software Engineering-Themen spielen in allen Informatikstudiengängen eine große Rolle, die Anzahl der Forschungsgruppen auf diesem Gebiet ist gewachsen, erfolgreicher Technologietransfer ist keine Seltenheit mehr. Dennoch sind noch immer große *Defizite* festzustellen, insbesondere bei Berücksichtigung der Bedeutung des Gebiets, wie die nachfolgenden Erläuterungen zeigen.

Wachsende Bedeutung insbesondere im Sekundärbereich

Software ist *wettbewerbsentscheidender* Faktor geworden, nicht nur in der Primärbranche, die durch die Entwicklung eigenständiger Softwareprodukte gekennzeichnet ist. Dies gilt weitaus stärker in allen so genannten Sekundärbranchen (eingebettete Software in Produkten und Dienstleistungen, z.B. Maschinen- und Fahrzeugbau, Elektrotechnik, Telekommunikation, aber auch Unterhaltungsbranche, Medizin u.a.), in denen 80 % der Softwareingenieure arbeiten [Evasoft2000]. Für die Wertschöpfung im Produktionsgüter- und Dienstleistungsbereich ist Software Engineering entscheidend als „Produktionstechnik des 21. Jahrhunderts“.

Unternehmensbefragungen in Deutschland zeigen eine durch Software erzielte Wertschöpfung im Jahr 2000 von mehr als 28 Mrd. Euro mit steigender Tendenz [Evasoft2000]. Gerade in den *ingenieurorientierten Sekundärbranchen* liegt die traditionelle Stärke der deutschen Industrie, etwa im Fahrzeugbau oder Maschinen- und Anlagenbau. Die Innovations- und Weltmarktführerschaft wird immer öfter durch ingenieurmäßige, softwareintensive Lösungen geprägt. Der Umsatz ist abhängig von der Software, die das Alleinstellungsmerkmal innovativer Produkte bestimmt. Das Spektrum möglicher neuer Produkte ist bereits enorm und weitet sich ständig aus. Um allerdings solche strategischen Markterfolge zu erzielen (die Internet-Telefonie-Firma Skype oder der deutsche Weltmarktführer im Digital Rights Management, die Firma Coremedia, seien hier als zwei aktuelle Beispiele genannt), bedarf es einer engen Verzahnung von Software Engineering und Unternehmensstrategie, die nichts mehr zu tun hat mit den überzogenen Spekulationen der Internet-Modewelle zur Jahrtausendwende.

Im Widerspruch dazu wird in vielen Unternehmen der Sekundärbranche die Entwicklung und Pflege von Software noch als reiner Kostenfaktor betrachtet. Nur wenige Unternehmen haben bereits die *strategische Bedeutung von Software* als Umsatzgenerator und „Business-Enabler“ erkannt und betrachten Softwareprojekte unter dem Aspekt ihrer Unternehmensstrategie.

Die künftigen neuen Produkte werden ohne hochgradige Qualitätssicherungen, die nur durch *qualitätsorientierte Entwicklungs- und Produktionsprozesse* erreicht werden, nicht absetzbar sein. Weitere Erfolgsfaktoren sind Steigerung der Produktivität und kurze Markteinführungszeiten. In der Praxis werden die diesbezüglichen Potenziale innovativer Software-techniken in Deutschland nur unzureichend oder zu spät 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.

Ergebnisse einer BMBF-Studie [Evasoft2000] belegen, dass

- nur ca. 30% der Unternehmen überhaupt über definierte Softwareentwicklungsprozesse verfügen, nur sehr wenige davon (Ausnahmen sind in der Avionik und Telekommunikation zu finden) erreichen einen qualitätsorientierten höheren Reifegrad – im Gegensatz zu traditionellen Ingenieurdisziplinen,
- aber nur ca. 22% der Unternehmen auf Kooperation mit externen Forschungseinrichtungen setzen, um ihre Situation zu verbessern.

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 ausreichendes softwaretechnisches Know-how 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 trotz einer mittlerweile vorhandenen Arbeitslosigkeit im minder qualifizierten Segment des IT-Sektors deutsche Unternehmen einige der höchstqualifizierten hiesigen Arbeitsplätze im SE-Bereich derzeit nur mit Ausländern besetzen. Im Widerspruch dazu, erfordert jedoch die strategische Bedeutung von Software (wie z.B. in der Fahrzeugindustrie erkannt) dringend eine Umkehr zur Eigenentwicklung von Software im Kernkompetenzbereich. Diese ist natürlich auch durch eine entsprechende Kompetenzweiterentwicklung in Hochschulen und außeruniversitären Forschungseinrichtungen zu begleiten.

Wandel des Gebiets

Die umfassende 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 betrachten, dessen Qualität messbar sein muss und das mit ebenfalls messbaren kosteneffizienten Verfahren entwickelt und produziert wird.

Dies bedeutet natürlich auch, dass *spezifische Techniken*, nämlich Methoden, Verfahren und Werkzeuge für *unterschiedliche Anwendungen* bzw. Branchen (z.B. Automobilindustrie vs. Web-basierte Systeme) entwickelt und erprobt werden müssen sowie als Basis für solche Entwicklungen umfangreiche Kooperationen zwischen Forschung und Industrie notwendig werden. Zwar fanden solche Entwicklungen oder Kooperationen zur Entwicklung spezifischer Softwaretechniken bereits statt, dies geschah aber nicht in der erforderlichen Breite und Tiefe, die der Bedeutung der Anwendungen entsprochen hätten.

3. Zur Situation der Forschung im Software Engineering

Forschungsgruppen in Deutschland in Wissenschaft und Industrie

Es gibt ca. 50 *Forschungsgruppen* zum Thema Software Engineering an Universitäten in Deutschland, die nahezu alle als Unterzeichner dieses Manifests vertreten sind. Diese Zahl ist obwohl beachtlich, deutlich zu klein, da diese Software Engineering-Gruppen sowohl Hauptverantwortung für die Ausbildung der zukünftigen Softwareingenieure tragen als auch die oben angesprochenen speziellen Softwaretechniken für verschiedene Anwendungsbereiche sowie Arten von Systemen entwickeln sollen.

Ein kürzlich fertig gestelltes Web-Portal gibt Übersicht über die speziellen *Felder* und *Projekte* der *Forschung* dieser Forschungsgruppen [SEComp].

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 breite Bereiche der Softwaretechnik ab. Auch die Max-Planck-Gesellschaft beginnt seit kurzem, 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, selten 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.

Positionierung im internationalen Vergleich

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 Herausgebern 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 und Analyse 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 „Automotive Software Engineering“, „Embedded or Hybrid Systems“ u.Ä. aufgegriffen, d.h. Fragestellungen mechatronischer Systeme und damit das *Zusammenspiel* zwischen *maschinentechnischen* und *elektrotechnischen* sowie *informatischen* Erzeugnissen diskutiert.

Zur Situation der 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 zu stark 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 als wissenschaftlich anerkannt werden wie die systematische Konstruktion eines mathematischen Beweises. Dies bedarf allerdings einer Kulturänderung in der Sichtweise potenzieller Antragsteller und deren Gutachter und einer Schwerpunktsetzung der DFG-Forschungspolitik im Bereich der Informatik.

Anzumerken ist auch, dass die zukünftige *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 sein wird. Schritte in diese Richtung sind bisher aus dem Ministerium nicht bekannt. Hier besteht offensichtlicher Handlungsbedarf. Ebenso ist kritisch zu vermerken, dass trotz der immer stärker werdenden Bedeutung von Software, und insbesondere von hochwertiger Software in unseren technischen Prozessen und Produkten, die Förderung hardwarenaher Technologien einen beständig größeren Anteil der Forschungsmittel bindet. Dem Argument, dass dies durch die kontinuierlich fortschreitende Miniaturisierung und dem dadurch entstehenden Innovations- und Wettbewerbsdruck bedingt ist, muss entgegengehalten werden, dass gerade auch 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. Auch diese Diskussion muss vermehrt mit der Politik und den Gremien der Forschungs- und Wirtschaftsförderung geführt werden.

Beobachtenswert ist stets auch die Förderpolitik anderer europäischer Länder. So hat beispielsweise in Irland die Softwareforschung bezogen auf Landesgröße und Bruttonsozialprodukt einen weitaus höheren Stellenwert als in Deutschland und hat 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 Vor-

schlagen 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-Reports 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 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.

4. Zur Situation der Lehre im Software Engineering

Software Engineering spielt in der *Ausbildung* an deutschen *Universitäten* bereits derzeit 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 existiert die wachsende Tendenz zum projektorientierten Studium, bei dem jeder Studierende die Notwendigkeit und Vorzüge softwaretechnischer Grundlagen hautnah selbst erlebt.

Durch die Einführung der Bachelor- und Masterstudiengänge im Rahmen des Bologna-Prozesses 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 Bologna-Prozess ebenfalls stets ein obligatorisches *Praxisprojekt* gefordert, das den Studierenden auch praktische Erfahrungen im Einsatz von softwaretechnischen Prinzipien, Methoden und Werkzeugen vermittelt. Teamfähigkeit und Projektmanagement verstärken diesen Praxisbezug. Auch Masterstudiengänge sehen Praktika vor, die inhaltlich und methodisch noch tiefer gehen. Zudem wird der *Umgang* mit ausgereiften *Werkzeugen* (Entwicklungsumgebungen, Dokumentations- und Versionsverwaltungssysteme) in der gesamten Informatikausbildung gestärkt.

Die Erfahrungen mit eigenen *Studiengängen* mit dem Schwerpunkt *Softwaretechnik* sind sehr gut. Versuche, wie beispielsweise 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 der Industrie 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 entwickeln.

Ausbau aufgrund verstärkter Nachfrage

Es gibt einen geschätzten Bedarf von 385.000 Softwareentwicklern in Deutschland [Eva-soft2000]. Auch bei vorsichtiger Bewertung dieser Zahl ergibt sich daraus, dass ein weiterer *Ausbau* der *Softwaretechnik* 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 (hoher Anteil Seiteneinsteiger, oft ohne Hochschulabschluss) tätig sind, ist die *Qualität und Kompetenz* des *Personals* oft verbesserungsfähig und verbesserungsbedürftig. Der hohe Anteil der Seiteneinstei-

ger, aber auch der bereits erwähnte Import hoch qualifizierter Softwareingenieure aus dem Ausland, ist eine Konsequenz nicht verfügbarer Absolventen mit erster Qualifikation in der Vergangenheit.

Es gibt weiterhin eine große *Nachfrage* nach hoch qualifizierten *Softwareentwicklern*, insbesondere für Projekte, die kritische Produkte entwickeln. Hier werden erst seit gut 10 Jahren verstärkt Informatiker herangezogen. In der Sekundärbranche werden auch in stärkerem Maße Abgänger mit interdisziplinären Vorbildungen gesucht. In diesen Firmen gibt es zudem oftmals zahlreiche 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.

5. Strategische Bedeutung des Software Engineering in der Zukunft

Die sich verstärkende Bedeutung des Software Engineering wurde bereits mehrfach betont. Wir beschreiben hier die vorhersehbare *zukünftige Situation*, die sich durch Extrapolation der Gegenwart ergibt.

Neue Funktionen durch Software

Moderne Produkte enthalten als wesentliche Komponente den Werkstoff Software in unterschiedlichen Funktionen und Rollen. Einerseits erlauben Software-basierte Lösungen neue 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 ein wichtiges *Beispiel* sei die so genannte *Context-Awareness-Qualität* genannt, die Funktionen verschiedener Komponenten in einer Art und Weise zusammenbringt, dass diese Umgebungsinformation zu erkennen und kooperativ und intelligent verarbeiten. Diese Möglichkeiten sind instrumentell, um wichtige fortgeschrittene Produkte und Services beispielsweise in Automobil- und Flugzeugbau aber auch elektronischen Dienstleistungen (z.B. eHealth) in Wirtschaft und Verwaltung zu realisieren..

Daraus ergibt sich die wesentliche Herausforderung: Neben der Konstruktion einzelner Komponenten aus Hardware- und Softwarebestandteilen ist die *Integration* und *Adaption* dieser *Komponenten* heute nicht vollständig und nicht in der gewünschten Qualität beherrscht und verstanden. Während die Konstruktion einzelner Komponenten – beispielsweise eines ABS 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 der beteiligten Disziplinen*.

Neue Qualität und Quantität

Insbesondere im Bereich eingebetteter Systeme und *eingebetteter Software* ist ein erhöhter Forschungsbedarf festzustellen, denn der Software-Anteil in eingebetteten Systemen spielt eine zunehmend größere Rolle. 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. Ähnlich wie das exponentielle Wachstum im Hardwarebereich mit Moore's Law charakterisiert wird, beobachtet man ein analoges *exponentielles Wachstum* bei eingebetteter Software.

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 intelligent, zuverlässig und effizient steuern.

Die Rolle des Software Engineering bei unterschiedlich 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 (ganz analog zum PISA-Problem). Längst verstandene Verfahrensweisen müssen in der vollen Breite der Wirtschaft so ausgerollt werden, dass gelungene Softwarekonstruktion eine *routinemäßige Kompetenz* allerorten wird.

Teil B: Forderungen und Maßnahmen

6. Verstärkung der Kompetenz in der Zukunft

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

- in der Industrie für Softwareingenieure: Sie müssen weitergebildet werden bzw. gezielt auf die herausfordernden Anforderungen hin ausgebildet werden,
- des Managements 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,
- auf Unternehmensebene in Wirtschaft und Verwaltung: Der Beitrag des Software-Personals zum Unternehmenserfolg muss erkannt und gewürdigt werden,
- in Forschung: Die Software Engineering Forschung muss international wettbewerbsfähig sein: die Besten müssen nach Deutschland kommen wollen. Darüber hinaus müssen die Forschungsfragestellungen aus den oben beschriebenen wichtigen Anwendungen transdisziplinär aufgegriffen werden.
- in der Lehre: Die universitäre Lehre muss international wettbewerbsfähig sein. Sie sollte international zusammenarbeiten. Trotz Grundlagenorientierung muss auch auf die gegenwärtige Praxis vorbereitet werden.

Aus diesen Forderungen ergeben sich die *organisatorischen Maßnahmen* zur Verstärkung dieser Kompetenz, die in den nachfolgenden Abschnitten beschrieben sind.

7. Forschungsthemen und Förderprogramme

Zukunftsorientierte Forschungsthemen

Deutschland und Europa müssen insbesondere darauf achten, dass der Technologievorsprung im Bereich *eingebetteter softwarebestimmter Systeme* gehalten und ausgebaut werden kann. Gleichzeitig sollte der Rückstand in Systemsoftware verringert werden. Um Software von zunehmender Größe und Komplexität (siehe Beispiel Automotive) in kürzeren Zyklen mit höherer Qualität und für neue Ansprüche (siehe Beispiel elektronisches Auge) zu entwickeln, müssen die dazu verwendeten Prozesse zur Entwicklung, aber auch zur Wartung weiter verbessert werden.

Damit erfolgversprechende Forschungsprojekte realisiert werden können, muss eine *Zusammenarbeit* mit *Ingenieuren*, insbesondere Maschinenbauern, erfolgen. Disziplinen wie der Maschinenbau unterscheiden sich allerdings in Bezug auf ihr Wissenschaftsverständnis merklich von der Informatik. Während in der Informatik allgemein das Bestreben nach verallgemeinerbaren Methoden vorherrscht, besteht im Maschinenbau ein vorrangiges Ziel darin, funktionierende (große) Systeme zu bauen. Disziplinübergreifende Forschungsprojekte müssen sowohl den methodischen Erkenntnisfortschritt als auch eine empirische Bestätigung beinhalten und werden daher in der Zukunft deutlich höhere Forschungsvolumina benötigen.

Neben diesen Fragestellungen, die sich im Umfeld eingebetteter Software ergeben, existieren noch eine Reihe *weiterer Themenstellungen*, die dringend genauer untersucht werden müssen:

Prozessfragestellungen

- Innovative Prozesse, um schneller auf sich ändernde Anforderungen zu reagieren und früher Ergebnisse zu erzielen (z.B. agile und inkrementelle Entwicklungsprozesse)

- Frühe und umfassende Integration in den Systementwicklungsprozess (Systems Engineering) bei der Entwicklung von Software in Produkten (eingebettete Systeme)
- Räumlich und organisatorisch verteilte sowie 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 und 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

- Wartungsprozesse für Legacy-Systeme, um deren langfristige Weiterentwicklung zu ermöglichen (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, und deren Querbezüge zur Marktforschung
- Usability Engineering und Entwicklung intuitiver und leistungsfähiger Mensch-Maschine-Schnittstellen
- Betriebs- und Monitoringmodelle von (insbesondere verteilten) Softwaresystemen, Informationslogistik
- Endbenutzer-Beherrschbarkeit Software-intensiver Produkte einschließlich von Möglichkeiten selbstbestimmter Ausgestaltung (z.B. Eigenprogrammierbarkeit von Mobiltelefonen und Autos)
- Werkzeugunterstützung zur Verifikation und Validierung (Theorembeweiser, Modellprüfer, Programmprüfer, Testmanagement und Testfallprüfung)
- 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

Hier ist insbesondere die *interdisziplinäre Zusammenarbeit* mit Ingenieuren und Forschern mit Markt- und Produktentwicklungs-Know-how zu fördern. Im Mittelpunkt steht dabei die Entwicklung eingebetteter Systeme und eingebetteter Software. 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 arbeitsintensiver 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 weiterer 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 wäre hier eine Förderung durch das BMBF und die EU sinnvoll.

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.

8. Forschung und Lehre

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

Es ist wichtig, dass Softwaretechnik als *wissenschaftliche Disziplin* akzeptiert wird, die einen eigenständigen Kern an wissenschaftlichen Grundlagen vermittelt. Notwendig hierfür ist, dass sich längerfristig eine allgemein akzeptierte „Konstruktionslehre“ für Software etabliert. Wichtig aber ist insbesondere, dass sich spezifische Softwaretechniken (z.B. für Automotive, Telekommunikation usw.) entwickeln.

Neben der Vermittlung praxisrelevanter Inhalte muss die Lehre auch gezielter auf *Forschungsinhalte vorbereiten*. Beides erfordert das Verfassen von Rahmen-Curricula, die festlegen, welche der von nationalen und internationalen berufständigen Organisationen (Fakultä-

tentag, 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 denkbar: 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 landesweiten „Doktoranden-Netzwerken“.

Aktivitäten zur Förderung einer praxisrelevanten Lehre

Um dem Anspruch der Praxisrelevanz gerecht zu werden, müssen *berufsfeldrelevante Inhalte* in Curricula noch stärker 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“ werden derzeit schon vermittelt; deren Anteil ist zu verstärken. Hierbei wird das Einbeziehen und Erproben innerhalb fachbezogener Lehrveranstaltungen empfohlen.

Eine stärkere Praxisorientierung erfordert auch 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 integriert. Ebenfalls denkbar sind Zusatzlehreangebote 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.

9. Zusammenarbeit mit der Industrie und Transfer

Für Innovation kommt dem Technologietransfer, der Umsetzung von angewandten Forschungsergebnissen in die industrielle Praxis, entscheidende Bedeutung zu. Insbesondere beim Software Engineering stellen wir einen gewaltigen Abstand zwischen Erfolg versprechenden Forschungsergebnissen und industriell eingesetzten Methoden fest. Dieser Abstand hat im Wesentlichen zwei Ursachen:

- zum einen ist die für Technologietransfer von Methoden/Prozessen notwendige enge Kooperation zwischen Forschung und Industrie in Deutschland nicht sehr ausgeprägt
- zum anderen existieren für viele Methoden keine oder nur unzureichend belegbare Aussagen über deren Nutzen und Risiken, so dass die Industrie vor dem Risiko des Einsatzes zurückscheut
- die heute notwendigen schnellen Innovationszyklen in der Industrie erfordern beschleunigten Technologietransfer. Dies erfordert verstärkte Förderung der angewandten Forschung und eine engere Kooperation zwischen Forschung und Industrie.

Einladung an die Industrie zur Zusammenarbeit

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. All 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 in die Industrie

Ein wesentlicher Aspekt von Software Engineering Technologietransfer 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 bestimmten 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.

Wir müssen also „Labore“ schaffen, in denen neue Methoden auf Ihre Eignung hin untersucht werden können. Kontrollierte Experimente mit ihren oft künstlichen Randbedingungen tragen zwar zum wissenschaftlichen Verständnis, nicht aber zur Reduktion des Risikobewusstseins in der Industrie bei. Deshalb müssen wir – ähnlich wie in anderen Disziplinen – eine industrienaher Erprobungs-/Testmöglichkeit für neue Methoden schaffen. 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 erscheinen eine (wenn nicht die einzige) Erfolg versprechende Idee. 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. Während Produktionsarbeitsplätze im globalen Zeitalter weltweit nach Kostengesichtspunkten verteilt werden, muss es unser aller Bestreben sein, die Entwicklung neuer Produkte und Prozesse im eigenen Land zu halten. Alles andere käme einem Ausverkauf deutscher Ingenieurskompetenz gleich. Die langfristigen Folgen wären katastrophal.

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 Förderungsmaßnahmen 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.

Teil C: Aufruf

Die in Teil B vorgeschlagenen *Forderungen richten sich* an die Politik, an die Industrie, an die Drittmittelgeber, an die Universitätsleitungen, aber auch an die Informatik-Fachbereiche und schließlich an die Forschungsgruppen des Software Engineering selbst. Software Engineering ist nicht nur eine Spezialdisziplin der Informatik. Sie ist in besonderem Maße für den Einsatz des Werkstoffs Software in komplexen Produkten verantwortlich. Sie trägt auch Verantwortung für die Güte der Ausbildung der zukünftigen Softwareingenieure.

Forschungsförderung Software Engineering

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 klar werden, dass Grundlagenorientierung – wie in allen Ingenieurwissenschaften – nicht nur Theorie bedeutet.

Forschung und Lehre: Ausbau in Richtung 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 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.

Forschung und Lehre: Umstrukturierung innerhalb des Software Engineering

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.

Zusammenarbeit mit Industrie und Technologietransfer

Die Industrie ist eingeladen, 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.

Referenzen

[Evasoft. 2000] GfK Marktforschung GmbH ; Fraunhofer-Institut für Experimentelles Software Engineering IESE ; Fraunhofer-Institut für Systemtechnik und Innovationsforschung ISI: Analyse und Evaluation der Softwareentwicklung in Deutschland. http://www.dlr.de/pt_it/sw/evasoftware_abschlussbericht.pdf. Dezember 2000. (Studie für das Bundesministerium für Bildung und Forschung)

[SeComp] <http://www-swe.informatik.uni-heidelberg.de/gi-fragebogen/index.php> Übersicht über die Software Engineering Landschaft in Deutschland