



Christian Stary (MLS), Klaus Fuchs-Kittowski (MLS)

Zur Wiedergewinnung des Realismus als notwendige Grundlage einer am Menschen orientierten Informationssystemgestaltung und Softwareentwicklung

Veröffentlicht: 4. Dezember 2020

Abstract

*Technik bringt das „Noch-nicht-Seiende zum Vorschein“
sagt uns Ernst Bloch¹*

Im Licht dieser Aussage von Ernst Bloch wird deutlich, warum es gerade bei der gegenwärtigen Entwicklung der Naturwissenschaften und Technik wichtig ist, zu präzisieren, was unter Realismus zu verstehen ist und die Notwendigkeit der Überwindung des naiven Realismus und jeder Form von Anti-Realismus, tiefer zu begründen. Die notwendige Wiedergewinnung des Realismus ist für uns als Informatikerinnen/Wirtschaftsinformatikerinnen kein allein innerphilosophisch zu diskutierendes Problem, sondern eine Forderung an modernes philosophisches Denken. Eine Forderung, die sich aus der Tatsache ergibt, dass ohne Realitätsbezug, ohne Anerkennung einer unabhängig von uns existierenden Realität, die von uns in schrittweiser Annäherung erkannt werden kann, keine sicher funktionierende Software geschaffen, kein menschengerechter Einsatz der Informations- und Kommunikationstechnik, die heute alle unsere Lebensbereiche durchdringt, erreicht werden kann. Mit der Diskussion der philosophischen Problematik des Realismus geht es uns insbesondere auch um die Betonung der Bedeutung und um die Weiterentwicklung der Einführungsmethodik – um die Entwicklung der Methodologie der soziotechnischen Informationssystemgestaltung und Softwareentwicklung - generell und speziell bezogen auf die besonders aktuelle Problematik der Entwicklung und des Einsatzes von autonomen Robotern – des Einsatzes humanoider Systeme.

In der Informatik/Wirtschaftsinformatik konzentriert man sich bis heute in Forschung und Lehre weithin auf die Schaffung bestimmter Artefakte und berücksichtigt u.E. immer noch nicht genügend, dass diese informationstechnologischen Systeme ihre Leistungsfähigkeit nur bei einer sachgerechten Einführung in den Arbeitsprozess, in die soziale Organisation in der und für die sie funktionieren sollen, wirklich entfalten können. Dies gilt bei dem hier diskutierten Einsatz humanoider Systeme umso mehr, da die Interaktion zwischen Automaten und Menschen noch intensiver wird. Daher wird die Entwicklung einer am Menschen orientierten Methodik umso wichtiger. So wird beim Einsatz von Robotern, die nicht mehr hinter Gittern, sondern gemeinsam mit den Menschen agieren sollen, die bisherigen Erfahrungen aus der IT-Unterstützung für qualifizierte Arbeit wichtig, entgegen dem Top-down-Ansatz, ein Bottom-up-Ansatz unumgänglich und daher auch der Realitätsbezug.

In diesem Artikel zeigen wir die Bedeutsamkeit des sog. „Neuen Realismus“ im Sinne der Wiedergewinnung des Realismus in Richtung eines konstruktiven Realismus für die Entwicklung und insbesondere Gestaltung zunehmend digitalisierter Lebenswelten. Neben der geistes- und sozialwissenschaftlichen Reflexion räumen wir der methodischen Gestaltbarkeit sozio-technischer Systeme einen bedeutsamen Platz ein. Wir zeigen, in welcher Form technologische Möglichkeiten bei humanzentrierter Lösungsentwicklung mit Hilfe der Design Science Einsatz finden können.

¹ Bloch, Ernst. (1970), Tübinger Einleitung in die Philosophie, Gesamtausgabe, Band 13, Frankfurt a. M., S. 356.

1. Vom Postmodernismus zum Realismus

Heute wird in der Philosophie viel über einen „Neuen Realismus“ gesprochen. Es geht dabei nicht von vornherein darum, ob alles was dazu gesagt wird, wirklich neu ist, sondern es geht vielmehr darum, nach einer Zeit philosophischen Denkens, in der der Realismus weithin abgelehnt wurde, Gründe zu finden, ihn wieder zu akzeptieren. Daher sprechen z.B. Hubert Dreyfus und Charles Taylor u.E. zu Recht von vornherein von einer „Wiedergewinnung des Realismus“² und nicht wie Markus Gabriel u.a. Autoren von „Neuem Realismus“³. Die Einführung des Begriffs „Neuer Realismus“ hat jedoch, über seinen agitatorischen Wert, sicher auch Vorteile. Zum einen kann man sogar einem genauen Termin und Ort für die Einführung des Begriffs angeben und zum anderen kann man möglicherweise den in der Philosophiegeschichte angesammelten großen Ballast unterschiedlichen Vorstellungen zu diesem Thema einfach hinter sich lassen. So datiert Maurizio Ferraris in seinem Beitrag: „Was ist Neuer Realismus?“ die Einführung des Begriffs genau auf den „23. Juli 2011, um 13:30“⁴ in Neapel. Er sagt richtig: „Der Realismus ist – genau wie der Idealismus, Empirismus oder Skeptizismus – ein dauerndes Thema der Philosophie. Der *Neue Realismus* erfüllt deswegen eine wiederkehrende Funktion, und zwar die Reaktion auf eine vorherrschende antirealistische Hegemonie.“⁵

Eben dies wollen wir in unserem Beitrag aufgreifen. Denn der Postmodernismus mit seiner Beliebigkeit, seinem Antirealismus war kein Zufall und hatte entsprechende Wirkungen auch im Denken von InformatikerInnen bzw. (digitaler) SystementwicklerInnen. Andererseits aber trägt die Informatik auch entschieden dazu bei, den Antirealismus zu überwinden. Denken wir z.B. an die jüngsten Ereignisse in der Raumfahrt. Die Landung von „Schiaparelli“ auf dem roten Planeten, speziell die sechs banger Minuten der Mars-Mission, in der vom Menschen nicht eingegriffen werden konnte, sondern alles vollautomatisch ablaufen musste, ist eine großartige Leistung der Automatisierungstechniker, der Hard- und Softwarespezialisten. Es ist eine besonders beeindruckende Forschungsleistung, beweist sie doch, dass es objektive Naturgesetze gibt, die wir erkannt haben und beherrschen, für unsere Zwecke nutzen können. Wenn heute über das Internet und dem Internet der Dinge aus der virtuellen Welt wieder in die materielle Produktion eingegriffen werden kann (Industrie 4.0), dann verlassen sich die InformatikerInnen bzw. (digitale) SystementwicklerInnen darauf, dass es eine Wirklichkeit unabhängig von ihrer Existenz gibt. Sie sind also Realisten, unabhängig davon, wie weit der Postmodernismus, der solipsistische radikale Konstruktivismus und andere antirealistische Positionen, in ihrem Wirkungskreis vertreten werden.

Dass sie vertreten wurden und vertreten werden hat natürlich Gründe. Ob die bisherigen Bemühungen um eine Wiedergewinnung des Realismus für die Erfordernisse der Informatik, speziell für eine menschengerechte Informationssystemgestaltung und Softwareentwicklung ausreichen, ist zu prüfen.

Seit Plato bemüht sich die Philosophie aufzuklären, wie es sich mit unserer Erkenntnis, mit unserer Erfahrung über die Welt konkret verhält. Plato verdeutlicht seine Sicht in seinem berühmten Höhlengleichnis. Danach sind die Menschen in der Situation von Gefangenen, die in einer Höhle gefesselt mit dem Rücken zum Eingang sitzen. Von allem, was vor und in der Höhle geschieht, sehen sie nur die Schatten, die von dem Licht vom Höhleneingang auf die ihnen gegenüberliegende Wand geworfen werden. Da die Menschen nichts anderes sehen, halten sie diese Schatten für die Wirklichkeit selbst. Sie werden also eigentlich betrogen. Daher hat der Philosoph die Aufgabe den Menschen ihre Situation zu erklären. Nur der Philosoph ist in der Lage sich umzudrehen. Er kann aus dem Höhlengang hinaus schauen und die Dinge selbst sehen, nicht nur deren Schatten an der Wand.

² Dreyfus, Hubert; Taylor, Charles. (2016), Die Wiedergewinnung des Realismus, Suhrkamp Verlag, Berlin.

³ Gabriel, Markus, Krüger, Malte Dominik. (2018), Was ist Wirklichkeit? – Neuer Realismus und Hermeneutische Theologie, Mohr Siebeck.

⁴ Ferraris, Maurizio. (2014), Was ist Neuer Realismus?, in: Markus Gabriel (Hrsg.): Der Neue Realismus, Suhrkamp Verlag, Berlin, S. 52.

⁵ Ebenda, S. 53.

2. Überwindung des naiven Realismus und des Anti-Realismus. Der neue Realismus – ein konstruktiver Realismus

2.1. Neuer Realismus?

In seinem erst kürzlich (2018) erschienen Buch: „Unaufgeregter Realismus“ betont Julian Nida-Rümlin, dass es Maurizio Feraris in Italien gelungen sei „eine philosophische Bewegung zu begründen, die unter dem Namen Novo Realismus läuft und einige Aufmerksamkeit im Fach und in der philosophisch interessierten Öffentlichkeit auf sich gezogen hat.“⁶ Er schreibt weiter: Er „nimmt seit Jahren Motive aus der analytischen Philosophie auf, um einen neuen Realismus zu entwickeln – nach der Transformation postmodernen Denkens in rechts-gewirkten Medienpopulismus.“⁷

In seiner Schrift: „Unaufgeregter Realismus“ wendet sich J. Nida-Rümelin speziell gegen den von den Repräsentanten der Postmoderne vertretenen Anti-Realismus, was ihm im Internet eine geharnischte Kritik eingebacht hat, mit der These, diese Schrift sei gescheitert.⁸ Dem ist nicht so. Julian Nida-Rümlin hat eine Tagung in München zu dem Thema durchgeführt, auf der u.a. Charles Lamore aufgetreten ist, der das Buch: „Vernunft und Subjektivität“⁹ sowie Paul Boghossian, der das Buch: „Angst vor der Wahrheit“¹⁰ zu unserem Thema geschrieben hat. Verwiesen sei hier auch auf Markus Gabriel und Mitautorinnen mit ihren Arbeiten zu „Neue Realismus“.^{11, 12} Als Vertreter auch der Disziplin Informatik und Gesellschaft wollen wir weiterhin auf das Buch von Julian Nida-Rümelin und Nathalie Weidenfeld: „Digitaler Humanismus“¹³ verweisen, denn wie zu zeigen ist, sind Realismus und Humanismus, speziell bei der Entwicklung und beim Einsatz der modernen Informations- und Kommunikationstechnologien (IKT), nicht zu trennen.

Wir wollen und können uns hier nicht zu dem ganzen Spektrum der hier angestoßenen philosophischen Diskussionen äußern. Anstoß für unseren Diskussionsbeitrag war eine Konferenz, die von Rainer Zimmermann¹⁴ im Rahmen der Ernst-Bloch-Assoziation zum Thema „Neuer Realismus“ durchgeführt wurde, auf der insbesondere die Frage besprochen wurde, ob dieser „neue Realismus“ wirklich neu ist? Inwieweit auch nur unterschiedliche Richtungen postmodernen Denkens aufeinandertreffen? Uns bewegt aber insbesondere auch das Buch von Hubert Dreyfus und Charles Taylor: „Die Wiedergewinnung des Realismus“.¹⁵ Die Arbeiten von H. Dreyfus und seinem Bruder Stuart, ihre Kritik an der klassischen KI-Forschung, hatten immer Wirkung auf die philosophisch-erkenntnistheoretische und methodologische Debatte in der Informatik/Wirtschaftsinformatik.¹⁶

Uns interessiert weniger die innerphilosophische Diskussion, sondern wir wollen darauf hinweisen, warum der naive Realismus und der Anti-Realismus unbedingt überwunden und der Realismus in der Tat wiedergewonnen werden muss. Wir wollen auch, auf der Grundlage unsere philosophisch-erkenntnistheoretischen Überlegungen und theoretischen und methodologischen Erkenntnisse sowie praktischen Erfahrungen bei der Entwicklung und beim Einsatz der Informations- und Kommunikationstechnologien, Überlegungen dazu ausführen, wie ein Realismus zu konzipieren wäre,

⁶ Nida-Rümelin, Julian. (2018), Unaufgeregter Realismus – Eine philosophische Streitschrift, metis verlag, S. 140.

⁷ Ebenda.

⁸ Gronemeyer, Matthias, Julian Nida-Rümelin scheitert mit seinem neuen Buch, <https://www.matthias-gronemeyer.de/matthias-gronemeyer/>

⁹ Lamore, Charles. (2012), Vernunft und Subjektivität, suhrkamp taschenbuch verlag, Berlin.

¹⁰ Boghossian, Paul. (2013), Angst vor der Wahrheit: Ein Plädoyer gegen Relativismus und Konstruktivismus Suhrkamp Verlag, Berlin.

¹¹ Gabriel, Markus, Krüger, Malte Dominik. (2018), Was ist Wirklichkeit? – Neuer Realismus und Hermeneutische Theologie, Mohr Siebeck.

¹² Gabriel, Markus (Hrsg.) (2014), Der Neue Realismus, Suhrkamp Verlag, Berlin.

¹³ Nida-Rümelin, Julian; Weidenfeld, Natalie. (2018), Digitaler Humanismus – Eine Ethik für das Zeitalter der künstlichen Intelligenz, Piper Verlag, München.

¹⁴ Zimmermann, Rainer E. (2016), Neuer Realismus als Neue Beliebigkeit - Postmodern gegen die Postmoderne, Schriftliche Fassung des mündlichen Beitrages im Workshop „Spekulativer Materialismus, Neuer Realismus“ der Ernst-Bloch-Assoziation, Berlin, 24.9.2016.

¹⁵ Dreyfus, Hubert, Taylor, Charles. (2016), Die Wiedergewinnung des Realismus, Suhrkamp Verlag, Berlin.

¹⁶ Dreyfus, Hubert L.; Dreyfus, Stuart E. (1986), Mind over Machine, The Power of Intuition and Expertise in the Era of the Computer, The Free Press, New York.

der dem Anliegen einer menschengerechten Gestaltung dieser technischen und damit verbundenen sozialen Entwicklungen gerecht werden könnte.

Die Notwendigkeit der Wiedergewinnung des Realismus ist für uns als InformatikerInnen/WirtschaftsinformatikerInnen kein allein innerphilosophisch zu diskutierendes Problem, sondern eine Forderung an modernes philosophisches Denken. Eine Forderung, die sich aus der Tatsache ergibt, dass ohne Realitätsbezug, ohne Anerkennung einer unabhängig von uns existierenden Realität, die von uns schrittweise annähernd erkannt werden kann, keine sicher funktionierende Software geschaffen, kein menschengerechter Einsatz der Informations- und Kommunikationstechnik, die heute alle unsere Lebensbereiche durchdringt, erreicht werden kann.

Diese Einsicht ist es, die die in Gang gesetzte philosophische Bewegung, für einen „Neuen Realismus“, zur Überwindung des naiven Realismus wie des Anti-Realismus im philosophischen Denken, hoffentlich erfolgreich werden lässt.

2.2. Erkennen oder Konstruieren?

In den Naturwissenschaften erfolgt die Modell- und Theorienbildung zur Erklärung oder Vorhersage von Phänomenen der Realität. Die Rolle der Formalisierung und Mathematisierung im Falle der Softwareentwicklung hat eine andere Funktion. Die Software soll bestimmten Anforderungen gerecht werden. Zwischen den informalen Anforderungen und formaler Spezifikation besteht eine mathematisch nicht fassbare Beziehung. „Die formale Spezifikation macht sich vom Realitätsbezug unabhängig“ schreibt Peter Schefe¹⁷ und hebt hervor: „Erst die informale Anforderung liefert den Realitätsbezug. Diese intensionale Bedeutungszuweisung ist grundlegend verschieden von der einer mathematisch-logischen Interpretation. Wir nennen sie daher Sinnzuweisung.“

Die von InformatikerInnen/WirtschaftsinformatikerInnen als EntwicklerInnen human-orientierter Systeme zu bewerkstellende Anforderungsanalyse ist also nicht mit der Erkenntnissituation eines Naturwissenschaftlers/einer Naturwissenschaftlerin vergleichbar. Er/Sie hat bestimmte handlungsorientierte Konzeptualisierungen der Realität zu beschreiben, durch einen Text zu erfassen. Es geht also nicht darum, auf der Grundlage bestimmter Fakten zugrundeliegende Naturgesetze zu erschließen, sondern darum Aufgaben, Bedürfnisse, Intentionen zu verstehen, Sinn zu erfassen.

Die bekannten Schwierigkeiten der Informationssystemgestaltung und Softwareentwicklung, speziell die Schwierigkeiten, die mit der Anpassbarkeit von Standardsoftware bzw. generell mit der Wiederverwendbarkeit von Software verbunden sind, haben ihre Grundlage in dieser Erkenntnissituation. Es geht, wie gesagt, nicht um die Erschließung allgemeingültiger Gesetze, sondern um die Schaffung informationstechnologischer Mittel, die an allgemeingültige und individuelle Situationen mit sich verändernden Zielen und Zwecken anpassbar sind.

Damit stellt sich die Frage: Wenn Informationssystemgestaltung und Softwareentwicklung keine Entdeckung von Gesetzmäßigkeiten sind, handelt es sich damit um reine Schöpfung? Oder, wenn Informationssystemgestaltung und Softwareentwicklung Realitätskonstruktion in gemeinschaftlicher Arbeit, auf der Grundlage der ermittelten Anforderungen ist, hat man es dann nicht auch mit der Aufdeckung wesentlicher Zusammenhänge, Gesetzmäßigkeit, zu tun? Es geht um Konstruktion und (Re-) Organisation, aber auf der Grundlage der Erkenntnis wesentlicher bzw. allgemein-notwendiger Zusammenhänge. Wesentliche Zusammenhänge der menschlichen Tätigkeit, der Arbeitsorganisation und betrieblichen Organisation als Ganzes sind zu berücksichtigen und nicht alles zu erfinden, zu erzeugen. Wie dies aus der Sicht eines weithin vertretenen Anti-Realismus, wie von Vertretern eines radikalen bzw. solipsistischen Konstruktivismus, behauptet wird.

2.3. Entdecken und konstruieren – konstruktiver Realismus

Für einen vagen Anti-Realismus gilt es schon als ausgemacht, nur Erzeuger und nicht Entdecker von Wirklichkeit zu sein. Eine genauere und weniger voreilige Besinnung auf das Verhältnis von „Medialität und Realität im Erkennen“, sollte uns dazu führen einen Konstruktivismus zu entwickeln, der einen philosophischen Realismus nicht verwirft und einen Realismus, der nur gemeinsam mit einem philosophischen Konstruktivismus vertreten werden kann, so dass die u. E. für die Informatik/Wirtschaftsinformatik wenig fruchtbare theoretische Alternative: naiver Realismus oder

¹⁷ Schefe, Peter. (1999), Softwaretechnik und Erkenntnistheorie, Informatik-Spektrum, 22(2), S. 122–135.

solipsistischer Konstruktivismus überwunden werden kann. Wie dies schon vor einiger Zeit von John McDowell,¹⁸ Crispin Wright,¹⁹ Hilary Putnam,²⁰ Robert Brandom,²¹ vorgezeichnet und von Martin Seel,²² besonders klar herausgearbeitet wurde. Aus der hier aufgezeigten Sicht ist also weder ein naiver Realismus noch ein Anti-Realismus zu vertreten sondern eine erkenntnistheoretisch-methodologische Position zu entwickeln, die den naiven wie den Anti-Realismus überwindet, in dem anerkannt wird, dass die Natur vor dem Menschen und der Mensch vor der Wissenschaft existierten, dass aber die Erkenntnis der wesentlichen Zusammenhänge der Natur des menschlichen Denkens und der Gesellschaft, die aktive sinnlich verändernde Tätigkeit des Menschen, Experiment und Industrie zur Voraussetzung hat. Es geht also um ein Erfassen der Realität und Konstruktion, um einen konstruktiven bzw. strukturellen Realismus, wie dies schon von Friedrich Wallner²³ unter Mitarbeit Christian Stary und Markus Peschl [ebenda, Fußnote, S. 246 sowie Vorwort, S. viii] vorgeschlagen wurde. F. Wallner stützt sich auf die Arbeiten von Huberto R. Maturana²⁴ zur Autopoiese. Die er aber erkenntnistheoretisch nicht wie dieser im Sinne eines nicht solipsistischen Konstruktivismus auslegt, sondern eben im Sinne eines von Christian Stary und Markus Peschl konzipierten, konstruktiven Realismus²⁵ (siehe auch Hinweis auf die Mitarbeit von Stary und Peschl).²⁶

Alan F. Chalmers²⁷ konnte gegenüber einem vagen Anti-Realismus verdeutlichen, dass trotz Schwierigkeiten der Realismus doch wissenschaftlich begründbar ist. Die theoretisch unfruchtbare Alternative: naiver Realismus oder solipsistischer Konstruktivismus ist also überwindbar.

Wenn heute versucht wird sich diesen Fragen immer mehr zu stellen, wird offensichtlich, dass der logische Positivismus als erkenntnistheoretische Grundlage, die der Informatik aus historischen Gründen weithin inhärent ist, nicht genügen kann, um die Fragen zu beantworten. Um die Probleme der Softwareentwicklung und ihrer Nutzung zu verstehen und zu lösen, bedarf es wesentlich reichhaltigerer Ansätze. Dies wurde ins besonders auf der von Christiane Floyd und Heinz von Förster 1991 durchgeführten Tagung: „Software Development and Reality Construction“²⁸ deutlich, auf der InformatikerInnen, u.E. erstmalig selbst, ihre philosophisch-erkenntnistheoretischen Probleme diskutierten.

Man suchte nach neuen Antworten zunächst in erkenntnistheoretischen Positionen, wie sie in den Lebensphilosophien, in der Hermeneutik, der Phänomenologie und in der Existenzphilosophie von M. Heidegger²⁹ entwickelt worden waren. Das Buch von Terry Winograd und Fernando Flores: „Understanding Computers and Cognition“³⁰, welches darüber hinaus von John R. Searle's

¹⁸ McDowell, John. (1994), *Mind and World*, Cambridge, Mass

¹⁹ Wright, Crispin. (1992), *Truth and Objectivity*, Cambridge, Mass – London C.

²⁰ Putnam, Hilary. (1994), *The Question of Realism*, in: ders. *Words and Life*. Hgv, J. Conant, Cambridge, Mass, S. 295–312.

²¹ Brandom, Robert B. (1995), *Making it Explicit. Reasoning, Representing and Discursive Commitment*, Cambridge, Mass. – London.

²² Seel, Martin. (1998), *Bestimmung und Bestimmenlassen – Anfänge einer medialen Erkenntnistheorie*, in: *Deutsche Zeitschrift für Philosophie*, Heft 3, Akademie Verlag, Berlin.

²³ Wallner, Friedrich. (1990), *Konstruktiver Realismus als Konsequenz eines autopoetischen Ansatzes*, in: Karl W. Kratky, Friedrich Wallner (Hrsg.): *Grundprinzipien der Selbstorganisation*, Wissenschaftliche Buchgesellschaft, Darmstadt, S. 236–248.

²⁴ Maturana, Huberto, R. (1982), *Erkennen: Die Organisation und Verkörperung der Wirklichkeit*, Braunschweig.

²⁵ Wallner, Friedrich. (1990), *Konstruktiver Realismus als Konsequenz eines autopoetischen Ansatzes*, in: Karl W. Kratky, Friedrich Wallner (Hrsg.): *Grundprinzipien der Selbstorganisation*, Wissenschaftliche Buchgesellschaft, Darmstadt, S. 236–248.

²⁶ Ebenda, Fußnote S. 246 sowie Vorwort S. viii.

²⁷ Chalmers, Alan F. (2001), *Wege der Wissenschaft – Einführung in die Wissenschaftstheorie*, Springer, Berlin.

²⁸ Floyd, Christiane; Züllighoven, Heinz; Budde Reinhard; Keil-Slawik, Reinhard (Editors). (1992) *Software Development and Reality Construction*, Springer Verlag, Berlin, Heidelberg, New York.

²⁹ Dreyfus, H.L.; Dreyfus, S.E. (1986), *Mind over Machine, The Power of Intuition and Expertise in the Era of the Computer*, New York.

³⁰ Winograd, Terry; Flores, Fernando. (1989), *Erkennen Maschinen Verstehen – Zur Neugestaltung von Computersystemen*, Rotbuch Verlag, Berlin.

Sprechakttheorie³¹ und der „Theorie des kommunikativen Handelns“³² von Jürgen Habermas beeinflusst war, gewann großen Einfluss. Zur Bewältigung der, durch die Informationssystemgestaltung und Softwareentwicklung aufgeworfenen neuen erkenntnistheoretisch-methodologischen Probleme gewann der von Huberto Maturana, Heinz von Förster u.a. vertretenen Konstruktivismus an Bedeutung. Zu gleich fand jedoch auch eine deutliche Hinwendung zur Tätigkeitstheorie statt, wie sie, ausgehend vom Marx'schen³³ Verständnis der Praxis als Grundlage der Erkenntnis, insbesondere von der russisch-kulturellen Schule von Wygotski³⁴ und Leontjew³⁵ entwickelt worden war und in der modernen Psychologie³⁶, in den Arbeitswissenschaften³⁷ sowie in der Wissenschaftstheorie weitergetragen wurde. Dies wurde insbesondere in der nordischen Schule der Informatik deutlich.^{38, 39, 40} und auch in der KI-Forschung in den U.S.A. reflektiert.⁴¹ Speziell auch mit der Entwicklung der Theorie selbstorganisierender Systeme, wie sie in den modernen Naturwissenschaften von Glansdorf⁴² und Prigogine mit Isabelle Stengers^{43, 44} von Manfred Eigen, Erich Jantsch, Werner Ebeling⁴⁵ entwickelt wurde und schrittweise für die hier aufgeworfenen Fragestellungen verallgemeinert wird erhielt diese erkenntnistheoretische Richtung starke Impulse.

Auf der Grundlage einer für den sozialen Bereich verallgemeinerten Theorie der Selbstorganisation, in Verbindung mit dem Tätigkeitskonzept, kann ein entsprechen des theoretisch-methodologischen Konzept entwickelt werden, welches den hier angesprochenen Dimensionen der Softwareentwicklung Rechnung tragen kann. Dies ist auch erforderlich. Denn standen bisher vielfach die rein technischen Gesichtspunkte der Softwareerstellung im Vordergrund, so tritt jetzt, mit dem dezentralen und vernetzten Einsatz der modernen IKT, verstärkt die Einbettung von IKT-Systemen in die menschliche Arbeitstätigkeit hervor. Speziell geht es beispielsweise um die IKT-Unterstützung wissensintensiver Arbeitsprozesse.⁴⁶

Es kann damit eine erkenntnistheoretisch-methodologische Position entwickelt werden, die uns:

- Besonderheiten der Softwareentwicklung als Wirklichkeitskonstruktion
- Besonderheiten der Unternehmensorganisation als kreativ-lernende Organisation besser erkennen lässt,
- eine vertiefte Sicht der für die Informatik und Wirtschaftsinformatik auf die zentralen Wechselbeziehungen zwischen Menschen, Aufgabe, Informationstechnologie und Organisation ermöglicht.

³¹ Searle, John, R. (1971), Sprechakte. Ein sprachphilosophischer Essay, Suhrkamp, Frankfurt.

³² Habermas, Jürgen. (1981), Theorie des kommunikativen Handelns, Suhrkamp, Frankfurt.

³³ Institut für Marxismus - Leninismus. (1978), Marx-Engels-Werke (MEW), Bd. 3, Dietz Verlag, Berlin, S. 5–7.

³⁴ Wygotski, Lew Semjonowitsch. (1964), Denken und Sprechen. Russische Originalausgabe 1934, deutschsprachige Ausgabe Akademie-Verlag, Berlin.

³⁵ Leontjew, Alexei Nikolajewitsch. (1982), Tätigkeit, Bewusstsein, Persönlichkeit. Köln, Pahl-Rugenstein.

³⁶ Hacker, Winfried. (1992), Allgemeine Arbeits- und Ingenieurpsychologie. Psychische Struktur und Regulation von Arbeitstätigkeiten. VEB Deutscher Verlag der Wissenschaften, Berlin 1973; Ulich, Eberhard, Arbeitspsychologie, Schäffer, Pöschel, Stuttgart.

³⁷ Raeithel, A. (1992), Activity Theory as a Foundation for Design. In: Floyd et al., Software Development and Reality Construction, Springer Verlag, Berlin 1992, S. 391–415.

³⁸ Nygaard, K. (1986), Program Development as a Social Activity. In: Kugler, H. G. (ed.): Information Processing 86. Proceedings of IFIP 10th World Computer Congress. Amsterdam: North Holland, S. 189–198.

³⁹ Nygaard, K. (1992), How Many Choices Do We Make? How Many Are Difficult? In: Floyd et al. 1992, S. 52–59.

⁴⁰ Ehn, P. (1988), Work-Oriented Design of Computer Artifacts. Stockholm: Arbetslivscentrum.

⁴¹ Arge, P.E. (1997), Computer and Human Experience. Cambridge University Press, 1997.

⁴² Glansdorff, Peter; Prigogine, Ilya. (1971), Thermodynamic Theory of Structure, Stability and Fluctuations, Wiley Interscience.

⁴³ Prigogine, Ilja; Stengers, Isabelle. (1981), Dialog mit der Natur – Neue Wege naturwissenschaftlichen Denkens, Piper Verlag, München.

⁴⁴ Prigogine, Ilja; Stengers, Isabelle. (1993), Das Paradox der Zeit, Piper Verlag, München.

⁴⁵ Ebeling, W. (2016), Physik, Biologie, Technik und Selbstorganisation von Information. In: Frank Fuchs-Kittowski, Werner Kriesel (Hrsg.): Informatik und Gesellschaft – Festschrift zum 80. Geburtstag von Klaus Fuchs-Kittowski, Peter Lang Verlag, Frankfurt a.M, 2016, S. 48–57.

⁴⁶ Fuchs-Kittowski, Frank. (2007), Integrierte IT-Unterstützung der Wissensarbeit, EUL Verlag.

Sieht man den Arbeitsprozess als den Hauptaspekt, dann wird deutlich, dass praktisch eingesetzte Software keine der Bedingungen für eine Ware, vom sozialen Kontext ihrer Herstellung unabhängig austauschbarem Gut, entspricht. Diese Feststellung ist besonders wichtig für die komplexen Softwaresysteme der künstlichen Intelligenz, den wissensbasierten Systemen bzw. Expertensystemen.

Wissen ist ein soziales Produkt. Seine Wahrheit gilt in der sozialen Gemeinschaft und kann auch nur wahr im Austausch mit der sozialen Gemeinschaft von WissenschaftlerInnen bleiben in der es geschaffen wurde, für längere Zeit bewahrt und genutzt wird. Wissen ist wahrscheinlich nur im engeren naturwissenschaftlich-technischen Bereich, als ein personenunabhängiges, reproduzierbares und vom sozialen Kontext unabhängig übertragbares Gut anzusehen.

Es ist insbesondere das Scheitern der mit dem Konzept der Vollautomatisierung und des Strukturalismus unterstellten Idee von einer abgeschlossenen Welt, die in relativ selbständige funktionale Bestandteile gegliedert werden kann und es daher ermöglicht vorbedachte Strukturen durch Programme nachzubilden, das die Ablösung der strukturierten Analyse und des strukturierten Entwurfs durch die objektorientierte Sichtweise und evolutionäre Konzepte der Systemgestaltung und Softwareentwicklung, des Paradigmas der Struktur durch das Paradigma der Selbstorganisation und Informationsentstehung, erforderlich macht. Das objektorientierte Paradigma der Informatik hat die Modellierung revolutioniert.⁴⁷ Mit dem WAM-Leitbild, dem Werkzeug, Automat, Material- Ansatz⁴⁸ wurde auf der Grundlage der objektorientierten Sichtweise eine spezielle Methode zur Entwicklung von Software zur Unterstützung qualifizierter Arbeit erarbeitet. Die objektorientierte Sichtweise eignet sich zur Realisierung von Softwareprodukten (objektorientierte Programmiersprachen) wie zugleich zur Beschreibung von Modellen, die im Zuge der Softwareentwicklung erforderlich sind. Ein objektorientiertes System besteht aus Klassen und Instanzen sowie deren Beziehungen z.B. Vererbung, Benutzung, semantische Relationen. Die Bildung von Klassen unterstützt die für die Modellierung erforderliche Abstraktion. Wichtig sind die einfachen semiformalen Ausdrucksmittel mit denen objektorientierte Systeme beschrieben werden können. Generell wollen wir hier Modelle dieser Art von den Erkenntnisssystemen in der Wissenschaft unterscheiden, denn ihre Aufgabe ist die Funktionsteilung zwischen Automaten und Mensch zu unterstützen.

Die Vorstellung von einer „Softwarefabrik“ im Sinne einer tayloristischen Produktion von Software, wie dies zunächst zur Lösung der „Softwarekrise“ durch die strukturierte Programmierung vertreten wurde, ist aufgegeben worden. Die Vorstellung, dass der Problemhorizont als geschlossener Raum erscheint und die Lösung sich in Form einer „gefrorenen Spezifikation“ angeben lässt ist gescheitert, dagegen kann man sich mit dem Konzept der Objektorientierung auf die eigenständige Dynamik des Softwareentwicklungsprozesses orientieren, der es meist nicht mit abgeschlossenen, zuvor spezifizierbaren Problemen zu tun hat. Der mit der Systemgestaltung und Softwareentwicklung einhergehende Erkenntnisprozess muss in flexibel erweiterbaren Artefakten darstellbar und anzureichern sein. In der Tat werden die Erfahrungen aus den Arbeitsprozessen, speziell die Spezifik der qualifizierten Arbeit, der eigenverantwortlichen Expertentätigkeit,⁴⁹ zunehmend in der Welt der Softwareentwicklung reflektiert. Die Objekte tauschen Nachrichten aus, nehmen Dienste in Anspruch und halten Kontrakte ein.

Gegenwärtig wird allgemein die Frage nach einer humanen Gestaltung rechnerunterstützter Werkzeuge bzw. Arbeitssysteme gestellt. Dies führt dann weiter zu der Frage, was Menschen sinnvoll mit diesen rechnerunterstützten Arbeitssystemen tun können. Dies weitet sich zu der Frage aus, was Menschen im Bereich, koordinierter Tätigkeiten, linguistischer Aktionen in wissensintensiven Arbeitsprozesse Zen Sinnvolles machen können. Die theoretischen Grundlagen, die Methodologie der Informationssystemgestaltung und Softwareentwicklung verändern sich grundlegend, wenn man die modernen Informations- und Kommunikationstechnologien aus der Sicht ihrer möglichen sinnvollen

⁴⁷ Horn, Erika. (2005), Software Engineering und Modellierung, Vortrag in der Klasse Naturwissenschaften der Leibniz-Sozietät.

⁴⁸ Züllighoven, Heinz. (1998), Das objektorientierte Konstruktionshandbuch – nach dem Werkzeug & Materialansatz, dpunkt.verlag, Heidelberg.

⁴⁹ Ebenda.

Nutzung durch den Menschen in seinem Arbeitsprozess betrachtet. Man spricht heute⁵⁰ generell von einer traditionellen, technikzentrierten Herangehensweise gegenüber einer auf die Arbeitsprozesse und den Menschen orientierten Herangehensweise bei der Informationssystemgestaltung und Softwareentwicklung. Hier gilt es zu entdecken und zu konstruieren.⁵¹

3. Die Entwicklung und der Einsatz integrierter humanoider Systeme – der Dreifachcharakter der Software setzt die Anerkennung der Realität voraus

3.1. Hybride Automatisierung

„Eine umfassender „Realitäts-Skepsis ist allenfalls als ein philosophisches Gedankenexperiment möglich“ schreibt J. Nida Rümelin u. E. zurecht: „nicht als Lebensform. Außerhalb philosophischer Seminare werden alle wieder zu Realisten.“⁵²

Die Leugnung einer Realität unabhängig von unserer Existenz erfolgt aber eben nicht nur im Sinnen philosophischer Gedankenexperimenten, sondern sicher auch unter deren Einfluss von namhaften Einzelwissenschaftlern. Als Beispiel sei hier nur Helmut Wilke eingeführt, der in seinem Buch: „Systemisches Wissensmanagement“⁵³, welches nicht ohne Einfluss auf die Wirtschaftsinformatik blieb, schreibt, dass es keine Wahrheit gibt. An dieser Stelle fragen wir in der Vorlesung über Wissensmanagement die StudentInnen, ob sie mit dem Fahrstuhl in den fünften Stock gekommen sind und sich damit auf die Wahrheit der erkannten physikalischen Gesetze verlassen haben?

Bei der Informationssystemgestaltung und Softwareentwicklung reicht aber, bei all dem was heute durch Computer, Computernetze, durch sog. intelligente Systeme, im Sinne der zu realisierenden Visionen einer smart Factory und smart City zu gewährleisten ist, dieser nur instinktive Realismus nicht aus. *Der Dreifachcharakter der Software setzt die bewusste Anerkennung der Realität voraus.*

Die gilt insbesondere, wenn wir es mit Arbeitssystemen, mit Mensch-Maschine-Systemen zu tun haben, in dem der Mensch aktiv werden muss, wenn Störungen oder Unterbrechungen des automatischen Prozessablaufes eintreten. Der im System involvierte Mensch muss die Ursachen dafür ermitteln. Das gelingt ihm, wenn er die ihm über die Bildschirmoberfläche eines Computers übermittelten Informationen mit Hilfe seines tätigkeitsspezifischen Gedächtnisbesitzes zu einem aktuellen „Bild“ über den Prozesszustand integriert und diesem jene Handlungsprogramme zuordnet, die zur Erreichung vorgegebener Produktions- oder Dienstleistungsziele führen.

Er muss sich auf die wahre Erkenntnis der zugrundeliegenden wesentlichen – gesetzmäßigen - Zusammenhänge verlassen können. Wie die Arbeiten von M. R. Endsley⁵⁴ zur „situation awareness“ verdeutlichen, ist dies die Voraussetzung für ein adäquates Handeln des Menschen im MMS unter der Bedingung, dass der automatisierte Prozessablauf nicht durch eine stabile Software gesteuert wird.

Für den weiteren Typ künftiger Arbeitssysteme, der auf der Grundlage moderner KI-Systeme, speziell autonome Roboter sog. androiden Systemen realisiert wird, gilt die Forderung nach Realitätsbezug bei der Systemgestaltung und Softwareentwicklung umso mehr, damit die kreativen Fähigkeiten des Menschen mit der Exaktheit und Geschwindigkeit der technischen Systeme sinnvoll kombiniert werden – eine hybride Automatisierung wirklich realisiert werden kann.

Es gibt eine Vielzahl von Innovationen in der digitalisierten Produktion, durch die ein effizientes hybrides Zusammenwirken von Menschen und Computer auf allen Stufen zwischen manueller Arbeit und vollautomatisierter Fertigung realisiert werden kann.

Dieser Typ von Arbeitssystemen, wie er auf der Basis *integrierter humanoider Systeme* realisiert werden kann, ist sowohl durch eine hohe Intensität der Technologie, durch neue Formen der visuellen

⁵⁰ Floyd, Chr.; Klaren, H. (1998), Informatik: gestern, heute, morgen, Informatik und Gesellschaft, Fernstudium, Universität Tübingen.

⁵¹ Foerster, H. v. (1991), Entdecken oder Erfinden – Wie läßt sich Verstehen verstehen? In: Einführung in den Konstruktivismus, Serie Piper, München Zürich.

⁵² Nida-Rümelin, Julia. (2018), Unaufgeregter Realismus – Eine philosophische Streitschrift, mentis Verlag, S. 138.

⁵³ Wilke, Helmut. (1998), Systemisches Wissensmanagement, Lucius & Lucius, Stuttgart.

⁵⁴ Endsley, M. R. (2000), Theoretical underpinnings of situation awareness. A critical review. In: M. R. Endsley & D. J. Garland, Situation awareness analysis and measurement, Mahwah: Lawrence Erlbaum Associates, S. 3–28.

Verbindung von Mensch und Computer, durch Nutzung der Geschwindigkeit, Exaktheit und Wiederholbarkeit des Roboters, als auch durch eine hohe Intensität des Sozialen, durch Nutzung der spezifischen sensomotorischen und kognitiven Fähigkeiten des Menschen, die Nutzung komprimierter Erfahrungen für intuitive Entscheidungen, charakterisiert.

Diese Kombination der spezifischen Fähigkeiten des Menschen mit den spezifischen Fähigkeiten des Computers, zu einer noch leistungsfähigeren Einheit, die auch hochleistungsfähige Software (Algorithmen und auch entsprechend entwickelter Orgware zu Grundlage hat, wird als *hybride Automatisierung* bezeichnet.

Die immer kürzeren Entwicklungszyklen intelligenter mechatronischer und cyber-physischer Systeme verlangen eine ständige weitere Qualifikation der Arbeitenden.

Das neuartige Zusammenwirken von adaptiven Systemen höherer Stufe, den sog. autonomen Robotern, die als intelligente Systeme neue Interaktionsmuster und Fähigkeiten realisieren können, verlangt auch von den Arbeitenden eine neue Qualität der Interaktion. Heute weiß man aber, dass insbesondere aus ökonomischer Sicht die Höhe des Automatisierungsgrades durch die Flexibilität der Roboter begrenzt wird (siehe Spath⁵⁵).

Die intelligenten Maschinen und die autonomen Roboter haben Möglichkeiten zur Anpassung an den Menschen. Dies verlangt jedoch zugleich auch neue Formen der Anpassung seitens der Arbeitenden. Es können sich somit Formen einer Koevolution herausbilden, die jedoch auch mit Gefährdungen verbunden sein können (siehe Brödner^{56,57}). Dies verlangt, wie wir sagen, dass die flexible Automatisierung durch Robotereinsatz, durch eine dynamische Automatisierung ergänzt bzw. erweitert wird, d.h. entsprechende organisatorische Maßnahmen, wie beispielsweise die Möglichkeit des Eingriffs des Menschen über einen indirekten Dialog – Regulations-Instanz – zu etablieren (siehe K. Fuchs-Kittowski⁵⁸).

Im Arbeitssystem dieses Typs kann die Steuerungssoftware auf Grund ihrer Vernetzung mit anderen Programmen im Internet Lösungen für die Behebung von Störungen und Unterbrechungen selbst generieren. Wegen des emergenten Charakters dieser Lösungen besteht aber die Gefahr, dass vorgegebene Produktions- und Dienstleistungsziele verfehlt werden. Daher muss der Mensch das Prozessgeschehen überwachen und unter Umständen regulierend eingreifen können. Allerdings ist bisher unklar, wie und unter welchen Bedingungen der Mensch im System dieses Typs eine adäquate situation awareness im Sinne von M. R. Endsley bilden kann (siehe Rothe⁵⁹).

Der Mensch muss daher Subjekt in diesem neuen Zusammenwirken mit dem Computer sein und bleiben! Das heißt, Beherrscher und Gestalter des Geschehens bleiben. Er muss sich auf wahre Aussagen über die von ihm zu bewältigenden Aufgaben verlassen können!

⁵⁵ Spath, D. (2013), Der Automatisierungsgrad hat in der Flexibilität seine Grenzen, Interview zu Industrie 4.0. <https://www.ingenieur.de/technik/fachbereiche/automation/der-automatisierungsgrad-in-flexibilitaet-grenzen/>.

⁵⁶ Brödner, Peter. (2019), Grenzen und Widersprüche der Entwicklung und Anwendung ‚Autonomer Systeme‘. In: H. Hirsch-Kreinsen & A. Karačić (Hrg.): *Autonome Systeme und Arbeit. Perspektiven, Herausforderungen und Grenzen der Künstlichen Intelligenz in der Arbeitswelt*, Bielefeld: transcript, S. 69–97.

⁵⁷ Brödner, Peter. (2020), Paradoxie der Ko-Aktion von Experten und adaptiven Systemen, in: Peter Brödner, Klaus Fuchs-Kittowski (Hrsg.): *Zukunft der Arbeit – Soziotechnische Gestaltung der Arbeitswelt im Zeichen von Digitalisierung und Künstlicher Intelligenz*, Abhandlungen der Leibniz-Sozietät, Band 67, trafo Verlag, Berlin.

⁵⁸ Fuchs-Kittowski, Klaus. (2020), Informationssystem-, Arbeits- und Organisationsgestaltung in Produktion und Verkehr – Das Orgware-Konzept, die Paradoxie der Sicherheit, des Wächters, der Beherrschung großer Datenmengen, in: *Zukunft der Arbeit – Soziotechnische Gestaltung der Arbeitswelt im Zeichen der ‚Digitalisierung‘ und ‚Künstlichen Intelligenz‘*, Abhandlungen der Leibniz-Sozietät der Wissenschaften, Band 67, trafo Wissenschaftsverlag, Berlin, S. 83–113.

⁵⁹ Rothe, Hans-Jürgen. (2020), Arbeit 4.0 – alte und neue arbeitswissenschaftliche und ingenieurwissenschaftliche Probleme, in: *Zukunft der Arbeit – Soziotechnische Gestaltung der Arbeitswelt im Zeichen der ‚Digitalisierung‘ und ‚Künstlichen Intelligenz‘*, Abhandlungen der Leibniz-Sozietät der Wissenschaften, Band 67, trafo Wissenschaftsverlag, Berlin, S. 21–39.

3.2. Dreifachcharakter der Software

Ein spontaner Realismus reicht also dazu nicht aus, um die in der Informatik/Wirtschaftsinformatik noch weithin vorherrschend Tendenz, Software vorrangig als ein eigenständiges Produkt, ohne wirklichen Bezug zum Kontext der Herstellung und des Einsatzes zu sehen, zu überwinden. Will die Softwaretechnik die Software und den Prozess der Softwareentwicklung tiefer verstehen, so muss die Softwareentwicklung als ein Prozess der Modell- und Theorienbildung verstanden und die Software als vergegenständlichte Methode und informationelles Arbeitsmittel gesehen werden.

Die Softwareentwicklung kann von verschiedenen Sichtweisen aus betrachtet werden. Das mit der Modellmethode verbundene neue Selbstverständnis der Wissenschaft hat auch besonderes Gewicht für die weitere Entwicklung der theoretischen und methodologischen Grundlagen der Softwareentwicklung. Es wird deutlich, dass Software ein Doppel-, ja sogar Dreifach-Charakter besitzt. Sie kann als Modell eines Gegenstandes und als informationelles Arbeitsmittel sowie als vergegenständlichte Methode verstanden werden. Sie ist im wachsenden Maße auch Medium und damit ein Mittel zur Kommunikation.

Als Modell eines Objektbereiches, der in Software zu transformieren ist, hat die Softwareentwicklung ebenfalls Beobachtung und Experiment sowie Hypothesen- und Theorienbildung zur Grundlage, um den Gegenstandsbereich zu erschließen.

Als informationelles Arbeitsmittel gesehen, wird mit der Software eine Tätigkeit vergegenständlicht, wozu es u.a. Erkenntnisse aus den Arbeits- und Organisationswissenschaften, insbesondere aus der Tätigkeitstheorie der Allgemeinen- und Arbeitspsychologie bedarf.

Als Mittel zur Unterstützung einer kommunikativen Tätigkeit gesehen, bedarf es der Kommunikationstheorie, den Sprachwissenschaften, der Semiotik und der Theorie kommunikativer Tätigkeiten.

Die Modellwelt des Software Engineering muss mit der Welt der Technik und mit der Welt der Anwender übereinstimmen. Dazu bedarf es entsprechender Kommunikations- und Lernprozesse, die die Schaffung einer gemeinsamen Sprache zur Voraussetzung hat.

Zumindest diese drei verschiedenen Dimensionen der Softwareentwicklung: a) Gegenstandsbezug, b) Tätigkeitsbezug und c) Kommunikationsbezug, gilt es zu berücksichtigen (vgl. Dahme⁶⁰).

Auf dem Gebiet des Software Engineering und der damit verbundenen Modellierung hat sich ein entscheidender Paradigmenwechsel vollzogen. Dieser wurde als Wechsel von der Produktorientierung auf eine Prozessorientierung bei der Softwareentwicklung charakterisiert (vgl. Floyd⁶¹). Dies ist insbesondere auch als ein Wechsel von einer allein auf die formale Struktur orientierte Sichtweise, zu einer gleichzeitigen Sicht auf die realen Arbeitsprozesse zu verstehen.⁶²

Aus der Beachtung des Doppel- bzw. Dreifachcharakters der Software ergibt sich eine grundsätzliche Komplementarität der Sichtweisen in der Informatik.

Zu beachten ist die Komplementarität von formaler (syntaktischer), technischer, produktorientierter Sichtweise und informaler, inhaltlicher (semantischer), sozialer, Prozess orientierter Sichtweise in der Informatik. So muss zum Beispiel Software formal korrekt und zugleich als Werkzeug im Arbeitsprozess diesem angemessen sein. Die Komplementarität von Sozialem und Technischen in der Informatik muss sich auch auf das Verständnis der Modellmethode im Methodengefüge der Softwareentwicklung auswirken.

Die Herausbildung theoretischer Grundlagen der Softwareentwicklung hat sich ursprünglich nicht oder nur in geringem Maße aus den Erfahrungen der Softwareherstellung und ihrer Anwendung ergeben, sondern die theoretische Fundierung wurde lange Zeit einfach mit der Entwicklung mathematischer Beweisverfahren und Spezifikationstechniken gleichgesetzt. Damit wurde jedoch nur ungenügend das Wesen der Software, ihre Herstellung und ihr Einsatz reflektiert. Die sehr rasche

⁶⁰ Dahme, Christian. (2000), Wissenschaftstheoretische Positionen in Bezug auf die Gestaltung von Software. In: Klaus Fuchs-Kittowski, Heinrich Parthey, Walter Umstätter, Roland Wagner-Döbler (Hrsg.): Organisationsinformatik und digitale Bibliothek in der Wissenschaft, Wissenschaftsforschung, Jahrbuch 2000, S. 167–177.

⁶¹ Floyd, Christiane. (1987), Outline of a Paradigm Change in Software Engineering. In: Computers and Democracy, a Scandinavian challenge. Hrsg.v. Bjerknes / Ehn / Kyng, Aldershot, Brookfield USA.

⁶² Ebenda.

Entwicklung der Hardware, die Erschließung immer neuer Einsatzgebiete für die modernen Informations- und Kommunikationstechnologien haben dazu geführt, dass es bisher auch nur ungenügend empirische Untersuchungen gibt, auf deren Grundlage Hypothesen zur Softwareentwicklung überprüft werden können. (siehe Oskar Grün, Lutz J. Heinrich⁶³).

Für die Tätigkeit des Softwareentwicklers ist der Bezug zu einer von ihm zu beachtenden Realität, die Bedeutung des Modells für die Gewinnung wahrer Erkenntnis deutlich zu machen. Zugleich kann ein tieferes Eindringen in das Wesen der Software zu einem besseren Verständnis der Modellmethode beitragen. Denn die hier zu vollziehende Transformation von einem konkreten Objektbereich in die Welt der Symbole (geistiges Modell) hat gegenüber anderen Modellen dieses Typs viele Besonderheiten. Die Softwareentwicklung erweist sich hier als ein interdisziplinärer Prozess, bei dem zu den Informatikdisziplinen die Disziplinen zu integrieren sind, die zur Erschließung des Objektbereiches erforderlich sind. Nur so kann das Gebiet der Softwaretechnik (Software Engineering) den gewachsenen gesellschaftlichen Anforderungen entsprechen und in der theoretischen Orientierung wie auch in der empirischen Fundierung zuvor zwangsläufig sich ergebende Defizite (vgl. Keil-Slawik⁶⁴) überwinden.

Will die Softwaretechnik die Software und den Prozess der Softwareentwicklung tiefer verstehen, so muss die Softwareentwicklung als ein Prozess der Modell- und Theorienbildung verstanden und die Software als vergegenständlichte Methode und informationelles Arbeitsmittel gesehen werden. Erst dann wird es möglich sein, die zu enge Orientierung des klassischen Software Engineering zu überwinden, welches ursprünglich als Kern der Informatik, losgelöst von den Erfahrungen der Herstellung und Anwendung der Software, auf einer abstrakten mathematischen Ebene, entwickelt wurde.

Aus technologischer Sicht ist der zentrale Ansatz zum Verständnis der Software ihr Gebrauchswert. Der allgemeine Gebrauchswert der Software besteht darin, dass sie eine Steuerinformation in Bezug auf die Hardware darstellt. Ihr besonderer Gebrauchswert liegt jedoch in der durch die Software vermittelten Konstruktion bzw. (Re-) Organisation des modellierten Realitätsbereiches, als Arbeitsmittel, als Rationalisierungs-, Erkenntnis-, Planungs-, Organisations- und Kommunikationsmittel.

Softwareentwicklung ist somit eine *spezifische Konstruktion sozialer Realität* mit dem Ziel, durch die Rechnerunterstützung ein neues Niveau körperlicher und geistiger Tätigkeit des Menschen zu erreichen, *welches Produktivitäts- und Persönlichkeitsentwicklung ermöglicht*.

Software ist das Ergebnis einer Transformation von Tätigkeiten in Handlungen und diese in Operationen, die nach ihrer Formalisierung als maschinelle Operationen ausgeführt werden können. Software wird damit zum Mittel einer Tätigkeit. Dazu müssen die maschinellen Operationen wieder in die Komplexität der menschlichen Tätigkeit, entsprechend ihrer Ziele und Motive, integriert werden können.

Software ist somit im Zusammenwirken verschiedener Prozesse, als: a) Arbeitsmittel, b) Arbeitsgegenstand und c) als Produkt zu erfassen. Vorherrschend war und ist auch noch weithin die Tendenz, die Software vorrangig als ein eigenständiges Produkt, ohne wirklichen Bezug zum Kontext der Herstellung und des Einsatzes zu sehen. Genau diese Haltung gilt es zu überwinden. Eine Umorientierung von der Sicht des isolierten Produktaspektes auf die gleichzeitige Sicht auf die realen existierenden Arbeitsprozesse bringt ein wesentlich anderes, vertieftes Verständnis der Softwaretechnik, durch den stärkeren Bezug Softwareerstellung und die Softwarenutzung, speziell auch der Organisation der Softwarenutzung.⁶⁵

Erst mit einer konsequenten Orientierung auf den realen Arbeitsprozess als den Hauptaspekt dessen, was bei der Herstellung und Benutzung der Software wirklich geschieht, auf die damit verbundenen Kommunikations- und Lernprozesse, kann eine technizistische Position wirklich

⁶³ Grün, Oskar; Heinrich Lutz J. (Hrsg.). (1997), Ergebnisse empirischer Forschung, Springer Verlag Wien.

⁶⁴ Keil-Slawik, Reiner (1989), Aufgabenbezogene Anforderungsermittlung. In: Floyd, Caster, Keil-Slawik, Einführung in SE B: IFIP-STU.WP.

⁶⁵ Floyd, C. (1995), Outline of an Paradigm Change in Software Engineering. Technische Universität Berlin, Institut für Angewandte Informatik.

überwunden werden und eine echte technologische Bearbeitung geistiger Arbeitsprozesse und ihrer Produkte erfolgen.

Die Herstellung und Anwendung von Softwareprodukten und ihre technologische Beurteilung muss stets von der Wechselwirkung zwischen dem Prozess- und Produktaspekt ausgehen, wobei der Prozessaspekt die primäre Seite darstellt. Die damit verbundenen Kommunikations- und Lernprozesse aller Beteiligten – der Anwender, die Softwareentwickler und Betreiber – verlangen die Entwicklung einer gemeinsamen Sprache. Erst durch die Herausbildung einer gemeinsamen Sprache lassen sich die drei unterschiedlichen „Welten“ des Entwicklungsprozesses: die Welt der Anwendung, die Welt der Methoden und Modelle und die Welt der Technik, schrittweise zusammenzuführen. Es ist häufig so, dass Softwareentwickler und die künftigen Anwender eine sehr unterschiedliche Sprache sprechen. Um eine möglichst größere Übereinstimmung zu erreichen können verschiedenen Methoden eingesetzt werden, wie Prototyping, Qualitätszirkel, ethnographische Methoden der kooperativen Arbeit, gemeinsame Arbeit an der Entwicklung von Ontologien. Es ist ein Prozess kooperativer Wissenserzeugung. Es geht um die Schaffung einer gemeinsamen Sprache, der Einbettung der Software und ihrer Nutzung in einen gemeinsamen sozialen Kontext, um die Schaffung eines sozialen Raums in dem die Beteiligten ihre kooperative Arbeit koordinieren können. Dies sind selbst wissensintensive Arbeitsprozesse für die Wissen bereitgestellt und in denen Wissen in Prozessen der Selbstorganisation erzeugt wird.

Die Neuorientierung im Software Engineering und Modellierung hat einen entscheidenden Wandel in den erkenntnistheoretisch-methodologischen Grundlagen zur Voraussetzung, der nur durch eine entsprechende Reflexion über Wissenschaft und Technik zu gewinnen war.

Auf der Grundlage eines naiven Realismus wird weithin von einer Isomorphie oder zumindest Homomorphie von Realität und ihrer Rekonstruktion durch Software aus gegangen. Diese Auffassung basiert auf philosophisch-erkenntnistheoretischen Grundsätzen, die von Vertretern des Wiener Kreises, des Logischen Positivismus oder Empirismus schon einige Jahrzehnte vor der Entwicklung der Informatik und Wirtschaftsinformatik und der Disziplin der Informationssystemgestaltung und Softwaretechnik entwickelt wurden.

Doch der sich in vielen Wissenschaften vollziehende Wechsel des Leitgedankens (Paradigmenwechsel), der das Prinzip der Selbstorganisation in den Mittelpunkt rückt, hat, wie aufgezeigt wurde, ebenfalls starken Einfluss auf die Wissenschaftstheorie der Gegenwart und führt insbesondere zu einer kritischen Haltung zu den wissenschaftstheoretischen Grundsätzen des Positivismus.

Heinz Zemanek machte schon früh darauf aufmerksam, dass, wenn wir Wurzeln der Informatik retrospektiv im „Tractatus logico-philosophicus“ sehen, man dann auch berücksichtigen muss, dass der ältere Wittgenstein kritisch zu dieser Arbeit des jungen Wittgenstein stand. Er schloss aus der Tatsache, dass L. Wittgenstein das Konzept des „Tractatus logico philosophicus“ später selbst als gescheitert erklärte, dass dann auch die Informatik auf eine breitere philosophische Grundlage zu stellen ist.^{66, 67}

Es ist somit auch weiterhin zu prüfen, inwieweit die wissenschaftstheoretischen und wissenschaftsmethodologischen Positionen, wie sie im Zusammenhang mit dem Leitgedanken der Selbstorganisation und der Tätigkeitstheorie entwickelt wurden, einer solchen Erweiterung bzw. Vertiefung der philosophischen, erkenntnistheoretisch-methodologischen Grundlagen der Informatik und insbesondere der Wirtschaftsinformatik dienen kann. Wie aufzuzeigen versucht wurde, ist auf dieser Grundlage in der Tat eine fruchtbare erkenntnistheoretische Position – ein konstruktiver Realismus – zu entwickeln.

⁶⁶ Zemanek, H. (1973), Philosophie und Informationsverarbeitung, NTZ,26,1973/8, S. 384–389.

⁶⁷ Zemanek, H. (1993), Philosophische Wurzeln im Wiener Kreis, in: Scheffe, P. et al. (Hrsg.), Informatik und Philosophie, Wissenschaftsverlag, Mannheim, Zürich.

4. Gestaltungsorientierte Entwicklung integrierter humanoider Systeme – Design Science als methodische Grundlage des konstruktiven Realismus

Wir wollen in der Folge zeigen, in welcher Form die Welt der Anwendung, die Welt der Methoden und Modelle und die Welt der Technik, schrittweise im Sinne human-orientierter sozio-technischer Systementwicklung zusammenzuführen sind. Wir erläutern zuerst das derzeitige Verständnis humanoider Systeme ehe wir mittels Design Science den Mehrfachcharakter sozio-technischer Systementwicklung strukturiert entsprechen.

Humanoide Systeme sind menschenähnliche Systeme, vornehmlich aus der Robotik – dort auch als Androide bezeichnet – bekannt. Sie verfügen über *künstlich intelligente Systemkomponenten, die nach menschlichem Vorbild aufeinander abgestimmt werden. An ihrem Einsatzspektrum lässt sich untersuchen, welches menschenähnliche Verhalten eines künstlichen Systems Menschen in welchem Ausmaß in welchen Situationen akzeptieren und schließlich sinnbringend nutzen. So ist das Robotik-System Pepper⁶⁸ in der Lage, nicht nur auf Fragen zu reagieren und zur Bewältigung von Aufgaben Anleitungen zu geben, sondern auch Handlungen zu setzen, wie etwa den Weg weisen. Mit der Fähigkeit, auch kleinteilig Lasten dank Leichtbauarmen und ‚Vierfingerhänden‘ zu bewegen (vgl. Rollin’ Justin⁶⁹), steht einer durchdringenden Vernetzung von humanoiden Systemen mit menschlichen Wirkbereichen nichts im Weg.*

Humanoide Systeme verfügen demnach über eine mobile Basis, die einen autonomen Betrieb in große Reichweite erlaubt, etwa im Gegensatz zu mobilen Endgeräten, die ein physikalisches Trägersystem benötigen. Kernkomponenten humanoider Systeme sind Bewegungsdetektions-Sensoren und Stereokameras, welche auch in unstrukturierten, sich dynamisch ändernden Umgebungen ermöglichen, 3D-Rekonstruktionen der Umgebung zu erstellen. In den nun sich vermehrt herausbildenden kollaborativen Wirkbereichen und Technologien⁷⁰ ist das grundsätzlich selbstständige und ohne menschliche Unterstützung agierende humanoide System nicht isoliert im Einsatz. Es ist mit anderen, insbesondere Internet-of-Things⁷¹-Anwendungen zu verbinden. Somit werden humanoide Systeme Teil föderierter Systeme, d.h. sie können sowohl autonom als auch in Kombination mit anderen Systemen arbeiten.

Die Gestaltung integrierter humanoider Systeme steht vor der Herausforderung, unterschiedliche Designansätze zu berücksichtigen. So führt der steigende Reifegrad humanoider Systeme durch entsprechend sichtbare Freiheitsgrade zum Merkmal, mehrere Ziele gleichzeitig zu verfolgen und dabei eine Aufgabenhierarchie einzuhalten. Damit kann ein Robotersystem beispielsweise Getränke servieren und dabei die Umgebung beobachten, um Kollisionen in der Umgebung zu vermeiden. Soll diese Funktionalität nutzbringend in sozio-technischen Umgebungen eingesetzt werden, ist zum einen die Abstimmung mit den menschlichen Tätigkeiten und zum anderen mit bereits eingesetzten Technologien erforderlich. Es hängt vom Einsatzbereich ab, ob und wie Personen humanoide Systeme konfigurieren und zur Bewältigung von Aufgaben anpassen. Diese organisationalen Anforderungen führen gegebenenfalls zu zusätzlicher technischer Funktionalität, die auf der technologischen Integration basiert (vgl. Rosly et al.⁷²).

⁶⁸ www.SoftBankRobotics.com.

⁶⁹ www.dlr.de

⁷⁰ Siehe z.B. <https://www.cc.gatech.edu/social-machines/>

⁷¹ Li, S.; Da Xu; L., Zhao, S. (2018), 5G Internet of Things: A survey. *Journal of Industrial Information Integration*, 10, S. 1–9.

⁷² Rosly, M. A.; Miskam, M. A.; Shamsuddin, S.; Yussof, H.; Zahari, N. I. (2020), Data Linking Testing Between Humanoid Robot and IoT Network Server for Autism Telerehabilitation System Development. In: *RITA 2018*. Springer, Singapore, S. 161–169.

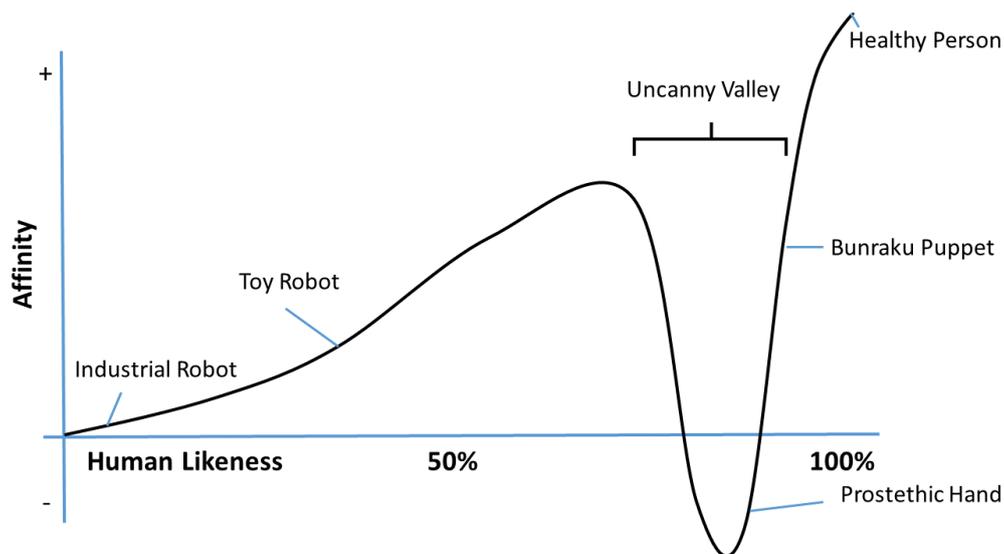


Abb. 1 Akzeptanz humanoider Systeme – aus Mori et al.⁷³ – Bunraku bedeutet japanisches Puppenmusiktheater (17. Jahrhundert).

Ein typisches Beispiel stellt der home healthcare Bereich dar, der mittels M-IoT (Medical Internet of Things)⁷⁴-Anwendungen ausgestattet ist. Hier müssen etwa Pflegekräfte die Bereitschaft entwickeln, humanoide Systeme auf Patienten einzustellen und diese derart konfigurieren, dass Patientendaten für Notfallsorge übergeben werden, um eine effektive Entlastung zu ermöglichen. Für die Akzeptanz bedeutet dies, dass integrierte Systeme nicht wie isolierte humanoide Systeme wie etwa von Mori et al.⁷⁵ untersucht (siehe Abbildung), verhalten werden.

Darüber hinaus kann die Konstruktion eines funktionellen humanoiden Roboters im Sinne einer menschenähnlichen künstlichen Intelligenz zu neuartigen Gestaltungs- und Entwicklungsaufgaben führen (vgl. Andersen et al.⁷⁶). Konfiguration und Anpassung werden zu einem Lernprozess, der mit Hilfe selbstlernender Algorithmen in einem humanoiden System implementiert wird. Auf Basis von Beobachtungen und lernpsychologischen Daten kann die Teilnahme an sozialen Prozessen ermöglicht werden (vgl. Can et al.⁷⁷). Damit sind neben den funktionalen Verrichtungstätigkeiten soziale Aufgaben von humanoiden Systemen übernehmbar.

Da bisherige Entwicklungsansätze eher isoliert und produktorientiert erfolgen (cf. Matteus et al.⁷⁸) wird in der Folge ein Design Science Ansatz vorgestellt, der sozio-technisches Systemdesign integrierter humanoider Systeme ermöglicht. Gestaltungsorientierte ("design science") orientierte Entwicklung hat Problemlösung zu Ziel. Dabei soll ein zu schaffender Artefakt (d.i. ein künstliches oder vom Menschen

⁷³ Mori, M.; MacDorman, K. F.; Kageki, N. (2012), The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2), S. 98–100.

⁷⁴ Sadoughi, F.; Behmanesh, A.; Sayfour, N. (2020), Internet of things in medicine: a systematic mapping study. *Journal of Biomedical Informatics*, 103, S. 103–383.

⁷⁵ Mori, M.; MacDorman, K. F.; Kageki, N. (2012), The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2), S. 98–100.

⁷⁶ Andersen, R. E.; Madsen, S.; Barlo, A. B.; Johansen, S. B.; Nør, M.; Andersen, R. S.; Bøgh, S. (2019), Self-learning Processes in Smart Factories: Deep Reinforcement Learning for Process Control of Robot Brine Injection. *Procedia Manufacturing*, 38, S. 171–177.

⁷⁷ Can, W.S.R.; Seibt, S.D.J. (2016), Making place for social norms in the design of human-robot interaction. *What social robots can and should do: Proceedings of Robophilosophy 2016/TRANSOR 2016*, 290, 303.

⁷⁸ Mateus, J. C.; Claeys, D.; Limère, V.; Cottyn, J.; Aghezzaf, E.H. (2019), A structured methodology for the design of a human-robot collaborative assembly workplace. *The International Journal of Advanced Manufacturing Technology*, 102(5-8), S. 2663–2681.

konstruierter materieller oder immaterieller Gegenstand) Nutzen herbeiführen, anhand dessen sich eine praktisch umgesetzte Lösung bewerten lässt.

Nach Hevner et al.⁷⁹ wirft ein Designer bei gestaltungsorientierter Forschung Fragen auf, welche für menschliche Probleme relevant sind. Durch die Schaffung innovativer Artefakte wird neues Wissen generiert und anhand von Demonstratoren im Sinne einer Lösungsinstanziierung bewertet. Bei Grundprinzip der design-wissenschaftlichen Forschung ist, dass bei der Konstruktion und Anwendung eines Artefakts Wissen generiert wird, das dem Verständnis für ein Gestaltungsanliegen und dessen Lösung dient.

Gestaltungsorientierte Entwicklung fokussiert im Sinne der Design-Based Research auf die Schaffung von (entworfenen) Artefakten in der Absicht, die Erfüllung von Anforderungen sukzessive zu verbessern (vgl. Peffers⁸⁰). Artefakte sind vielfältig. Sie können unter anderem Roboter, Software, Prozesse, Methoden, Verfahren oder Modelle sein. Sie werden in vier Kategorien unterteilt: Konstrukte, Modelle, Methoden und Instanziierungen.

Die durchgeführten Gestaltungs- und Entwicklungsaktivitäten sollen nicht ein gesellschaftsrelevantes Thema behandeln, sondern einen innovativen Beitrag zur bereits vorhandenen Wissensbasis der beteiligten Disziplinen darstellen. Folglich sollten vergleichbare Lösungen nicht bereits im Vorfeld existieren. Das geschaffene Artefakt ist im Rahmen von Design-Zyklen zu evaluieren, um sicherzustellen, dass die gesetzten Ziele und daraus abgeleitete Anforderungen an den Artefakt erreicht werden. Design Science ermöglicht folglich, auf Basis von wissenschaftlichen und praktischen Erkenntnissen ein Artefakt zu erstellen.

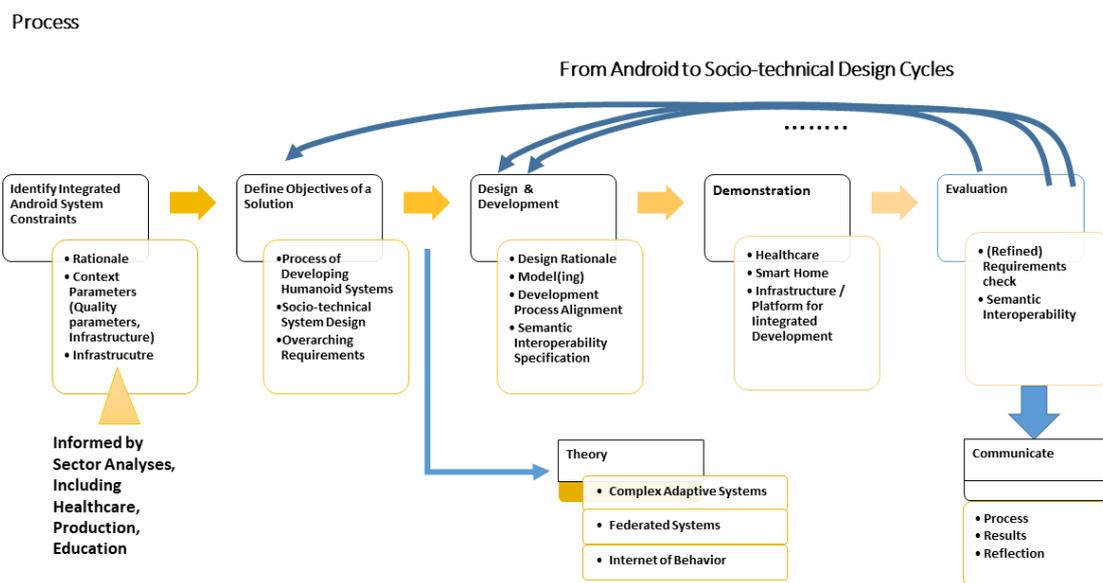


Abb 2. Gestaltungsorientiertes Vorgehen bei der integrativen Gestaltung von androiden Systemen (Operationalisierung nach Peffers⁸¹)

Gestaltungsorientiertes Vorgehen betrifft nach Hevner⁸² drei Zyklen: Der wissenschaftlichen (Rigor) Zyklus, welcher das Wissen von der Theorie herleitet, wird mit dem praktischen (Relevance) Zyklus,

⁷⁹ Hevner, A.R.; March, S.T.; Park, J.; Ram, S. (2004), Design Science in Information Systems Research. In: MIS Quarterly, Vol. 28 (2004) No. 1, S. 75–105.

⁸⁰ Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. (2007), A Design Science Research Methodology for Information Systems Research. In: Journal of Management Information Systems, Vol. 24 (2007) No. 3, S. 45–77.

⁸¹ Ebenda.

⁸² Hevner, A.R. (2007), A Three Cycle View of Design Science Research. In: Scandinavian Journal of Information Systems, Vol. 19 (2007) No. 2, S. 87–92.

welcher den Stand der Technik einbezieht, verknüpft. Die Verknüpfung von Theorie und Praxis leitet ein Artefakt ab. Das Artefakt wird iterativ von Erkenntnissen aus Theorie und Praxis verbessert. Die Verbesserung des Artefakts liefert einen Beitrag zu wissenschaftlichen und praktischen Erkenntnissen. Hevner⁸³ beschreibt die drei Zyklen wie folgt:

- *"The Relevance Cycle inputs requirements from the contextual environment into the research and introduces the research artifacts into environmental field testing.*
- *The Rigor Cycle provides grounding theories and methods along with domain experience and expertise from the foundations knowledge base into the research and adds the new knowledge generated by the research to the growing knowledge base.*
- *The central Design Cycle supports a tighter loop of research activity for the construction and evaluation of design artifacts and processes."*

Der Gestaltungsprozess setzt sich aus der gekoppelten Schaffung und Evaluierung zusammen. Die Schaffung betrifft Aktivitäten, die das Artefakt erstellen und anhand eines Schaustücks (Demonstrator) erfahrbar machen. Die nachfolgende Evaluierung inkludiert die Bewertung des Artefakts und liefert Informationen dazu, wie Anforderungen an die Lösung erreicht wurden. Dadurch kann das Verständnis des adressierten Problems und im nächsten Gestaltungszyklus das Artefakts verbessert werden. Design – und Evaluierungsaktivitäten werden solange wiederholt bis die Anforderungen erfüllt werden.

Design Science ermöglicht aus mehreren Gesichtspunkten einen integrativen konstruktiven Ansatz, die Welt der Anwendung, die Welt der Methoden und Modelle und die Welt der Technik schrittweise im Sinne human-orientierter sozio-technischer Systementwicklung zusammenzuführen: Zum ersten ist es das Konzept des Artefakts, das die Möglichkeit eröffnet, über den technischen Gestaltungsgegenstand (z.B. das Robotiksystem) hinaus Lösungs'gegenstände' im Rahmen wiederkehrender Gestaltungsaktivitäten zu berücksichtigen. Dies können Methoden der Anpassung oder auch Richtlinien sein, welche sich im Laufe von Designzyklen verfestigen oder aufgrund der Adaptierung von Anforderungen (siehe Abbildung zur Operationalisierung nach Peffers⁸⁴). Damit können Entwicklungsergebnisse unterschiedliche materielle wie immaterielle Elemente beinhalten und je nach Anforderung mehrdimensional sein.

Die Modellwelt wird vor allem bei der Gestaltung cyber-physischer Systeme⁸⁵ relevant, sobald digitale Zwillinge ganzheitlich eingesetzt werden. Cyberphysische Systeme sind aufgrund der Heterogenität ihrer Komponenten und derer Interaktionen (Kommunikation, Kollaboration, Kontrolle, Berechnung) nicht nur komplex, sondern unterliegen einer starken Dynamik⁸⁶. Sie weisen daher einen hohen Grad an Selbstreferenz auf, die es auf Modellebene in Sinne der Gestaltung als auch des integrativen Engineering und der Ausführung zu beachten gilt (vgl. Grieves et al.⁸⁷).

Zum zweiten eignet sich Design Science als Entwicklungsparadigma aufgrund der integrativen Berücksichtigung von Theorien und Praxiserfahrungen für die schrittweise Entwicklung humanoider Systeme. Damit wird es möglich, unterschiedliche Theorien als Grundlage der Gestaltung zur Anwendung zu bringen bzw. gezielt auszuwerten. Aus sozio-technischer Sicht ein wünschenswerter Ansatz: Nicht alles, was technisch möglich ist, muss für Menschen von Nutzen sein. So wird beispielsweise ein Alarmzustand eines Robotik-Systems im Gesundheitsbereich im eigenen Haushalt wird vorverarbeitet bzw. (nach Rollen) gefiltert Informationen an die jeweiligen Stakeholder (Produzent, Betreiber, Patient, Arzt) weiterleiten und nach Möglichkeit im Maschinenverbund autonom verarbeiten, um nicht jede messbare Ausnahmesituation zu einer menschlichen Intervention eskalieren lassen zu müssen. Diese

⁸³ Ebenda, S. 87.

⁸⁴ Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. (2007), A Design Science Research Methodology for Information Systems Research. In: Journal of Management Information Systems, Vol. 24 (2007) No. 3, S. 45–77.

⁸⁵ Liu, Y.; Peng, Y.; Wang, B.; Yao, S.; Liu, Z. (2017), Review on cyber-physical systems. *IEEE/CAA Journal of Automatica Sinica*, 4(1), S. 27–40.

⁸⁶ "complex systems with organic integration and in-depth collaboration of computation, communications and control (3C) technology" in Liu, Y., Peng, Y., Wang, B., Yao, S., & Liu, Z. (2017), Review on cyber-physical systems. *IEEE/CAA Journal of Automatica Sinica*, 4(1), 27-40, S. 27.

⁸⁷ Grieves, M.; Vickers, J. (2017), Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary perspectives on complex systems*. Springer, Cham, S. 85–113.

Filterung bzw. sozio-technische Aufgabenteilung kann mittels Design-Zyklen zunächst getrennt gestaltet und bewertet werden, um die jeweilig möglichen technischen Möglichkeiten zunächst detailliert zu kennen und danach integrativ bzw. im Wechselspiel zu betrachten.

Schließlich erlaubt Design Science den strukturierten Aufbau von Wissen durch den Relevanz- (Relevance-) und Wissenschaftszyklus (Rigor Cycle). Die stellt zum einen sicher, dass der Kontext der jeweiligen humanoiden Systementwicklung transparent wird – sowohl die Anforderungen als auch die Evaluierungsparameter bestimmen sich daraus abhängig vom Organisations- und Technikbezug. Zum anderen stellt die Dokumentation erfolgreicher Entwicklungen evidenzbasiertes und nicht-redundantes Vorgehen sicher, wodurch der Effektivität wie auch der Effizienz gleichermaßen Rechnung getragen werden kann. Dies bedeutet in letzter Konsequenz, dass der wiedergewonnene Realismus im Sinne eines methodisch fundierten konstruktiven Realismus menschengerechte Technologieentwicklung auch bei fortschreitender Digitalisierung ermöglicht.

5. Zukünftige Orientierung am Menschen auf der Grundlage konstruktiven Realismus‘

Im Vorangegangenen wurde betont, dass die Entwicklung der Informatik, speziell die Entwicklung des Informatik-nahen Konzepts der Industrie 4.0, mit dem über das Internet der Dinge mögliche Eingriffe von der virtuellen Welt in die materielle, industrielle Produktion, anti-realistische Positionen überwunden werden, ein am Menschen orientierter Einsatz dieser Technologien einen Realismus erforderlich machen, ja erzwingen. Dazu muss aber auch gesagt werden, dass sich damit auch ein naiver Realismus wieder verstärkt verbreitet, dem ebenfalls entgegengetreten werden muss. Mit der Darstellung einer Abfolge von industriellen Revolutionen und der These von der sich jetzt vollziehenden 4. technischen Revolution, wird oftmals ein schon überwunden geglaubter technischer Determinismus propagiert. Diese Sicht beruht meist auf naiven Realitätsvorstellungen: Dass wir das Sein so erkennen können, wie es unabhängig von unseren Erkenntnismittel existiert, halten ihre Vertreter für selbstverständlich. Wahr ist für sie, was funktioniert.

Wenn hier der Vorschlag gemacht wird, den „Neuen Realismus“ als einen „konstruktiven Realismus“ zu verstehen, dann richtet sich dieser Ansatz sowohl gegen den Anti-Realismus als auch gegen den naiven Realismus. Diese angestrebte Überwindung des naiven Realismus und des Antirealismus ist jedoch nicht als ein Einfaches „sowohl“ als „auch, sondern als „weder/noch“ zu verstehen. Es muss uns um eine echte Vermittlung zwischen den widerstreitenden Ansätzen gehen. Diese Vermittlung erfolgt im Prozess der Gestaltung und des Einsatzes der Informations- und Kommunikationstechnologien. In diesem Prozess geht es um die in Beziehung-Setzung von lebendigen, kreativ arbeitenden Menschen zu immer leistungsfähigeren, aber toten Artefakten. Es geht damit um die Beachtung der Dialektik von Gemeinsamkeit und Unterschied zwischen Menschen und Automaten. Der Mensch, der menschliche Geist ist weder nur ein komplexes kybernetisches Funktionssystem noch etwas von diesem völlig Unabhängiges, rein Geistiges, sondern ein Drittes – Ausdruck des ganzen Menschen.

Praktisch-methodologisch gedacht kommt folglich die Er- und Bearbeitung des Gestaltungs(spiel)raums vor jener des Lösungsraums in einem konstruktiv-realistischen Entwicklungsprozess. Diesbezüglich kommt der nordisch-europäischen Schule der Systementwicklung konstitutiver Charakter im Sinne humanzentrierter sozio-technischer Systemgestaltung zu. Prozessual ergänzt wird dieser durch den kreativ-explorativen Ansatz des Design Thinking, wie von Terri Winograd und der Stanfordschule propagiert, und zwar eingebettet in Design (Science) Zyklen: Entlang von Gestaltungszyklen materialisierte Designideen strukturieren Engineering-Aktivitäten zur Umsetzung von Demonstratoren. Diese werden ob ihrer Lösungsbeiträge evaluiert. Derart entstehende Artefakte erlauben die Berücksichtigung unterschiedlicher Theoriebildungen und Konzepte sowie die fokussierte Bearbeitung technischer und sozialer Elemente humanoider Systeme – das wohl wesentlichste Feature bei der konstruktiven Wiedergewinnung des Realismus als notwendige Grundlage einer am Menschen orientierten Informationssystemgestaltung und Softwareentwicklung.