

Arbeitsorientiertes Vorgehen zur Gestaltung menschengerechter Software.

Matthias Rauterberg, Oliver Strohm & Eberhard Ulich

Institut für Arbeitspsychologie (IfAP),
Eidgenössische Technische Hochschule Zürich (ETH)
Nelkenstrasse 11, CH-8092 Zürich

Dieser Beitrag ist eine erweiterte Fassung des Beitrages der GI Jahrestagung 1993 in Dresden.

Zusammenfassung

Ein arbeitsorientiertes Vorgehen bei der 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. Aufbauend 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 wird die Tech-

nologie zu einer Option, deren je unterschiedliche Nutzung mit möglicherweise dramatischen Folgen für die Stellung des Menschen im Produktionsprozess verbunden ist. 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. Mögliche Konsequenzen der Realisierung dieser Konzepte sind in Tabelle 1 einander gegenübergestellt.

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. Dieses Erkenntnis setzt sich in neuerer Zeit auch bei renommierten Produktionstechnikern durch. So etwa bei Spur, der noch vor wenigen Jahren als Exponent des Weges in die menschenleere Fabrik gegolten hat: "Je mehr die Produktionsziele auf die Erzeugung hochkomplexer Qualitätsartikel hinauslaufen und den breitflächigen Einsatz neuer Technologien erfordern, umso mehr bieten sich ganzheitlicher Aufgabenzuschnitt und breitere Verwendung von Qualifikationen an" [Spur-89:9]. Aus den Angaben in Tabelle 1 lässt sich ableiten, dass technikorientierte Gestaltungskonzepte zur mangelnden Nutzung oder sogar zum Verlust vor-

handener Qualifikationen führen können. So wies z.B. [Eidenmüller-87] darauf hin, dass Qualitätsregelung und Ausregelung von Störungen am schnellsten und sichersten vor Ort erfolgen und dort auch mit dem geringsten Aufwand verbunden

sind. Wo dies nicht geschieht bzw. durch die technische Auslegung der Anlagen verunmöglicht wird, sei mit einem Verlust vorhandener Qualifikationen zu rechnen.

Tabelle 1 Vergleich unterschiedlicher Konzepte für die Gestaltung rechnerunterstützter Arbeitssysteme (modifiziert nach [Ulich-91]).

	Technikorientierte Gestaltungskonzepte	Arbeitsorientierte Gestaltungskonzepte
Allokation der Steuerung	Zentralisierte Steuerung durch vorgelagerte Bereiche.	Dezentralisierte Steuerung im Fertigungsbereich.
Allokation der Kontrolle im Mensch-Maschine-System	Zentrale Kontrolle. Aufgabenausführung durch Rechnervorgaben inhaltlich und zeitlich festgelegt. Keine Handlungs- und Gestaltungsspielräume für Systembenutzer.	Lokale Kontrolle. Aufgabenausführung nach Vorgaben der Systembenutzer. Begrenzt durch Systembenutzer definierbare Handlungs- und Gestaltungsspielräume.
Informationszugang	Uneingeschränkter Zugang zu Informationen über Systemzustände nur auf der Steuerungs-, bzw. Managementebene. Kein, bzw. stark eingeschränkter Zugang zu Informationen über Systemzustände vor Ort durch den Systembenutzer.	Uneingeschränkter Zugang zu Informationen über Systemzustände auf der Steuerungs-, bzw. Managementebene. (Un)-eingeschränkter Zugang zu Informationen über Systemzustände vor Ort durch den Systembenutzer.
Mensch-Maschine-Funktionsteilung	Systembenutzer übernehmen die nicht automatisierten Resttätigkeiten.	Systembenutzer übernehmen ganzheitliche Aufgaben von der Arbeitsplanung bis zur Qualitätskontrolle.
Zuordnung von Regulation und Verantwortung	Steuerung der Arbeit durch Spezialisten, z.B. Programmierer, Einrichter, etc.	Regulation der Arbeit durch Systembenutzer mit der Verantwortung für Vorbereitungs-, Planungs-, Überwachungs- und Kontrolltätigkeiten.

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- oder 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.

2.1. Aufgabenorientierung und das Konzept der "vollständigen Tätigkeit"

Im Rahmen der soziotechnischen Systemkonzeption spielt der Begriff der Aufgabenorientierung ('task orientation') eine bedeutsame Rolle. Auf-

gabenorientierung 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 Vollendung oder Fortführung der Arbeit auslösen.

Fasst man die Angaben von [Emery-74], [Cherns-76] sowie [Emery-76] und [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 (vgl. Tabelle 2).

Tabelle 2 Merkmale der Arbeitsgestaltung, die Aufgabenorientierung bewirken.

Gestaltungsmerkmal	Ziel / Absicht Vorteil / Wirkung	Realisierung durch...
Ganzheitlichkeit	<ul style="list-style-type: none"> • Mitarbeiter erkennen Bedeutung und Stellenwert ihrer Tätigkeit. • Mitarbeiter erhalten Rückmeldung über den eigenen Arbeitsfortschritt aus der Tätigkeit selbst. 	... Aufgaben mit planenden, ausführenden und kontrollierenden Elementen bzw. der Möglichkeit, Ergebnisse der eigenen Tätigkeit auf Übereinstimmung mit gestellten Anforderungen zu prüfen.
Anforderungsvielfalt	<ul style="list-style-type: none"> • Unterschiedliche Fähigkeiten, Kenntnisse und Fertigkeiten können eingesetzt werden. • Einseitige Beanspruchungen können vermieden werden. 	... Aufgaben mit unterschiedlichen Anforderungen an Körperfunktionen und Sinnesorgane.
Möglichkeiten der sozialen Interaktion	<ul style="list-style-type: none"> • Schwierigkeiten können gemeinsam bewältigt werden. • Gegenseitige Unterstützung hilft Belastungen besser ertragen. 	... Aufgaben, deren Bewältigung Kooperation nahelegt oder voraussetzt.
Autonomie	<ul style="list-style-type: none"> • Stärkt Selbstwertgefühl und Bereitschaft zur Übernahme von Verantwortung. • Vermittelt die Erfahrung, nicht einfluss- und bedeutungslos zusein. 	... Aufgaben mit Dispositions- und Entscheidungsmöglichkeiten.
Lern- und Entwicklungsmöglichkeiten	<ul style="list-style-type: none"> • Allgemeine geistige Flexibilität bleibt erhalten. • Berufliche Qualifikationen werden erhalten und weiterentwickelt. 	... problemhaltige Aufgaben, zu deren Bewältigung vorhandene Qualifikationen erweitert bzw. neue Qualifikationen angeeignet werden müssen.

Bei [Rice-58] und [Emery-59] findet sich eine Anzahl von Hinweisen auf die besondere motivationale Bedeutung der Ganzheitlichkeit bzw. Vollständigkeit ("wholeness") von Aufgaben. In der jüngeren Psychologie findet sich der Terminus 'vollständige Aufgabe' bei [Tomaszewski-81], wäh-

rend bei [Hacker-86, -87] von der 'vollständigen Tätigkeit' und bei [Volpert-87] von der 'vollständigen Handlung' die Rede ist. Merkmale der Vollständigkeit, die es bei Massnahmen der Arbeitsgestaltung zu berücksichtigen gilt, sind in Tabelle 3 zusammengefasst.

Tabelle 3 Merkmale vollständiger Tätigkeiten (nach Angaben von [Tomaszewski-81], [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 adäquaten 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.

Bei unvollständigen Tätigkeiten oder partialisierten Handlungen im Sinne von [Volpert-74] - "fehlen weitestgehend Möglichkeiten für ein eigenständiges Zielsetzen und Entscheiden, für das Entwickeln individueller Arbeitsweisen oder für ausreichend genaue Rückmeldungen" [Hacker-87:35]. Vollständige Tätigkeiten bieten demgegenüber Möglichkeiten der Setzung von Zielen und Teilzielen und Entscheidungsmöglichkeiten in allen

Phasen der Aufgabenerledigung, gewähren also Tätigkeits- bzw. Handlungsspielraum. Damit werden vollständige Tätigkeiten zu einer Grundvoraussetzung für die Realisierung des Konzepts der differentiellen Arbeitsgestaltung [Ulich-78, -88, -89b]. Vollständige Tätigkeiten sind zudem in *sequentieller* und *hierarchischer* Hinsicht vollständig (vgl. Kasten 1).

Kasten 1 Die sequentielle und hierarchische Vollständigkeit von Tätigkeiten [Hacker 87].

"Eine vollständige Tätigkeit ist zum ersten in sequentieller Hinsicht vollständig. Neben blossen Ausführungsfunktionen umfasst sie

- Vorbereitungsfunktionen (das Aufstellen von Zielen, das Entwickeln von Vorgehensweisen, das Auswählen zweckmässiger Vorgehensvarianten)
- Organisationsfunktionen (das Abstimmen der Aufgaben mit anderen Menschen) und
- Kontrollfunktionen, durch die der Arbeitende Rückmeldungen über das Erreichen seiner Ziele sich zu verschaffen in der Lage ist.

Zum zweiten sind vollständige Tätigkeiten in hierarchischer Hinsicht vollständig, indem sie Anforderungen auf verschiedenen, einander abwechselnden Ebenen der Tätigkeitsregulation stellen. Zu denken ist beispielsweise an das Abwechseln von routinisierten Operationen der Zuordnung von Bedingungen zu Massnahmen mit algorithmisch vorgegebenen Denkvorgängen und mit Problemfindungs- und -lösungsprozessen. Eine Mindestforderung scheinen Mischformen zu sein, die etwa zur Hälfte der Arbeitszeit intellektuelle Verarbeitungsanforderungen einschliessen" [Hacker-87:43].

Nun ist ganz offensichtlich, dass vollständige Tätigkeiten oder Aufgaben in dem hier beschriebenen Sinn in zahlreichen Fällen - vermutlich sogar mehrheitlich - wegen des damit verbundenen Umfanges nicht als Einzelarbeitstätigkeiten gestaltbar sind sondern nur als Gruppenaufgaben. Darauf haben die Vertreter der soziotechnischen Systemkonzeption schon sehr früh aufmerksam gemacht. Arbeit in Gruppen kommt hauptsächlich aus zwei - miteinander zusammenhängenden - Gründen psychologisch ein besonderer Stellenwert zu:

- (1) Das Erleben ganzheitlicher Arbeit ist in modernen Arbeitsprozessen mehrheitlich nur möglich, wenn interdependente Teilaufgaben zu vollständigen Gruppenaufgaben zusammengefasst werden.
- (2) Die Zusammenfassung von interdependenten Teilaufgaben zur gemeinsamen Aufgabe einer Gruppe ermöglicht ein höheres Mass an Selbstregulation und sozialer Unterstützung.

Vergleicht man die hier genannten Merkmale vollständiger Tätigkeiten mit den von [Hellpach-22] formulierten Aufgabenmerkmalen, so werden Über-

einstimmungen deutlich. Gravierende Unterschiede bestehen allerdings im Grad der Präzisierung und Differenzierung sowie in der theoretischen Fundierung der Herleitung. Weitere Differenzierungen sollten u.a. der Instruktions- und der Feedbackkomponente vermehrt Beachtung schenken. So verweist [Tomeszewski-81:24] auf eine Arbeit von Materska (1972), nach deren Ergebnissen sich Unterschiede in der Aufgabenausführung ergeben, je nachdem, ob "Informationen über das zu erreichende Ergebnis oder aber Informationen über den Verlauf der auszuübenden Tätigkeit dominieren". In Zusammenhang mit Fragen der Rückmeldung sollten wir uns an einen Hinweis von [Ammons-56] erinnern, demzufolge das Optimum an exakter Information keineswegs immer mit deren möglichem Maximum übereinstimmt; von einem bestimmten Punkt ab könne zusätzliche Information unter Umständen sogar eine Verschlechterung der Leistung bewirken.

2.2. 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 [Naur-85], [Reisin-91], [Weltz-91], also gemeinsam verbindliche Antworten auf die Fragen "ob", "wo" und "wie" des geplanten Technikeinsatzes zu finden.

Hierzu gehört insbesondere die Bestimmung aller Eigenschaften des neu zu gestaltenden Arbeitssystems. Um die gemeinsame Optimierung des sozialen und des technischen Teilsystems durchzuführen, bedarf es menschengerechter Kriterien für die Gestaltung von Arbeitstätigkeiten [Ulich-93] und valider Kriterien für die Mensch-Maschine-Funktionsteilung [Hoyos-90].

Eine sehr verbreitete Strategie zur Funktionsverteilung zwischen Mensch und Maschine ist die "maximale Automatisierung". Sie entspricht dem in Tabelle 1 dargestellten technikorientierten Gestaltungskonzept. 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 die Autoren solcher Listen sich 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) "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].

Tabelle 4 Vergleich der Fähigkeiten von Menschen und Maschinen (MABA-MABA - Liste; aus [Lanc-75])

Funktionen, die der Mensch besser bewältigt als die Maschine	Funktionen, die eine Maschine besser bewältigen kann als ein Mensch
<ol style="list-style-type: none"> 1. Detektion energetisch sehr schwacher Signale und deren Verstärkung; 2. Flexibilität und Improvisation (schnelles Finden einer Alternativlösung); 3. Wechsel von einer bestimmten Strategie zu einer anderen (Übergang zu einer anderen Lösung); 4. Langfristiges Behalten von großen Informationsmengen ($2,8 \times 10^{20}$ bit, nach Neumann) und schnellerer Suchvorgang; 5. Räumliche Wahrnehmung (Wahrnehmung von Raumtiefe und Formen); 6. Interpolation (Bestimmung der Werte zwischen fixen Punkten bzw. Werten); 7. Prädiktion und Antizipation (Vorhersage weiterer Entwicklung in logisch schwer definierbaren Bedingungen); 8. Induktive Urteilsprozesse (Verallgemeinerung) bzw. Bildung einer Ansicht; 9. Realisierung homöostatischer Prozesse (Beibehalten einer stabilen Lage bei Änderung der äusseren Bedingungen); 10. Adaptation und Lernen; 11. Durchführung komplexer Entscheidungen; Lösung komplizierter unvollkommen definierter Situationen bzw. unvorhergesehener Situationen; 	<ol style="list-style-type: none"> 1. Lösung einfacher arithmetischer Aufgaben mit großer Geschwindigkeit, Fähigkeit zu sehr schnellen Reaktionen (10^{-6} - 10^7 s); 2. Differenzierung, d. h. Durchführung der mathematischen Operation d/dt; 3. Integrierung, d. h. Durchführung des Integrals einer Funktion; 4. Einsatz großer Kraft oder Leistung bei großer Präzision und genau definiertem Ablauf (bei der Maschine ist die Leistung im Praxisbezug unbeschränkt); 5. Exakte Wiederholung bestimmter Prozesse nach einem vorgegebenen Programm über einen beliebigen Zeitraum; 6. Langfristige Wachsamkeit, keine Ermüdungserscheinungen; 7. Kurzzeitiges Behalten einer Information, kurzzeitige Speicherung; 8. Durchführung von komplexen simultanen Funktionen mit großer Geschwindigkeit bzw. nach genauer zeitlicher Abfolge; 9. Deduktive Urteilsprozesse; 10. Einfache Entscheidungen von dem Typ ja-nein mit großer Geschwindigkeit (allerdings mit weniger Möglichkeit, die Ergebnisse zu korrigieren); 11. Detektion von Signalen, deren Qualität mit den menschlichen Sinnesorganen nicht wahrgenommen werden kann, mit wesentlich größerer Genauigkeit, als dies der Mensch in seinem Bereich kann.

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 [Hoyos-90] vorgenommen; diese Strategie setzt jedoch voraus, daß sich der Mensch mit der Maschine vergleichen läßt [Zölch-91] [Dunckel-93].
- (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 Technikeinsatz dient lediglich der Unterstützung und Kompensation menschlicher Fähigkeiten.
- (5) *flexibel allocation*: hierbei kann der Benutzer weitgehend frei entscheiden, wie und mit welchen Mitteln, bzw. Werkzeugen er seine Aufgaben erledigt; durch dies Strategie wird dem differentiell-dynamischen Gestaltungsprinzip nach [Ulrich-78] optimal Rechnung getragen.

Eines der Hauptprobleme traditioneller Softwareentwicklung liegt darin, daß die bisher primär an Softwareentwicklungen beteiligten Personen-Gruppen 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, -92a] [Waeber-91]. Wegen der umfangreichen Qualifikation in dem jeweiligen Fachgebiet ist es nur sehr begrenzt möglich, auf eine interdisziplinäre Zusammenarbeit zu verzichten.

3. Gestaltung des Software-erstellungprozesses

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 [Rauterberg-91] [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].

3.1. 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, mindestens 100 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-81a: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.

3.2. 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 (siehe Abb. 1). Das technische Teilsystem lässt sich somit durch die Projektaufgaben auf strategischer und operativer Ebene wie auch durch mögliche Methoden, Instrumente und Werkzeuge zur Erfüllung dieser Aufgaben definieren [Beck-93]. Das soziale System 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 Rolle nehmen die verschiedenen Projektbeteiligten ein? Welche Kompetenzen haben die einzelnen 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 themenspezifischen 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 wäre etwa ein iterativ-zyklisches oder inkrementelles Vorgehen unter Einsatz von Prototyping oder Versioning ein adäquates Pendant zu einer solchen Aufbau-

organisation. 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.

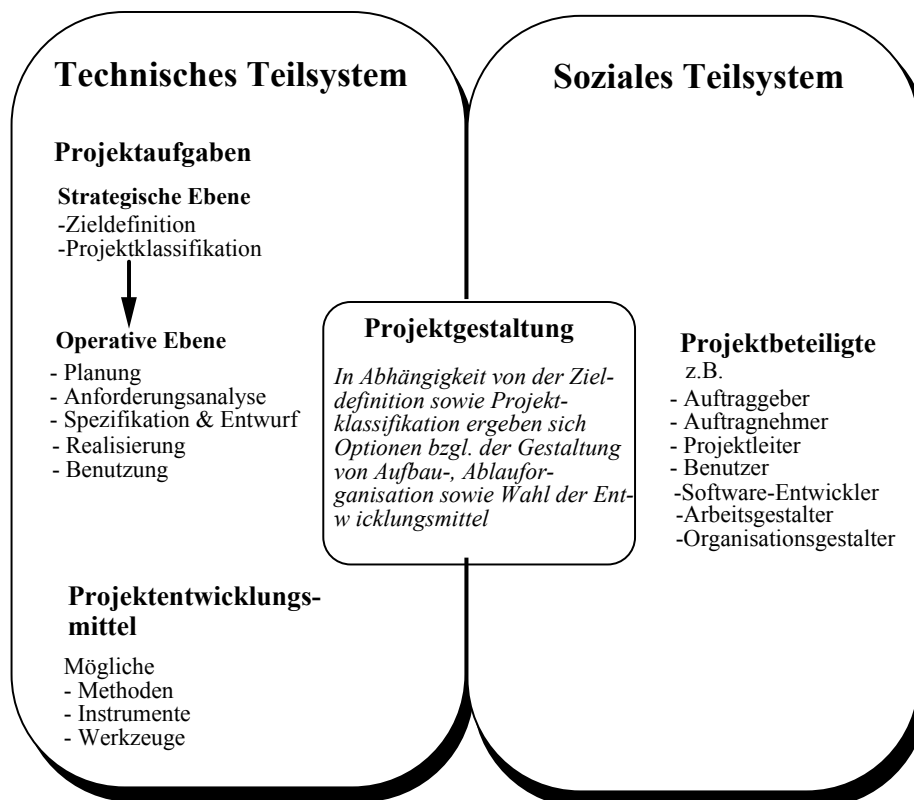


Abbildung 1 Eine soziotechnische Betrachtung der Organisation von Software-Entwicklungsprojekten.

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. Frühzeitige Transparenz für alle Beteiligten über das Projekt, ein auf die Projektziele und Rahmenbedingungen abgestimmtes Vorgehen, sowie die verringerte Gefahr, dass im Laufe des Projektes "unvorhergesehenes" oder "nichtbedachtes" eintritt, sind die wesentlichen Vorteile.

3.3. 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. Projektteams sind häufig funktional und sehr arbeitsteilig organisiert. In grossen Projekten werden z.T. Teams für die Analyse und Konzeptentwicklung, andere für die Spezifikation und Realisierung und wiederum andere für die Qualitätssicherung eingerichtet. In kleineren Projekten dagegen wird zwar häufig nur ein Projektteam eingesetzt, das die gesamten Aufgaben des Software-Lebenszyklus bearbeitet; solche Teams sind jedoch häufig intern funktionsorientiert und arbeitsteilig ausgerichtet und zusammengesetzt. Programmierer werden vorwiegend mit programmier-technischen und testenden Aufgaben beauftragt, dagegen selten mit analytischen und konzeptionellen Aufgaben, die vorwiegend zum Aufgabenbereich des Projektleiters und Systemanalytikers gehören. Programmierer, die unter solch arbeitsteiligen Bedingungen arbeiten, sind mit ihrer Arbeitssituation häufig sehr unzufrieden, da für sie weder Kooperationsmöglichkeiten mit anderen Programmieren noch mit Benutzern speziell in der Analyse- und Konzeptionsphase bestehen. Durch mangelnden Überblick über eine komplexe Aufgabe erleben sie ihre Arbeit als partialisiert und entfremdet [Hacker-89]. Dies ist gut nachvollziehbar, wenn man bedenkt, dass Software-Entwickler selbst vor allem die analytischen und konzeptionellen Aufgaben - im Unterschied zu den realisierenden Aufgaben - als fachlich und sozial herausfordernd und somit qualifizierend erleben! [Frese-91].

Starke Funktions- und Arbeitsteilung in und zwischen Projektteams resultiert in 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. Dass dies in der betrieblichen Praxis mancherorts erkannt worden ist, zeigt sich daran, dass verstärkt Analytiker-Programmierer - die mit der Bearbeitung möglichst vieler Aufgaben des Software-Entwicklungsprozesses beauftragt werden können - ausgebildet bzw. auf dem Arbeitsmarkt gesucht werden.

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 Projektteam.

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, 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. Dies beinhaltet, dass das Team z.B. auch Reviews zu den Anforderungen, zur Spezifikation und zum Entwurf durchführt. Dazu sollten zuverlässige Methoden, Verfahren und Werkzeuge zur Qualitätssicherung zur Verfügung stehen [Wallmüller-90].

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 wird. Neben einer individuellen und kollektiven Aufgabenorientierung können auch gemeinsame Aktivitäten des Projektteams ausserhalb des Projektalltages eine positive Teamentwicklung fördern. In grösseren Projekten, die entsprechend den Einsatz mehrerer solcher Teams erfordern, wäre 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 z.B. 6 selbstregulierten Projektteams bedingt, ist die Schaffung einer

Projektstruktur überlappender Gruppen eine Notwendigkeit, um verteiltes Wissen und verschiedene Arbeitsergebnisse zu integrieren (siehe Abb.2).

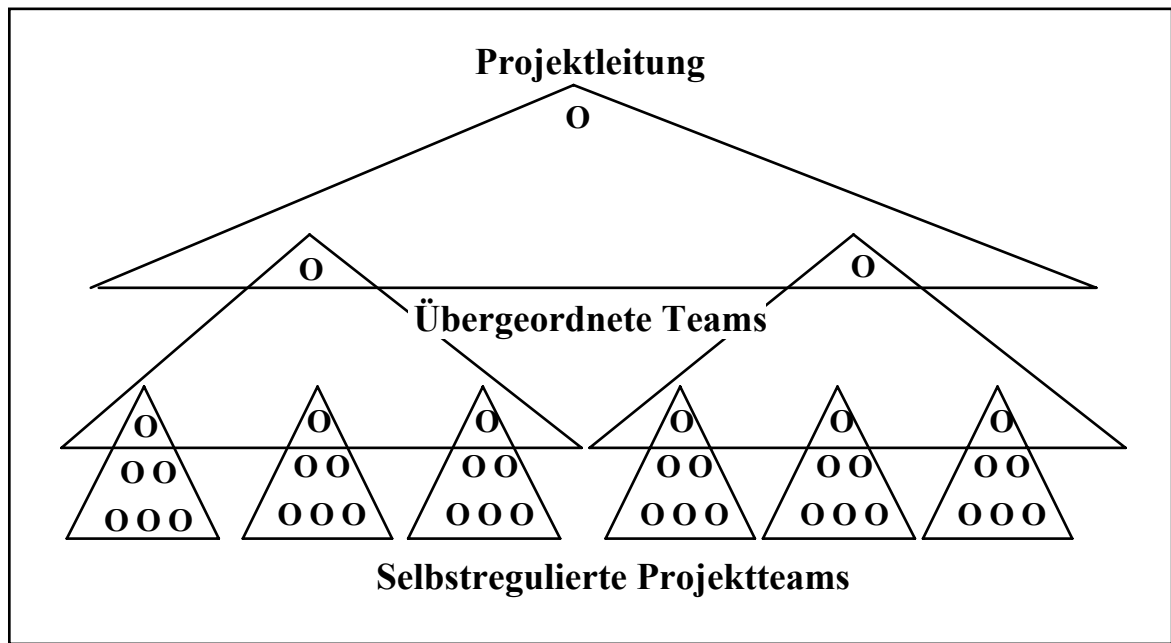


Abbildung 2 Ein Modell überlappender Projektteams (in Anlehnung an [Likert-72]).

Übergeordnete Teams werden dabei aus Vertretern der verschiedenen Projektteams gebildet. Die Vertreter in diesen übergeordneten Gruppen werden entweder durch das Team fest bestimmt oder wechseln nach einem Rotationsprinzip. Die Aufgaben dieser übergeordneten Teams können Koordination, Wissens- und Ergebnistransfer oder spezifische Aufträge beinhalten. Die Entwicklung eines Gesamtkonzeptes des Systems oder eines übergreifenden Konzeptes zur Qualitätssicherung sind Beispiele für Projektaufgaben, welche in übergeordneten Teams ausgeführt werden können bzw. müssen. Vertreter mehrerer übergeordneter Teams bilden - falls notwendig - mit der Projektleitung eine weitere Instanz, in der projektübergreifend Absprachen und Koordination stattfinden können.

Eine Projektorganisation mit selbstregulierten Teams kann folgende Vorteile aufweisen:

- sie fördert die Identifikation des Teams mit der Projektaufgabe,
- die Arbeit in einem solchen Team ist qualifizierend in fachlicher wie sozialer Hinsicht,
- die Motivation, Flexibilität, Kreativität und gegenseitige Unterstützung werden gefördert,

- diese Form der Projektorganisation wirkt der Bürokratisierung entgegen.

Kontinuität in der personellen Zusammensetzung von Projektteams, durch die Qualifikationen und Erfahrungen erhalten werden und nicht z.B. infolge von Fluktuationen verlorengehen, wird durch selbstregulierte Projektteams mit individueller und kollektiver Aufgabenorientierung ebenfalls wahrscheinlicher bzw. besser erreicht. Damit sich diese Vorteile einstellen, sind jedoch einige Bedingungen zu erfüllen:

- die Projektaufgabe sollte für die Teammitglieder überschaubar sein,
- die einzelnen Teilaufgaben sollten einen inneren Zusammenhang aufweisen,
- das Projektteam sollte über vereinbarte Output-Ziele verfügen,
- das Projektteam sollte über den Grad der Zielerreichung laufend Feedback erhalten,
- die Teamgrösse sollte der Aufgabenkomplexität optimal angepasst sein.

Erfahrungsgemäss zeigt sich, dass Arbeitsgruppen bzw. Projektteams eine Grösse von 8-10 Personen nicht übersteigen sollten, damit neben der geforderten aufgabenbezogenen Transparenz auch

die Transparenz des "sozialen Systems" erhalten bleibt. 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. Eine weitere wichtige Voraussetzung für diese Arbeitsform ist ein Entlohnungskonzept, das weniger auf einer individuellen Leistungsbemessung basiert, als vielmehr die gesamte Teamleistung sowie die Qualifikation der einzelnen Teammitglieder - das heisst ihr Können und damit ihre Flexibilität und Einsetzbarkeit - honoriert (Pay for knowledge). Folgende Kriterien können für die Berechnung des individuellen Lohnanteils in Betracht gezogen werden:

Wieviele Programmiersprachen/Entwicklungswerkzeuge beherrscht das Teammitglied?

Welche Aufgaben des Software-Lebenszyklus kann das Teammitglied bearbeiten?

Welche Branchen-/Fachbereichskennnisse hat das Teammitglied?

Individuelle Lohnanteile nach einem solchen Könnenskonzept unterstützen eine individuell wie auch kollektiv erlebte Lohngerechtigkeit und fördern die Bereitschaft der Teammitglieder Qualifizierungsmaßnahmen zu nutzen. Die Anwendung dieses Konzeptes setzt einen Konsens im Team über folgende Fragen voraus [Ulich-91]:

Welche Ansätze der Arbeitsbewertung werden angewendet?

Welche einheitlichen und überprüfbaren Kriterien werden dabei verwendet?

Wer nimmt in welchem zeitlichen Abständen eine Überprüfung der Beurteilung sowie mögliche Höherstufungen vor?

Mit welchem Gewicht findet die Qualifizierung der Teammitglieder bei der Festlegung der Tätigkeit des Vorgesetzten z.B. des Projektleiters und deren Bewertung Berücksichtigung?

Teamleistungsanteile, die z.B. für das Erreichen vereinbarter Qualitäts- oder Budgetziele vorgesehen sind, sollten dagegen gleichmässig auf alle Teammitglieder verteilt werden, damit auch die kollegiale Unterstützung im Projektteam gefördert wird.

3.4. Ablaufmodell für den Softwareentwicklungsprozeß

Ein wesentliches Problem der adäquaten Verzahnung der verschiedenen Optimierungs-Zyklen im Quadranten-Modell iterativer Softwareentwicklung (siehe Abb. 3) 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 [Zelkowitz-79:5] [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 Zyklusdauer 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 nämlich 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 [Waeber-91].

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 [Budde-84]. Auf diesen Erkenntnissen aufbauend wurden die Organisation benutzerorientierter Softwareentwicklungsprozesse in konkreten Entwicklungsprojekten untersucht und mitgestaltet.

In einem Softwareprojekt zur Entwicklung und Einführung eines Bürokommunikationssystem war das mit einem innovativen Vorgehen (Arbeitsanalysen, Prototyping, Benutzerbeteiligung) entwickelte System dem mit traditionellem Vorgehen (Phasenmodell/keine Partizipation) entwickelten Produkt hinsichtlich Benutzerfreundlichkeit und Aufgabenangemessenheit deutlich überlegen.

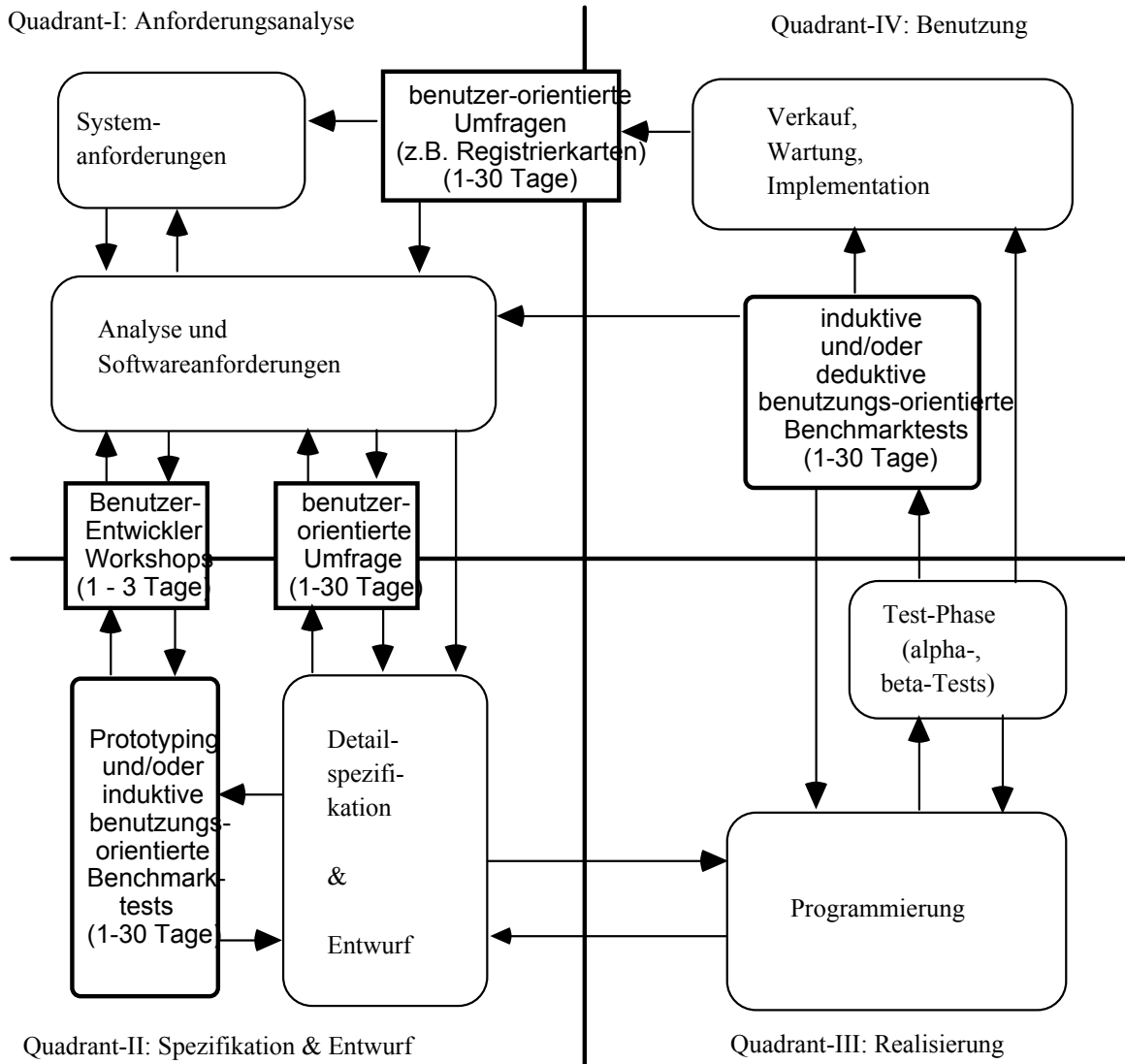


Abbildung 3 Übersicht über verschiedene Methoden der Benutzerbeteiligung bei Softwareentwicklung im Rahmen des zyklischen Quadrantenmodells [Rauterberg-91]¹.

¹ Die bei [Hesse-92] vorgestellten vier "Sektoren der Software-Technologie-Landschaft" sind mit den Quadranten weitgehend deckungsgleich. Leider ist ein entsprechender Literaturhinweis auf [Rauterberg-91] nicht zu finden.

4. Zukünftiger Forschungs- und Entwicklungsbedarf

Es werden Methoden zum "Management der sozialen Prozesse im Arbeitssystem" benötigt, welche zur Zeit von Unternehmensberatern und Organisationsgestaltern mit einer leider oftmals ausschließlich organisatorischen Perspektive eingesetzt werden. Es fehlen Methoden, welche die Analyse, Bewertung und Gestaltung von Unternehmen ermöglichen, bei denen von Anfang an systematisch der Einsatz von Technik mitgedacht und geplant wird. Hierzu ist es notwendig, daß speziell qualifizierte Arbeits- und Organisationsgestalter mit fundierten Kenntnissen über formale Spezifikationsmethoden eingesetzt werden. Es werden Methoden zum "Management der sozialen Prozesse im Softwareentwicklungsprozeß" benötigt, welche primär auf die Gestaltung der sozialen und kommunikativen Aspekte ausgerichtet sind. Diese Methoden können über entsprechend qualifizierte Projektmanager zum Einsatz gelangen und weiterentwickelt werden. Es fehlen weitere grundlegende Methodenvergleichsstudien, in denen die noch offenen Problembereiche geklärt werden könnten (vgl. etwa: [Boehm-81b] [Müller-Holz-91] [Kirsch-91] [Jeffries-91] [Jeffries-92]). Die oftmals veröffentlichten Falldarstellungen sind in ihrer Aussagekraft viel zu begrenzt, als daß sich daraus zwingende Schlussfolgerungen ableiten ließen. Es bedarf verstärkt der "Hilfe zur Selbsthilfe". Projekte, bei denen die Aufgabe der Begleitforscher auf die Rolle von Beratern begrenzt ist, sind unterrepräsentiert. Für diesen Projekttyp sind Experten für Organisations- und Technikgestaltung unabdingbar.

Literaturangaben

- [Ammons-56] Ammons, R.B. (1956). Effects of knowledge of performance: a survey and tentative theoretical formulation. *Journal of General Psychology* 54:279-299.
- [Beck-93] Beck, A. & Janssen, C. (1993) Vorgehen und Methoden für aufgaben- und benutzerangemessene Gestaltung von grafischen Benutzungsschnittstellen. in: W. Coy, P. Gorny, I. Kopp & C. Skarpelis (Hrsg.) *Menschengerechte Software als Wettbewerbsfaktor*. (German Chapter of the ACM Berichte 40, S. 200-221). Stuttgart: Teubner.
- [Boehm-81a] Boehm, B. (1981) *Software Engineering Economics*. Englewood Cliffs: Prentice Hall.
- [Boehm-81b] Boehm, B.W., Gray, T. & Seewaldt, T. (1981) Prototyping versus specifying: a multiproject experiment. *IEEE Transactions on Software Engineering*, SE-10(3):224-236
- [Boehm-86] Boehm, B. (1986) *Wirtschaftliche Software-Produktion*. Wiesbaden: Forkel.
- [Brodbeck-93] Brodbeck, F. & Sonntag, S. (1993) Arbeitsanforderungen und soziale Prozesse in der Software-Entwicklung. in: W. Coy, P. Gorny, I. Kopp & C. Skarpelis (Hrsg.) *Menschengerechte Software als Wettbewerbsfaktor*. (German Chapter of the ACM Berichte 40, S. 248-258). Stuttgart: Teubner.
- [Budde-84] Budde, R., Kuhlenkamp, K., Mathiassen, L. & Züllighoven, H. (1984, eds.) *Approaches to Prototyping*. Berlin: Springer.
- [Cherns-76] Cherns, A. (1976) The principles of organizational design. *Human Relations*, 29:783-792.
- [Dunckel-93] Dunckel, H., Volpert, W., Zölch, M., Kreutner, U., Pleiss, C. & Hennes, K. (1993) *Konstrastive Aufgabenanalyse im Büro*. (Mensch Technik Organisation, Band 5; E. Ulich, Hrsg.). Stuttgart: Teubner.
- [Eidenmüller-87] Eidenmüller, B. (1987) Auswirkungen neuer Technologien auf die Arbeitsorganisation. *Fortschrittliche Betriebsführung und Industrial Engineering*, 36(H.1):4-8.
- [Emery-59] Emery, F.E. (1959) Characteristics of socio-technical systems. Document No. 527. London: Tavistock Institute of Human Relations.
- [Emery-74] Emery, F.E. & Emery, Y. (1974) *Participative Design*. Canberra: Centre for Continuing Education, Australian National University.
- [Emery-76] Emery, F.E. & Thorsrud, E. (1976) *Democracy at Work*. Leiden: Martinus Nijhoff.
- [Emery-82] Emery, F.E. & Thorsrud, E. (1982) *Industrielle Demokratie*. (Schriften zur Arbeitspsychologie, Band 25; E. Ulich, Hrsg.). Bern: Huber.
- [Frese-91] Frese, M., Fritz, A., Stolte, W. & Brodbeck, F. (1991) Eine erste Analyse von Tätigkeiten und ihre Anforderungen. Posterbeitrag auf der Arbeitstagung "Software für die Arbeit von morgen". München.
- [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, Band 41; E. Ulich, Hrsg.). 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.
- [Hacker-89] Hacker, W. (1989) Designing the designer's task: participative analysis and evaluation of software development tasks. In: M. Smith & G. Salvendy (eds.), *Work with Computers: Organizational, Management, Stress and Health Aspects* (p. 163-168). Amsterdam: Elsevier.
- [Hellpach-22] Hellpach, W. (1922) Sozialpsychologische Analyse des betriebstechnischen Tatbestandes "Gruppenfabrikation". In R. Lang und W. Hellpach: *Gruppenfabrikation* (S. 5-186). Berlin: Springer.
- [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.
- [Jeffries-91] Jeffries, R., Miller, J., Wharton, C. & Uyeda, K. (1991) User interface evaluation in the real world: a comparison of four techniques. In: S. Robertson, G. Olson & J. Olson (eds.) *Human Factors in Computing Systems: Reaching through technology CHI'91* (pp.119-124). New York: ACM.
- [Jeffries-92] Jeffries, R. & Desurvire, H. (1992) Usability testing vs. heuristic evaluation: was there a contest? *SIGCHI Bulletin*, 24(4):39-41.
- [Kirsch-91] Kirsch, C. (1991) Benutzerbeteiligung bei der Datenmodellierung. *Softwaretechnik-Trends*, 11(3):93-103.
- [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.
- [Müller-Holz-91] Müller-Holz auf der Heide, B., Ascherleben, G., Hacker, S. & Bartsch, T. (1991) Methoden zur empirischen Bewertung der Benutzerfreundlichkeit von Bürosoftware im Rahmen von Prototyping. In: Frese, M., Kasten, C., Skarpelis, C. & Zang-Scheucher, B. (eds.) *Software für die Arbeit von morgen* (S. 409-420). Berlin Heidelberg New York: Springer.
- [Naur-85] Naur, P. (1985) Programming as theory building. *Microprocessing and Microprogramming*, 15:253-261.
- [Rauterberg-91] Rauterberg, M. (1991) Partizipative Konzepte, Methoden und Techniken zur Optimierung der Softwareentwicklung. *Softwaretechnik-Trends*, 11(3):104-126..
- [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.
- [Reisin-91] Reisin, M. (1991) Kooperativer Aufbau einer gemeinsamen Referenztheorie. In: P. Brödner, G. Simonis & H. Paul (Hrsg.), *Arbeitsgestaltung und partizipative Systementwicklung* (S. 59-79). Opladen: Leske & Budrich.
- [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, Band 3; E. Ulich, Hrsg.). Zürich: Verlag der Fachvereine.
- [Spur-89] Spur, G. (1989) *Unternehmensführung in der künftigen Industriegesellschaft*. *Siemens Zeitschrift*, 63(6):4-9.
- [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.
- [Tomaszewski-81] Tomaszewski, T. (1981). Struktur, Funktion und Steuerungsmechanismen menschlicher Tätigkeit. In: T. Tomaszewski, (Hrsg.): *Zur Psychologie der Tätigkeit*. Berlin (DDR): Deutscher Verlag der Wissenschaften, 11-33.
- [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 & 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) *Arbeitspsychologie*. Stuttgart: Poeschel-Verlag.

- [Ulich-93] Ulich, E. (1993) Gestaltung von Arbeitstätigkeiten. In: H. Schuler (Hrsg.) Lehrbuch Organisationspsychologie (S. 189-208). Göttingen: Huber.
- [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 (S. 1-42). Göttingen: Hogrefe.
- [Volpert-74] Waeber, D. (1991) Aufgabenanalyse und Anforderungsermittlung in der Softwareentwicklung: Zur Konzeption einer integrierten Entwurfsstrategie. In: M. Frese, C. Karsten, C. Skarpelis & B. Zang-Scheucher (Hrsg.), Software für die Arbeit von morgen. Bilanz und Perspektiven anwendungsorientierter Forschung. Ergänzung zum Tagungsband (S. 35-45). Krefeld: Vennekel & Partner.
- [Wallmüller-90] Wallmüller, E. (1990) Software-Qualitätssicherung in der Praxis. München: Oldenbourg.
- [Weltz-91] Weltz, F., Lullies, V. & Ortmann, R G. (1991) Softwareentwicklung als Prozeß der Arbeitsstrukturierung. In: Ackermann, D. & Ulich, E (Hrsg.) Software-Ergonomie '91 (S. 70-75). (Berichte des German Chapter of the ACM, Vol. 33). Stuttgart: Teubner.
- [Weltz-92] Weltz, F. & Ortmann, R. (1992) Das Softwareprojekt – Projektmanagement in der Praxis. Frankfurt: Campus.
- [Zelkowitz-79] Zelkowitz, M., Shaw, A. & Gannon, J. (1979) Principles of Software Engineering and Design. Englewood Cliffs: Prentice Hall.
- [Zölch-91] Zölch, M. & Dunckel, H (1991) Erste Ergebnisse des Einsatzes der "Kontrastiven Aufgabenanalyse". In: D. Ackermann & E. Ulich (Hrsg.) Software-Ergonomie '91 (S. 363-372). (Berichte des German Chapter of the ACM, Vol. 33). Stuttgart: Teubner.