

# Arbeitsorientierte Gestaltung von Informationsprozessen.

M. Rauterberg, O. Strohm & E. Ulich

Institut für Arbeitspsychologie, Eidgenössische Technische Hochschule Zürich

**Zusammenfassung:** Die arbeitsorientierte Gestaltung von Informationsprozessen umfaßt sowohl die Gestaltung des Arbeitssystems, in dem neue Technologien eingesetzt werden, bzw. eingesetzt werden sollen, als auch die Gestaltung des Erstellungs- und Einführungsprozesses selbst. Im vorliegenden Beitrag werden zunächst Konzepte und Kriterien für die Gestaltung von Arbeitssystemen vorgestellt, welche sowohl für Arbeitssysteme allgemein, als auch auf Softwarehäuser im Besonderen zutreffen. Dabei spielen Aufgabenorientierung und das Konzept der "vollständigen Tätigkeit" eine zentrale Rolle. Daraus abgeleitet sind Fragen zur Mensch-Maschine-Funktionsteilung zu klären, bevor Arbeitsgestaltung durch Softwaregestaltung sinnvoll ergänzt werden kann. Basierend auf diesen generellen Konzepten und Gestaltungskriterien wird eine Aufbau- und eine dazu passende Ablauforganisation für die Softwareerstellung vorgestellt und diskutiert.

## 1. Einleitung

Für Arbeitspsychologen, Informatiker und betriebliche Praktiker resultieren aus den technologischen Entwicklungen der achtziger Jahre neuartige Fragestellungen, deren Beantwortung umso dringlicher ist, als die inzwischen verfügbare Technologie selbst weder die Ablauf- noch die Aufbauorganisation zwingend determiniert. So bestehen auf der einen Seite Möglichkeiten, neue Kombinationen von fortgeschrittener Technologie und qualifizierter menschlicher Arbeit - mit herausfordernden Arbeitsinhalten und weitgehender Selbstregulation in Gruppen - zu schaffen. Auf der anderen Seite bestehen aber auch vielfältige Möglichkeiten, mit Hilfe der gleichen Technologie vorhandene Formen der Arbeitsteilung zu unterstützen oder sogar zu verstärken. Damit stehen viele Unternehmen vor der Entscheidung, ob - prinzipiell formuliert - der Mensch als verlängerter Arm der Maschine mit Restfunktionen in einer 'Automatisierungslücke' - und potentieller Störfaktor - zu betrachten ist oder die Maschine als verlängerter Arm des Menschen mit Werkzeugfunktion zur Unterstützung der menschlichen Fähigkeiten und Kompetenzen. Diese entgegengesetzten Positionen bezeichnen wir als 'technikorientiert' bzw. 'arbeitsorientiert'. Technikorientierte Konzepte zielen in erster Linie darauf ab, *Technik* zu gestalten. Die Strukturierung von Aufbau- und Ablauforganisation ist hier ebenso wie die Entwicklung und der Einsatz menschlicher Qualifikationen dem Vorrang der Technik nachgeordnet. Arbeitsorientierte Konzepte zielen demgegenüber darauf ab, *Arbeitssysteme* zu gestalten, d.h. die Entwicklung und den Einsatz von Technik, Organisation und Qualifikation gemeinsam zu optimieren. Um einem Mißverständnis vorzubeugen: arbeitsorientierte Konzepte sind keine technikfeindlichen Konzepte. Vielmehr werden durch die gleichzeitige Berücksichtigung von Organisation und Mitarbeiterqualifikation die Voraussetzungen für erfolgreichen Technikeinsatz überhaupt erst geschaffen.

## 2. Gestaltung des Arbeitssystems

Für [Hacker-86:61] ist der Arbeitsauftrag bzw. seine Interpretation oder Übernahme als Arbeitsaufgabe "die zentrale Kategorie einer psychologischen Tätigkeitsbetrachtung..., weil mit der 'objektiven Logik' seiner Inhalte entscheidende Festlegungen zur Regulation und Organisation der Tätigkeiten erfolgen". Im Beitrag von [Volpert-87:14] heißt es dazu: "Der Charakter eines 'Schnittpunktes' zwischen Organisation und Individuum macht die Arbeitsaufgabe zum psychologisch relevantesten Teil der vorgegebenen Arbeitsbedingungen". Beide Zitate machen deutlich, daß für tätigkeits- bzw. handlungstheoretisch orientierte Arbeitspsychologen die Arbeitsaufgabe zum wichtigsten Ansatzpunkt der Arbeitsgestaltung wird. Deshalb ist auch vom "Primat der Aufgabe" die Rede [Ulich-91]. Damit läßt sich eine Brücke schlagen zu den Konzepten soziotechnischer Systemgestaltung, auch wenn diese von Hacker und Volpert nicht explizit berücksichtigt wurden.

**Aufgabenorientierung und das Konzept der "vollständigen Tätigkeit":** Im Rahmen der sozio-technischen Systemkonzeption spielt der Begriff der Aufgabenorientierung ('task orientation') eine bedeutsame Rolle. Aufgabenorientierung bezeichnet einen Zustand des Interesses und Engagements, der durch bestimmte Merkmale der Aufgabe hervorgerufen wird. [Emery-59:53] beschreibt zwei Bedingungen für das Entstehen von Aufgabenorientierung: (1) Die arbeitende Person muß Kontrolle haben über die Arbeitsabläufe und die dafür benötigten Hilfsmittel. (2) Die strukturellen Merkmale der Aufgabe müssen so beschaffen sein, daß sie in der arbeitenden Person Kräfte zur Vollen- dung oder Fortführung der Arbeit auslösen.

Faßt man die Angaben von [Emery-59] sowie [Emery-82] zusammen, so sind es im wesentlichen die folgenden Merkmale von Arbeitsaufgaben, die das Entstehen einer Aufgabenorientierung begünstigen: Ganzheitlichkeit, Anforderungsvielfalt, Möglichkeiten der sozialen Interaktion, Autonomie, Lern- und Entwicklungsmöglichkeiten. Bei [Rice-58] und [Emery-59] findet sich eine Anzahl von Hinweisen auf die besondere motivationale Bedeutung der Ganzheitlichkeit bzw. Vollständigkeit ("wholeness") von Aufgaben. Merkmale der Vollständigkeit, die es bei Maßnahmen der Arbeitsgestaltung zu berücksichtigen gilt, sind in Tabelle 1 zusammengefaßt. Nun ist ganz offensichtlich, daß vollständige Tätigkeiten oder Aufgaben in dem hier beschriebenen Sinn in zahlreichen Fällen - vermutlich sogar mehrheitlich - wegen des damit verbundenen Umfangs nicht als Einzelarbeitstätigkeiten gestaltbar sind sondern nur als Gruppenaufgaben.

**Tabelle 1** Merkmale vollständiger Tätigkeiten (nach Angaben von [Hacker-86], [Volpert-87], [Ulich-89a]).

- 
- (1) Das selbständige Setzen von Zielen, die in übergeordnete Ziele eingebettet werden können.
  - (2) Selbständige Handlungsvorbereitungen im Sinne der Wahrnehmung von Planungsfunktionen.
  - (3) Auswahl der Mittel einschließlich der erforderlichen Interaktionen zur Zielerreichung.
  - (4) Ausführungsfunktionen mit Ablauffeedback zur allfälligen Handlungskorrektur.
  - (5) Kontrolle mit Resultatfeedback und der Möglichkeit, Ergebnisse der eigenen Handlungen auf Übereinstimmung mit den gesetzten Zielen zu überprüfen.
- 

**Arbeitsgestaltung und Softwaregestaltung:** Eines der wichtigsten Probleme der Arbeitssystemgestaltung mit Einsatz moderner Technologie besteht darin, ein gemeinsames Verständnis aller betroffenen Personengruppen über die Mensch-Maschine-Funktionsteilung und damit über den zu automatisierenden Anteil im Arbeitssystem herzustellen. Um die gemeinsame Optimierung des sozialen und des technischen Teilsystems zu realisieren, bedarf es unter anderem valider Kriterien für die Mensch-Maschine-Funktionsteilung [Hoyos-90] [Dunckel-93]. Eine verbreitete, rein technikorientierte Strategie zur Funktionsverteilung zwischen Mensch und Maschine ist die "maximale Automatisierung". Will man jedoch den spezifischen Fähigkeiten des Menschen Rechnung tragen, so wird ein Vergleich zwischen Mensch und Maschine angestellt [Hoyos-90]. [Hoyos-90] weist zu Recht darauf hin, daß solche Vergleiche sich zumeist ausschließlich an den Leistungsmöglichkeiten von Mensch und Maschine orientieren und damit die "Gesamtrolle des Menschen im System" nur ungenügend erfassen. Ausserdem gäben allgemeine MABA-MABA-Listen (men-are-better-at, machines-are-better-at; [Lanc-75]) "zunächst nur ein vages Bild und eine erste Orientierung; erst im Gesamtspektrum einer Aufgabenzuteilung läßt sich die Rolle des Menschen erkennen und bewerten". Schließlich sei "die Aufgabenverteilung zwischen Mensch und Maschine bei bestimmten Aufgabentypen und konkreten Problemlösungen keine einmalige Entscheidung..., sondern eine kontinuierliche Tätigkeit, die iteratives Vorgehen und laufende Überprüfung erfordert" [Hoyos-90:15].

Nach [Bailey-89:189] lassen sich die folgenden fünf Strategien für eine Mensch-Maschine-Funktionsteilung (MMF) unterscheiden: (1) *comparison allocation*: die MMF wird gemäß den MABA-MABA-Listen [Lanc-75] vorgenommen; diese Strategie setzt jedoch voraus, daß sich der Mensch mit der Maschine vergleichen läßt [Hoyos-90]; (2) *leftover allocation*: diese – im technischen Kontext sehr beliebte – Strategie ist auf maximale Automatisierung ausgerichtet und beläßt lediglich die nicht automatisierbaren Funktionen beim Menschen; (3) *economic allocation*: bei dieser Strategie wird versucht, die insgesamt preiswerteste Lösung zu erreichen; (4) *humanized task approach*: bei dieser Strategie sollen durch die gefundene Lösung primär die menschlichen Fähigkeiten gefördert werden; der Technischeinsatz dient lediglich der Unterstützung und Kompensation menschlicher Fähigkeiten; (5) *flexible allocation*: hierbei kann der Benutzer weitgehend frei entscheiden, wie und mit welchen Mitteln, bzw. Werkzeugen er seine Aufgaben erledigt; durch diese Strategie wird dem differentiell-dynamischen Gestaltungsprinzip nach [Ulich-78] optimal Rechnung getragen.

Eines der Hauptprobleme traditioneller Softwareentwicklung liegt darin, daß die bisher primär daran beteiligten Personengruppen häufig nicht erkennen, daß Softwareentwicklung zumeist auch Arbeits- und/ oder Organisationsgestaltung ist. Dies erfordert notwendiger Weise eine interdisziplinäre Zusammenarbeit zwischen Arbeits- und Organisations-Experten einerseits und Softwareentwicklungs-Experten andererseits [Rauterberg-91], [Rauterberg-92a]. Wegen der umfangreichen Qualifikation in dem jeweiligen Fachgebiet ist es nur sehr begrenzt möglich, auf eine interdisziplinäre Zusammenarbeit zu verzichten.

### 3. Gestaltung des Softwareerstellungsprozesses

Viele Softwareprojekte sind durch komplexe, innovative und zeitkritische Problemstellungen gekennzeichnet. Das Projektmanagement dient dazu, diese schwierige Situation zu bewältigen und sollte daher als ganzheitliches Vorhaben verstanden werden, das die für ein Projekt notwendigen planenden, steuernden, überwachenden, methodischen und personalbezogenen Aktivitäten umfaßt. Als Erfolgskriterien sind ergebnisbezogen die Leistung und Qualität des Systems, der Kosten- und Zeitaufwand für dessen Entwicklung sowie prozeßbezogen die Zufriedenheit aller Projektbeteiligten und -betroffenen mit dem Projektverlauf und dessen Ergebnis von Relevanz [Weltz-92] [Spinas-93]. Man kann feststellen, daß traditionelle Organisationsmodelle bei der Softwareentwicklung mit Problemen verbunden sind, welche Zweifel daran aufkommen lassen, ob in solchen Projektstrukturen in effizienter Weise benutzer- und aufgabenangemessene Software entstehen kann [Weltz-92] [Spinas-93].

**Traditionelle Softwareentwicklung:** Analysiert man Softwareentwicklungsprozesse, so erkennt man eine Reihe von Problemen und Schwachstellen. Die Ursachen hierfür sind sowohl in den verwendeten theoretischen Konzepten und den traditionellen Vorgehensweisen (insbesondere Projektmanagement), als auch in unzureichenden Kostenrechnungsmodellen begründet. Aufgrund der Analyse zahlreicher Beispiele aus der Praxis der Softwareentwicklung liegt in der Literatur eine Sammlung von Lösungsmöglichkeiten vor, welche übereinstimmend auf die Bedeutung der Partizipation aller betroffenen Gruppen hinweist. Die Analyse dieser Fallbeispiele läßt drei wesentliche Barrieren erkennen: die Spezifikations-Barriere, die Kommunikations-Barriere und die Optimierungs-Barriere [Rauterberg-91]. Die Tatsache, daß die Behebungskosten für Fehler, welche erst in der Benutzungsphase entdeckt werden, ca. 100 bis 1000 mal größer sind als die Kosten für ihre Behebung in der Anforderungsphase, läßt eine Optimierung vor allem in den frühen Phasen des Softwareentwicklungsprojektes als zwingend notwendig erscheinen [Boehm-81:40]. Da jedoch die frühen Phasen primär durch das Management sozialer Prozesse gekennzeichnet sind, müssen hier entsprechende Konzepte und Methoden aus den Sozial- und Arbeitswissenschaften angewendet werden.

**Ein soziotechnisches Konzept für die Softwareentwicklung:** Nach dem soziotechnischen Systemansatz bestehen Organisationen aus einem sozialen Teilsystem (den Organisationsmitgliedern mit deren individuellen und kollektiven Interessen und Qualifikationen) und einem technischen Teilsystem (den Betriebsmitteln und technologischen Bedingungen). Unter dieser Perspektive läßt sich auch die Organisation von Softwareentwicklungsprojekten betrachten. Das technische Teilsystem ist dabei durch die Projektaufgaben auf strategischer und operativer Ebene wie auch durch mögliche Methoden, Instrumente und Werkzeuge zur Erfüllung dieser Aufgaben definiert. Das soziale Teilsystem ist durch die Projektbeteiligten mit ihren projektspezifischen Bedürfnissen, Interessen und Qualifikationen definiert. Für die konkrete Gestaltung der Aufbau- und Ablauforganisation, sowie die Wahl der Entwicklungsmittel ergeben sich in Abhängigkeit von den Projektzielen und der Projektklassifikation vielfältige Optionen, die im Sinne der Gestaltungskriterien des soziotechnischen Systemansatzes sowie einer interessenausgleichenden Berücksichtigung der verschiedenen Projektbeteiligten und -betroffenen genutzt werden sollten. Hinsichtlich der Aufbauorganisation eines Projektes sind diese Optionen z.B. mit folgenden Fragen verbunden: Aus welchen Personen setzt sich das Projektteam zusammen? Welche Rollen nehmen die verschiedenen Projektbeteiligten ein? Welche Kompetenzen haben die Projektbeteiligten? Liefern die Benutzer lediglich fachspezifische Informationen und evaluieren ad hoc die Lösungsvorschläge der Entwickler oder arbeiten sie aktiv an der Konzeption und Realisierung des Systems im Projektteam mit? Werden neben dem Projektteam themenspezifische Arbeitsgruppen eingerichtet?

Eine starke Integration von Benutzern in die Aufbauorganisation mittels gemischten Projektteams und Arbeitsgruppen ist natürlich auch mit Implikationen für die Ablauforganisation und die benötigten Entwicklungsmittel verbunden. So ist etwa ein iterativ-zyklisches oder inkrementelles Vorgehen unter Einsatz von Prototyping oder Versioning ein adäquates Pendant zu einer Aufbauorganisation mit Teamarbeit. Diese Formen der Ablauforganisation werden wiederum durch den Einsatz von Projektentwicklungsmitteln wie Aufgabenanalysen, Programmiersprachen der 4. Generation, Werkzeugen wie Masken- und Dialoggeneratoren etc. am besten unterstützt bzw. überhaupt erst praktikabel. Spätestens zu diesem Zeitpunkt kommen Fragen bzgl. Investitionen in die Entwicklungsumgebung und Qualifizierungsmaßnahmen für die Projektarbeit auf. Diese Fragen bzw. Implikationen zeigen die Notwendigkeit und zugleich auch die Vorteile eines soziotechnischen bzw. ganzheitlichen Vorgehens bei der Planung und Gestaltung von Software-Projekten. Nur diese Sichtweise ermöglicht schließlich, die einzelnen Elemente des Projektmanagements gemeinsam zu optimieren und aufeinander abzustimmen.

**Teamorganisation bei Softwareentwicklungsprojekten:** In zahlreichen neueren Projektmanagementbüchern wird übereinstimmend Teamarbeit als die adäquate Arbeitsorganisation für die Abwicklung von Projekten dargestellt. Die Ergebnisse einer Meta-Analyse zu Projekterfolgskriterien

[Gemünden-90] zeigen, daß sehr verschiedene Merkmale eines Projektteams wie z.B. Fachkompetenz, Entscheidungskompetenzen, Partizipation, Motivation, Fluktuation/Kontinuität im Zusammenhang mit Projekterfolgskriterien stehen. "Es fällt auf, daß personalen Faktoren ein wesentlich höheres Gewicht zukommt als technokratischen Instrumenten" [Gemünden-90:13]. Das Verständnis darüber, wie Teamarbeit zu gestalten ist, damit eine kompetente und motivierte Aufgabenausführung in einem Projektteam erfolgt, ist allerdings in der betrieblichen Praxis sehr unterschiedlich. Starke Funktions- und Arbeitsteilung in und zwischen Projektteams führt zu einem Mangel an Transparenz, Eigeninitiative und Verantwortungsübernahme bei den Teammitgliedern, die Lern- und Entwicklungsmöglichkeiten bei der Arbeit sind eingeschränkt, geringe Motivation und Produktivität können die Folge davon sein. Die Gestaltung von Projektteams sollte sich an folgenden sozio-technischen Prinzipien orientieren (nach [Ulich-91]): (1) relative Unabhängigkeit des Projektteams, (2) innerer Aufgabenzusammenhang im Projektteam, sowie (3) Einheit von Produkt und Projektorganisation. Aus der Anwendung dieser drei Kriterien folgt, daß einer Gruppe von Analytiker-Programmierern eine ganzheitliche Projektaufgabe - welche idealtypischerweise möglichst viele interdependente Teilaufgaben eines Software-Lebenszyklus umfaßt - zur weitgehend selbständigen Bearbeitung übertragen wird [Spinas-93]. In gemischten Teams wird eine solche Gruppe z.B. durch Benutzervertreter ergänzt. Die Übertragung einer ganzheitlichen Projektaufgabe bzw. eines ganzheitlichen Auftrages ist eine wesentliche Voraussetzung für das Verständnis einer gemeinsamen Aufgabe im Team. Dieses Verständnis ermöglicht ein höheres Maß an Selbstregulation und gegenseitiger Unterstützung [Ulich-91].

In einem selbstregulierten Projektteam werden neben der ganzheitlichen Projektaufgabe auch organisatorische Aufgaben wie die interne Koordination und Aufgabenverteilung, Feinbudgetierung und -planung etc. wahrgenommen und ausgeführt. Im Sinne der arbeitswissenschaftlichen Forderung für die industrielle Produktion, Qualität nicht zu erprüfem, sondern zu erzeugen, ist die begleitende Qualitätssicherung (QS) über alle Projektschritte des Software-Lebenszyklus, eine weitere Aufgabe des Projektteams. Eine wichtige Voraussetzung für die Arbeit in selbstregulierten Projektteams ist die soziale Integration. Deshalb sollte das Projektteam - für die Zeit der Projektarbeit - in einem gemeinsamen Büro oder zumindest räumlich nahe zusammenarbeiten, damit Kommunikation und Kooperation im Team erleichtert werden. In größeren Projekten, die den Einsatz mehrerer solcher Teams erfordern, ist eine wesentliche Funktion des Projektleiters die Koordination der verschiedenen Teams, damit konsistente Teilergebnisse entstehen, die zu einem Ganzen zusammengeführt werden können.

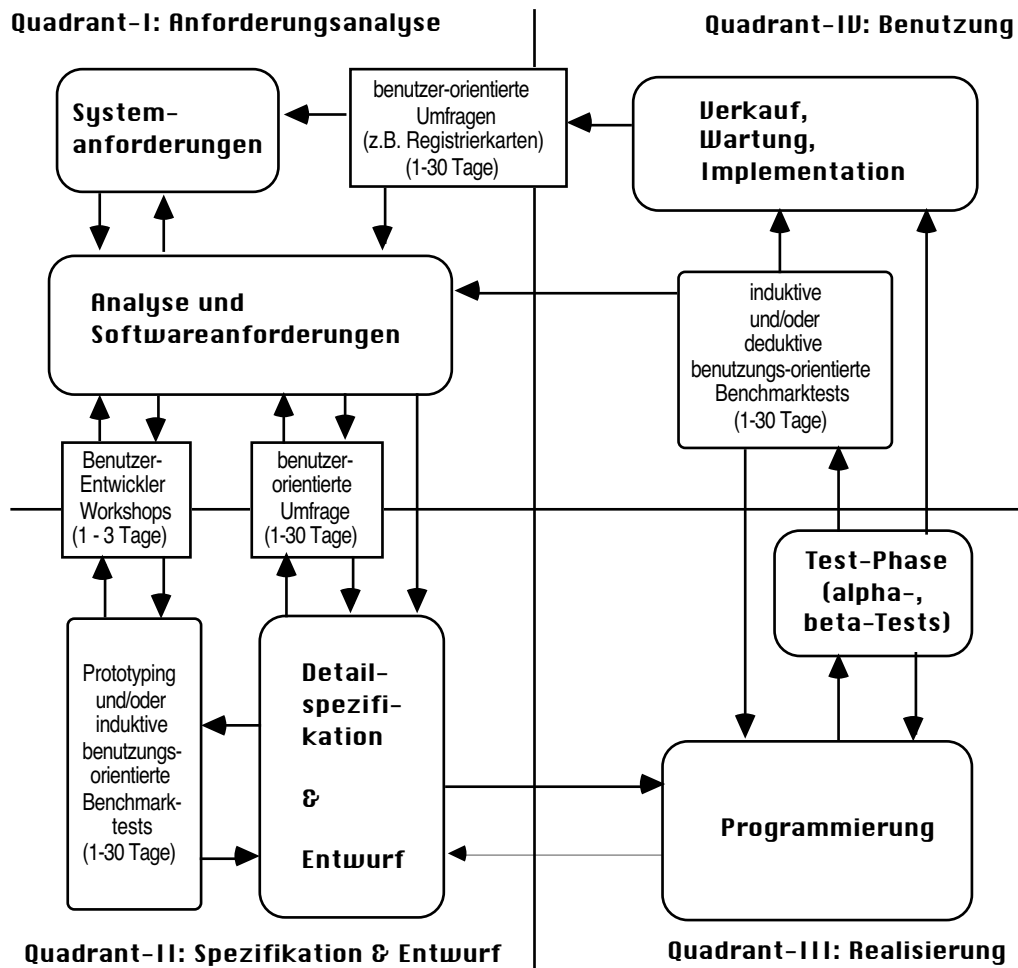
Der Aufteilung von großen Projektaufträgen in kleine, überschaubare und ganzheitliche Teilaufträge im Sinne einer produktorientierten Arbeitsteilung - welche auf der Basis einer modularen Systemkonzeption erfolgen kann - kommt in diesem Zusammenhang eine zentrale Bedeutung zu [Scacchi-91:308]. In einem sehr großen Projekt, welches die Einrichtung von mehreren selbstregulierten Projektteams erfordert, ist die Schaffung einer Projektstruktur überlappender Gruppen eine Notwendigkeit, um verteiltes Wissen und verschiedene Arbeitsergebnisse zu integrieren.

Damit sich die Vorteile der Arbeit in Projektgruppen einstellen, sind jedoch einige Bedingungen zu erfüllen: (1) die Projektaufgabe sollte für die Teammitglieder überschaubar sein, (2) die einzelnen Teilaufgaben sollten einen inneren Zusammenhang aufweisen, (3) das Projektteam sollte über vereinbarte Output-Ziele verfügen, (4) das Projektteam sollte über den Grad der Zielerreichung laufend Feedback erhalten, (5) die Teamgröße sollte der Aufgabenkomplexität optimal entsprechen. Ergebnisfeedback schon während der Entwicklung ist wiederum am besten durch den Einsatz von Prototyping und eine frühe und enge Zusammenarbeit mit den Benutzern zu erreichen.

**Ablaufmodell für den Softwareentwicklungsprozeß:** Ein wesentliches Problem der adäquaten Verzahnung der verschiedenen Optimierungs-Zyklen im Quadranten-Modell iterativer Softwareentwicklung (siehe Abb. 1) besteht in der *Synchronisation* dieser Zyklen. Sind an verschiedenen Stellen im partizipativen Softwareentwicklungs-Konzept gleichzeitig mehrere Optimierungs-Zyklen aktiv, so müssen diese adäquat synchronisiert werden [Rauterberg-91] [Hesse-92]. Dieser Aspekt ist deshalb besonders wichtig, weil nur so das Ausmaß an Inkonsistenzen innerhalb des gesamten Entwicklungsprozesses minimiert werden kann (→ 'simultaneous engineering'). Wenn z.B. parallel zur Implementationsphase weitere Rücksprachen und Anforderungsanalysen mit dem Anwender stattfinden, passiert es leicht, daß die Entwickler gemäß der stets veralteten Spezifikationen oftmals für den 'Papierkorb' programmieren. Die Ursache hierfür ist bei fehlender oder mangelnder Synchronisation dieser beiden Optimierungs-Zyklen in ihrer unterschiedlichen Zyklus-Dauer zu sehen [Rauterberg-91]. Einen wesentlichen Einfluß auf die Qualität des Softwareproduktes hat die Art und Weise der Kommunikation zwischen Entwickler und Anwender, sowie der Entwickler untereinander [Kraut-90].

Mittels Fallstudien und einer schriftlichen Umfrage in mehr als 100 Betrieben wurden Entwicklungsprozesse für Applikationssoftware untersucht [Strohm-91]. Die Ergebnisse zeigen deutlich, daß ein auf die frühen Phasen zentrierter Entwicklungsprozeß, aktive Benutzerbeteiligung, sowie der Einsatz

von Prototyping zur Reduktion des Wartungsaufwandes, der Termin- und der Kostenüberschreitungen beiträgt [Rauterberg-92b]. Die Grundannahme des Wasserfall-Modells, daß alle Anforderungen in der Anfangsphase vollständig bekannt sind, exakt beschrieben werden können und sich nicht verändern, kann nicht länger aufrecht erhalten werden [Rauterberg-91]. Das Endprodukt entfernt sich, z.B. durch Modifikationen oder Restriktionen bei der Programmierung, von den ursprünglichen Benutzeranforderungen. Ferner wurde deutlich, daß schriftliche Systembeschreibungen, Pflichtenhefte und Diagramme von den Benutzern schwer verstanden und – insbesondere negative – Konsequenzen für die spätere Systembenutzung kaum erkannt werden können.



**Abbildung 1** Übersicht über verschiedene Methoden der Benutzerbeteiligung bei Softwareentwicklung im Rahmen des zyklischen Quadrantenmodells [Rauterberg-91] [Spinas-93].

Innovative, erfolgreiche Entwicklungsprojekte hingegen weisen folgende Merkmale auf: ganzheitliche Aufgaben für die ProjektmitarbeiterInnen, aktive Benutzerbeteiligung, einkalkulierter Mehraufwand für analytische und konzeptionelle Arbeiten in den Anfangsphasen, Durchführung von Aufgabenanalysen, iterativ-zyklische Vorgehensweise im Projektablauf sowie Einsatz von Prototyping zur Anforderungspräzisierung und zur Gestaltung der Benutzungsoberfläche. Prototypen eignen sich nicht nur als anschauliche 'Verhandlungsobjekte' für Diskussionen zwischen Benutzern und Entwicklern, sondern tragen auch zur besseren Strukturierung von Ideen der Entwickler bei. Auf diesen Erkenntnissen aufbauend wurden die Organisation benutzerorientierter Softwareentwicklungsprozesse in konkreten Entwicklungsprojekten untersucht und mitgestaltet. In einem Softwareprojekt zur Entwicklung und Einführung eines Bürokommunikationssystems war das mit einem innovativen Vorgehen (Arbeitsanalysen, Prototyping, Benutzerbeteiligung) entwickelte System dem mit traditionellem Vorgehen (Phasenmodell/keine Partizipation) entwickelten Produkt hinsichtlich Benutzungsfreundlichkeit und Aufgabenangemessenheit deutlich überlegen. Am wichtigsten ist jedoch, daß wir anfangen zu lernen, Technik, Organisation und den Einsatz menschlicher Qualifikation gemeinsam zu planen und zu optimieren.

ren. Betrachten wir also die Technik als eine Option, welche es uns gestattet, unsere Lebens- und Arbeitsräume menschengerecht und lebenswert zu gestalten.

## Literaturangaben

- [Boehm-81] Boehm, B. (1981). *Software Engineering Economics*. Englewood Cliffs: Prentice Hall.
- [Dunckel-93] Dunckel, H., Volpert, W., Zölch, M., Kreutner, U., Pleiss, C. & Hennes, K. (1993). Das KABA-Verfahren. Schriftenreihe Mensch-Technik-Organisation (Hrsg.: E. Ulich), Band 5. Zürich: Verlag der Fachvereine.
- [Emery-59] Emery, F.E. (1959). *Characteristics of Socio-Technical Systems*. Document No. 527. London: Tavistock Institute of Human Relations.
- [Emery-82] Emery, F.E. & Thorsrud, E. (1982). *Industrielle Demokratie*. Schriften zur Arbeitspsychologie (Hrsg.: E. Ulich), Band 25. Bern: Huber.
- [Gemünden-90] Gemünden, G. (1990). Erfolgsfaktoren des Projektmanagements - eine kritische Bestandsaufnahme der empirischen Untersuchungen. *Projekt Management*, 1&2:4-15.
- [Hacker-86] Hacker, W. (1986). *Arbeitspsychologie*. Schriften zur Arbeitspsychologie (Hrsg.: E. Ulich), Band 41. Bern: Huber.
- [Hacker-87] Hacker, W. (1987). Software-Ergonomie: Gestalten rechnergestützter Arbeit? In: W. Schönplug & M. Wittstock (Hrsg.), *Software-Ergonomie'87: Nützen Informationssysteme dem Benutzer?* (S. 31-54). Stuttgart: Teubner.
- [Hesse-92] Hesse, W., Merbeth, G. & Frölich, R. (1992). *Software-Entwicklung*. München, Wien: Oldenbourg.
- [Hoyos-90] Hoyos, C. (1990). Menschliches Handeln in technischen Systemen. In: C. Hoyos & B. Zimolong (Hrsg.), *Enzyklopädie der Psychologie*, Band D III 2, *Ingenieurpsychologie* (S. 1-30). Göttingen: Hogrefe.
- [Kraut-90] Kraut, R. & Streeter, L. (1990). Satisfying the need to know: interpersonal information access. In: D. Diaper et al. (eds.), *Human-Computer Interaction - INTERACT'90* (pp. 909-915). Amsterdam: Elsevier.
- [Lanc-75] Lanc, O. (1975). *Ergonomie*. (Urban Taschenbücher, Band 197). Stuttgart: Kohlhammer.
- [Likert-72] Likert, R. (1972). Neue Aspekte der Unternehmensführung. Schriftenreihe "Führung und Organisation der Unternehmung", Band 14. Bern: Paul Haupt.
- [Rauterberg-91] Rauterberg, M. (1991). Partizipative Konzepte, Methoden und Techniken zur Optimierung der Softwareentwicklung. In: P. Brödner, G. Simonis & H. Paul (Hrsg.), *Arbeitsgestaltung und partizipative Systementwicklung* (S. 95-125). Opladen: Leske & Budrich.
- [Rauterberg-92a] Rauterberg, M. (1992). Partizipative Modellbildung zur Optimierung der Softwareentwicklung. In: R. Studer (Hrsg.), *Informationssysteme und Künstliche Intelligenz: Modellierung* (S. 113-128). Berlin: Springer.
- [Rauterberg-92b] Rauterberg, M. & Strohm, O. (1992). Work Organization and Software Development. In: P. Elzer & V. Haase (eds.), *Proceedings of 4th IFAC/IFIP Workshop on "Experience with the Management of Software Projects"*. Annual Review of Automatic Programming. 16 (2):121-128.
- [Rice-58] Rice, A.K. (1958). *Productivity and Social Organization: the Ahmedabad Experiment*. London: Tavistock.
- [Scacchi-91] Scacchi, W. (1991). Understanding software productivity: towards a knowledge-based approach. *International Journal of Software Engineering and Knowledge Engineering*. 1(3):293-321.
- [Spinas-93] Spinass, P., Rauterberg, M., Strohm, O., Waeber, D. & Ulich, E. (1993, in Druck). *Benutzerorientierte Software-Entwicklung. Konzepte, Methoden und Vorgehen zur Benutzerbeteiligung*. Schriftenreihe Mensch-Technik-Organisation (Hrsg.: E. Ulich), Band 3. Zürich: Verlag der Fachvereine, Stuttgart: Teubner.
- [Strohm-91] Strohm, O. (1991). Projektmanagement bei der Softwareentwicklung. In: D. Ackermann & E. Ulich (Hrsg.), *Software-Ergonomie '91. Benutzerorientierte Software-Entwicklung* (S. 46-58). Stuttgart: Teubner.
- [Ulich-78] Ulich, E. (1978). Über das Prinzip der differentiellen Arbeitsgestaltung. *Industrielle Organisation*, 47, 566-568.
- [Ulich-88] Ulich, E. (1988). Arbeits- und organisationspsychologische Aspekte. In: H. Balzert, H.U. Hoppe, R. Oppermann, H. Peschke, G. Rohr und N. Streitz (Hrsg.), *Einführung in die Software-Ergonomie* (S. 49-66). Berlin: de Gruyter.
- [Ulich-89a] Ulich, E. (1989). Arbeitspsychologische Konzepte der Aufgabengestaltung. In: S. Maaß und H. Oberquelle (Hrsg.), *Software-Ergonomie '89: Aufgabenorientierte Systemgestaltung und Funktionalität* (S. 51-65). Stuttgart: Teubner.
- [Ulich-89b] Ulich, E. (1989). Individualisierung und differentielle Arbeitsgestaltung. In: C. Graf Hoyos und B. Zimolong, (Hrsg.): *Ingenieurpsychologie. Enzyklopädie der Psychologie*. Göttingen: Hogrefe.
- [Ulich-91] Ulich, E. (1991, 2. Auflage 1992). *Arbeitspsychologie*. Zürich: Verlag der Fachvereine, Stuttgart: Poeschel-Verlag.
- [Volpert-74] Volpert, W. (1974). Handlungsstrukturanalyse als Beitrag zur Qualifikationsforschung. Köln: Pahl-Rugenstein.
- [Volpert-87] Volpert, W. (1987). Psychische Regulation von Arbeitstätigkeiten. In: U. Kleinbeck und J. Rutenfranz (Hrsg.): *Arbeitspsychologie. Enzyklopädie der Psychologie*. Göttingen: Hogrefe, 1-42.
- [Weltz-92] Weltz, F. & Ortmann, R. (1992). *Das Softwareprojekt - Projektmanagement in der Praxis*. Frankfurt: Campus.

Horst Reichel (Hrsg.)

# Informatik – Wirtschaft – Gesellschaft

23. GI - Jahrestagung  
Dresden, 27. September - 1. Oktober 1993



Springer

1993

Berlin Heidelberg New York London Paris Tokyo  
Hong Kong Barcelona Budapest