

# Das Produktivitätsparadoxon der Computertechnik

Peter Brödner

## 1 Einführung: Computer als universelle Rationalisierungsmaschine?

Rationalisierung ist ein schillernder Begriff. Die Vieldeutigkeit steckt schon im lateinischen Wortstamm ›*ratio*‹, der je nach Kontext sowohl ›Berechnung‹ als auch ›Vernunft‹ bedeuten kann. Ersteres trifft auf Computer genau zu: Sie führen nichts weiter als berechenbare Funktionen aus. In diesem Sinne können Computer und vernetzte Computersysteme zurecht als komplexe Maschinen zur Rationalisierung aufgefasst werden. So heißt im Angelsächsischen die wissenschaftliche Lehre, die deren Entstehung und Betrieb sowie ihre logisch-mathematischen und physikalischen Grundlagen zum Gegenstand hat, ebenso zutreffend ›Computer‹ bzw. ›Computing Science‹.

Im üblichen Sprachgebrauch wird unter Rationalisierung jedoch noch etwas anderes verstanden. Häufig wird mit Rationalisierung zunächst in einengender Weise ein Bündel von Maßnahmen zur Kostensenkung und zur Reduktion von lebendiger Arbeit bezeichnet, die in der Folge zu höherer Arbeitsproduktivität und Abbau von Arbeitsplätzen führen. Diese enge Sicht wird aber der tatsächlichen vielseitigeren wirtschaftlichen Dynamik industrieller Entwicklung seit Ende des 18. Jahrhunderts nicht gerecht, dazu muss der Rationalisierungsbegriff weiter gefasst werden. *Rationalisierung* meint dann in diesem weit gefassten Sinn, Arbeits- und Wertschöpfungsprozesse nach jeweils verfügbarem Wissen so zu gestalten, dass sozial und wirtschaftlich gewünschte Effekte mit dem jeweils geringst möglichen Aufwand erzielt werden. Das umfasst nicht nur zweckmäßige Mittel, sondern bezieht auch die »Vernünftigkeit von Zwecken« in die Betrachtung ein (so bereits Weber 1921). Mithin richtet sich Rationalisierung auf in der Gesellschaft als relevant erachtete Ziele, etwa Produktivität und Wettbewerbsfähigkeit von Unternehmen oder auch sozialverträgliche Gestaltung von Arbeit und Technik, und hängt dabei von jeweils vorgefundenen Umständen ab.

So hat uns die durch Zwänge der Kapitalverwertung angetriebene Industrialisierung, anfangs aufgrund ausgeprägter Arbeitsteilung in eng spezialisierte Verrichtungen bei Hand- und Kopfarbeit gleichermaßen (Smith 1776, Babbage 1832), im weiteren Verlauf auch durch massiven Einsatz von energie- und stoffumwandelnden Maschinen (»Kraft- und Arbeitsmaschinen«) eine enorme Steigerung der Arbeitsproduktivität und mit ihr beträchtlichen Wohlstand beschert. Seit 1850 hat sich die Arbeitsproduktivität in entwickelten Gesellschaften um mindestens den Faktor 20 erhöht, bei gleichzeitiger Verdoppelung der Lebenserwartung, erheblichem Wachstum der Bevölkerung und starker Urbanisierung. Begleitet wurde diese in der Kulturgeschichte singuläre Entwicklung durch die ›Verwissenschaftlichung‹ der (Re-)Produktion, d.h. durch deren analytische Durchdringung und Explikation von Erfahrung in begriffliches Wissen. Als Folge und weitere Triebkraft zugleich wachsen Bildungs- und Forschungseinrichtungen und zeichenbasierte Wissensarbeit breitet sich überall stark aus. Heute ist der Wert der spezifischen Verwendung von Wissen in Produkten oder Diensten oftmals schon größer als der eingesetzte Materialwert; ohne adäquates Wissen gibt es weder Innovation noch produktive Arbeit.

Vor dem Hintergrund dieser Entwicklung wird nun heute die produktivitätssteigernde Wirkung des Einsatzes von energie- und stoffumwandelnden Maschinen zumeist umstandslos auf den Einsatz von Computern als dominanter Basistechnik im Umgang mit zeichenbasierter Wissensarbeit in Wertschöpfungsprozessen übertragen (heute irreführend als ›Digitalisierung‹ bezeichnet). So werden Computersysteme gemeinhin als technische Artefakte betrachtet, mittels derer jedwede Wissensarbeit neu gestaltet und leistungsfähiger, insbesondere produktiver gemacht werden kann.

Dank ihrer Programmierbarkeit und vielfältigen Einsatzfähigkeit werden sie auch als universell nutzbare ›enabling technology‹ gesehen, von der hohe Effizienzgewinne erwartet werden.

So werden seit inzwischen über vier Dekaden immer wieder große Summen in den Einsatz von Computersystemen und deren Vernetzung investiert in der Hoffnung auf ebenso große Steigerungen der Arbeitsproduktivität, meist freilich ohne die erwarteten Wirkungen tatsächlich zu überprüfen. Regierungen legen stets aufs neue gut ausgestattete Förderprogramme auf, um die Innovationskraft und Wettbewerbsfähigkeit von Unternehmen zu sichern und die Leistung öffentlicher Dienste zu steigern. Manche Beobachter sprechen in diesem Kontext wiederholt sogar von einem entstehenden »globalen Informationsraum«, der sich durch die Nutzung via Internet global vernetzter Computer bilde und einen »Produktivkraftsprung« bewirke (z.B Boes et al. 2014). Entsprechend wird auch immer wieder technologische Arbeitslosigkeit befürchtet (›Computer als Jobkiller‹), diesmal auch bei qualifizierten Wissensarbeitern. Während in den 1980er Jahren im Zusammenhang mit Forschungen zu wissensbasierter »symbolischer künstlicher Intelligenz (KI)« von der »menschenleeren Fabrik« die Rede war, bestimmen heute ›maschinelles Lernen‹ und ›künstliche neuronale Netze‹, sog. subsymbolische ›KI‹, den öffentlichen Diskurs um Folgen der sog. ›Digitalisierung‹ für Arbeit und Beschäftigung.

Dem stehen freilich empirische Befunde in großer Zahl und erdrückender Aussagekraft entgegen, die auf einzelwirtschaftlicher Ebene nur in einzelnen Fällen unter jeweils bestimmten Kontextbedingungen Produktivitätssteigerungen konstatieren und gesamtwirtschaftlich trotz massiver Investitionen, die seit längerem diejenigen in Produktionstechnik übersteigen, kaum zusätzliche produktivitätssteigernde Effekte nachzuweisen vermögen. Dieses Phänomen wird als Produktivitätsparadoxon der Computertechnik bezeichnet und wurde erstmals von Robert Solow (1987) konstatiert: »You can see the computer age everywhere but in the productivity statistics«.

Diesem Paradoxon will der vorliegende Beitrag nachgehen, indem zunächst die wichtigsten empirischen Befunde zu den Produktivitätseffekten des Computereinsatzes auf gesamt- wie auf einzelwirtschaftlicher Ebene referiert werden. Danach wird den Gründen für die darin aufscheinende offenkundige Selbsttäuschung der Akteure nachgegangen, die im wesentlichen in einem falschen Verständnis der Besonderheiten der Funktionsweise von Computern und ihres Einsatzes zu suchen sind und das Phänomen hinreichend zu erklären vermögen. Abschließend werden Schlüsse gezogen, wie künftig diese Technik besser gestaltet und genutzt werden kann.

## **2 Das Produktivitätsparadoxon: Empirische Befunde**

Seit der berühmt gewordenen Aussage von Solow über die paradoxe, besser gesagt: kontraintuitive Wirkung der Computertechnik auf die Arbeitsproduktivität ist eine Fülle empirischer Untersuchungen über das Phänomen erschienen, zunächst v.a. in den USA (zur Übersicht vgl. etwa Accenture 2015). Aufgrund der vergleichsweise vielfältigeren Analysen und besseren Datenlage wird hier vorwiegend auf US-bezogene Studien zurückgegriffen (zumal die Entwicklung und Anwendung der Computertechnik im jüngeren Verlauf amerikanisch dominiert ist); die Befunde können grundsätzlich aber auch für deutsche und europäische Verhältnisse Geltung beanspruchen. Wie die empirischen Befunde zeigen, ist es dabei aufgrund von Unterschieden in der Auswirkung angemessen, zwischen der gesamt- und der einzelwirtschaftlichen Perspektive zu unterscheiden.

### ***2.1 Gesamtwirtschaftliche Perspektive***

Nachdem sich die Computertechnik trotz ihrer grundsätzlich universellen Eignung für die Rationalisierung jedweden zeichenbasierten Arbeitsprozesses (was in Kap. 3 näher erläutert wird) anfangs über zwei Dekaden nur sehr zögerlich und wenig dynamisch verbreitet hat, nimmt deren

Diffusion ab den frühen 1970er Jahren beträchtlich an Fahrt auf und führt im weiteren Verlauf zu massiv wachsenden Investitionen in einer Vielzahl von Sektoren von Produktion und Dienstleistung. In der gleichen Zeit hat sich die Leistung der Hardware-Komponenten (Prozessoren, Speicher, Datentransfer-Kapazität) entsprechend dem Mooreschen ›Gesetz‹ exponentiell gesteigert, was das Preis-Leistungsverhältnis stark verringert und damit die Investitionsdynamik deutlich befeuert. So wird die Hardware bezogen auf ihre Leistung nun ständig billiger – ein Umstand, dem bei der Berechnung der Investitionssummen durch statistische Ermittlung eines Deflators auch Rechnung getragen wird. Andererseits steigt mit der Vielfalt der Einsatzfälle und Aufgaben der notwendige Aufwand für die Entwicklung passender und komplexer werdender Software stark an und macht den Hauptanteil an der Investition aus. Das entgegen allen Erwartungen Überraschende ist nun, dass exakt in dem Moment zu Beginn der 1970er Jahre, ab dem die Investitionsdynamik in Computertechnik massiv von unter 10 auf über 200 Mrd. \$ p.a. (1990) zunimmt, die Arbeitsproduktivität in den Dienstleistungssektoren zu stagnieren beginnt, während sie in der Produktion in gewohntem Maße weiter wächst, was insgesamt das gesamtwirtschaftliche Produktivitätswachstum wegen des zunehmenden Gewichts der Dienstleistungen stark bremst (Brynjolfsson 1993).

Mitte der 1990er Jahre setzt dann ebenso überraschend eine sprunghafte Zunahme der Arbeitsproduktivität von rd. 1% p.a. auf 2,3 % p.a. ein (Stiroh 2002; in Europa deutlich schwächer), was viele Beobachter zu der Vermutung veranlasst, dass sich endlich, nach jahrzehntelanger Verzögerung, die produktivitätssteigernden Wirkungen der Computertechnik auch in der Wirtschaftsstatistik niederschlagen. Das hat sich aber in der Folge als großer Irrtum herausgestellt (Acemoglu et al. 2014). Wie das McKinsey Global Institute (2002) aufgrund von Sektordaten des Bureau of Economic Analysis (BEA) nachweisen kann, lässt sich nämlich der ganze Produktivitätssprung auf einmalige strukturelle Veränderungen in nur 6 Wirtschaftszweigen, darunter ganz überwiegend im Groß- und Einzelhandel (sog. ›Big Box-Strategie‹), zurückführen, die nur sehr bedingt überhaupt mit Computertechnik verbunden sind.

Dementsprechend ist die insgesamt nur eine Dekade andauernde Produktivitätszunahme auf über 2% p.a. im Jahre 2005 schon wieder zu Ende. Seitdem liegt sie zunächst bei 1% und ist zuletzt trotz weiterer Investitionen in Computertechnik auf anhaltend hohem Niveau sogar deutlich unter diesen Wert gefallen (Gordon 2014, 2016). Das gilt trotz aller Unterschiede im Detail grundsätzlich auch für andere entwickelte Länder, etwa die G7-Länder, wie nachstehende Abb. 1 ausweist.

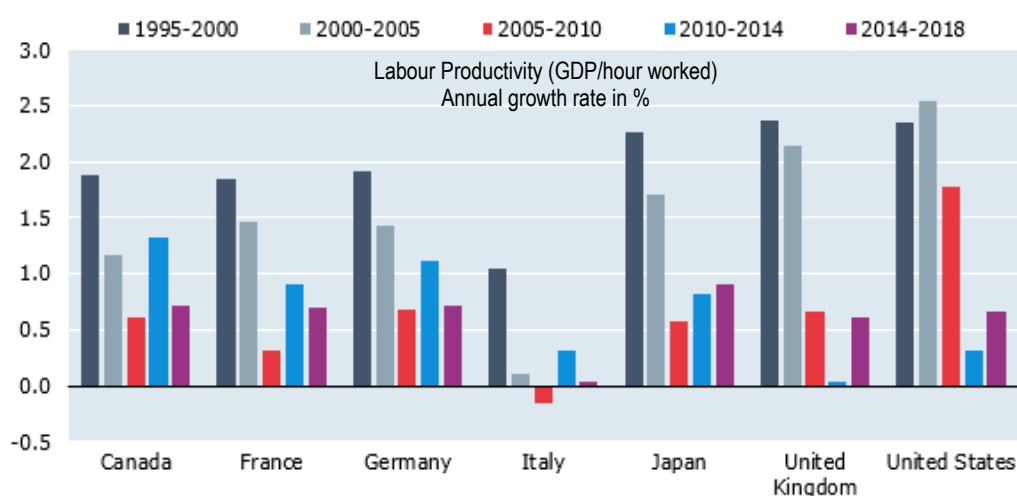


Abb.1: Entwicklung der Arbeitsproduktivität in G7-Ländern 1995- 2018 (OECD 2019)

Zum besseren Verständnis der Verbreitungsdynamik der Computertechnik ist darauf zu verweisen, dass 1966 mit dem Erscheinen der Mainframe-Rechner IBM 360 und PDP 10 von Digital Equipment, deren ›Virtual Machine‹ genannte Betriebssysteme über Terminals den unmittelbaren Zugang zu Datenverarbeitungskapazität am Arbeitsplatz ermöglichen, ganz neue Arbeitsabläufe bei deren Nutzung eröffnet werden. Außerdem kommen zu Beginn der 1970er Jahre die ersten von Texas Instruments entwickelten Mikroprozessoren auf den Markt und erleben mit dem Z 80 von Zilog 1976 einen Durchbruch. Durch sie wird auch die Entwicklung von Personal Computern möglich (Apple 1976, IBM 1981 mit Betriebssystem von Microsoft, Macintosh von Apple 1984). Grafische Benutzungsoberflächen zur Interaktion mit Computern und gebrauchstaugliche Software für Tabellenkalkulation, Grafik- und Textedition machen Computer nun zu weithin verwendbaren Werkzeugen für Wissensarbeiter. Durch beide Neuerungen wird der Einsatz von Computern aus den Fesseln der von Hohepriestern der Datenverarbeitung bewachten Heiligtümer der Rechenzentren befreit und dem allgemeinen Gebrauch in den vielfältigen zeichenbasierten Arbeitsprozessen von Produktion und Dienstleistung zugänglich. Zudem wird Ende 1982 mit dem Protokoll TCP/IP ein internationaler Quasi-Standard zum Datentransfer zwischen Computern gesetzt, der nach und nach im Zuge des Ausbaus von Übertragungskapazitäten der Telekommunikation die weltweite Vernetzung von Computersystemen mittels kostengünstiger ›Datenfernübertragung‹ ermöglicht.

Bei der Erklärung dieses paradox oder kontraintuitiv erscheinenden Phänomens sinkender Produktivitätszuwächse bei anhaltend sehr hohen Investitionen in Computertechnik als basaler ›enabling technology‹ ist zunächst zu berücksichtigen, dass die gesamtwirtschaftliche Arbeitsproduktivität von einer Vielzahl auch gegenläufiger, sich teilweise wechselseitig kompensierender Einflüsse abhängt. Daher ist es ohne zusätzliche Analysen kaum möglich, einzelne Faktoren wie den Einsatz von Computertechnik soweit zu isolieren, dass deren spezielle Wirkung aufgezeigt werden kann. So zeigt sich in einer Vielzahl von Studien immer wieder, dass der Zusammenhang zwischen Aufwendungen für Computersysteme und deren Wirkungen auf die Arbeitsproduktivität außerordentlich diffus ist (Übersichten über empirische Untersuchungen zur Makro- und Mikroebene geben Potthoff 1998, Accenture 2015).

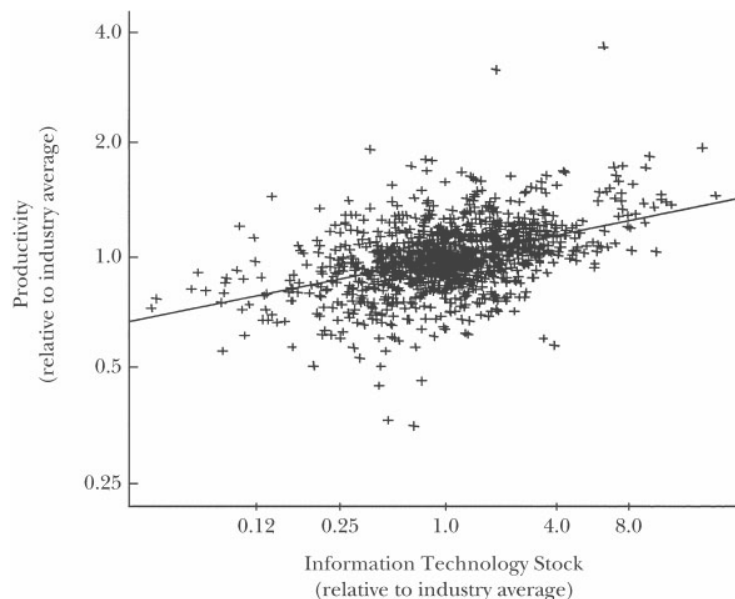
Dessen eingedenk haben sich in jüngerer Zeit die Untersuchungen von gesamtwirtschaftlich orientierten zu einzelwirtschaftlichen Analysen verlagert. Im Umgang einzelner Organisationen mit Computertechnik zeigen sich denn auch erstaunliche Unterschiede.

## **2.2 Einzelwirtschaftliche Perspektive**

Zunächst liefern diesbezüglich verfügbare Daten aus 400 großen US-Unternehmen ein eindruckliches Bild davon, wie höchst unterschiedlich diese Unternehmen produktiven Nutzen aus dem Einsatz von Computersystemen (›IT‹) ziehen (vgl. Abb. 2). Obwohl sie mehr oder weniger alle gleiche oder zumindest funktionsähnliche Systeme nutzen, allerdings in sehr unterschiedlichem Umfang (›IT stock‹), variiert die Arbeitsproduktivität enorm: Z.B. ist sie bei einem Unternehmen am oberen Rand der Punktwolke um den Faktor 4 größer als bei einem am unteren Rand (die Achsen sind logarithmisch skaliert). Zudem ist die Korrelation zwischen dem kumulierten Computereinsatz (›IT stock‹) und der Produktivität insgesamt vergleichsweise schwach. Zwar ist die mit Blick auf die Verfügbarkeit entsprechender Daten getroffene Auswahl der Unternehmen nicht repräsentativ, wegen des fortgeschrittenen Computereinsatzes aber umso instruktiver (Brynjolfsson & Hitt 2000, Brynjolfsson 2003).

Deutlich verweist dieses Bild darauf, dass neben dem Umfang des Einsatzes von Computertechnik noch andere Faktoren die Wirkung auf die Arbeitsproduktivität beeinflussen, etwa die soziale und organisatorische Einbettung der Computersysteme, Managementpraktiken im Umgang mit ihnen oder die Befähigung der Nutzer für deren wirksamen Gebrauch. Weitere Einflüsse sind in einem

unzulänglichen Verständnis der Funktionsweise dieser Systeme und ihrer Nutzung, in unpassend definierten Computerfunktionen oder in deren wirksamer Nutzung entgegenstehenden Organisationsformen zu suchen.



*Abb. 2: Korrelation zwischen Arbeitsproduktivität und Computereinsatz  
(Daten aus 400 großen US-Unternehmen; Brynjolfsson & Hitt 2000)*

Bereits früh wurde denn auch versucht, diesen Zusammenhängen mittels ökonomischer Analysen auf die Spur zu kommen. So zeigt sich etwa, dass die Wirkungen isolierter Einsätze von Computersystemen auf die Arbeitsproduktivität für sich genommen eher gering sind. Erst wenn sie mit komplementären organisatorischen Veränderungen und Qualifizierungsmaßnahmen zur Ertüchtigung der Benutzer für die neuen Arbeitsstrukturen einhergehen, stellen sich deutliche Steigerungen der Arbeitsproduktivität ein. Aus diesen Analysen ergibt sich insbesondere, dass mögliche Leistungssteigerungen in verschiedener Hinsicht von solchen komplementären Maßnahmen abhängen (Bresnahan et al. 2002, Brynjolfsson 2003):

- Computersysteme vermögen v.a. dann die Leistungsfähigkeit von Unternehmen zu steigern, wenn deren Einführung mit durchgreifender Dezentralisierung, objektorientierter Reorganisation in multifunktionalen Teams und Investitionen in ›Humankapital‹ einhergeht.
- Unternehmen mit derart dezentralisierten Organisationsformen erreichen höhere Produktivität in der Nutzung von Computersystemen als solche, die nur Computer einsetzen.
- So genannte ›intangible assets‹, also z.B. Managementfähigkeiten, Erfahrung oder kollektive Handlungskompetenz, beeinflussen stark den realisierten Nutzen von Computersystemen.
- In der Regel ist der Aufwand für organisatorische Erneuerung und Qualifizierung um ein Vielfaches höher als die Ausgaben für Hard- und Software.

Ähnliche Befunde haben sich auch aus eigener Forschung am Institut Arbeit und Technik ergeben. Als in dieser Hinsicht sehr aufschlussreich haben sich Fallstudien zum Einsatz von Produktionsplanungs- und -steuerungssystemen (PPS- bzw. ERP-Systemen) in produzierenden Unternehmen erwiesen. Als integrierte betriebswirtschaftliche Systeme für Materialwirtschaft, Auftragssteuerung, Finanzbuchhaltung und Personalplanung durchdringen sie Organisationen nahezu vollständig und strukturieren deren Arbeitsabläufe. An ihnen lassen sich daher vor allem

organisationsbezogene Einsatzprobleme und Herausforderungen wirksamer Aneignung besonders gut studieren.

Beim Einsatz dieser Systeme zeigt sich, dass sieben von zehn Unternehmen einem auf zentrale Planung und Kontrolle ausgerichteten, dabei auf rein technische Funktionalität fixierten Blick auf den Computereinsatz folgen – mit außerordentlich schädlichen Folgen für deren wirtschaftliche Leistung: Projekte der Implementierung überziehen die Zeit- und Kostenbudgets beträchtlich und trotz hoher Kosten verbessern sich wettbewerbsrelevante Leistungsgrößen wie Produktivität, Durchlaufzeiten oder Bestände kaum (was meist ohne Konsequenzen hingenommen wird). Implementierungsprozesse konzentrieren sich ganz überwiegend auf die einmalige Bestimmung der Systemanforderungen zu Beginn, während Fragen der Nutzung und kollektiven Aneignung der Systeme kaum in Betracht gezogen werden. Dabei bleiben viele Funktionen ungenutzt, notwendiges Wissen über Geschäftsprozesse wie über Bedingungen und Folgen kollektiven, systemgestützten Handelns im Organisationszusammenhang ist unter den verschiedenen Akteuren unzureichend vermittelt und oft entstehen fehlerhafte, zu Fehlentscheidungen verleitende Datenhalden (Maucher 1998, 2001).

Nur eine kleine Minderheit von Unternehmen setzt demgegenüber adäquat an organisatorischen Veränderungen an. Mit großem wirtschaftlichem Erfolg richten sie Wertschöpfungsprozesse konsequent am Kundennutzen aus und beginnen mit deren objektorientierter Restrukturierung, um eine dementsprechend angepasste Konzeption und Nutzung von Computersystemen als Arbeitsmittel und Medium der Kooperation zu entwickeln. Zur Bewältigung der organisatorischen Veränderungen und wirksamen Aneignung der Systeme beteiligen sie betroffene Fachleute und organisieren schon in der Konzeptions- und Einführungsphase kollektive Lernprozesse. Nur wenn die Arbeitsweise, mithin die Regeln gemeinsamen Handelns, im Gebrauch der Systeme angemessen verändert werden, lassen sich deren Nutzenpotenziale wirksam aktivieren (Maucher 1998, 2001).

Entsprechend lässt sich der derzeitige Forschungsstand aufgrund einer Metastudie zur Auswertung von über 50 Einzelanalysen wiederum aus den USA wie folgt zusammenfassen: »... *the wide range of performance of IT investments among different organizations can be explained by complementary investments in organizational capital such as decentralized decision-making systems, job training, and business process redesign. IT is not simply a tool for automating existing processes, but is more importantly an enabler of organizational changes that can lead to additional productivity gains*« (Dedrick et al. 2003). Zu einem ganz ähnlichen Schluss gelangen auch Jorgenson et al. (2008) in einer neueren Analyse der Entwicklung der US-Arbeitsproduktivität: »... *a wealth of microeconomic evidence emphasizes the complexity of the link from technology to productivity. To leverage information technology investments successfully, firms must typically make large complementary investments and innovations in areas such as business organization, workplace practices, human capital, and intangible capital.*«

### **2.3 Permanenz der Software-Krise**

Die Kosten für Computer Hardware haben seit langem eher sinkende Tendenz, denn deren Leistungsfähigkeit wächst bis in die jüngste Zeit exponentiell; so verdoppelt sich beispielsweise die Rechenleistung von Prozessoren etwa alle zwei Jahre (>Moore's Gesetz«, allerdings nähern sich neuerdings die stofflichen Strukturen physikalischen Grenzen). Die Aufwendungen für Entwicklung, Erprobung und Wartung der Software machen daher schon längst den ganz überwiegenden Teil der Gesamtkosten von Computertechnik aus, v.a. auch, weil Softwaresysteme dank der hohen Hardware-Leistung immer funktionsreicher und komplexer werden können. In Form von immer mehr aufeinander aufbauenden Schichten türmen sich immer vielfältiger zusammengesetzte, abstraktere Funktionen, die jeweils zur Ausführung spezifischer Aufgaben

kaskadenförmig auf generische Funktionen tieferer Schichten zurückgreifen. Das gilt insbesondere auch für die komplexen, für den Einsatz in Organisationen bestimmten Computersysteme, die hier im Fokus stehen.

Damit verlagern sich auch die Kostenanteile für die Software selbst mehr und mehr auf deren Wartung und Pflege (einschließlich immer aufwendigerer Fehlerbehebung und Tests, die gleichwohl keine Korrektheit garantieren) sowie die Anpassung an erneuerte Betriebssysteme und veränderte Entwicklungsumgebungen. Wachsende Anteile beansprucht auch die Abwehr und Bewältigung äußerer Angriffe infolge weitreichender Vernetzung. Mit Blick auf die große Bedeutung von Softwarefehlern schätzt die Zeitschrift für Computertechnik IX beispielsweise für das Jahr 2005 den jährlichen Aufwand für Fehlerbeseitigung in deutschen Unternehmen auf 14,4 Mrd. € (36% des IT-Budgets) und die durch Softwarefehler verursachten Verluste auf rd. 84 Mrd. € im Jahr (IX-Studie 1/2006: Software-Testmanagement, Hamburg: Heise). Ähnlich schätzt das US Institute of Standards and Technology (NIST) die Folgekosten von Softwarefehlern im Jahre 2002 auf rd. 60 Mrd. USD (oder 0,6% des US-BIP). Auch wenn das nur Schätzungen sind, geht es hier nicht um ›Peanuts‹.

Bereits im Jahr 1968 wurde auf einer von der NATO gesponserten Konferenz erstmals eine ›Software-Krise‹ ausgerufen (Naur & Randell 1968). Seither haben sich, um der immer wieder neuen Schwierigkeiten im Umgang mit Funktionsvielfalt und Komplexität Herr zu werden, die Methoden des Software Engineering rasant entwickelt, etwa mit systematischen Vorgehensweisen zu weitreichender Abstraktion und Modularisierung, mit abstrakten Datentypen und strukturierter, objektorientierter Analyse, Entwurf und Programmierung oder mit vielfältigen Bemühungen um Weiter- und Wiederverwendung von erprobten Software-Modulen in Gestalt von Programmbibliotheken und Software-Frameworks.

Einerseits haben diese zweifellos hilfreichen methodischen Verbesserungen jedoch im Software Engineering beträchtliche Rebound-Effekte zur Folge: Je zuverlässiger und wirksamer sich diese Methoden zeigen, desto größere und komplexere Projekte werden anzugehen gewagt. Andererseits ändert das an dem in der Softwaretechnik leider gewohnten, verglichen mit anderen Technikfeldern extrem hohen Anteil an Projekten wenig, die entweder vollständig oder zumindest teilweise scheitern, indem sie ihre Zeit- und Geldbudgets bei weitem überziehen, ohne den geplanten Funktionsumfang je zu erreichen (dazu gibt es eine Fülle von Einzelbeispielen, vgl. Landauer 1995; Emam & Koru 2008). Auf Basis der Daten aus über 50.000 Software-Projekten weltweit ergibt sich, dass die Anteile komplett gescheiterter Projekte (um 20% aller Projekte), der teilweise gescheiterten Projekte (um 53%) und schließlich der erfolgreichen Projekte (um 27%) im Zeitverlauf zwar leicht schwanken, sich aber nicht grundsätzlich bessern; sie hängen hauptsächlich von der Projektgröße ab: je größer, umso gefährdeter (Standish Group International 2016). Mehr oder weniger ist in großen Teilen der Software-Entwicklung inzwischen ein Zustand ›rasenden Stillstands‹ erreicht, in dem viel Aktivität ohne substantiellen Fortschritt entfaltet wird.

Diese bei der Nutzung von Computertechnik in Organisationen in besonderem Maße hervortretenden Probleme und trotz aller methodischen Verbesserungen nun schon seit fünf Dekaden andauernden Krisenerscheinungen im Software Engineering verweisen offenbar auf tiefer liegende systematische Ursachen, die nicht einfach durch einzelne Fehlleistungen zu erklären sind. Nach den vorgestellten empirischen Befunden ist vielmehr zu vermuten, dass wichtige Grundlagen der Fachdisziplin nicht hinreichend geklärt sind, damit grundsätzlich falsche, zu Illusionen führende Annahmen über die Funktionsweise und Nutzungsmöglichkeiten der Systeme wirksam sind, darüber hinaus auch unangemessene Vorgehensweisen beim Projektmanagement zur Gestaltung und Implementation der Systeme zum permanenten Elend beitragen. Diesen Überlegungen wird im folgenden näher nachgegangen.

### 3 Hintergrund: Einblicke in den Maschinenraum der Computertechnik

#### 3.1 Die Maschine als Fetisch – das verkehrte Verhältnis von Mensch und Maschine

Wenn in entwickelten Gesellschaften über Technik gesprochen oder berichtet wird, dann wird darunter in der Regel schlicht eine Ansammlung technischer Artefakte oder Maschinen verstanden. Sie sind plötzlich wie aus heiterem Himmel da, führen anscheinend ein Eigenleben und ›verändern‹ unser Leben grundlegend, indem sie, wie es heißt, ›etwas mit uns machen‹. Dieses Phänomen wird gelegentlich als »technologischer« oder »Technikdeterminismus« (Lutz 1987) apostrophiert und ist in diesen Tagen, insbesondere im Zusammenhang mit Computertechnik, nach einer Phase zumindest partieller Aufklärung, wieder vorherrschend. Ständig lesen und hören wir von der »Macht der Algorithmen«, die etwas ›prognostizieren‹ (Rückfälligkeit von Straftätern), ›entscheiden‹ (Kreditvergabe) oder ›empfehlen‹ (Routenplanung, Kauf bestimmter Artikel), dass angeblich ›autonome Agenten‹ (z.B. ›selbstfahrende Autos‹) unsere Arbeit übernehmen und uns im Grunde überflüssig machen. Darin kommt ein zwar weit verbreitetes, aber gänzlich fehlgeleitetes Verständnis von Technik zum Ausdruck, das Grundfragen nach dem unterliegenden Menschenbild, unserem Verhältnis zur Welt und zu uns selbst aufwirft.

Im ursprüngliche Wortsinn wird mit *téchne* eine List bezeichnet, die List, Wirkungen der äußeren Natur für eigene Zwecke zu nutzen. Aristoteles sieht darin einen eigenen Teil unserer praktischen Vernunft, die Fähigkeit, etwas Nützliches herstellen zu können, beruhend auf Erfahrung, Übung und Einsicht in Naturverhältnisse. Damit gerät das Verhältnis des Menschen zur äußeren Natur wie auch sein Verhältnis zu sich selbst als sozialem Gattungswesen in den Fokus. In dieser Perspektive erscheint die kulturelle Evolution als die Fortsetzung der natürlichen mit anderen Mitteln: Den Menschen als Produkt natürlicher Evolution kennzeichnet die durch seine bewusste Tätigkeit geschaffene Welt der Kultur, deren Fortentwicklung sich, getrieben von jeweils gesellschaftlich dominanten Interessen, niederschlägt in Formen der Herstellung und des Gebrauchs von

- Werkzeugen als Mitteln zweckmäßiger Nutzung von Naturkräften (*homo faber*: Werkzeuge sowie energie- und stoffumwandelnde Maschinen vermitteln kausale Wirkungen),
- Zeichen als Mitteln sozialer Interaktion, Kommunikation und Reflexion (›semiotisches Tier‹: Sprache, Schrift und Buchstabendruck vermitteln intentionale Bedeutungen).

So zeigt eine genauere Analyse, dass das auf Artefakte fixierte Technikverständnis in die Irre führt: Technische Artefakte fallen nicht vom Himmel, sondern müssen für bestimmte Zwecke mittels Einsicht in Prozesse der Natur oder sozialer Praxis mühsam konzeptionell entworfen und stofflich hergestellt werden. Als solche sind sie aber bloß tote, nutzlose Gegenstände, solange sie nicht für bestimmte Aufgaben zweckgemäß eingesetzt, mithin dafür angeeignet und praktisch wirksam verwendet werden. Das alles geschieht stets im Spannungsfeld des technisch Machbaren, der Gestaltbarkeit von Natur bzw. sozialer Praxis, und des sozial Wünschenswerten, abhängig von jeweils herrschenden Interessen. In dem auf Artefakte fixierten Technikverständnis verkehrt sich stattdessen das Verhältnis von Menschen zu einander und zur äußeren Natur: Durch Verdinglichung personaler Verhältnisse und Personifizierung von Dingen werden Maschinen zum Fetisch erhoben. Hierin wurzelt die »Macht der Machwerke über die Machenden« (Haug 2005) oder, wie Anders (1973) sagt, die »prometheische Scham« des Menschen vor der Perfektion seiner eigenen Artefakte. Dieser kollektive Wahn führt letztlich zu Infantilisierung und Selbstentmündigung, wie sie etwa in Formen der Verhaltenssteuerung durch Smartphone-Gebrauch bereits zum Ausdruck kommen.

Im Unterschied dazu wird nach allgemeinem professionellen Verständnis Technik definiert als die Gesamtheit von Maßnahmen zur Herstellung und zum Gebrauch künstlicher Mittel für gesellschaftliche Zwecke. Ihr werden damit nicht nur die Artefakte und Sachsysteme selbst



zugerechnet, sondern auch deren sozial konstruierte und kulturell vermittelte Herstellung und Anwendung (Ropohl 1991, VDI 1991). Als geronnene Erfahrung verkörpern sie begriffliches, explizites Wissen über die Natur bzw. soziale Praxis und als Arbeitsmittel stellen sie Handlungsanforderungen an ihren Gebrauch, durch den die Artefakte erst ihren Sinn erhalten und in ihrer Qualität zu beurteilen sind. Gerade in den Prozessen der Entwicklung und Herstellung technischer Artefakte sowie ihrer Aneignung zu praktisch wirksamer Verwendung liegen die eigentlichen Probleme von Technik als der ›Anstrengung, Anstrengungen zu ersparen‹, dem Sinn technischen Handelns; eben hierin liegen auch die Wurzeln missbräuchlichen Umgangs.

Vor diesem Hintergrund müssen nun die fundamentalen Unterschiede zwischen klassischen Maschinen der *Energie und Stoffumwandlung* (›Kraft- und Arbeitsmaschinen‹, chemische oder biologische Prozesse) und Computern als *semiotischen* Maschinen betrachtet werden. Erstere nutzen durchweg Kenntnisse der Thermo- bzw. Elektrodynamik und Mechanik, um zweckmäßig gestaltete Funktionen der *Kraftübertragung* zu realisieren und so Naturkräfte und -effekte nutzen zu können. Im Unterschied dazu greifen Computer in Zeichenprozesse sozialer Praxis ein, sind damit programmgesteuerte, ›Daten‹ verarbeitende Maschinen. Energie- und stoffumwandelnde Maschinen übertragen Kräfte, semiotische Maschinen manipulieren bedeutungslose Zeichenträger.

Die fundamentalen Unterschiede zwischen beiden Maschinenklassen liegen mithin in deren Wirkbereichen, Funktionsweisen und Zwecken. Der *Wirkbereich* von Kraft- und Arbeitsmaschinen (wie auch von artifiziellen chemischen und biologischen Prozessen) liegt in der Natur und nutzt natürliche Kräfte zweckgemäß für Prozesse der Energie- und Stoffumwandlung, während der Wirkbereich semiotischer Maschinen ganz in zeichenbasierter sozialer Interaktion liegt und auf wohl determinierten Funktionen der Verarbeitung von Daten als Zeichenträgern im Rahmen zugrunde liegender Zeichenprozesse beruht. Mit semiotischen Maschinen wird der soziale Raum der Zeichenprozesse und Interaktion nirgends verlassen. Im Unterschied dazu beruht die *Funktionsweise* von Maschinen und Prozessen der Energie- und Stoffumwandlung auf der Einsicht in natürliche Effekte als Ergebnis von Naturerkenntnis und ihr Zweck ist das Nutzen von Naturkräften. Die Funktionsweise semiotischer Maschinen beruht dagegen auf Vorschriften zur Manipulation von Daten, gewonnen durch Analyse, Modellierung und Formalisierung von Zeichenprozessen kognitiver Arbeit und sie dient der Organisation und Koordination kollektiven Handelns. Den zwecks Maschinisierung körperlicher Arbeit geschaffenen mechanischen Funktionen der Kraftübertragung als dem Kern maschineller Energie- und Stoffumwandlung entsprechen dann bei Computern als semiotischen Maschinen die algorithmischen Funktionen für Datenverarbeitung, -speicherung und -transfer zwecks Maschinisierung kognitiver Wissensarbeit.

Beiden Klassen technischer Artefakte gemeinsam ist zunächst, dass beide stets das zur Problemlösung erforderliche explizite methodische und funktionale Wissen vergegenständlichen, das sich der natürlichen analytischen Intelligenz ihrer Konstrukteure verdankt. Beiden Klassen gemeinsam ist ferner ihre enge Verwandtschaft zur Sprache, indem sie auf der Basis von Begriffsbildung und expliziertem Wissen absichtsvoll gestaltete, wohl bestimmte Funktionen verkörpern, die durch Menschen in deren Handlungskontext zu interpretieren sind, um sie wirkungsvoll zu gebrauchen (die funktionale ›Sprache‹ der Artefakte). Dabei sind die Wirkungen kraft der wohlbestimmten Funktionen durch die Eingaben determiniert. Um sinnvolle Eingaben machen und deren kausale Wirkungen interpretieren zu können, müssen Handlungen ihres Gebrauchs in der funktionalen ›Sprache der Artefakte‹ zum Ausdruck gebracht werden. Das gilt für alle technischen Artefakte, vom Faustkeil bis zum Computer.

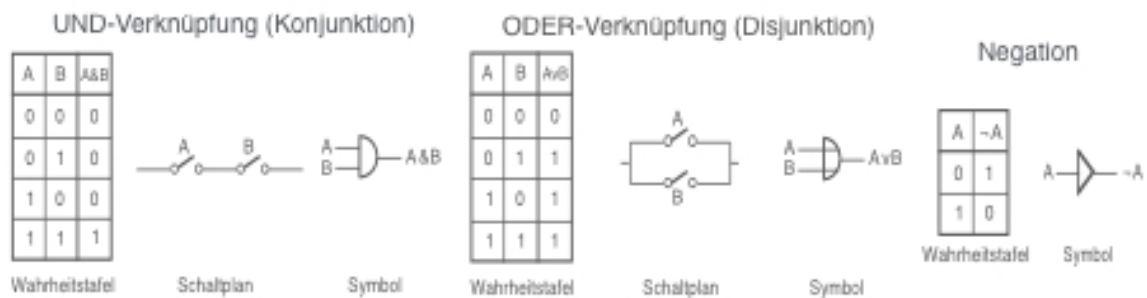
Auf der Grundlage dieser Unterschiede und Gemeinsamkeiten können nun die Besonderheiten der Funktionsweise von Computern als semiotischen Maschinen in Betracht gezogen und die Gründe für ihre verbreitete Mystifizierung aufgezeigt werden.

### 3.2 Wider die Mystifizierung des Computers

In der Computertechnik führen falsche Bezeichnungen und Metaphern vom Beginn an in die Irre. Seit ihren ersten Implementierungen am Ende des zweiten Weltkriegs bis in unsere Tage mit der Realisierung vermeintlich ›künstlicher Intelligenz‹ und ›maschinellen Lernens‹ mystifizieren sie die semiotische Maschine. Unzählige Berichte sprechen damals vom Computer als »Elektronengehirn«, wie einige ausgewählte Beispiele illustrieren: »30-Tonnen-Elektronengehirn an der Philadelphia Universität denkt schneller als Einstein« (Philadelphia Evening Bulletin 15.02.1946), »Giant Brains or Machines That Think« (Buchtitel von E.C. Berkeley 1949), »Can Man Build a Superman?« (Titel des Time Magazine mit Bild des Mark III vom 23.01.1950) oder »The Computer and the Brain« (Buchtitel von J.v. Neumann 1958).

Die Folgen dieser falschen Metaphorik, wie sie auch im äußerst wirkmächtigen sog. ›computational model of the mind‹ zum Ausdruck kommt, sind bis heute schwerwiegend: Damals konzipiert auch Turing (1950) mit der Konstruktion des »imitation game« seinen Turing-Test, der bis heute als das Maß aller Dinge gilt (wenn er auch zur Beurteilung der Frage »can machines think?« gänzlich wertlos ist). Gefangen im damaligen Zeitgeist des Behaviorismus wird damit vom äußerlich beobachtbaren Verhalten einer Maschine auf deren ›Intelligenz‹ geschlossen, wenn Menschen deren Verhalten nicht mehr von dem eines Menschen zu unterscheiden vermögen. Abgesehen davon, dass es sich hier um eine reine Zuschreibung handelt, ist menschliches Urteilsvermögen auch sehr leicht zu täuschen, wie etwa Weizenbaums (1966) Experiment mit dem Computerprogramm »Eliza« eindrücklich gezeigt hat, das Menschen für einen einfühlsamen Psychiater halten, obgleich es nur simple Antworten schematisch generiert. Gleichwohl laufen die vielen vergeblichen Versuche, ›künstliche Intelligenz‹ (›KI‹) zu definieren, im Kern darauf hinaus, in heillos zirkulärer Weise ›KI‹ zu bestimmen als »making a machine behave in ways that would be called intelligent if a human were so behaving« (so schon McCarthy 1955) bzw. »... to create systems that are capable of performing tasks commonly thought to require intelligence« (Autorengruppe 2018). Mit diesen oder ähnlich unsinnigen Definitionen gelingt es nicht einmal, ›KI‹-Systeme von konventionellen Computersystemen zu unterscheiden. Deren Verhalten verkörpert, wie bei Maschinen immer, stets lediglich das durch die natürliche Intelligenz ihrer Konstrukteure erst gewonnene Wissen und Verständnis zugrunde liegender Sachverhalte (und kann natürliche Intelligenz folglich nicht ›erklären‹). Erzählungen über ›künstliche Intelligenz‹ sind mithin reine Selbsttäuschung.

So bleibt, um dieser Art Mystifikation von Computersystemen zu entkommen, nichts weiter übrig, als sich ihrer tatsächlichen Funktionsweise zu vergewissern. Bekanntlich führen Computer nach dem ideellen Modell der Turingmaschine (Turing 1936) beliebige berechenbare Funktionen aus – und nichts sonst. Dazu besteht ihre Hardware aus binären Schaltsystemen (heute in Gestalt mikroelektronischer, hoch integrierter Prozessoren und Speicherchips hoher Taktfrequenz), auf deren Speicherzellen (›Registern‹) mittels elementarer Anweisungen programmgesteuert einfachste logische und arithmetische Operationen ausführbar sind. Benötigt werden dazu – basierend auf der theoretischen Grundlage der Booleschen Algebra – lediglich die Logikbausteine für Konjunktion, Disjunktion und Negation, die während eines Schalt-Takts auch die Addition beliebiger Dualzahlen realisieren (die Arithmetik mit Dualzahlen wird so auf logische Operationen der Booleschen Algebra zurückgeführt, veranschaulicht in Abb. 3). Auf dieser Basis lässt sich für die Ausführung beliebiger arithmetischer Operationen, letztlich auch mit negativen oder Gleitkomma-Zahlen, das anschließend beschriebene Rechnermodell einer minimalen abstrakten Maschine angeben, das diese Operationen durch wiederholte Ausführung nur ganz weniger elementarer Anweisungen ermöglicht. Zum Verständnis dieser Anweisungen ist noch wichtig, dass sich Logikbausteine in einem anfangs unbekanntem Zustand befinden, der jeweils nur geändert werden kann (durch das Symbol := markiert) und daher vor Gebrauch erst in einen definierten Zustand versetzt werden muss.



Ein binäres **Schaltssystem** aus diesen Logikgattern, das zwei n-stellige Dualzahlen korrekt addiert, besteht aus insgesamt einem Halbaddierer und n - 1 Volladdierern:

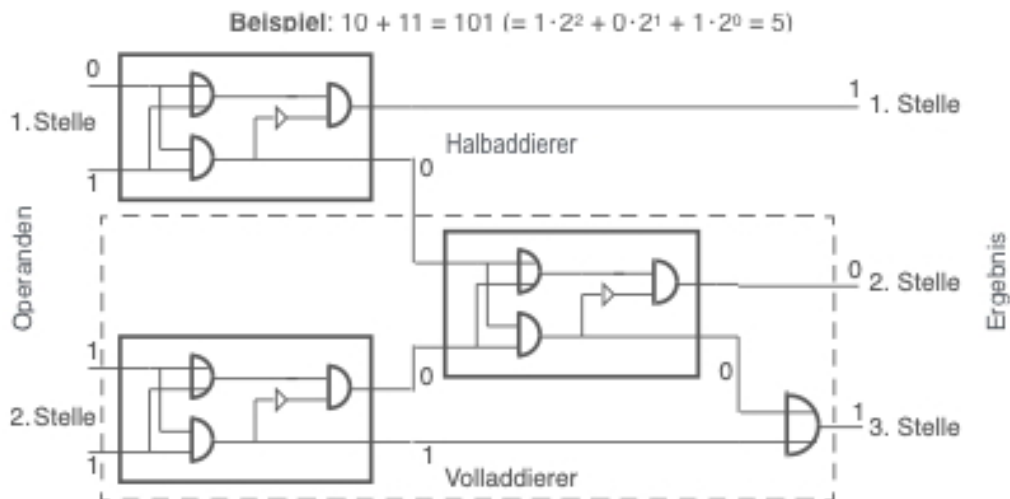


Abb. 3: Elementare Logikbausteine und damit realisiertes Schaltssystem zur Addition von Dualzahlen (Brödner 1997)

Das Modell einer minimalen abstrakten Maschine ist gemäß Rekursionstheorie gleichmächtig wie die Turingmaschine, mithin wie diese zur Ausführung beliebiger berechenbarer Funktionen (Algorithmen) in der Lage (und setzt wie diese beliebig große Daten- und Programmspeicher für natürliche Zahlen inklusive der 0 voraus). Dazu genügen vier elementare Anweisungen:

- $x := 0$                       Löschen des Inhalts einer Speicherzelle,
- $x := x + 1$                 Erhöhen des Inhalts einer Speicherzelle um 1,
- $x := x - 1$                 Erniedrigen des Inhalts einer Speicherzelle um 1,
- while  $x \neq y$  do ... end    Rekursion: Die While-Schleife wird solange durchlaufen bis  $x = y$ .

Damit lassen sich weitere, darauf aufbauende Anweisungen konstruieren; Beispiele:

- Laden einer Speicherzelle mit einer Konstanten n ( $x := n$ ) durch Iteration:  
 $x := 0$ ;  $x := x + 1$ ; ...;  $x := x + 1$  (x wird n-mal um 1 erhöht),
- Zuweisen einer Variablen y an eine Variable x ( $x := y$ ) mittels Rekursion:  
 $x := 0$ ; while  $x \neq y$  do  $x := x + 1$  end,
- Erhöhen einer Variablen x um eine Variable y ( $x := x + y$ ) mittels der Hilfsvariablen h durch Rekursion:  
 $h := 0$ ; while  $h \neq y$  do  $x := x + 1$ ;  $h := h + 1$  end,
- Addition ( $z := x + y$ ) mittels Variablenzuweisung ( $z := x$ ) und -erhöhung ( $z := z + y$ ),
- Boolesche und Relationsausdrücke lassen sich in arithmetische transformieren mit 1 für *true* und 0 für *false*.

Darin zeigt sich ein in der Computertechnik durchgängig genutztes Konstruktionsprinzip, das aus eindeutig bestimmten und vielfach erprobten einfachen Berechnungsfunktionen komplexere und reichhaltigere Funktionen zu bilden erlaubt (z.B. für Suchen, Sortieren, Bildverarbeitung oder Datentransfer u.v.a. mehr). Somit beruhen auch die komplexesten Programme und Softwaresysteme letztlich stets auf der Ausführung der hier beschriebenen Elementarfunktionen, allerdings operativ ausgeführt auf immer leistungsfähigeren Schaltsystemen, inzwischen in Milliarden Takten pro Sekunde. Das ist im Kern das ganze ›Geheimnis‹.

Bezeichnenderweise wurden fast alle dieser theoretisch-konzeptionellen Grundlagen der Computertechnik bereits im Zuge der Industrialisierung im 19. Jhdt. entwickelt, zuletzt im Verlauf der Grundlagenkrise der Mathematik in den 1930er Jahren, jedenfalls lange bevor die ersten Computer physisch realisiert werden konnten (vgl. nachstehende Übersicht). Daher ist auch die derzeit verbreitete Rede von der »vierten industriellen Revolution« geschichtsvergessener und ignoranter Unfug. Die seither erfolgte Entwicklung beruht denn auch weitgehend auf ständigen Verbesserungen der Hardware und im Software Engineering.

#### *Historische Übersicht zu den Grundlagen der Computertechnik*

- 1792-1801 Gaspard de Prony entwickelt und nutzt ein formularbasiertes Verfahren zur extrem arbeitsteiligen Neuberechnung mathematischer Tafeln (als Basis-Werkzeug für Ingenieurarbeit) im Dezimalsystem; das Formularschema der Rechenoperationen bildet die Urform eines Algorithmus (noch im WK II werden z.B. Tragwerke, V2-Flugbahnen u.v.a. so berechnet).
- 1805 Jacquard-Webstuhl, erste digital gesteuerte Arbeitsmaschine (mit Lochbrettern).
- 1812 Charles Babbage konzipiert die »Difference Engine« zur einfachen Berechnung von Polynomen:  $f(x) = a_n x^n + \dots + a_1 x + a_0$  (1822 prototypisch realisiert).
- Um 1830: Charles Babbage entwirft und programmiert die »Analytical Engine«; sie nimmt die von-Neumann-Architektur programmierbarer Universalrechner (Prozessor - Speicher - Steuerung) vorweg (scheitert aber an der Mechanik).
- 1847: Die de Morganschen Gesetze [ $\neg(a \wedge b) = \neg a \vee \neg b$  und  $\neg(a \vee b) = \neg a \wedge \neg b$ ] aufgreifend publiziert George Boole einen Logikkalkül (um 1888 von G. Peano als »Boolesche Algebra« axiomatisiert); er bildet das logisch-funktionale Fundament für binäre Schaltsysteme (heutige Computer-Hardware, s. oben).
- 1860-1880: C.S. Peirce entwickelt erstmals einen Prädikatenkalkül 1. Stufe, arbeitet an »logischen Maschinen« und entwickelt eine triadische Zeichentheorie, ohne die der allg. Computereinsatz nicht zu verstehen ist (äquivalent: G. Freges »Begriffsschrift« 1879; s. unten »algorithmisches Zeichen«)
- 1931: Kurt Gödel beweist u.a. unter Verwendung rekursiver Funktionen die Unvollständigkeit formaler Systeme wie das der *principia mathematica* von B. Russell & A.N. Whitehead.
- 1936: Alan Turing publiziert das ideelle Modell der »Turingmaschine«, definiert damit formal die Begriffe Algorithmus und berechenbare Funktion (äquivalent:  $\lambda$ -Kalkül von A. Church & S. Kleene).
- 1941/45: Konrad Zuse entwickelt die Z3 als ersten binären Rechner (Relaistechnik) und den Plankalkül als erste (quasi-funktionale) Programmiersprache (markiert die Geburt des modernen Computers).

Damit stellt sich die entscheidende Frage, die letztlich über Wohl und Wehe des Einsatzes von Computersystemen bestimmt: Wie gelangt man von der sozialen Praxis in der Welt der Wissensarbeit und Wertschöpfung, ihren zumeist impliziten Handlungs- und Entscheidungsregeln, zu den Turing-berechenbaren bzw. partiell rekursiven Funktionen, die allein auf binären Schaltsystemen ausführbar sind? Und wie gelangt man durch deren Anwendung wieder zurück zu Wissensarbeit höherer Leistung? Bei der Analyse dieser Übergangsprozesse stellt sich heraus, dass die vermeintliche »Macht der Algorithmen« tatsächlich die Macht der Interessen und Perspektiven ihrer Konstrukteure ist (bzw. der Einflüsse, denen diese ihrerseits im Kapitalverhältnis unterliegen).

Seit jeher erfordert nämlich die Gestaltung von Computersystemen für den praktischen Gebrauch eine sorgfältige und aufwendige Modellierung der stets zeichenbasierten sozialen Praktiken von kognitiver bzw. Wissensarbeit. Die der Wissensarbeit zugrunde liegenden Zeichenprozesse (»Semiosen«) müssen in einer zweckorientierten Perspektive sachgerecht modelliert werden durch partielle Explikation der tatsächlichen sozialen Praktiken in Gestalt begrifflichen Wissens über deren Gegenstände, Strukturen und Abläufe. Wesentliche theoretische Grundlage dafür ist die triadische Zeichentheorie von C.S. Peirce (1983), die Zeichen sozialer Praxis als dreistellige Relation ((R – O) – I) von physischem Zeichenkörper (»Repräsentamen« R), bezeichnetem Objekt (O) und Bedeutung zuweisendem Begriff (»Interpretant« I) auffasst. Als eigene soziale Praxis unterliegt auch die Modellbildung – die den eigentlichen Kern des Software Engineering ausmacht – stets den Interessen und Machteinflüssen beteiligter Akteure und durchläuft folgende Schritte der Abstraktion und Formalisierung (Andelfinger 1997):

- **Semiotisierung:** Die zunehmend präzisierete Beschreibung der betrachteten sozialen Praxis mittels Zeichen liefert ein perspektivisch reduziertes Abbild derselben als Ergebnis gemeinsamer Reflexion und Kommunikation der Akteure (Sprachanalyse, →»Ontologie«). Ergebnis ist ein im wesentlichen sprachlich artikuliertes *Anwendungsmodell*.
- **Formalisierung:** Dessen Formalisierung durch Abstraktion von situations- und kontextgebundenen Bedeutungen mittels Verwendung standardisierter Zeichen und Operationen, durch Festlegung relevanter Bedingungen und Funktionen (funktionale Spezifikation) liefert ein *formales Modell*.
- **Algorithmisierung:** Die Überführung von Objekten, Abläufen und Funktionen des formalen Modells in auto-operational ausführbare Prozeduren in Form von Daten und berechenbaren Funktionen (Algorithmen) liefert schließlich das *Berechnungsmodell*.

Die berechenbaren Funktionen des in ausführbare Software transformierten Berechnungsmodells operieren auf Daten als auf rein syntaktische Aspekte reduzierten »Quasi-Zeichen« (Nöth 2002), deren binär kodierte »Repräsentamen« sie per Anweisung verändern, ohne Ansehen ihrer Bedeutung. Im Computersystem implementiert bilden sie »auto-operationale Formen« als Ausdruck abstrakter, formalisierter Handlungen (Floyd 2002). Auf diesem Wege der Modellierung und Formalisierung kann mittels Computersystemen »Kopfarbeit maschinisiert« werden (Nake 1992), wenigstens partiell (übrigens verhalten sich Menschen ebenfalls wie eine Maschine, wenn sie ein Berechnungsverfahren wie etwa die schriftliche Division ausführen). Der Sinn der dabei ausgeführten »auto-operationalen Formen« muss freilich durch Aneignung für den praktisch wirksamen Gebrauch erst wieder erschlossen werden. So werden die Computersysteme durch Interpretation ihrer Funktionen im Handlungskontext der Arbeitspraxis, wieder in einen – freilich eben dadurch veränderten – Praxiszusammenhang gestellt (vgl. Abb. 4).

In der Interaktion mit Computern werden von Benutzern Zeichen (für Daten und damit operierende Funktionen) eingegeben, die für sie im jeweiligen Handlungskontext gewohnte Bedeutung tragen. Innerhalb des Computersystems werden diese außen sinnvoll interpretierbaren Zeichen auf binäre Signale als deren physischen Verkörperungen (R) reduziert, die mittels Programm nach vollständig festgelegten Anweisungen – dem Algorithmus – verarbeitet werden. Das mithin kausal

determinierte Resultat dieser Signalverarbeitung kann dann bei Erscheinen an der Systemoberfläche erneut als Zeichen interpretiert werden. Über das mittels Kodierung beiden Zeichenprozessen gemeinsame Repräsentamen R auf der Benutzungsoberfläche sind diese fest gekoppelt. Intern verarbeitete Signale werden anstelle der intentionalen Interpretation durch Benutzer durch die Anweisungen des Algorithmus als kausalem Interpretanten determiniert, dessen Ergebnis  $R'$  das so kausal bestimmte Objekt bildet. Genau dies ermöglicht außen eine sinnvolle Interpretation (Abb. 4). So ist Interaktion mit Computern gekennzeichnet durch kausale Determination sinnfreier Signalverarbeitung im Innern und durch sinngebende Interpretation der an der Oberfläche als Zeichen gedeuteten Signale außerhalb. Der soziale Raum der Zeichenprozesse wird dabei nicht verlassen (weshalb Computer zurecht auch als semiotische Maschinen bezeichnet werden; Nake 2001).

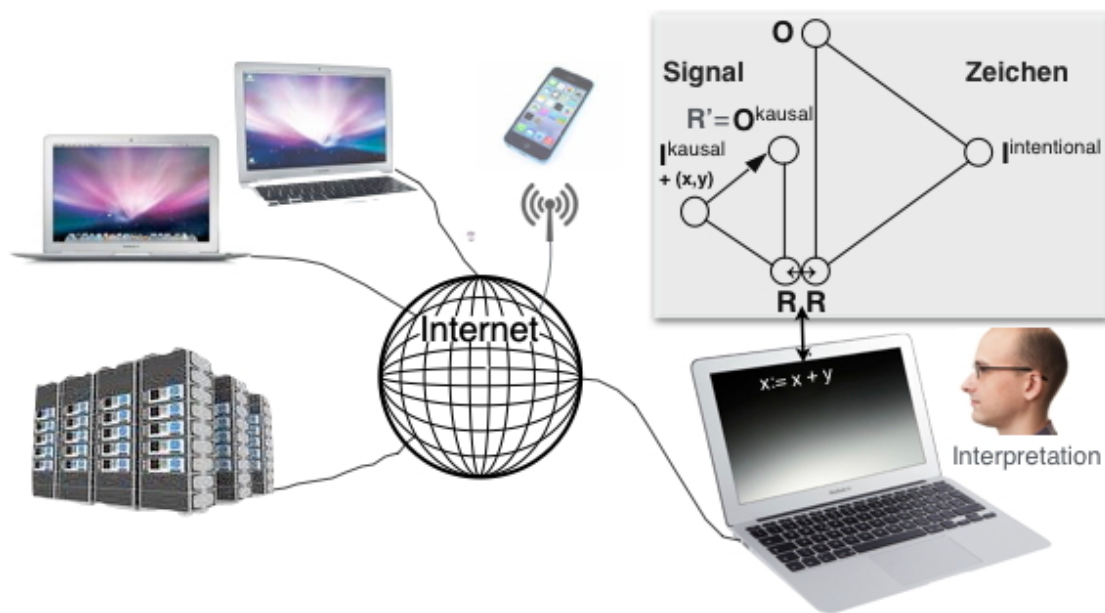


Abb. 4: »Algorithmisches Zeichen« (Nake 2001): Als Einheit von Signal und Zeichen in vernetzten Computersystemen vermittelt es zwischen Signal und Sinn

Damit wird auch die eigentlich neue Qualität deutlich: Verglichen mit bisherigen Medien handelt es sich bei global vernetzten Computersystemen um ein *instrumentelles Medium*, das in sich interaktiv nutzbare Werkzeuge zum Herstellen, Bearbeiten, Speichern und Auffinden beliebiger zeichenbasierter Gegenstände mit der Möglichkeit beliebigen Datentransfers über das Netzwerk zu anderen Teilnehmern vereint (Denning 2003). So können durch dieses integrierte Medium, im Zusammenspiel von Berechnung, Interaktion und Verbindung und unter Beachtung passender Standards und Handlungsvereinbarungen, weltweit räumlich verteilte, virtuelle Arbeitsräume computervermittelter Kooperation (CSCW; vgl. Schmidt 2011) sowie virtuelle Bibliotheken zur Nutzung kodifizierter Wissensbestände (WWW, verteilter Hypertext) geschaffen werden. Welchen gesellschaftlichen Nutzen das qualitativ neue Medium hat, entscheidet sich jedoch, wie bei anderen Medien auch, v.a. durch die Regeln, die sich eine Gesellschaft für den Umgang damit gibt.

Abschließend ist hier noch anzumerken, dass viele Computer nicht in Organisationen, sondern als Steuerungen für energetische, stoffliche oder logistische Prozesse eingesetzt werden. Mittels Sensoren zur Messung relevanter Prozessgrößen und Aktoren (Stellglieder) vermögen Steuerungen gezielt in Energie- und Stoffströme der Prozesse einzugreifen, aufgrund ihrer Algorithmen maschinelle Operationen zu koordinieren und damit den Prozessen ein zweckgemäß gewünschtes Verhalten zu verleihen, mit denen sie dann zu einem sog. »cyber-physischen System« (CPS)

verschmelzen (daher werden sie auch »embedded systems« genannt). So können Steuerung und Prozess in gewünschter Weise zusammenwirken, um im Ergebnis Qualität und Mengenleistung der Prozesse zu steigern. Dies gelingt aber nur, weil von dem zu steuernden Prozess zuvor ein hinreichend genaues mathematisches oder heuristisches Modell in Zeichenform erstellt wurde, mittels dessen das Berechnungsmodell der Steuerung entworfen werden kann. Eben dies macht die Steuerung selbst ebenfalls zur semiotischen Maschine (sie verarbeitet Signale, aber keine Kräfte). Mithin können aus dieser Perspektive Computer auch als universelles Steuerungspotential verstanden werden, das jeweils per Programm als prozessspezifische Steuerung eingerichtet wird. Nach diesen Einblicken in die tatsächliche Funktionsweise von Computern bleibt noch die Frage zu klären, was aus diesen Einsichten für die Erklärung des Produktivitätsparadoxons für eine künftig verbesserte Praxis geschlossen werden kann.

#### 4 Schlussfolgerungen: Erklärung des Produktivitätsparadoxons und Perspektiven der Besserung

Wie aus der bisherigen Darstellung hervorgeht, wird mit Software massiv in soziale Praktiken organisierter Wissensarbeit interveniert. Durch Beobachtung und begriffliche Explikation vorgefundener Praktiken, v.a. der ihnen innewohnenden impliziten Regeln und Entscheidungslogik (>rationale<) wird zunächst kodifiziertes Wissen über diese generiert. Auf dieser Basis kann zweckorientiert und interessengeleitet die zum Einsatz eines lauffähigen Softwaresystems führende Modellbildung und Formalisierung vorangetrieben und letztlich als nutzbares Computersystem implementiert werden. Um dessen Funktionen aber tatsächlich praktisch wirksam nutzen zu können, müssen diese freilich erst durch die Nutzer mühsam angeeignet werden, bevor sie routiniert verwendet werden können (vgl. Abb. 5). Bei komplexen Systemen, die verbundene Aufgaben vieler Wissensarbeiter betreffen, kann dieser Aufwand den für die eigentliche Entwicklung der Software bei weitem übersteigen (wie die einzelwirtschaftlich orientierte Empirie zum Paradoxon zeigt).

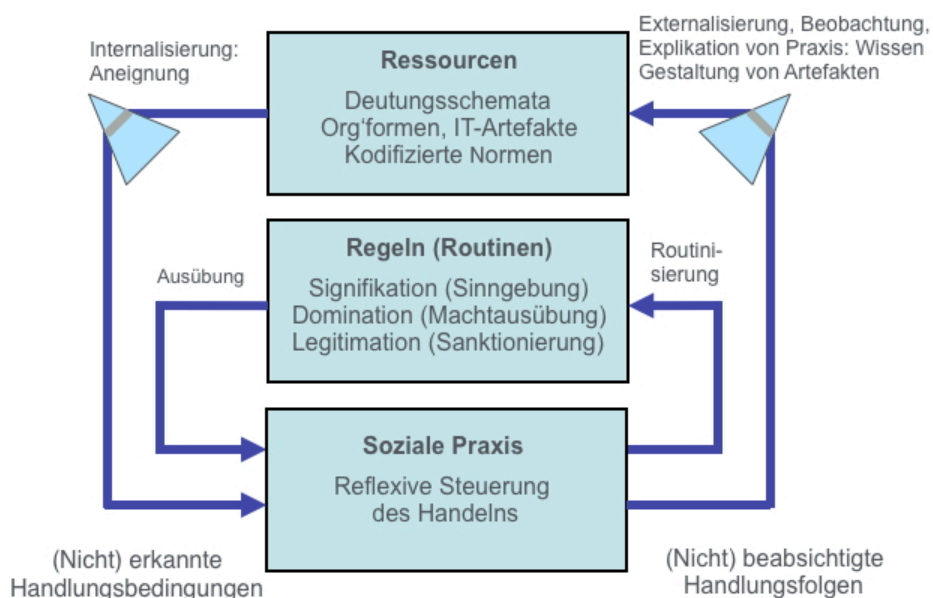


Abb. 5: Praxistheorie: Rekursive Konstitution von Handeln und Struktur (eigene Darstellung)

Gestützt auf diese grundlegenden praxistheoretische Erkenntnisse (vgl. die Übersicht bei Reckwitz 2003; im einzelnen z.B. Orlikowski 2000), ist im vorliegenden Zusammenhang von großer Bedeutung zu erkennen, dass die Tätigkeiten, die im Kreislauf von Modellierung und

Formalisierung, über den Einsatz als Computersystem bis zur Aneignung von dessen Funktionen vollzogen werden, insgesamt die organisationalen Praktiken stark verändern. Dabei ist zu reflektieren, dass sowohl Modellbildung und Implementierung des Computersystems als auch die Aneignung von dessen Funktionen für die Nutzung jeweils kreative, zweckorientierte und interessen geleitete Tätigkeiten sind, deren Verlauf und Ergebnis nicht vorherzusehen sind. Zudem ist dieser Kreislauf in hohem Maße *selbstbezüglich*, indem er maschinelle Funktionen für eine Praxis zu schaffen anstrebt, die am Ende eine andere ist als die, für die sie konzipiert wurden. Soziale Praktiken der Wissensarbeit als Gegenstand der Modellierung geraten durch den Vorgang des Modellierens in Bewegung.

Damit erweist sich der Entwicklungsprozess der Software zugleich auch als Prozess der Organisationsentwicklung: Software ist Orgware, ein Medium des Organisierens. Dessen inhärente Selbstbezüglichkeit stellt die an der Entwicklung beteiligten und von ihr betroffenen Akteure vor enorme Herausforderungen. Zunächst lassen sich die Gebrauchstauglichkeit der Systemfunktionen wie auch die produktiven Wirkungen überhaupt nur im praktischen Einsatz, d.h. nach deren Aneignung für die eben dadurch aber veränderte Praxis der Wissensarbeit, angemessen beurteilen. Darüber hinaus gilt es im Verlauf des Prozesses möglichst früh die wechselseitige Ignoranz der Hauptakteure – das Unverständnis der Software-Ingenieure für die eigentlichen Aufgaben der Wissensarbeiter wie deren mangelndes Verständnis für die Möglichkeiten der Modellierung und Formalisierung – zu überwinden. Schließlich ist seitens des Projektmanagements dem Umstand Rechnung zu tragen, dass sich infolge der Selbstbezüglichkeit Anforderungen immer wieder ändern (Brödner 2008, Rohde et al. 2017).

So ergibt sich aus der semiotischen Natur von Computersystemen und ihrer Eigenschaft als Medium des Organisierens, dass Modellierung, Einführung oder auch Anpassung eines komplexen Softwaresystems nur gelingen können, wenn sie als integraler Teil eines umfassenderen Prozesses der Organisationsentwicklung verstanden und organisiert werden. Bislang noch zu oft praktizierte Vorgehensweisen, die sich sequentiell auf eine einmalige, umfassende Anforderungsanalyse mit nachfolgenden Entwicklungs- und Einführungsphasen gründen, sind dabei von vornherein zum Scheitern verurteilt. Vielmehr erfordert der Umstand der sozialen Einbettung semiotischer Maschinen ein *partizipativ und reflexiv angelegtes, evolutionäres Vorgehen* mit kurzen, überschaubaren Revisionschleifen, in dem wiederholt Zyklen der Anforderungsanalyse, Softwareentwicklung, Implementation, Erprobung sowie der *formativen Evaluation* erreichter Resultate durchlaufen werden. In einer solchen Entwicklungsspirale werden die Systemfunktionalität und deren Aneignung im Rahmen restrukturierter Prozesse in jeweils kleinen, bewusst begrenzten Entwicklungszyklen hervorgebracht und so wiederholt der Bewertung und Einflussnahme unterworfen. So bleiben Anforderungen und Entwicklungsaufgaben der einzelnen Zyklen überschaubar und halten die Risiken des Scheiterns in Grenzen (ggf. ist nur der letzte Zyklus zu revidieren). Dabei sind bewährte Methoden des Software Engineering zu nutzen, reichen aber allein nicht aus. Vielmehr müssen sich die Beteiligten dabei insgesamt über alle Aspekte ihrer im Entstehen begriffenen neuen sozialen Struktur verständigen, was insbesondere die produktive Verbindung verschiedener Sichtweisen, die Bewältigung von Konflikten und den Ausgleich unterschiedlicher Interessen einschließt.

Dazu ist es erforderlich, die wechselseitige Ignoranz der Akteure zu überwinden, indem die praktische Erfahrung aus der wirklichen Arbeit, bereits expliziertes Prozesswissen und das Wissen über technische Möglichkeiten und Grenzen von Computersystemen produktiv miteinander verbunden werden. Dazu bedarf es hinreichender Analyse der Praxis, etwa mit ethnografischen Methoden, und der Initiierung eines Kommunikations- und Lernprozesses, in dem die Beteiligten unterschiedliche Perspektiven zusammenführen, ein geteiltes Verständnis des Arbeitsprozesses



entwickeln und dabei eine geteilte ›Sprache der Artefakte‹ nutzen. Darin lassen sich explizites Wissen über den sich ändernden Arbeitsprozess und die darin zu nutzenden Softwareartefakte artikulieren und gemeinsam reflektieren. Dabei müssen sich die Akteure zunächst auch über Ziele, Aufgaben und Grundsätze ihrer künftigen Zusammenarbeit – über ihre Geschäftsstrategie – sowie über dazu passende Strukturen und Abläufe der Arbeits- und Wertschöpfungsprozesse verständigen, um daraus abzuleiten, wie diese durch Programme und Daten wirksam unterstützt werden können. Wer Softwaresysteme adäquat gestalten und produktiv nutzen will, muss Organisationsentwicklung betreiben, unter Beteiligung aller betroffenen Akteure von Beginn an.

Vor diesem Hintergrund ist auch leicht nachzuvollziehen, dass sich derzeit sog. ›agile Methoden‹ des Software Engineering (vgl. Hanser 2010) zunehmend verbreiten, denen ein derartiges zyklisch evolutionäres Entwicklungsmuster zugrunde liegt und – sofern sie der dargelegten Herausforderungen eingedenk auch konsequent umgesetzt werden – auch beträchtliche Erfolge verzeichnen. Allerdings wird diese notwendige Bedingung oft nicht eingehalten und durch Rückfall in alte gewohnte Praktiken werden mögliche Erfolge verschenkt. Allerdings stellen die einschneidenden, aber notwendigen Veränderungen das Management vor ein schwer zu bewältigendes Dilemma: Derart weit reichende kollektive Lernprozesse sind grundsätzlich nach Verlauf und Ergebnis offen und bedeuten daher aus herkömmlicher Managementsicht einen kaum erträglichen Kontrollverlust. Andererseits sind sie den hier angestellten Überlegungen zufolge notwendige – wenngleich auch nicht hinreichende – Bedingung für den Projekterfolg. So ist traditionell denkendes Management hin- und hergerissen zwischen dem Risiko eines mit hoher Wahrscheinlichkeit scheiternden Organisations-Entwicklungsprojektes und der Befürchtung eines riskanten Kontrollverlusts über den Projektverlauf. Genau hierin ist auch der Grund auszumachen, warum – trotz wiederkehrender Erfahrungen des Scheiterns – so wenige Manager wagen, Projekte zur Organisationsentwicklung reflexiv und evolutionär anzugehen. Das Dilemma wird jedoch drastisch gemildert, wenn man die Prozedur einer zyklischen formativen Projektevaluation ernst nimmt und sich auf diese neue Form der Kontrolle durch reflexive Steuerung einlässt.

Aus der sozialen Einbettung von Softwaresystemen und deren engem Verwobensein mit den sozialen Praktiken von Organisationen ergibt sich ferner, dass Produktivitätsfortschritte nicht aus erhöhter Leistungsfähigkeit der Hardware, sondern letztlich vor allem aus der Reorganisation der zugrunde liegenden Geschäftsprozesse und der dabei benötigten Wissensarbeit entstehen können. Eben hierin unterscheiden sich semiotische Maschinen grundsätzlich von klassischen Maschinen der Energie- und Stoffumwandlung, die zwecks Produktivitätssteigerung zusätzlich Naturkräfte und -effekte erschließen und nutzen können.

Selbst wenn viele einzelne Aufgaben der Wissensarbeit durch Implementierung umfangreicher Berechnungsverfahren – etwa bei der Berechnung der Festigkeit oder Dynamik mechanischer Strukturen (FEM), der Steuerung von komplexen logistischen Prozessen oder der Simulation von Wettervorgängen – weitgehend automatisiert werden können, wird die dadurch eingesparte Arbeit durch Rebound-Effekte vermehrter oder vielfacher Anwendung zumeist wieder kompensiert. Nicht selten stehen aber auch algorithmisch festgelegte Prozeduren bei der Bewältigung von unvorhergesehenen oder Ausnahmesituationen eher als Hindernis im Wege und erfordern zusätzliche Arbeit sie zu umgehen. Zudem ist zu berücksichtigen, dass sowohl die Modellierung und Formalisierung bis hin zur Konstruktion der Algorithmen der Berechnungsmodelle als auch die Aneignung von deren Funktionen für die praktisch wirksame Nutzung und Interpretation beträchtlichen Aufwand erfordern. Gerade in diesen arbeitsaufwendigen Prozessen wird über Erfolg oder Misserfolg des Einsatzes von Softwaresystemen entschieden. Umfangreiche zusätzliche Wissensarbeit wird darüber hinaus auch für die fortlaufende Entwicklung und Vermittlung von Methoden des Software Engineering und der Berechnungsverfahren benötigt. Zusammengenommen erklärt das recht gut das Ausbleiben nennenswerter gesamtwirtschaftlicher Produktivitätseffekte.

Schließlich wird damit auch verständlich, warum das unreflektierte Gerede von ›Digitalisierung‹, insbesondere von ›künstlicher Intelligenz‹ und ›maschinellern Lernen‹, die eigentlichen Herausforderungen weithin ignoriert und die wirklichen Vorgänge beim Einsatz fortschrittlicher Computersysteme nicht angemessen beschreibt. Die verwendeten Metaphern führen direkt ins Reich der Mythen und verbergen – wie beim trickbasierten Zaubern – die eigentlich banale Funktionsweise auf doppelte Weise: das, was versteckt wird, und den Vorgang des Versteckens selbst. So verbergen diese Metaphern, dass die ganze Intelligenz jeweils in den Tätigkeiten der Modellierung und Formalisierung sozialer Praxis und der Aneignung der Softwarefunktionen für den praktisch wirksamen Gebrauch steckt, während das Computersystem lediglich die Algorithmen des so gewonnenen Berechnungsmodells ausführt. So laufen gebräuchliche Verfahren ›maschinellen Lernens‹ (etwa Verfahren mit Stützvektoren oder Entscheidungsbäumen, künstliche neuronale Netze, K-Means Clustering oder lineare Regression) im Kern auf bloße Funktions-Approximation an vorhandene Daten hinaus (das hat mit ›Lernen‹ im herkömmlichen Sinn nichts zu tun). Dabei werden meist alt bekannte mathematische Methoden, ausgeführt als eine Art ›Hochgeschwindigkeits-Statistik‹, auf extrem schneller Hardware eingesetzt – allerdings mit stets nur wahrscheinlichen und daher unsicheren Ergebnissen, zudem oftmals auf Basis fragwürdiger Qualität der zur Anpassung benutzten Daten. Zugleich bleibt der Einsatz der Verfahren auf den Problemtyp beschränkt, für den sie geschaffen sind. Mithin sind diese Metaphern ein irreführender Etikettenschwindel.

Der Erfolg des Einsatzes fortschrittlicher Computersysteme hängt folglich entscheidend von der analytischen und methodischen Kompetenz der Entwickler und dem Verständnis der Nutzer ab, sie für ihre Wissensarbeit sinnvoll zu verwenden. Für die künftige Entwicklung dieser Systeme ist dementsprechend ein grundlegender Perspektivwechsel angesagt: von einem daten- und methodengetriebenen Vorgehen zu problemorientierter Herangehensweise. Statt zu fragen, wie oft nur zufällig verfügbare Daten und Methoden der Modellierung sozialer Praxis auf unterschiedliche Aufgaben der Wissensarbeit angewandt werden können, um sie durch Automaten zu ersetzen, ist es weit zielführender, praktisch relevante Probleme der Steigerung von Flexibilität und Produktivität von Wissensarbeit organisatorisch anzugehen, sie bei der Bewältigung ggf. auch, aber nicht nur, mit methodisch sorgfältig entwickelten Berechnungsmodellen effektiv zu unterstützen. Erforderlich ist nichts weniger als ein Wandel im Verständnis und der Perspektive von Entwicklung und Gebrauch der Computertechnik, der Perspektivwechsel vom Automatisierungsmittel zum Medium des Organisierens, methodisch von der Systemgestaltung zur Strukturierung sozialer Praxis. Allein auf diesem Wege können Computersysteme auch künftig zur effektiven Bewältigung gesellschaftlicher Herausforderungen beitragen (z.B. zu sozial-ökologischer Transformation). Das belegt jedenfalls die ganze bisherige Entwicklung, sowohl durch die Erfolge bei der Steigerung der Leistung computerunterstützter Wissensarbeit als auch durch die Sackgassen misslungener Automatisierung, während entgegen stehende Behauptungen lediglich dem Fetisch des Computers als ›intelligenter Maschine‹ huldigen.

## Literatur

Accenture (2015): Economic Literature Review: Impact of Technology on Productivity / Labor Productivity, <https://pdfs.semanticscholar.org/1c85/219b3f962cb8f9884ec93a4c8e2556ed5fc7.pdf>

Acemoglu, D.; Autor, D.; Dorn, D.; Hanson G.H. & Price, B. (2014): Return of the Solow Paradox? IT, Productivity, and Employment in US Manufacturing, *American Economic Review: Papers & Proceedings* 104 (5), 394–399

Andelfinger, U. (1997): Diskursive Anforderungsanalyse. Ein Beitrag zum Reduktionsproblem bei Systementwicklungen in der Informatik, Frankfurt/M: Peter Lang

Anders, G. (1973): Die Antiquiertheit des Menschen. Bd. 1, München: Beck

- Autorengruppe (2018): *The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation*, Oxford: Future of Humanity Institute u.a. 02/2018, <https://arxiv.org/pdf/1802.07228.pdf>
- Babbage, C. (1832): *Die Ökonomie der Maschine*, Nachdruck der Originalübersetzung von 1833, hg. von P. Brödner, Berlin: Kulturverlag Kadmos 1999
- Boes, A.; Kämpf, T.; Langes, B. & Lühr, T. (2014): Informatisierung und neue Entwicklungstendenzen von Arbeit, *Arbeits- und Industriesoziologische Studien* 7 (1), 5-23
- Bresnahan, D.F.; Brynjolfsson, E. & Hitt, L.M. (2002): Information Technology, Workplace Organization, and the Demand for Skilled Labor: Firm-level Evidence, *The Quarterly Journal of Economics* 117 (1), 339–376
- Brödner, P. (2008): Das Elend computerunterstützter Organisationen, in: D. Gumm (Hg.): *Mensch – Technik – Ärger? Zur Beherrschbarkeit soziotechnischer Dynamik aus transdisziplinärer Sicht*, Münster: Lit-Verlag, 39-60
- Brödner, P. (1997): *Der überlistete Odysseus. Über das zerrüttete Verhältnis von Menschen und Maschinen*, Berlin: edition sigma
- Brynjolfsson, E. (2003): The IT Productivity Gap, *Optimize*, Issue 21, July 2003
- Brynjolfsson, E. (1993): The Productivity Paradox of Information Technology, *CACM* 36 (12), 67-77
- Brynjolfsson, E. & Hitt, L.M. (2000): Beyond Computation: Information Technology, Organizational Transformation and Business Performance, *Journal of Economic Perspectives* 14 (4), 23-48
- Dedrick, J.; Gurbaxani, V. & Kraemer, K.L. (2003): Information Technology and Economic Performance: A critical review of the empirical evidence, *ACM Computing Surveys* 35 (1), 1-28
- Denning, P.J. (2003): Great Principles of Computing, *CACM* 44 (11), 15-20
- Emam, K.E. & Koru, A.G. (2008): A Replicated Survey of IT Software Project Failures, *IEEE Software* 25 (5), 84-90
- Floyd, C. (2002): Developing and Embedding Autooperational Form, in: Y. Dittrich; C. Floyd & R. Klischewski (Eds.): *Social Thinking – Software Practice*, Cambridge (MA): MIT Press, 5-28
- Gordon, R.J. (2016): *The Rise and Fall of American Growth. The U.S. Standard of Living since the Civil War*, Princeton: Princeton University Press
- Gordon R.J. (2014): *The Demise of U.S. Economic Growth: Restatement, Rebuttal, and Reflections*, NBER Paper
- Haug, W. F. (2005): *Vorlesungen zur Einführung ins »Kapital«*, Hamburg: Argument
- Hanser, E. (2010): *Agile Prozesse: Von XP über Scrum bis MAP*, Berlin Heidelberg: Springer
- Jorgenson, D.W.; Ho, M.S. & Stiroh, K.J. (2008): A Retrospective Look at the U.S. Productivity Growth Resurgence, *Journal of Economic Perspectives* 22 (1), 3-24
- Kretschmer, T. (2012): *Information and Communication Technologies and Productivity Growth: A Survey of the Literature*, OECD Digital Economy Papers, No. 195, Paris: OECD Publishing
- Landauer, T.K. (1995): *The Trouble with Computers. Usefulness, Usability, and Productivity*, Cambridge (MA): MIT Press
- Lutz, B. (1987): Das Ende des Technikdeterminismus und die Folgen. Soziologische Technikforschung vor neuen Aufgaben und neuen Problemen, in: Lutz, B. (Hg.): *Technik und sozialer Wandel. Verhandlungen des 23. Deutschen Soziologentages in Hamburg 1986*, Frankfurt/M: Campus
- Maucher, I. (2001): *Komplexitätsbewältigung durch Entwicklung und Gestaltung von Organisation* München: Hampp
- Maucher, I. (Hg.) (1998): *Wandel der Leitbilder zur Entwicklung und Nutzung von PPS-Systemen*. München: Hampp
- McKinsey Global Institute (2002): *How IT Enables Productivity Growth. The US experience across three sectors in the 1990s*, San Francisco: MGI

- Nake, F. (2001): Das algorithmische Zeichen, in: W. Bauknecht; W. Brauer & T Mück (Hg.): Informatik 2001. Tagungsband der GI/OCC Jahrestagung, 736-742
- Nake, F. (1992): Informatik und Maschinisierung von Kopfarbeit, in: W. Coy; F. Nake; J.-M. Pflüger; A. Rolf; J. Seetzen; D. Siefkes & R. Stransfeld (Hg.): Sichtweisen der Informatik, Braunschweig Wiesbaden: Vieweg, 181-201
- Naur, P. & Randell, B. (eds.) (1968): Software Engineering. Report on a conference sponsored by the NATO Science Committee, Brussels: NATO
- Nöth, W. (2002): Semiotic Machines, Cybernetics and Human Knowing 9 (1), 5-22
- Orlikowski, W. J., 2000: Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations, Organization Science 11(4), 404-428
- Peirce, C.S. (1983): Phänomen und Logik der Zeichen, Frankfurt/M: Suhrkamp
- Potthoff, I. (1998): Empirische Studien zum wirtschaftlichen Erfolg der Informationsverarbeitung. In: Wirtschaftsinformatik 40, 54-65
- Reckwitz, A. (2003): Grundelemente einer Theorie sozialer Praktiken. Eine sozialtheoretische Perspektive, Zeitschrift für Soziologie 32 (4), 282-301
- Rohde, M.; Brödner, P.; Stevens, G.; Betz, M. & Wulf, V. (2017): Grounded Design – a Praxeological IS Research Perspective, Journal of Information Technology 32 (2), 163-179
- Ropohl, Günter (1991): Technologische Aufklärung. Beiträge zur Technikphilosophie, Frankfurt/M: Suhrkamp.
- Schmidt, K. (2011): Cooperative Work and Coordinative Practices. Contributions to the Conceptual Foundations of Computer-Supported Cooperative Work (CSCW), London: Springer
- Smith, A. (1776): Der Wohlstand der Nationen – eine Untersuchung seiner Natur und seiner Ursachen, aus dem Engl. übertragen u. mit einer Würdigung von H.C. Recktenwald, München: Beck 1974
- Standish Group International (2016): CHAOS Report 2015, [https://www.standishgroup.com/sample\\_research\\_files/CHAOSReport2015-Final.pdf](https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf)
- Stiroh, K. J. (2002). Information Technology and the U.S. Productivity Revival: What Do the Industry Data Say? American Economic Review 92 (5), 1559-1576
- Solow, R.M. (1987): We'd Better Watch Out, New York Times Book Review, July 12
- Turing, A.M. (1950): Computing Machinery and Intelligence, Mind 49, 433-460
- Turing, A.M. (1936): On Computable Numbers. With an Application to the Entscheidungsproblem, in: M. Davis (Ed.): The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems, and Computable Functions, New York 1965, 116-151
- Verein Deutscher Ingenieure (Hg.) (1991): Technikbewertung – Begriffe und Grundlagen. Erläuterungen und Hinweise zur VDI-Richtlinie 3780, VDI Report 15, Düsseldorf: VDI.
- Weber, M. (1921). Wirtschaft und Gesellschaft. Tübingen: Mohr Siebeck 1976
- Weizenbaum, J. (1966): ELIZA – A Computer Program for the Study of Natural Language Communication Between Man and Machine, CACM 9 (1), 36-45