

Auszug aus **Grundrisse** bei **Context XXI**

(http://contextxxi.org/wie-die-edv-konzerne-ihre.html)

erstellt am: 27. Oktober 2020

Datum dieses Beitrags: September 2002

Wie die EDV Konzerne ihre Mehrwertproduktion zu kontrollieren versuchen – ein Erfahrungsbericht

■ FRANÇOIS NAETAR

Ist der Kapitalismus eine Schranke für die Produktivkräfte?

Aufgrund seiner Vorstellungen über das Verhältnis von Produktivkräften und Produktionsverhältnissen schreibt Marx: *„In der gesellschaftlichen Produktion ihres Lebens gehen die Menschen bestimmte, notwendige, von ihrem Willen unabhängige Verhältnisse ein, Produktionsverhältnisse, die einer bestimmten Entwicklungsstufe ihrer materiellen Produktivkräfte entsprechen ... Eine Gesellschaftsformation geht nie unter, bevor alle Produktivkräfte entwickelt sind, für die sie weit genug ist, und neue höhere Produktionsverhältnisse treten nie an die Stelle, bevor die materiellen Existenzbedingungen derselben im Schoß der alten Gesellschaft selbst ausgebrütet worden sind.“* [1] Angesichts der Entwicklung des Kapitalismus im letzten Jahrhundert scheint die Vorstellung, dass es noch weiterer Entwicklung der Produktivkräfte bedürfe, um die Existenzbedingungen für neue (kommunistische) Produktionsverhältnisse zu schaffen, eine beunruhigende Vorstellung zu sein. Mit einer alle Grenzen überschreitenden Dynamik hat sich die Masse an Ware, die in der Produktion oder im Konsum vernutzt werden, vervielfacht. Noch in den letzten zwei Jahrzehnten des vergangenen Jahrhunderts konnten große Territorien mit über zwei Milliarden Menschen

(China, Indien, tw. der ehemalige Ostblock, Südostasien) mehr oder weniger vollständig in den Verwertungsprozess des internationalen Kapitals eingegliedert werden. Gleichzeitig fand ein „neuer Schub der Durchkapitalisierung und inneren Landnahme in Form einer Unterwerfung weiterer gesellschaftlicher Sektoren unter den Kapitalverwertungsprozess durch ‚Kommodifizierung‘, d.h. der warenförmigen Inwertsetzung von Arbeitsprodukten und Naturressourcen“ statt. [2]

An anderer Stelle schreibt Marx, dass das Kapital *„auch, während es die Tendenz hat, die Produktivkräfte ins Maßlose zu steigern, ebenso die **Hauptproduktivkraft den Menschen selbst, vereinseltigt, limitiert, etc. ...**“* [3] (Hervorhebung vom Autor). Scheinen diese zwei Bestimmungen von Marx nicht im völligen Widerspruch zu liegen? Muss nicht eine Gesellschaft, die sich vom Kapitalismus verabschieden will, weniger produktiv sein, die Entwicklung der Produktivkräfte zurücknehmen, das heißt weniger Naturstoffe umwälzen, mehr auf die Arbeit, auf den Menschen, als auf Technologie setzen?

Die Aussage, dass der Kapitalismus von einer Produktionsweise abgelöst werden soll, welche die Fesseln, die von den Produktionsverhältnissen den Produktivkräften angelegt werden, beseitigen soll, scheint in der momentanen Diskussion den Bestrebungen aller Gegner des herrschenden Systems völlig

zuwiderzulaufen. Als Beispiel dazu ein Zitat von Slavoj Žižek, der als in Slowenien im „real existierenden Sozialismus“ aufgewachsener Linker die Verbindung von Produktivkraft und Kapitalismus folgendermaßen einschätzt:

Gerade als Marxisten und aus Treue zu Marx' Werk sollten wir seinen Fehler beim Namen nennen: Marx erkannte, dass der Kapitalismus die atemberaubende Dynamik einer sich selbst steigernden Produktivität freisetzte — man erinnere sich nur an seine von großer Faszination geprägten Beschreibungen des Kapitalismus als größtem Revolutionär der Menschheitsgeschichte; er sah aber auch, wie diese kapitalistische Dynamik von ihrem eigenen inneren Hindernis oder Antagonismus vorangetrieben wird — die ultimative Grenze des Kapitalismus, der kapitalistischen, sich selbst vorantreibenden Produktivität ist das Kapital selbst, das heißt, die unaufhörliche kapitalistische Entwicklung und Revolutionierung ihrer eigenen materiellen Bedingungen, der irre Tanz ihrer bedingungslosen Produktivitätsspirale, ist letztlich nichts als eine verzweifelte Flucht nach vorn, um dem ihr selbst inhärenten und sie schwächenden Widerspruch zu entkommen. Marx' grundsätzlicher Fehler bestand darin, eine neue, höhere Gesellschaftsordnung, den Kommunismus, für möglich zu halten, eine Ordnung, die das Potential der sich selbst steigernden Produktivität, die im Kapitalismus aufgrund seines inhärenten Widerspruchs immer wieder durch gesellschaftlich zer-

störerische Wirtschaftskrisen vereitelt wird, nicht nur aufrechterhalten, sondern weiter steigern und schließlich voll zur Entfaltung bringen würde. ... Wenn wir das Hindernis, den inhärenten Widerspruch des Kapitalismus, beseitigen, erhalten wir nicht den freigesetzten Produktivitätstrieb, sondern wir büßen genau diese Produktivität ein, die vom Kapitalismus zugleich generiert und zerstört worden ist — wenn wir das Hindernis beseitigen, löst sich das von diesem Hindernis vereitelte Potential auf. ...

Kapitalismus und Kommunismus sind nicht zwei verschiedene historische Realisierungen, zwei Arten der „instrumentellen Vernunft“, sondern die instrumentelle Vernunft als solche ist kapitalistisch, gründet auf kapitalistischen Verhältnissen, und der „real existierende Sozialismus“ ist daran gescheitert, dass er letztlich eine Unterart des Kapitalismus war, ein ideologischer Versuch, beides gleichzeitig zu haben, aus dem Kapitalismus ausbrechen und zugleich sein zentrales Element bewahren zu wollen. Wir wetten also darauf, dass, selbst wenn wir auf den teleologischen Begriff des Kommunismus (die Gesellschaft der gänzlich entfesselten Produktivität) als den impliziten Standard, an dem Marx die Entfremdung der existierenden Gesellschaft misst, verzichten, der Hauptteil seiner „Kritik der politischen Ökonomie“, die Einsicht in den sich selbst vorantreibenden Teufelskreis der kapitalistischen (Re)Produktion, überlebt. Die Aufgabe des heutigen Denkens ist daher eine doppelte: Einerseits die marxistische „Kritik der politischen Ökonomie“ zu wiederholen, aber ohne den utopisch ideologischen Begriff des Kommunismus als ihres inhärenten Standards; andererseits, sich vorzustellen, wie man effektiv aus dem kapitalistischen Horizont ausbrechen kann, ohne in die Falle einer Rückkehr zur eminent vormodernen Vorstellung einer ausgeglichenen, (selbst)beherrschten Gesellschaft zu gehen (der „vorkartesianischen“ Versuchung, der ein Großteil der heutigen Ökologie erliegt). [4]

Eine Diskussion über die Frage der Bedeutung der Analysen und Einschätzungen von Marx darf sich der Frage der Produktivkräfte nicht entziehen. Bis in die 60er und 70er Jahre des vorigen Jahrhunderts wurde erst von SozialdemokratInnen und dann von

KommunistInnen die Überlegenheit des Sozialismus und Kommunismus unter anderem mit der schnelleren Entwicklung der Produktivkräfte argumentiert. In den 30er Jahren sang Ernst Busch: „Die Krise fasst das Kapital, der Kommunismus steht wie Stahl“. Vollmundig wurde noch von Chruschtschow die „kommunistische Losung“ ausgegeben: „den entwickelten Kapitalismus einholen und überholen“. [5]

Dieser „klassischen“ Leseart der Wechselwirkung von Produktivkraftentwicklung und Produktionsverhältnissen, steht allerdings eine andere Interpretation des Begriffes „Produktivkraft“ entgegen, die im obigen Zitat über die wichtigste Produktivkraft den Menschen selbst seinen Ausdruck findet. Die höhere „Produktivkraft“ bedeutet in dieser anderen Leseart, dass der allseitig entwickelte und kooperierende freie Produzent erstens die zur Reproduktion notwendige Arbeit, die Marx immer als das „Reich der Notwendigkeit“ bezeichnete, minimieren und die allein oder in Kooperationen zur freien Verfügung stehende Zeit – das „eigentliche Reich der Freiheit“ – maximieren wird. In diesem Sinn kann die Aussage, dass eine nachkapitalistische Produktionsweise Produktivkräfte freisetzen wird, noch immer Sinn machen. Ein wesentlicher Grund, warum neben der brutalen Verschwendung (von Mensch und Natur) die „Ineffizienz“ des Kapitalismus (im Sinne der Verringerung der notwendigen Arbeit) in letzter Zeit so völlig aus dem Blickfeld verschwand, ist die völlig im Vordergrund stehende Kritik an der Kapitalverwertung und ihrer Verselbständigung. Der Automatismus des Kreislaufes des Kapitals, der jedes Feld der Gesellschaft mit seiner Verwertungslogik durchdringt, stand und steht im Vordergrund der Betrachtungen. Bevor der Tauschwert der kapitalistisch produzierten Waren und der in ihnen steckende Mehrwert als Profit am Markt realisiert werden kann, muss dieser Mehrwert jedoch erst produziert werden. Um den Kapitalismus von diesen beiden Seiten, nämlich der Verschwendung und der Ineffizienz, zu verstehen, genügt es nicht, die Bewegung des Kapitals zu betrachten. Wir müssen uns jene Orte näher anschauen, an denen der Mehrwert produziert wird – die Fabrik, das Büro und andere Orte, an denen

„die gesellschaftliche GesamtarbeiterIn“ zur Produktion des Mehrwerts beiträgt.

Was wissen wir über die Produktion von Mehrwert im Postfordismus?

Wenn wir die Orte betrachten, an denen Mehrwert produziert wird, dann sehen wir, das sie vom Doppelcharakter des Arbeitsprozesses gekennzeichnet sind: es wird abstrakte, (mehr)wertschöpfende Arbeit geleistet und es wird durch stoffliche, technische und organisatorische Gegebenheiten bestimmte konkrete Arbeit geleistet. Den Prozess, diese konkrete Arbeit auch in ihrer stofflichen Form dem Ziel der Mehrwertproduktion unterzuordnen, bezeichnet Marx als „die reelle Subsumption unter das Kapital“. Im Gegensatz dazu wird die unveränderte Übernahme einer Produktionsform unter das Kommando des Kapitals als „formelle Subsumption“ bezeichnet. In der (linken) Öffentlichkeit ist in letzter Zeit die Diskussion darüber, wie sich das Kapital im Produktionsprozess zu diesem Doppelcharakter der Arbeit verhält, wenig behandelt worden. Auf sehr allgemeine Weise beschäftigt sich das Buch „Empire“ von Negri und Hardt (siehe Artikel über dieses Buch in der letzten Ausgabe) mit dieser Frage, indem es feststellt, dass die ganze Gesellschaft tendenziell real unter das Kapitalverhältnis subsummiert wird.

Der kritische Teil der Betriebssoziologie war hier eine Ausnahme. Er beschäftigt sich mit dem Kapitalismus unter einem anderen Gesichtspunkt, nämlich der „bürokratischen Herrschaft über die Arbeit“, bzw. der „Herrschaft durch Organisation“. [6] Auch wenn mir scheint, dass diese Untersuchungen die Tendenz haben, ihrerseits wieder den Verwertungszusammenhang aus den Augen zu verlieren, sind doch in den letzten 20 Jahren eine Vielzahl von Untersuchungen entstanden, die es wert wären, von der Linken mit Aufmerksamkeit behandelt zu werden. [7] Die älteren dieser Untersuchungen beschäftigen sich mit den IndustriearbeiterInnen – den klassischen ProletarierInnen, meist auch noch in der Autoindustrie. [8] Erst im letzten Jahrzehnt wurde erkannt, dass neue, „postfordistische“ Formen der

Gestaltung des Produktionsprozesses schnell an Bedeutung gewinnen. In den interessantesten Untersuchungen wird grundsätzlich auf den unlöslichen Widerspruch im Doppelcharakter der Arbeit verwiesen. Dieser Widerspruch zeigt sich (in jeder kapitalistischen Produktion) als „Widerspruch zwischen Autonomie und Heteronomie“. Einerseits ist die Arbeit im kapitalistischen Produktionsprozess immer fremdbestimmt (heteronom): „Kapitalismus bedeutet für die Arbeitenden immer zunächst und grundlegend Ausschluss: Ausschluss von der Verfügung über die Produktionsmittel und der Definition der Produktionsziele, drohender Ausschluss von der Verfügung über Reproduktionsmittel durch die Kündigungsoption. Das impliziert ein elementares Machtgefälle und setzt den Rahmen für ein spezifisches Herrschaftsverhältnis. Im Produktionsprozess selbst setzt sich diese primäre Ausschlusstendenz fort, zugleich wird sie modifiziert und begrenzt. Bürokratische Herrschaft über Arbeit intendiert und befördert einerseits in vielfältiger Weise den Ausschluss der Arbeitenden aus den Entscheidungen über Produktionsziele und -methoden. Sie ist der großangelegte Versuch, Tätigkeiten und Kooperation der Arbeitskräfte durch Dritte, von außen, durch Organisation und Technisierung zu programmieren – gleichsam fernzusteuern.“ [9]

Durch die Zerlegung der Arbeit in kleinste Handgriffe und das Herausziehen sämtlicher Intelligenz aus der Produktion in die Hierarchie von IngenieurInnen, ArbeitsvorbereiterInnen und MeisterInnen glaubten die ApologetInnen des Taylorismus, den Arbeiter / die Arbeiterin in eine „rational“ zu planende „Maschine“ verwandeln zu können und auf die autonome Intelligenz der Arbeiter nicht mehr angewiesen zu sein. Dieser Glaube entsprach aber nie der Realität. Selbst in den maximal entmenslichten Produktionsformen des Fließbands ist nämlich im tatsächlichen Produktionsprozess die kapitalistische Bürokratie darauf angewiesen, dass die ProduzentInnen autonom kooperieren: „Unabhängig davon, wie vollständig Tätigkeiten durch technische und bürokratische Kontrolle vorgeschrieben, bewertet und sanktioniert werden, die Kontrolle kann und darf die spontane Kooperation der Arbeiter, die für die Produktion notwendig ist, nicht eliminieren ... Falls es dem Kapitalismus jemals gelänge, Arbeiter

auf konditionierte Automaten zu reduzieren, so würde dies den unmittelbaren Zusammenbruch des Produktionsprozesses bedeuten. Der Traum der Manager würde zum Alptraum. Der Kapitalismus ist abhängig von der schöpferischen Teilhabe der unmittelbaren Produzenten.“ [10]

Die hier beschriebene Problematik gilt notwendigerweise in noch größerem Umfang für Bereiche, bei denen es aus technischen oder historischen Gründen nicht möglich war, eine „Mechanisierung“ (oder wie auch oft gesagt wird, ein Engineering) der Vorgehensweisen einzuführen. Es stellt sich die Frage: Wie konkretisiert sich dieser allgemeine Widerspruch in einem Bereich, der hochkomplex ist und eine entwickelte Interaktion der Produzenten erfordert, nämlich der **Produktion von Software**? Insbesondere: wie gingen und gehen die großen multinationalen Konzerne mit dieser Situation um?

- Wie gelingt es dort, Kontrolle über die Produzierenden auszuüben?
- Wie hat sich diese Kontrolle in der Phase des Postfordismus entwickelt; welche neue Formen von Kontrolle wurden eingeführt?
- Worin zeigt sich die Autonomie der ProduzentInnen, worin ihre Heteronomie?
- Was treibt die Entwicklung voran: der Widerstand der Multitude, der nicht zu bändigende Erfindungsreichtum der ProduzentInnen, die Verwertungsgesetze des Kapitals?
- Was kann über die Effizienz der Produktion innerhalb der multinationalen Konzerne ausgesagt werden? Sind sie die hochproduktiven Maschinen des Kapitalismus?
- Schließlich: was kann dabei über die Entwicklung der Produktivkraft gelernt werden? Hat Slavoj Žižek recht?

Es existieren nur sehr wenige Untersuchungen über die neue qualifizierte Arbeit, sei es in der EDV, der Softwareindustrie oder der Kulturindustrie. Ich werde versuchen, den Bereich der EDV bzw. der Softwareentwicklung aus meiner Sicht vorzustellen. Notwendigerweise wird dabei eine sehr persönliche Sicht der Entwicklung entstehen; ich denke aber, dass solche „Geschichten“ notwendig sind, um die Situation in ihrer Vielfalt und Einheit erkennen zu können. Ich arbeite seit fast 30 Jahren in verschiedenen multinationalen Konzer-

nen, die Hard- und Software produzieren und hatte im Laufe meiner Arbeit mit einer Vielzahl anderer Firmen in Europa und Nordamerika Kontakt. Das ist eine gute Basis für eine Einschätzung der Situation und Entwicklung in dieser Branche.

Meine ersten Erfahrungen in EDV Firmen

In den 60er und 70er Jahren des vorigen Jahrhunderts wurden die großen privaten und staatlichen Organisationen wie Banken, Versicherungen, Krankenkassen und großen Industriebetriebe auf Elektronische Datenverarbeitung umgestellt. Unter anderem war feststellbar:

- Die Software wurde zusammen mit der Hardware angeboten.
- Eine sogenannte Rechenstunde an einem Rechner kostete ein Vielfaches der Arbeitszeit der EntwicklerInnen. In einem meiner ersten Projekte war z.B. das Verhältnis von Kosten der Rechenstunde zu den Kosten, mit welcher die Firma eine Arbeitstunde kalkulierte 4:1 und höher.
- Ein großer Teil der Software wurde mit jedem Projekt neu entwickelt. Bis auf die damals sehr dünne Schicht von Programmen direkt über der Hardware, dem Betriebssystem, wurde fast alles „neu“ erfunden. In dem Projekt, in dem ich meine ersten Erfahrungen machte (einem Bahnhofsinformationssystem) wurde von der Datenhaltung („Datenbank“) bis zum Bildschirmmaskensystem und der Aufbereitung der Ausdrucke alles im Entwicklungsteam neu programmiert.
- Da es noch keine oder nur sehr wenige EntwicklerInnen mit Fachabschluss in Informatik gab (HTL oder Hochschule) wurden AbsolventInnen aus anderen Fächern angelernt.
- Die Hardware und das Betriebssystem der Rechner waren von HerstellerIn zu HerstellerIn verschieden. Daher konnte ein einmal erstelltes System auf den Rechnern anderer Hersteller nicht ablaufen. Das erzeugte eine große Abhängigkeit der AnwenderInnen (meist selbst große Unternehmen wie Versicherungen und Banken) von den HerstellerInnen – ein de facto Monopol.

Alle diese Faktoren erzeugten eine Situation, in der die EntwicklerInnen der EDV-HerstellerInnen im Wesentlichen im Alleingang mit den AnwenderInnen zusammenarbeiteten, wenn nicht überhaupt der Großteil der Entwicklung von einigen „Gurus“ des Anwenders selbst durchgeführt wurde. Da ich in einer Abteilung angestellt war, die sich Softwareprojekte nach dem zu leistenden Aufwand bezahlen ließ oder die EntwicklerInnen (sogenannte DienstleisterInnen) überhaupt an andere Projekte vermietete, andererseits aber in den Projekten die Kosten der Softwareentwicklung im Verhältnis zu den Kosten der Hardware gering waren, waren die Kontrollen minimal. In mehr oder weniger selbstorganisierter Weise (oft ziemlich chaotisch) wurden Projekte mit einer extrem flachen Organisationsstruktur durchgeführt. Solange die Zeiten für die Entwicklung nicht völlig von den Erwartungen abwichen, hatten die EntwicklerInnen freie Hand. Da der Großteil der EntwicklerInnen „angelernt“ wurde und die Diskussion über Technologien in der Softwareentwicklung gerade erst begonnen hatte (meist noch in Hochschulen), waren die entwickelten Programme oft von geringer Qualität — viele eigene, nicht ausgereifte Ideen. Jeder hatte seinen eigenen Stil. Die einzelnen, persönlichen Konzepte passten nur wenig zusammen. Hauptsache war, dass die fertige Software einigermaßen funktionierte. Bis auf wenige Basiskurse wurde die Ausbildung dem „training on the job“ überlassen und deshalb auch viele Leute mit Hochschulausbildung angestellt, da angenommen wurde, dass diese gelernt hatten, selbstständig aus Büchern zu lernen. [11]

Die Hierarchie der großen multinationalen Konzerne hatte für die Softwareentwicklung wenig Bedeutung. Die Abteilungsleiter unterschrieben die Stundensätze und machten ungefähre Schätzungen der Kosten, die den EntwicklerInnen nicht mitgeteilt wurden (und diese auch nicht sonderlich interessierten). Auf softwaretechnischer Ebene wurde inhaltlich alles von den EntwicklerInnen untereinander oder unter Mitwirkung eines Gruppen- bzw. Projektleiters besprochen. Wichtig war natürlich der Kontakt zu den AnwenderInnen und dessen eigenen EntwicklerInnen – falls er welche hatte. Am Be-

ginn eines solchen Projektes wurde ein Konto eingerichtet und auf dieses mussten die EntwicklerInnen ihre Stunden schreiben, um so wenigstens eine Zuordnung der Arbeitskosten zu den KundInnen zu haben. Darüber hinaus gab es keine „kommerziellen“ Aspekte der Arbeit. Der Hauptmechanismus zur Disziplinierung der EntwicklerInnen war neben dem Eigeninteresse etwas zu schaffen, elegante Lösungen zu finden, dass der AnwenderInnen/KundInnen mit der entwickelten Lösung einigermaßen zufrieden sein mussten. Die Entwicklung von Software ähnelte einem klassischen Handwerksbetrieb, der eine Reihe von mehr oder weniger fähigen angelernten ArbeiterInnen anstellt. Sie hatte auch etwas von einer Manufaktur, in der noch jede HandwerkerIn seine eigenen, selbstgelernten Tricks und Vorgehensweisen hatte, die den ChefInnen unbekannt waren und die teilweise auch gut gehütet wurden [12]

Die „Softwarekrise“ und die Bürokratisierung der Softwareproduktion

Im Laufe der 70er Jahre wurde immer klarer, dass die Softwareentwicklung in einer technischen Wachstumskrise steckte. Die Komplexität der Programme nahm zu und die Entwicklungszeiten explodierten. Die Kosten der Software, die im Vergleich mit den Kosten der Hardware lange Zeit bedeutungslos gewesen waren, begannen an Bedeutung zu gewinnen. Vor allem wurden die Zeiten, die für die Entwicklungen benötigt wurden, immer länger, die KundInnen unzufrieden und die Kosten waren oft doppelt und mehrfach so hoch wie vorher geschätzt. Und schon bald gab es für dieses Phänomen auch einen Namen: die Softwarekrise. [13] In einer Konferenz über diese Krise bei Philips, an der ich damals teilnahm, herrschte eigentlich Ratlosigkeit. Die Softwareleute versuchten verständlich zu machen, warum Softwareentwicklung eigentlich so komplex ist und sich schlecht planen lässt. Die HardwareentwicklerInnen hatten dafür keinerlei Verständnis und meinten, die SoftwareentwicklerInnen sollten sich einmal ordentlich hinsetzen um wie in anderen Disziplinen Engineering-Vorgehensweisen einzuführen. Seit dieser Zeit bis heute tobt die Diskussion, wie dieser

Prozess in den Griff zu bekommen ist. Das heißt natürlich auch bessere Verwaltbarkeit im Sinne von Kapitalverwertung. Die ManagerInnen, VerkaufschefInnen und zum Teil die ProjektleiterInnen lauschten gierig diversen Theorien, wie Softwareentwicklung in eine ganz normale Engineeringtätigkeit verwandelt werden kann.

Die späten 70er und der Beginn der 80er Jahre waren der Höhepunkt in der Welle der Bürokratisierung der Softwareentwicklung. Einerseits wurden Großprojekte wie z.B. computerbasierte Telefonzentralen entwickelt, bei denen Mannschaften mit mehreren hundert EntwicklerInnen eingesetzt wurden. Ohne genau definierte Schnittstellen zwischen den einzelnen Softwareteilen ist so eine Entwicklung nicht möglich. So wurden hierarchisch strukturierte, sehr bürokratisch aufgebaute EntwicklerInnenmannschaften gebildet, in denen der einzelne nur seinen eigenen Teil (Modul) und dessen Schnittstellen in die äußere Welt kannte. Trotz großer Schwierigkeiten gelang es teilweise, auf diese Weise diese komplexen Systeme zu entwickeln. Allerdings explodierten die Kosten, wie vorher niemand zu schätzen gewagt hätte. Trotz der (in der nicht austauschbaren Hardware versteckten) hohen Profite wurden schließlich auch für die großen multinationalen Konzerne diese Entwicklungen zu teuer. Mitte der 80er wurden z.B. bei Philips alle Entwicklungen in diesem Bereich eingestellt, nachdem vergeblich versucht wurde, mit anderen Multis zu kooperieren. [14]

Trotz diverser Bruchlandungen war diese Periode durch eine Zunahme an hierarchischer Planung gekennzeichnet. Es wurde vorgeschrieben, welche Projekte die (in den Zentralen) entstehenden Systeme, Datenbanken usw. verwenden müssen. In Firmen wie z.B. Nixdorf oder Siemens wurden langfristig verbesserte Versionen der Basissoftware geplant und mit bürokratischen Methoden die Qualität der Auslieferungen relativ gut gesichert. Basis dafür war, dass mit de-facto-Monopolen und entsprechenden Monopolprofiten – die einmal realisierten Projekte konnten nicht oder nur unter horrenden Kosten auf die Hardware und Systemsoftware andere Hersteller umgestellt werden – ein zwar extrem teurer, in einigen

Bereichen aber dennoch gut funktionierender bürokratisch organisierter Apparat aufgebaut werden konnte. Ende der 70er Jahre konnte bei einem Nixdorf oder Siemens Angestellten der Eindruck entstehen, dass alles, was für Computer erfunden wird, innerhalb des Konzerns entstanden ist. Die Forschungsergebnisse von Universitäten und anderen Firmen wurden für den firmeninternen Gebrauch an die spezielle Hardware und Systemsoftware angepasst und bei deutschen Firmen dazu auch noch eingedeutscht. Ich hatte zu diesem Zeitpunkt den Eindruck, überhaupt nicht zu wissen, wie die EDV-Welt außerhalb von Siemens aussieht. Ich arbeitete mit spezieller Hardware, speziellen Betriebssystemen und mit Computersprachen, die es nur bei Siemens gab.

Trotz der Vorgaben der Zentralen änderte sich außerhalb von Großprojekten in der Anwendungsentwicklung nur wenig. Die EntwicklerInnen waren weiterhin quer über alle Hierarchien auf die Kooperation mit ihren KollegInnen und auf ihre eigene Erfindungsgabe angewiesen. Bei den von der Zentrale vorgegebenen Produkten war es auch am besten, direkten Kontakt zu den Entwicklern der Produkte — sie saßen in der Regel in der eigenen Firma — zu haben. Die Disziplinierung der EntwicklerInnen änderte sich, im Vergleich mit der Zeit vor der Krise, nur wenig. Hauptsächlich war die KundInnenzufriedenheit maßgebend, wobei bei den KundInnen „Zufriedenheit“ auch einen weiten Interpretationsspielraum zuließ, da ja auch dort die Manager nicht genau wussten, was die Probleme und Sorgen ihrer Angestellten sind. [15] In großen Projekten konnte es schon auch passieren, dass ganze Gruppen ihre eigenen Wege gingen und Entwicklungen machten, von denen niemand wusste, wozu sie da sind.

Die de-facto-Monopole lösen sich auf

Was beendete eigentlich diese „beschauliche“ Entwicklung? Aus einer globaleren Sicht ist die Softwareentwicklung in den 70er und 80er Jahren nicht untypisch für die fordistische Phase des Kapitalismus nach dem zweiten Weltkrieg. Diese gar nicht „schlanken“ Apparate mit den Freiräumen,

die den Produzierenden zum Teil eingeräumt wurden, gab es in vielen Bereichen, in denen der Druck des Marktes wenig oder gar nicht spürbar war, oder die durch Monopolprofite gut finanzierbar waren. Weltweit sind die 80er Jahre durch das Ende der keynesianisch-fordistischen Regulationsweise und der Durchsetzung neoliberaler Politik gekennzeichnet. Eine wichtige Interpretation dieser Entwicklung ist, sie als erfolgreichen den Versuch zu sehen, trotz weiterhin sinkender Kapitalproduktivität (Geldwert der produzierten Waren / Geldwert des fixen Kapitals) die Profitrate wieder auf die alte Höhe zu treiben. [16] In einer konkreten Industrie wie der Softwareindustrie gibt es aber eine Vielzahl von Komponenten, die den Umschwung zu einem neuen Produktionsregime bewirken. Auch diese Einschätzung werde ich subjektiv aus einer Teilnehmerperspektive illustrieren.

Die „Revolution“ von unten: die Entwicklung der Mikrocomputer

Ende der 70iger Jahre begann ich für Philips in einem neuen Bereich zu arbeiten. In einer Soloaktion der österreichischen Philips wurde begonnen, unter Umgehung der Konzernspitzen und unter Verwendung der ersten am Markt (nicht in den Philips Forschungslabors) entstandenen Mikroprozessoren, einen Mikrocomputer zu bauen. Mit diesem Projekt wurde bei Philips eine Entwicklung aufgegriffen, die um 1976-77 herum vor allem in Kalifornien begonnen hatte. Dort hatte eine Vielzahl von BastlerInnen begonnen, mit einem, zuerst von Intel für ganz andere Zwecke gebauten Mikroprozessor, persönlich verwendbare Mikrocomputer zu bauen. Das ganze war eine „grass root“ Bewegung — sie entstand von unten auf der grünen Wiese. Zudem war sie über relativ lange Zeit mit vielen Vorstellungen und Ideen der 68er Bewegung in Kalifornien gekoppelt. So gab es Feste mit Rockbands, an denen die BastlerInnen ihre Entwicklungen vorführten. Auch die berühmten Apple Entwickler kamen aus dem Umfeld von ElektronikbastlerInnen, die versuchten, die ATT Telefonsysteme zu knacken und kostenlos um die ganz Welt zu telefonieren. [17] Während der ersten Jahre

dieser Entwicklung entstanden hunderte kleine und kleinste Firmen, viele mit neuen Ideen und einer kleinen Fangemeinde. Der Großteil dieser Firmen hatte keine oder wenige private GeldgeberInnen und die Interessen dieser „GründerInnen“ waren weitgehend dadurch bestimmt, ihre Konzepte realisieren zu können. Fremdes Geld spielte zu Anfang eher in der Form von Onkeln und Tanten als von professionellen InvestorInnen eine Rolle. Philips Österreich wollte hier mitmachen und hatte bei Microsoft, damals, Ende 1979, eine Firma mit 15 MitarbeiterInnen war, den Basic Interpreter lizenziert, der bei den BastlerInnen sehr beliebt war.

Trotz der sehr speziellen Welt der Elektronik- und Softwarefreaks war die Situation nicht untypisch für eine Anfangsphase beim Entstehen neuer Technologien und Produkte. underte von kleinen Firmen, die danach strebten, ihre Ideen zu realisieren und sich dabei auf ein Spezialpublikum stützten, das über den lokalen Markt die Produkte kaufte. Die kleinen Firmen hatten in der Regel gemeinsame EigentümerInnen, meist die freundschaftlich verbundenen EntwicklerInnen einer der interessanten Lösungen. In einem gewissen Sinn war das die „Wunderwelt des freien Marktes“: Kreativität, Kooperation, Wettbewerb — eventuell winkte sogar Reichtum. [18]

So wie Philips versuchten damals eine ganze Reihe von Konzernen von dieser Blüte zu profitieren und unter Ausnutzung ihrer Position den Markt zu übernehmen. Gelungen ist es nur einem Einzigen, nämlich IBM, der, gestützt auf seine damalige Vormachtstellung in der EDV, den Mikrocomputermarkt mit seinem Personalcomputer aufrollen konnte. Innerhalb weniger Jahre waren beinahe alle kleinen Hardwarefirmen verschwunden – Apple ist da eine Ausnahme, die ich hier nicht behandeln will. Es entstanden aber eine nicht weniger große Anzahl von Softwarefirmen, die Produkte für den nun vorherrschenden IBM Personalcomputer entwickelten.

Trotzdem führte die Entwicklung zum Ende des de-facto-Monopols in der alten Form nicht austauschbarer Hard- und Software. Das hatte folgende Gründe:

- IBM musste, um den Markt zu besetzen, fremde, nicht im eigenen Haus gefertigte Prozessoren verwenden – nämlich die von Intel.
- Auch das Betriebssystem konnten sie nicht selber entwickeln, sondern ließen mehrere Varianten von den einflussreichsten KleinanbieterInnen (unter ihnen Microsoft) entwickeln. [19]
- Da die generellen Konzepte von IBM nicht geschützt werden konnten – sie waren ja das Resultat einer von IBM völlig unabhängigen Entwicklung – konnten findige Nachbauer im sogenannten clean room Ansatz [20] die Hardware und Basissoftware nachkonstruieren (Compaq war der erste).

Resultat davon war, dass Firmen wie Compaq Hardware anbieten konnten, auf denen die ganze für den IBM-PC entwickelte Softwarepalette lauffähig war. Die Hardware verbilligte sich im Konkurrenzkampf sehr schnell und so begann sich erst bei den kleineren Firmen und im Privatbereich, später auch bei den großen Konzernen der Personalcomputer durchzusetzen. Erst dieser Prozess machte die Hardware zu einer Ware wie jede andere und gestattete es nicht mehr, die Kosten der Softwareentwicklung in den Monopolpreisen für die Hardware zu verstecken. Zwar wehrten sich Firmen wie IBM aber auch Siemens und Bull im Bereich der Großrechner zum Teil erfolgreich gegen diese Tendenz und erwirtschafteten in ihm weiterhin Sonderprofite, aber sie wussten, dass die Bedeutung der Großrechner für ihre Stärke am Markt immer geringer werden wird.

Der Markt für Software wird chaotisch

Erst durch die beschriebene „Kommodifizierung“ der Hardware, wurde die Software außerhalb des Großrechnerbereiches zu einer von der Hardware getrennten Ware mit einem eigenen Markt. Die bisherigen DominatorInnen des EDV-Marktes verloren daher im neuen, schnell expandierenden PC-Bereich ihre Sonderstellung. Wie bereits beschrieben, wurde der Softwaremarkt im PC-Bereich von Beginn an nicht von den EDV-Riesen, sondern von den kleinen innovativen Softwarefirmen beherrscht. [21] Im Zuge dieser En-

wicklungen mussten die großen EDV-Firmen feststellen, dass ihre Planungen nicht mehr funktionierten. Gab es bisher für die Produktentwicklung einigermaßen verlässliche Pläne, die über zwei, drei und mehr Jahre liefen, so vernichtete die einsetzende Konkurrenz und der Preisverfall alle Absichten einer geplanten Vorgehensweise. Bis auf ein bis zwei Firmen, die durch ihre dominante Stellung noch planen konnten, wurden die Treibenden zu Getriebenen. In den meisten Bereichen verkürzte sich der Planungshorizont auf 3-6 Monate. Weiters wurde versucht, sich auf jene Bereiche zu konzentrieren, die man noch zu dominieren glaubte.

Für die EDV-Firmen stellte sich daher, in noch wesentlich gravierenderer Weise als vorher, die Frage, wie sie unter diesen neuen Bedingungen konkurrenzfähige Softwareprodukte zu „attraktiven“ Preisen produzieren sollten, mit denen dann die AnwendungsentwicklerInnen unter Verwendung der möglichst firmeneigenen Hardware, bei den KundInnen Geld verdienen sollten. Eine der Strategien des Managements war eine noch stärkere Bürokratisierung und Hierarchisierung der Produktion, um billig genug für die neuen Marktverhältnisse zu sein:

- Wenige und teure „SpezialistInnen“ sollten die Konzepte mit den AnwenderInnen oder den AuftraggeberInnen diskutieren und Pflichtenhefte sowie Spezifikationen über die Softwarearchitektur schreiben.
- Danach sollten in einem eigenen Schritt von den schon etwas zahlreicheren AnalytikerInnen Detailspezifikationen erarbeitet werden,
- Schließlich sollte die Software dann von einer Heerschar relativ billiger ProgrammiererInnen realisiert werden – wobei letztere auch aus Niedriglohnländern wie Slowakei, Russland oder Indien kommen konnten. [22]
- Damit dieses Vorgehen überhaupt eine Chance hatte, mussten dicke Bände von Vorgehenshandbüchern und Hilfsprogramme geschrieben werden, die es gestatteten, alle Artefakte zu verwalten.

Daneben wurde, der in den 80er und 90er Jahren vorherrschenden Ideologie der „Vermarktung“ der Verhältnisse auch in den Betrieben folgend, sowie

um Profitabilität überhaupt zuordenbar zu machen, jede Abteilung in „profit- oder cost-center“ verwandelt.

- Jede Abteilung hat auf diese Weise ihre „KundInnen“ und „LieferantInnen“, die meist firmenintern sind.
- Den „KundInnen“ werden von den „LieferantInnen“ Kostenvoranschläge mit entsprechender Kalkulation gemacht. Ein „profit-center“ hat dabei die Aufgabe, den Profit – auch bei internen KundInnen – zu maximieren. Ein „cost-center“ hat die Aufgabe, seine „Preise“ so zu kalkulieren, dass seine Kosten abgedeckt sind.
- Die „KundInnen“ dürfen sich die „LieferantInnen“ frei aus den firmeneigenen Angeboten und denen am Markt aussuchen. Auf diese Weise soll auch innerhalb der Firma Konkurrenz erzeugt werden.

Wie wirkten sich die Maßnahmen in ihrem Zusammenspiel aus? Dazu zwei Beispiele aus ganz verschiedenen Bereichen:

Das erste Beispiel ist eine große Entwicklungsabteilung von Siemens in München. In dieser Abteilung waren in den 80er Jahren über 3000 Menschen beschäftigt. Sie stellten vor allem verschiedene Softwareprodukte her, auf deren Basis dann in Projekten konkrete Anwendungslösungen gebaut wurden. Die interne Struktur der Abteilung war, so wie oben beschrieben, hierarchisch, wobei hauptsächlich EntwicklerInnen am Standort eingesetzt wurden. Nur speziell ausgliederbare Teile wurden in anderen Ländern entwickelt. Die konkreten Kosten und die Profitstruktur (welche Abteilung mit ihren Produkten Gewinne macht und welche nicht) war dem Topmanagement weitgehend unbekannt. Früher war Software Teil des Hardwareangebots, musste daher in jeden Fall genommen werden und kostete eben soviel wie der Entwicklungsaufwand war. Die Software kam aber im Laufe der oben beschriebenen Entwicklung in Konkurrenz zu von relativ kleinen Firmen am Markt angebotenen Produkten. Als ein erster Schritt wurde eine Reihe von profit- und cost-centern eingerichtet, um überhaupt Kosten zuordnen zu können. Im weiteren war geplant, aussichtsreiche und erfolgversprechende Abteilungen weiterbestehen zu lassen und die an-

deren aufzulösen. Ein neuer Manager wurde eingesetzt, der die „guten“ von den „schlechten“ Produkten trennen, und eine neue generelle Perspektive entwickeln sollte. Das kommerzielle Ziel war, innerhalb von 2 Jahren schwarze Zahlen zu schreiben.

Die interessante Frage ist: Konnten die genannten Maßnahmen unter Beibehaltung der bürokratischen Struktur in einem auf den Erfindungsreichtum der EntwicklerInnen angewiesenen Produktionsbereich greifen? Tatsächlich scheiterte die Umstrukturierung kläglich. Erfolgreich war man in der Anfangsphase mit dem Schließen von Abteilungen, die Verluste machten und bei denen es keine Perspektive gab, diese Situation zu ändern. Daher sanken die Kosten des Bereichs. Das Problem, das in diesem und auch in den anderen mir bekannten Fällen auftrat, ist, dass nicht nur die hierarchisch konzipierten Abteilungen in ihrer Entwicklungsweise hohe Kosten verursachten, sondern auch, dass zwar „abgestorbene Äste“ liquidiert werden konnten, aber „wachsen“ mussten die Äste letztlich selber. Die EntwicklerInnen müssen gute Konzepte haben. In diesem Fall glaubte der Manager mit einigen Marketingideen einen Aufschwung lancieren zu können. Die eigenen GruppenleiterInnen und EntwicklerInnen hielten nicht viel davon, taten aber auch nichts dagegen. Letztlich änderte sich nichts an den roten Zahlen; die gesamte Abteilung wurde aufgelöst und der leitende Manager verließ die Firma. Die Folge bei Siemens war, dass die Entwicklung von Softwareprodukten, soweit nicht weiterhin eingebunden in Spezialhardware wie Mobiltelefone oder medizinische Apparate, völlig eingestellt wurde. Das Topmanagement erklärte, dass eine Firma wie Siemens keine Softwareprodukte herstellen könne und dass das andere am Markt besser könnten.

Das zweite Beispiel war mein eigener Arbeitsbereich in Österreich. Diese Entwicklungsgruppe war für eine große Firma untypisch. Unsere Entwicklung ähnelte der eines kleinen Softwarehauses. Ein durchsetzungsstarker Manager meinte eine gute Idee für ein Softwareprodukt zu haben und konnte in Zeiten großer Unsicherheit, wie es weitergehen soll, relativ große Mittel aufreiben, um das Produkt zu real-

isieren. Da er den großen Entwicklungsabteilungen misstraute, wurde die Entwicklung nahe am Vertrieb in einem kleinen Team begonnen. Die beteiligten EntwicklerInnen waren zwar nicht überzeugt, dass die Produktidee sich technisch und kommerziell erfolgreich realisieren lässt, aber sie waren erfreut, in einem völlig unbürokratischen Team, weitgehend selbstorganisiert, modernste Softwaretechnologien auszuprobieren und anwenden zu können. Im Verhältnis zu üblichen Entwicklungszeiten in großen Firmen war die Dauer für die Produktentwicklung kurz und das Ergebnis konnte technologisch mit den besten Entwicklungen Schritt halten. Leider stellte sich die Produktidee selber – wie von den Entwicklern von Beginn an vermutet — als Flop heraus. Nach einigen Versuchen der Korrektur, die zum Teil auch gelangen, wurde das Produkt schließlich erfolgreich, wenn auch nur an wenigen Orten eingesetzt. Leider kam das Projekt daraufhin in den Gegenwind der Firmenpolitik der Zentrale: Die Produktfamilie an der wir arbeiteten, wurde vollständig eingestellt bzw. verkauft, die EntwicklerInnengruppe aufgelöst. Generell zog sich die Firma auf den sogenannten Dienstleistungsbereich zurück und versuchte, unter Ausnutzung des Firmennamens Siemens und der Verwendung von am Markt gekauften Produkten, Projekte für und mit Kunden zu realisieren oder oft auch nur EntwicklerInnen, AnalytikerInnen usw. für KundInnenprojekte zu vermitteln.

Auswirkungen auf die EntwicklerInnen

Nachdem sich lange Zeit die konkreten Verhältnisse für die EntwicklerInnen nicht gravierend änderten, brachte die Änderung der Marktverhältnisse, die ich am Beispiel des Debakels in der Herstellung von Produkten aufzeigte, innerhalb von kurzer Zeit gewaltige Umstellungen. Heute schauen die Verfahren zur Sicherstellung der Profitabilität folgendermaßen aus:

- Den EntwicklerInnen wird beinahe im Wochenrhythmus der aktuelle Stand der Abteilung in Richtung Gewinn, Verlust, Auslastung bekanntgegeben.
- Eine Unmenge von Initiativen und

Programmen wird gestartet, um die Angebote gewinnsicherer und die Projekte besser steuerbar zu machen. In diesen Programmen geht es zu keinem Zeitpunkt um bestimmte inhaltliche Verfahren oder softwaretechnische Lösungen, sondern ausschließlich um kommerzielle Faktoren bzw. Frühwarnsysteme, um frühzeitig Maßnahmen treffen zu können, wenn Projekte in Gefahr kommen, Verluste zu machen.

- In großen Veranstaltungen werden die EntwicklerInnen mit diesen Programmen betrauert und dem kommerziellen Kauderwelsch ausgesetzt – was zum Teil auch Wirkung zeigt.
- Letztlich wurden neue Arbeitsverträge, sogenannte Beteiligungsverträge, mit dem Zuckerl einer Lohnsteigerung angeboten. In diesen werden ca. 70% des vorherigen Bruttogehalts als Fixgehalt festgelegt. Der Rest wird in Form von Beteiligungen ausgezahlt, wobei der Gewinn der Firma, der Abteilung, in erster Linie aber die selber geleisteten produktiven Stunden – das sind diejenigen Stunden für die „KundInnen“ (im bereits genannten Sinn) bereit war, zu zahlen – maßgeblich sind. Um 100% zu erreichen sind also eine gewisse Zahl von produktiven Stunden im Jahr notwendig, werden diese überschritten steigt der ausgezahlte Betrag rasch über 100% an; umgekehrt sinkt er. Zeitaufzeichnungen außerhalb der Aufzeichnung produktiver Stunden fallen weg. Es gibt weder Überstunden noch Zeitausgleich.

Von einem klassischen Lohnarbeitsverhältnis bleibt dabei nicht viel über:

- Der Arbeitsplatz ist meist nicht in der Firma, sondern dort, wo das Projekt läuft, oder wohin man/frau vermietet wurde. In die Firma kommt man/frau ein, zwei Mal in der Woche.
- Die gesamte technische Seite der Arbeit lastet auf den Schultern der Projektmannschaft oder des einzelnen Entwicklers.
- Schon der Vorgesetzte hat kaum eine Ahnung, um welche konkreten technischen Probleme es geht.
- Oft verschaffen sich die EntwicklerInnen selber den Folgeauftrag, der nur mehr in den kommerziellen Details

von den Vorgesetzten ausgehandelt wird.

Die Firma bekommt — überspitzt ausgedrückt — die Position einer Dienstleisterin für die EntwicklerInnen, die fallweise einen Arbeitsplatz zur Verfügung stellt und mittels des Vertriebsapparates den EntwicklerInnen beim Akquirieren von Aufträgen hilft. Die gesamte Situation stellt sich jetzt häufig so dar, dass die KundInnen Projekte bei einem Multinationalen Konzern bestellen, die auf Produkten beruhen, welche von einer Softwarefirma mit vielleicht 20 – 50 Beschäftigten erstellt wurde. Der Multi hat, wenn er Glück hat, sogar Leute, die das Produkt kennen und nicht erst kennen lernen müssen. [23]

1. Schon immer musste den Produzierenden in gewissem Ausmaß Autonomie zugestanden werden, bei der Produktion von Software ist das offensichtlich

Von der notwendig vorhandenen Autonomie innerhalb des fremdbestimmten Produktionsprozesses wurde schon in der Einleitung gesprochen: *„Demnach muss sich das Arbeitshandeln zum einem an fremdgesetzten Zielen und Methoden und zum anderen an der objektiven Logik des Arbeitsprozesses orientieren.“ Es muss „eine Stofflichkeit und Eigenlogik von Arbeitsprozessen (postuliert) werden, die als solche kein ‚formalisierbarer‘ Organisationsbestand ist und daher weder vom Kapital noch vom Management ‚besessen‘ und restlos instrumentiert werden kann.Diese arbeitsstoffliche Basis und notwendige Prozessstruktur setzt objektiv Ansprüche an Selbsttätigkeit, da die Prozessziele sonst nicht zu erreichen sind.“* [24]

Die notwendige Autonomie der Produzierenden ist bei den genannten komplexen Arbeiten offensichtlich. Eine Unzahl an Entscheidungen muss von den Entwicklungsmannschaften autonom, ohne jeden Einfluss des Managements, getroffen werden. Die Versuche, den Produktionsprozess von Software zumindest teilweise zu automatisieren, um so die Abhängigkeit von den EntwicklerInnen zu verringern, sind gescheitert. Ohne die Kreativität und den Einsatz der ProgrammiererInnen ist jedes Softwareprojekt zum Scheitern verurteilt.

2. Die „Vermarktlichung“ der internen Strukturen hat gravierende Auswirkungen auf die Arbeit in der EDV-Industrie

Relativ lange waren die EntwicklerInnen selber von den Umstrukturierungen der Industrie wenig betroffen. Zwar gab es in den Zeiten des Booms die Tendenz unter den EntwicklerInnen, sich selbstständig zu machen und als Freelancer mit Zeitkontrakten zu arbeiten, da es sich so besser verdienen ließ, aber die Kernmannschaften hatten weiterhin wenig mit Themen wie der Profitabilität der Abteilungen, Gruppen und der einzelnen EntwicklerInnen zu tun.

In den letzten Jahren hat aber der „Postfordismus“ gerade in der Softwareindustrie in vollem Umfang Einzug gehalten. Die Mechanismen der Beteiligungsverträge unterlaufen die in der Softwareindustrie immer schon schwach entwickelten Arbeitszeitgesetze und nähern die Angestellten den Freelancern an. Zum Teil entwickeln sich die Auseinandersetzungen um Kontrolle und Autonomie nun auf neuem Terrain. Auch mit dem Mittel der Abrechnung ausschließlich produktiver Stunden gelingt es nicht, die Eigeninitiative der EntwicklerInnen abseits von Verwertungsinteressen zu unterdrücken. So machen die EntwicklerInnen für die KundInnen mehr als die sie beschäftigende Firma für unbedingt notwendig erachtet – sie wollen einfach eine sinnvolle Lösung bauen – und verbrauchen dadurch mehr produktive Stunden als vorgesehen. Letztlich sieht sich die Geschäftsleitung dann genötigt, mit Appellen die EntwicklerInnen zum sparsamen Verbrauch produktiver Stunden zu bewegen. [25] Wie sich die neue Situation unter krisenhaften Verhältnissen, wie sie zur Zeit in der EDV-Industrie herrschen, auswirken wird, bleibt abzuwarten.

3. Läuft die Entwicklung in eine falsche Richtung, dann scheint nur Zusperrern als Lösung erfolgreich zu sein

Meine Erfahrungen zeigen, dass es den bürokratischen Organisationen im Regelfall nicht gelingt, Dinge, die global, das heißt von der Führung her, schief laufen, zu korrigieren. Es gibt natürlich Beispiele für große Firmen, die in der Lage waren, ihren Kurs zu ko-

rrigieren, wie z.B. Microsoft, die lange Zeit das Internet verschlafen hatten. Aber das lag dann an Einzelpersonen, wie in diesem Fall Bill Gates, der nicht nur die bürokratische Macht hatte, sondern auch eine „Vision“ durchsetzen konnte. Dies war nur möglich, weil die EntwicklerInnen Vertrauen in seine kommerzielle und technische Urteilsfähigkeit hatten.

Die übliche Entwicklung war so, wie ich sie z.B. bei Philips erlebte:

Mit viel Talent wurde ein neuer Personalcomputer entwickelt. Die Konzepte waren gut, die TechnikerInnen hatten ihr ganzes Wissen eingebracht. Den TechnikerInnen wurde während der Fertigstellung aber klar, dass die Maschine keine Chance auf Erfolg hat, wenn sie nicht steckerkompatibel in Hard- und Software zum sich als Standard entwickelnden Personalcomputer (damals von IBM dominiert) produziert wird. Auch ein Marktforschungsinstitut bestätigte diese Ansicht. Die EntwicklerInnen meinten, dass sie weitere sechs Monate benötigen, um dieses Ziel zu erreichen. Für die unmittelbar für die Entwicklung verantwortlichen ManagerInnen hätte das bedeutet, für weitere sechs Monate Budgets aufzreiben zu müssen und vom Topmanagement für diese hohen Verluste verantwortlich gemacht werden zu können. Daher wurde keine Entscheidung getroffen und weiter gemacht wie bisher, mit dem Endresultat, dass die Vorhersage der EntwicklerInnen eintraf und die Maschine sich nur schlecht bis gar nicht verkaufte. Schließlich wurde dann 18 Monate später konsequenterweise die ganze Entwicklung liquidiert, da es nicht gelungen war, schwarze Zahlen zu schreiben.

4. Die bürokratisch-hierarchische Organisation dient der Kapitalherrschaft, ihre „Produktivität“, selbst im Sinne der Kapitalverwertung, ist zweifelhaft

Es gibt eine Reihe von arbeitssoziologischen Untersuchungen, die zu zeigen versuchen, dass die hierarchische Organisation insgesamt keinerlei produktiven, sondern ausschließlich Herrschaftssinn hat: *„Der soziale Sinn hierarchisierter Arbeit liegt nicht in der technischen Rationalität, sondern im Akkumulationsinteresse des Kapitals.“* [26] Weiters schreibt Marlin: *„Kapitalistische Ar-*

beitsteilung, wie Adam Smith sie am berühmten Beispiel der Stecknadelmanufaktur darstellte, war nicht das Ergebnis der Suche nach einer technologisch überlegenen Arbeitsorganisation, sondern nach einer, die dem Unternehmer die zentrale Rolle im Produktionsprozess verschaffen sollte: Er integrierte die Teilarbeiten seiner Arbeiter zu einem marktwirksamen Produkt.“ Oder wie eine Untersuchung über das Entstehen von Arbeitsstrukturen in der US-amerikanischen Stahlindustrie des letzten Jahrhunderts zeigt: „Die Technologie bringt kein Arbeitssystem hervor. Technologien definieren nur einen Bereich von Möglichkeiten ... Die Entwicklung der Hierarchie innerhalb des Arbeitspersonals war keine Antwort auf anwachsende Komplexität von Arbeiten, sondern sie beruhte auf der Absicht, die entstehende Homogenität der Industriearbeit aufzuheben ... Die heutige Teilung in geistige und körperliche Arbeit ist eine künstliche und unnötige Auftrennung, die nur dazu dient, die Macht der Arbeitgeber über die Arbeiter zu behaupten.“ [27]

Interessant in diesem Zusammenhang ist eine neue Vorgehensweise in der Produktion von Software, die sich „eXtreme Programming“ nennt. Sie wurde von einzelnen, selbständigen EntwicklerInnen und BeraterInnen erarbeitet, die genug von den Pseudoargumenten der ManagerInnen hatten, die immer wieder mit neuen hierarchischen Tricks versuchten, den Produktionsprozess zu kontrollieren. Die Prinzipien des „eXtreme Programming“ können hier im Detail nicht erklärt werden, wesentlich sind aber folgende Punkte:

- Die EntwicklerInnen/ProgrammiererInnen selber werden ins Zentrum gerückt. Es gibt keine Organisation von Spezifizierern über und keine Programmiersklaven unter ihnen.
- Die Entwicklung wird von Anfang an als kooperativer Prozess verstanden, daher arbeiten während der Programmierung immer zwei Leute zusammen (pair programming). Auch die Planung wird als gemeinsame Aktivität verstanden, in der neben den EntwicklerInnen nur die AnwenderInnen sitzen (planning game).
- Die Komplexität des Entwicklungsvorganges wird anerkannt und es wird versucht, ihn durch extrem kurze Zyklen (incremental develop-

ment) von 2-3 Wochen gemeinsam zu steuern. Das heißt, dass alle 2-3 Wochen ein Entwicklungsstand an die AnwenderInnen übergeben werden kann, an dem dieser die Resultate der Entwicklung mit seinen Vorstellungen vergleichen kann.

Die großen Firmen sind allerdings trotz besseren Wissens nicht in der Lage diesen Entwicklungsprozess einzusetzen, da er sowohl mit der Firmenstruktur als auch mit ihren Vermarktungsbedürfnissen kollidiert. [28]

Auch die free und open Software Bewegung zeigt, dass es ohne bürokratisch-hierarchische Organisation möglich ist, hervorragende Software wie das Betriebssystem Linux zu realisieren. Die Erfolge dieser erst mit dem Internet möglichen Form von produktiver Arbeit und die Bedeutung, die sie politisch hat, können hier aber nicht besprochen werden.

5. Die „Überlegenheit“ des Kapitalismus gegenüber dem „realen Sozialismus“ lag nicht an der hohen Produktivität der multinationalen Konzerne

Für mich stellte sich damals, nach den Erfahrungen bei Siemens und Philips — so um 1984 — jedenfalls die Frage, wie es die Konzerne im Ostblock schaffen konnten, noch unfähiger und unorganisierter als die westlichen multinationalen Konzerne zu sein. Tatsächlich unterschieden sich die internen bürokratischen Strukturen der Firmen im Osten und Westen nur geringfügig. Sie waren beide dadurch gekennzeichnet, dass die Arbeit letztlich fremdbestimmt war. Die ProduzentInnen in beiden Welthälften wussten meist, woran es krankte, aber sie hatten im Rahmen des Systems weder das Interesse noch die Macht, die Verhältnisse zu ändern.

Eine Diskussion darüber, welche Faktoren letztlich die Konkurrenz zwischen Ostblock und westlichen Kapitalismus am Weltmarkt zugunsten letzterem entschied, kann hier nicht geführt werden. Soviel ist jedenfalls ist sicher: die bürokratische Ineffizienz, der aufgeblähte hierarchische Apparat und die Unfähigkeit Fehler zu korrigieren kennzeichnen sowohl die „real sozialistischen“ als auch die kapitalistischen Konzerne.

6. Der Kommunismus wird die „Hauptproduktivkraft – den Menschen selbst“ entwickeln

Die Beispiele zeigen, dass Konzepte zur Überwindung des Kapitalismus, die nicht den Hauptaspekt auf die Selbsttätigkeit der (Re)Produzierenden legen, keine Zukunft haben. Der Kommunismus muss diese Barrieren der Produktionsverhältnisse für die Produktivkräfte beseitigen. In diesem Sinne hat Slavoj Žižek trotz seiner berechtigten Kritik an den in den sozialistischen und kommunistischen Weltbewegungen vorherrschenden Form, Produktivkräfte zu denken, nicht recht.

[1] Karl Marx, Zur Kritik der politischen Ökonomie, MEW Band 13, S.9

[2] Joachim Hirsch, Postfordismus, Dimensionen einer neuen kapitalistischen Formation, aus trend.partisan.net 04/2002

[3] Karl Marx, Grundrisse der Kritik der politischen Ökonomie, S. 325

[4] Slavoj Žižek, Das fragile Absolute. Gebundene Ausgabe, Volk und Welt, Berlin 2000, S. 17-19

[5] Der dazu passende Witz ist: Nachdem der Redner die Losung „den Kapitalismus einholen und überholen“ erklärt hatte, fragt jemand aus dem Publikum: „Wenn wir meinen, dass der Kapitalismus sich in Richtung Abgrund entwickelt, warum wollen wir ihn dann einholen und überholen?“

[6] Siehe dazu im Anhang einige Referenzen zu aus dem Internet erhältlichen Untersuchungen.

[7] Das „Kapital“ von Marx ist neben der Entwicklung des Verwertungsprozesses voll von Analysen der (bürokratischen) Herrschaft über die Arbeit und die ArbeiterInnen, wenn auch zu Recht von vielen WissenschaftlerInnen angemerkt wird, dass erst mit Foucault und anderen ein besseres Verständnis der Art von Herrschaft in Organisationen wie der Fabrik, dem Büro etc. entstand.

[8] Die Motivation für diese Wahl war, dass die linken Industriesoziologen nach dem Entstehen des Klassenbewusstseins in den Produktionsbetrieben forschen wollten und die Autoindustrie ihnen dafür am geeignetsten schien.

[9] Harald Wolf, Prokrustes-Revolutionen und das Gespenst der Autonomie, Labour-

Net Germany

[10] Michael Burowoy, *Fabrik und Staat im Kapitalismus und im Sozialismus*, In: *Das Argument*, Nr. 140, S 508-524

[11] Einer der Entwickler mit denen ich zusammenarbeitete, war ursprünglich Assistent für Althilologie an der Universität gewesen, ein anderer hatte eben seine Dissertation aus Psychologie beendet. Danben wurden HTL AbsolventInnen diverser Richtungen angeleert. (Die Firma nannte das: „einen Mix einstellen“.)

[12] Einer der Entwickler in meiner ersten Firma sammelte die eigene Dokumentation der Programme zu Hause und da er auch noch der einzige war, der sich in den (chaotischen) Programmen auskannte, machte er sich selbstständig und verlangte von der Firma hohe Stundensätze zur Wartung der eigenen Software. Die Firma beugte sich diesen Bedingungen und versuchte aber so schnell wie möglich, das heißt innerhalb von 2-3 Jahren, die vorhandene Lösung zu ersetzen. Die notwendigen Umstellungen von Programmen, um sie für das Jahr 2000 geeignet zu machen, erforderte deshalb auch öfters kuriose Praktiken. So wurden EntwicklerInnen aus der Pension oder sogar aus dem Altersheim geholt, damit sie ihre Routinen für das Jahr 2000 anpassen.

[13] Das „Kultbuch“ dieser Zeit, das auch noch heute in vielen Aussagen Gültigkeit hatte, war „The mythical man month“ des Entwicklungsleiters des IBM 370 Betriebssystemes Frederick P. Brooks Jr.. Die horrenden Kosten dieser Entwicklung wurden nämlich um ein mehrfaches der ursprünglichen Schätzung überschritten. Das Buch macht sich über die Idee lustig, dass man nur mehr Leute in ein Projekt hineinstecken muss, um die Entwicklungszeiten zu verkürzen.

[14] Was mir bei der Innensicht über diese Monsterprojekte auffiel: Wie konnte man in der kommunistischen Weltbewegung glauben, dass man mit bürokratisch organisierter Planung die Produktion einer ganzen Gesellschaft organisieren kann, wenn man nicht einmal mit gut motivierten und bezahlten Programmierern die Software einer Telefonzentrale entwickeln kann, ohne alle Budgets zu sprengen und alle Zeitvorgaben über den Haufen zu werfen – auch wenn Softwareentwicklung in dieser Größenordnung sicher zu den komplexesten Tätigkeiten von Menschen in Teams zählen.

[15] Ich entwickelte z.B. für eine Versicherung ein System zum Erfassen der Millionen alten Karteikarten. Dafür wurden eine Reihe von Frauen eingestellt, die diesen monotonen Job 2- 3 Jahre tun sollten.

Beim Jahreswechsel stellte sich eine der Datumsroutinen von mir als fehlerhaft heraus und die Frauen konnten nichts mehr eingeben. Ich war aber auf Urlaub und 10 Tage nicht erreichbar. Die Enttäuschung war groß, als ich nach dem Urlaub den Fehler behob und die Frauen statt Bücher lesen zu können wieder Karteikarten erfassen mussten. Zur damaligen Zeit hatte ein solcher Programmierfehler und meine Nichterreichbarkeit wenig Auswirkungen. Ich erinnere mich nicht, dass mir irgendwer Vorwürfe machte und insgesamt wurde das Projekt als sehr erfolgreich eingeschätzt.

[16] Siehe dazu z.B. Alain Lipietz, *The Fortunes and Misfortune of Post-Foredooms*, http://perso.club-internet.fr/lipietz/INT/INT_FordistAll.html

[17] Steve Jobs, der kommerzielle Kopf von Apple, war z.B. früher Mitglied einer buddhistischen Kommune gewesen und ging noch während er ersten 5 Jahre in der Firma ausschließlich barfuß. Sein Partner Wozniak, der eigentliche technische Kopf, weigerte sich lange Zeit, irgendwelche Managementaufgaben zu übernehmen und zog es vor, weiterhin halbtags für Apple zu arbeiten und sonst seinen eigenen Interessen nachzugehen.

[18] Diese Phasen mit von unten entstehenden neuen Kleinunternehmen mit neuen Konzepten sind es die der kapitalistischen Propaganda des „freien Marktes“ eine gewisse Glaubwürdigkeit geben. Sie sind das Abziehbild, das in den Köpfen vieler Menschen als die „schöne Seite“ des Kapitalismus erlebt werden, auch wenn sie selber nie eine Chance hatten dabei zu sein. Sie sind die „andere“ Seite der „hässlichen Monopole und multinationalen Konzerne“, die „nur über ihr Monopol oder ihre Marktmacht“ Geschäfte machen.

[19] Die Mikrocomputerfirmen, auch die einflussreichsten, hatten damals kaum mehr als 30–40 Angestellte. Eine Ausnahme war Apple.

[20] Dabei sitzen unter Aufsicht eines Notars / einer Notarin zwei Gruppen von EntwicklerInnen in zwei getrennten „Räumen“. Im ersten wird die Hardware und/oder Software eines Produktes analysiert und danach dokumentiert. Im zweiten „Raum“ sitzen EntwicklerInnen,

die nachweislich das Produkt und seine Originaldokumentation nicht kennen, und realisieren unter Verwendung der im ersten „Raum“ erstellten Spezifikation ein „stecker- und softwarekompatibles“ Gerät. Das ist keine Patenverletzung, da bei Software nur das konkrete Programm geschützt ist. Ein im „clean room“ entwickeltes kompatibles Programm, bei der die EntwicklerInnen die Originalquelle nicht kennen, ist deshalb patentrechtlich erlaubt. Bei der Hardware ist es ähnlich.

[21] In extrem kurzer Zeit entstanden auf diesem Markt neue Monopole wie jenes von Microsoft, dessen Wert auf den Aktienmärkten alle Großkonzerne übertrifft. Es wäre interessant zu untersuchen, warum gerade der Softwarebereich für Monopolbildungen besonders anfällig ist.

[22] Diese Vorgehensweise wurde noch vor der „Internetrevolution“ eingeführt. Das Internet mit seinen verbesserten Kommunikationsmöglichkeiten über weite Distanzen erleichterte die Durchführung solcher „verteilter“ Projekte und macht Länder wie Indien als Entwicklungsstandort überhaupt erst möglich. Gleichzeitig verstärkte das Internet aber auch die Illusionen des Managements über die Effizienz solcher Vorgehensweisen und führte zu einer nicht unerheblichen Anzahl von gescheiterten Projekten.

[23] Die interessante Frage ist jetzt, welche Vorteile dann eine große Firma überhaupt noch gegen eine kleine Softwarefirma hat. Ein Vorteil ist, dass die große Firma eher in der Lage ist Risiko zu übernehmen und Projekte zu Fixpreisen anzubieten. Durch den gewaltigen Apparat, den die große Firma dafür benötigt, wird das aber für den Kunden eine teure Angelegenheit.

[24] Harald Wolf, *Arbeit und Autonomie*, Westfälisches Dampfboot 1999, S 99

[25] Die KundInnen haben in der Regel eine fertige Funktionalität (Fixpreis) gekauft und unterschreiben die für sie geleisteten produktiven Stunde gerne, solange das in ihrem Interesse ist.

[26] St.A. Marlin, *Was tun die Vorgesetzten? Ursprünge und Funktionen der Hierarchie in der kapitalistischen Produktion*. In: F. Duve (Hrsg.): *Technologie und Politik*, Bd. 8, Reinbeck 1977, S. 148 - 203

[27] Katherine Stone, *The origins of job structure in the steel industry*, zitiert nach Klaus Türk, *Die Organisation der Welt*,

Westdeutscher Verlag 1995, S 51-52

[28] Eine kanadische Kleinfirma, die dieses Vorgehen in der Softwareentwicklung angewandt und schließlich von IBM gekauft wurde, ließ IBM weiterhin ihre völlig unabhängige Struktur, da herausgefunden wurde, dass die Kleinfirma bei gleichen Problemstellungen fast um den Faktor 10 schneller war, um eine bestimmte Qualität und Funktionalität zu entwickeln, und dieser Vorteil bei Eingliederung verloren gehen würde. Das ist auch einer der Gründe, warum die open und free software Bewegung mit Linux und seinen Entwicklungen von IBM, Oracle etc. nicht einfach aufgekauft wird.

Anhang

Einige Papiere zur Frage von Heteronomie und Autonomie in der Produktion und die Auswirkungen der neuen Formen von Arbeitsorgani-

sation:

- Der Artikel „Prokrustes-Revolutionen und das Gespenst der Autonomie, Über den ‚neuen Geist des Kapitalismus‘ und seine Widersprüche“ von Harald Wolf befindet sich in: express – Zeitung für sozialistische Betriebs- und Gewerkschaftsarbeit, 3/2001 unter: <http://www.labournet.de/diskussion/wipo/wolf.html>
- Den Begriff des „Arbeitskraftunternehmers“ im Postfordismus haben G. Voß (Institut für sozialwissenschaftliche Information und Forschung München) und H. Pongratz (Technische Universität Chemnitz) entwickelt. Verschiedene Aufsätze der beiden Autoren gibt es unter <http://www.tu-chemnitz.de/phil/soziologie/voss/start.htm>.
- Einen Artikel über das Problem

der neuen Autonomie von Dr. Klaus Peters geschrieben im Auftrag des DGB gibt es unter: http://www.union-network.org/unisite/Sectors/IBITS/Industry/publication/NWAUT_g.PDF

- Wilfried Glißmann, Betriebsrat bei IBM in Düsseldorf war einer der ersten, der versucht gegen die neuen Formen der Arbeitsorganisation Widerstand zu organisieren. „Macht was ihr wollt, aber seid profitabel“ ist ein Flugblatt mit dem er gegen Überarbeitung bei IBM auftritt. Zu finden unter Glißmann im Netz.

Lizenz dieses Beitrags
Copyright

© Copyright liegt beim Autor / bei der Autorin des Artikels