

Eigenprogrammierung als Unterstützung individueller Arbeitsgestaltung

Arbeitskreis Software-Ergonomie Bremen *

Zusammenfassung

Der Arbeitskreis Software-Ergonomie Bremen stellt aufgrund eigener Erfahrungen bei der Einführung individualisierbarer Standard-Software zunächst die Notwendigkeit einer Arbeitsgestaltung mit einbeziehenden Ansätzen der Eigenprogrammierung dar. Eigenprogrammierung wird als von ExpertInnen unabhängige Eigenaktivität verstanden, die allerdings Restriktionen unterliegt. Mit Eigenprogrammierung werden insbesondere Aufgaben unterstützt, die ohne Realitätsverlust nicht exakt zu spezifizieren sind, und informelles Wissen berücksichtigt, das direkt ohne Bindung an formale Strukturen umgesetzt werden kann. Damit bleiben individuelle Arbeitsstile erhalten. Hieraus sich ergebende Formen individualisierter Software-Gestaltung werden auf der Basis eines vom Arbeitskreis entwickelten 5-Stufen-Modells aufgezeigt.

Hoffnungsträger zur Überwindung der Software-Krise

Die Automatisierung in Büro und Verwaltung ist seit einiger Zeit dort in eine Krise geraten, wo es darum geht, dispositive Tätigkeiten formalisieren und automatisieren zu wollen. Angesichts von Bedienungs- und Nutzungsproblemen, verursacht durch disfunktionale und intransparente Programme, sind Wirtschaftlichkeitsüberlegungen und Techniknutzungsprognosen zu reiner Spekulation geworden, nicht eingeplante zusätzliche Tätigkeiten in vor- und nachgelagerten Bereichen haben den erhofften Rationalisierungsgewinn längst kompensiert. In dieser Situation kommt die breite Diskussion um individualisierte Software-Gestaltung wie gerufen, die sogenannte "Software-Krise" scheint mittels Eigenprogrammierung überwunden werden zu können. Aufgabengerechte Dialogformen und Systemfunktionen, die von ExpertInnen aufgrund fehlender Detailkenntnisse und existierender Programmierengpässe nicht codiert werden können, sollen von den BenutzerInnen nun selbst realisiert werden.

* Im Arbeitskreis Software-Ergonomie Bremen arbeiten folgende WissenschaftlerInnen zusammen: Klaus Peter Hasler, Heinrich Riehl, Uwe Schläger, Johannes Schnepel, Wolfgang Taube, Gabriele Winker

Selbstverständnis von Eigenprogrammierung

Software-Ergonomie-ExpertInnen kommt diese Situation nicht ungelegen: Da Eigenprogrammierung der BenutzerInnen in der Praxis nicht so einfach umzusetzen ist, hat sich für WissenschaftlerInnen ein neues Betätigungsfeld ergeben, das den eigenen Arbeitsplatz- und Forschungsinteressen dient und gleichzeitig vielversprechend an Humanisierungsziele anknüpft. Dies gilt auch für den Arbeitskreis Software-Ergonomie Bremen, der ein Zusammenschluß von WissenschaftlerInnen ist, die in der öffentlichen Verwaltung in Bremen in verschiedenen Funktionen für die Umsetzung eines menschengerechten Technikeinsatzes verantwortlich sind.

Für uns hat sich die Auseinandersetzung mit Fragen der Eigenprogrammierung u.a. aus den Schwierigkeiten bei der Realisierung dieses Anspruches ergeben: Trotz unterschiedlichster Partizipationsmodelle in bremischen HdA-Projekten wie DEMOS (Dezentrales Einwohnermeldewesen-Online-System) und PROSOZ (Programm zur Unterstützung der Sozialhilfeverwaltung, vgl. Bergdoll u.a. 1991) und über Dienstvereinbarungen abgesicherter Beteiligungsstrukturen nehmen BenutzerInnen zu wenig an der Gestaltung ihrer eigenen Arbeitssituation und ihrer Arbeitsmittel teil. Trotz gefordertem Aufgabenbezug findet dezentraler Technikeinsatz weitgehend standardisiert statt. Dies ist umso problematischer, da sich nur mit realen Handlungs- und Ermessensspielräumen am Arbeitsplatz die immer vielfältiger und komplexer werdenden Verwaltungsaufgaben im Interesse der BürgerInnen sinnvoll bearbeiten lassen.

Eigenprogrammierung kann im Interesse der BenutzerInnen zur Arbeitsplatzhumanisierung eingesetzt werden und Eigenaktivität der Beschäftigten fördern. Dies setzt jedoch voraus, die Individualisierbarkeit nicht nur auf die Benutzungsoberfläche zu beschränken, sondern auch auf die Gestaltung der am Arbeitsplatz unterschiedlich benötigten Funktionalität auszuweiten. Mit der Untersuchung der Eigenprogrammierung verfolgen wir das Ziel, die Rolle der ExpertInnen zurückzudrängen zugunsten einer größtmöglichen Autonomie der BenutzerInnen bei der Gestaltung des eigenen Arbeitsmittels. Während in einem arbeitsteiligen Systementwicklungsprozeß eine Normierung und Umstrukturierung des Arbeitsprozesses angelegt ist, die oft nicht im Interesse der direkt Betroffenen liegt, können im Rahmen der Eigenprogrammierung die Betroffenen selbst ihren Arbeitsprozeß immer wieder neu gestalten. Da Eigenprogrammierung Stellvertreter-Partizipation überflüssig macht, ist sie sozusagen die höchste Form der Partizipation. Sofern Dritte betroffen sind, sind Beteiligungsstrukturen allerdings notwendig.

Hinter diesen Zielen verbirgt sich ein Menschenbild, das von dem Bedürfnis des Menschen nach Autonomie bzw. Handlungs- und Ermessensspielräumen ausgeht. Jeder Schritt, der den Beschäftigten ermöglicht, ihre Arbeitsinhalte, Arbeitsabläufe, Arbeitsmittel mitzugestalten, ist mittelfristig ein kleiner Schritt zur Demokratisierung der Arbeitswelt.

Voraussetzungen und Rahmenbedingungen

Aus der dargestellten Zielbestimmung lassen sich die dafür notwendigen Voraussetzungen direkt ableiten: Je mehr Gestaltungsspielräume bei der Arbeit existieren, desto sinnvoller wird Eigenprogrammierung. Ohne Gestaltungsspielräume verkommt Eigenprogrammierung zur technischen Spielerei, die von der Notwendigkeit der arbeitsorganisatorischen Umgestaltung solcher Arbeitsplätze eher ablenkt als daß sie eine sinnvolle Arbeitsgestaltung unterstützt. Auch sollten Kooperationsbereitschaft und -möglichkeiten unter den BenutzerInnen vorhanden sein, um der Gefahr der extremen Individualisierung zu begegnen und einen Austausch über sinnvolle Arbeitsstrukturen und technische Unterstützungsmöglichkeiten zu gewährleisten. Neben diesen aufgaben- und arbeitsplatzbezogenen Voraussetzungen müssen im Hinblick auf eine Arbeitsgestaltung mit einbeziehende Eigenprogrammierung weitere Rahmenbedingungen im Vorfeld geschaffen werden.

Arbeitssystembezogene und benutzerzentrierte Qualifizierungs- und Beratungskonzepte sind notwendig, um einerseits den Beschäftigten die Möglichkeit zu geben, ihre Fachkompetenz im Entwicklungsprozeß entsprechend einsetzen und nutzen zu können, andererseits aber die Vermittlung von aufgabenunabhängigem Vorrats- und Balastwissen zu vermeiden.

Eigenprogrammierung kann nur dann zu funktionalen und benutzungsfreundlichen Systemen führen, wenn entsprechende Softwareentwicklungsumgebungen zur Verfügung stehen. In dieser Hinsicht müssen im weiteren Verlauf der Auseinandersetzung um Eigenprogrammierung operationalisierbare Kriterien formuliert werden, die die Auswahl von Software-Tools aus Sicht der Beschäftigten erleichtern.

Eigenprogrammierung erfordert im Gegensatz zu traditioneller Systementwicklung eine weitgehende Abkehr von objektivierenden bzw. bedingungsbezogenen Arbeitsanalyseverfahren wie z.B. VERA-S oder TBS (vgl. Oesterreich, Volpert 1987, S. 55-61). Um inter- und intraindividuelle Unterschiede berücksichtigen und realisieren zu können, sind, wenn überhaupt, subjektive Verfahren wie STA einzusetzen (vgl. Ulich 1981). Es

erscheint uns fraglich, ob eine detaillierte Beschreibung des Istzustands auf der Basis handlungstheoretischer Modelle überhaupt angebracht ist (vgl. Holzkamp 1986). Die Systemspezifikation erhält bei der Eigenprogrammierung sowieso einen anderen Charakter. Da der Transformationsprozeß zum Programmierer wegfällt, benötigen Beschäftigte als direkt Betroffene des Istzustands nicht dessen formale Beschreibung, sondern eher Methoden, die die kollektive Auseinandersetzung zur Verbesserung des Istzustands unterstützen (z.B. Zukunftswerkstatt, verschiedene Visualisierungsmethoden, vgl. Baitsch u.a. 1989, S. 141-162).

Aus dem dargestellten Selbstverständnis und den Rahmenbedingungen ergibt sich aus unserer Sicht zusammenfassend folgende Definition: Eigenprogrammierung ist die von DV-ExpertInnen relativ unabhängige, eigenständige Gestaltung (Programmierung, Anpassung, Nutzung) von Software durch die BenutzerInnen hinsichtlich Systemfunktionalität und Benutzungsoberfläche unter Berücksichtigung der konkreten Fachaufgabe und der inter- und intraindividuellen Problemlösungsstrategien. Eigenprogrammierung findet im Rahmen von Arbeitsgestaltung auf der Grundlage eines Qualifizierungs- und Beratungskonzeptes statt und kommt verstärkt dort zum Einsatz, wo es individuelle Handlungs- und Ermessensspielräume gibt. Kooperationsbeziehungen bei der Eigenprogrammierung sind dort erforderlich, wo andere direkt oder indirekt betroffen sind. BenutzerInnen, die für andere BenutzerInnen gemeinsam zu nutzende Lösungen erstellen, sind allerdings für diese KollegInnen bereits wieder ExpertInnen.

Restriktionen und Probleme bei Eigenprogrammierung

AutorInnen, die sich durch Eigenprogrammierung lediglich die Lösung von Bedienungsproblemen versprechen, sehen als Schwächen dieses Ansatzes primär den notwendigen hohen Qualifizierungs- und Beratungsaufwand und befürchten suboptimale Technikausnutzung durch überforderte BenutzerInnen (vgl. Krause 1988, S.9-10). Ausgehend von motivierten und lernbereiten BenutzerInnen sind wir jedoch der Meinung, daß sich der Qualifizierungsaufwand langfristig positiv auf Arbeitsergebnisse auswirkt und arbeitssystembezogene Beratungsangebote Defizite bei der Technikenutzung ausgleichen können. Aus eigener Praxis sehen wir allerdings eine Reihe anderer möglicher Probleme, die bei Eigenprogrammierung auftreten können:

- Einschränkung der Software-Qualität hinsichtlich Wartung und Systemsicherheit,
- uneinheitliches Verwaltungshandeln aufgrund individueller Lösungen,
- Realisierungsprobleme bei zu komplexen Aufgabenstellungen,
- unzureichender ArbeitnehmerInnen- und KlientInnendatenschutz bei Anwendungen mit sensiblen personenbezogenen Daten,

- Zurückdrängen der Fachlichkeit zugunsten von technikzentrierten Fragestellungen,
- zu hoher Anteil an Bildschirmarbeit.

Diese Probleme sollten unserer Meinung nach jedoch nicht den Stellenwert von Eigenaktivität bei technischen Aufgabenstellungen von vornherein begrenzen, sondern vielmehr zu einer differenzierten Sicht der Eigenprogrammierung führen.

5-Stufen-Modell

Je nach konkreter Aufgabenstellung und Arbeitsteilung, vorhandenen Handlungs- und Ermessensspielräumen, individuellen Neigungen und Fähigkeiten der BenutzerInnen kann das sinnvolle Ausmaß der Übernahme technischer Aufgaben durch die BenutzerInnen unterschiedlich sein. Hierzu haben wir ein 5-Stufen-Modell entwickelt, dessen Stufen und deren Einsatzfelder im folgenden dargestellt werden.

zentral erstellte Software ohne Individualisierbarkeit	zentrale Software, keine Individualisierbarkeit der Funktionalität	zentral erstellte Programm-bibliothek und Schnittstellen	zentral erstellte Datenbasis	
	Gestaltung der Dialog-schnittstelle (z.B. Maskengenerator)	Kombination zentraler Moduln und Eigenprogrammierung	Eigenprogrammierung (z.B. Abfragesprachen)	Eigenprogrammierung (z.B. Makroprogrammierung)
Stufe 1	Stufe 2	Stufe 3	Stufe 4	Stufe 5

Stufe 1 stellt das klassische Modell der Software-Entwicklung dar. BenutzerInnen können am Entwicklungsprozeß beteiligt sein, übernehmen auf der technischen Ebene aber keine eigenständigen Aktivitäten. Eine partizipative Systementwicklung, die den BenutzerInnen die Aufgaben der Anforderungsermittlung und des Testens zuweist, ist die weitestgehende Ausprägung einer beteiligungsorientierten Systementwicklung auf dieser Stufe. Notwendig mag dieses traditionelle Vorgehen der Software-Entwicklung einerseits bei Aufgaben sein, die wegen erforderlicher Systemsicherheit hohe Anforderungen an die Software-Qualität stellen. Andererseits können Tätigkeiten, bei denen es aufgrund restriktiver gesetzlicher Regelungen sowieso keine Handlungs- und Ermessensspielräume gibt, von vornherein durch zentral erstellte vollautomatisierte Systeme abgewickelt werden (z.B. Batch-Verfahren für gesetzlich festgelegte regelmäßige Auszahlungen).

In der Stufe 2 wird den BenutzerInnen die Funktionalität des Systems in Form der Datenbasis und der Verarbeitungslogik vorgegeben, so daß lediglich Maskenaufbau und Dialogsteuerung gestaltbar sind. Individuelle Masken- und Dialoggestaltung setzt benutzungsfreundliche Maskengeneratoren und Dialog-Manager voraus. Diese Stufe kann bei Tätigkeiten zum Einsatz kommen, die ähnlich wie Stufe 1 aufgrund gesetzlicher Regelungen keine Ermessensspielräume zulassen, die jedoch nicht voll- oder halbautomatisch abgewickelt werden können, sondern Eingaben von BenutzerInnen verlangen (z.B. Lohn- und Gehaltsabrechnungsverfahren). Damit kann wenigstens die Benutzungsoberfläche entsprechend den inter- und intraindividuellen Unterschieden von den BenutzerInnen selbst gestaltet werden.

Auf Stufe 3 kann die BenutzerIn über die Gestaltung der Dialogschnittstelle hinaus auch die Funktionalität des Systems selbst beeinflussen. Innerhalb dieser Stufe sind weitere Unterscheidungen möglich: Zum einen kann unter vorgegebenen Moduln einer Programmbibliothek ausgewählt und eine eigene Lösung konfiguriert werden, zum anderen können selbstgeschriebene Programme der BenutzerInnen mit zentral erstellten Moduln kombiniert werden. Dabei ist eine genaue und für die BenutzerIn verständliche Beschreibung der Schnittstellen zwischen den Moduln sowie der Datenbasis notwendig. Diese Stufe setzt bereits Tätigkeiten mit hohen Handlungs- und Ermessensspielräumen voraus. Wenn für Teilbereiche einer Anwendung mehrere BenutzerInnen die gleiche Funktionalität benötigen, kann es sinnvoll sein, diese zentral von DV-ExpertInnen erstellen zu lassen, zumal bei hoher Komplexität der Aufgabenstellung. Teilbereiche, die je nach Arbeitsplatz unterschiedlich abgearbeitet werden, können dagegen individuell von den BenutzerInnen programmiert werden (z.B. ortsteilabhängige Heizkostenabrechnung innerhalb der Sozialhilfesachbearbeitung).

Stufe 4 stellt zentral nur die Datenbasis zur Verfügung, auf die die BenutzerIn mit Hilfe geeigneter Abfragesprachen zugreifen kann. Dieses Vorgehen bietet sich bei Tätigkeiten mit wiederum hohen Ermessens- und Handlungsspielräumen an, wobei die einzelnen Aufgaben auf Basis einheitlicher Datenstrukturen abzuwickeln sind (z.B. Vorgangssachbearbeitung bei Bauanträgen).

Auf Stufe 5 programmiert die BenutzerIn das gesamte Programm mit allen Komponenten (Datenbasis, Funktionalität, Dialogsteuerung) selbst. Diese Stufe kann sinnvollerweise dort zum Einsatz kommen, wo Tätigkeiten mit hohen Ermessens- und Handlungsspielräumen bei gleichzeitig geringer horizontaler oder vertikaler Arbeitsteilung vorkommen (z.B. Miet- und Wirtschaftlichkeitsberechnungen im Amt für Wohnungsförderung).

Während Stufe 1 den konventionellen Software-Entwicklungsprozeß darstellt, bezeichnen die Stufen 2 bis 5 das unterschiedliche Ausmaß von Eigenprogrammierung im Softwareentwicklungsprozeß. Ziel dieses Modells ist es, unter Beachtung der oben genannten Probleme und Restriktionen dennoch weitgehende Möglichkeiten der Eigenprogrammierung für die BenutzerInnen zu eröffnen. Mit der Umsetzung der Stufe 2 wird nur ansatzweise, bei der Realisierung der Stufe 5 dagegen eine völlige Abkehr von der Dominanz der Spezialisten erreicht. Entsprechend gestärkt wird die autonome Verfügung der BenutzerInnen über ihre Arbeitsmittel.

Erste Ergebnisse aus der eigenen Praxis

In der bremischen Verwaltung haben wir neben dem konventionellen Software-Entwicklungsprozeß der Stufe 1, über deren Entwicklungen und Auswirkungen es bereits eine Reihe von verallgemeinernden Darstellungen gibt (vgl. u.a. Brinkmann, Kuhlmann 1990; Grimmer 1986), bisher Erfahrungen mit der Flexibilisierung der Bedienoberfläche entsprechend Stufe 2 und der vollständigen Eigenentwicklung auf der Grundlage von Standardsoftware entsprechend Stufe 5 gesammelt, aus denen sich folgende Thesen ableiten lassen:

1. Mit arbeitssystembezogener Qualifizierung und Beratung ist ein aufgabenangemessener individualisierter Technikeinsatz möglich.

Ohne entsprechende Systemkenntnisse können sich MitarbeiterInnen kaum gegenüber jenen neuen Technikhierarchien durchsetzen, für die DV-Beschaffungen und DV-Anwendungen mehr eine Status- und Prestigefrage ist als das Ergebnis einer Erforderlichkeitsprüfung. Diese technikzentrierten MitarbeiterInnen passen Programme zentral für alle anderen BenutzerInnen an und standardisieren damit wiederum flexible Bedienoberflächen. Dies muß nicht unbedingt so sein: Durch arbeitssystembezogene Qualifizierung und regelmäßige Workshops, auf denen aufgetretene Probleme besprochen werden, sind BenutzerInnen in der Lage, sich ausgehend von ihrer fachlichen Aufgabenstellung mit Hilfe der PC-Standardsoftware kleine Anwendungen selbst zu erstellen und die Bedienoberfläche entsprechend ihrem Arbeitsstil anzupassen (Makros zur flexiblen Drucksteuerung, Tastenbelegungen zur Verknüpfung von einzelnen Dateien mit entsprechenden Texten, Makros zur Bearbeitung aufwendiger und rechenintensiver Aufstellungen mit gleichzeitigem Formularausdruck etc.).

2. Eigenprogrammierung eröffnet Frauen neue Gestaltungsmöglichkeiten an ihrem Arbeitsplatz.

Benutzerinnen haben ausgehend von ihrer Aufgabenstellung häufig Kritik an der Funktionalität wie am Bedienkomfort der Standard-Programme. Insbesondere wird jedoch gerade ihnen von den selbsternannten - in der Regel männlichen - DV-Experten bis zur letzten Tastenbelegung alles zentral vorgegeben, was beinahe zwangsläufig ihre Experimentierbereitschaft mit der Technik mindert. Wird Frauen jedoch die Kompetenz und die Zeit zur Eigengestaltung ihres Arbeitsmittels und ihres Arbeitsablaufs zugestanden, sind sie nach entsprechender Qualifikation durchaus motiviert, ihren Arbeits- und Programmablauf entsprechend der Aufgabenstellung zu verändern. Somit stellt sich die in der Literatur beschriebene andere Herangehensweise von Frauen an Technik (vgl. u.a. Schiersmann 1987) primär als ein Hierarchieproblem verknüpft mit selbstherrlichem Technik-Bluff einzelner Männer dar.

3. Bei Eigenprogrammierung wird die einmalige strukturierte Arbeitsanalyse von Dritten ersetzt durch permanente Optimierung der Arbeitsabläufe durch die Beschäftigten.

Mit Zunahme der Eigenprogrammierung verändert sich der Arbeitsanalyseprozeß. Die Notwendigkeit einer exakten Spezifikation für die DV-ExpertInnen entfällt. Nachdem in einer Vorlaufphase der eigentliche Gegenstand abgegrenzt ist, Probleme diskutiert und hieraus erste Veränderungen abgeleitet werden, können BenutzerInnen mit einer informellen, "weichen" Spezifikation arbeiten und diese in Programme umsetzen. Die Programmabnahme erfolgt dann nicht als Vergleich vorher beschriebener Anforderung mit der erstellten Software, sondern durch eigene Überlegungen der BenutzerIn, inwieweit sie aus ihrer Sicht die Arbeitsaufgabe mit dem System zufriedenstellend lösen kann. Sie kann dann weitere Veränderungen vornehmen, das System optimieren und so schrittweise ihren Arbeitsablauf besser organisieren (vgl. Denkmodell "lernendes System", Bonin 1988).

4. Eigenprogrammierung kann zur Enthierarchisierung der Arbeitssystemgestaltung beitragen.

Im Gegensatz zur arbeitsteilig organisierten Systementwicklung kann Eigenprogrammierung unter Umgehung formaler und hierarchischer Strukturen kurzfristig erste Prototypen hervorbringen, die dann wiederum Gegenstand weiterer Auseinandersetzung in der Gruppe sind. Hierdurch werden Abhängigkeiten von DV-ExpertInnen verringert und interessengeleitete Lösungen diskutiert bzw. realisiert, die ohne die

Möglichkeit der Eigenprogrammierung von vornherein durch höhere Hierarchieebenen, z.B. aus Angst vor Status- und Machtverlust, verhindert werden.

5. Eigenprogrammierung kann als Vehikel benutzt werden, um Fachlichkeit zu stärken und Arbeitsinhalte zu verbessern.

Die eigenverantwortliche Realisierung von Systemen zur Unterstützung komplexer Arbeitsaufgaben kann dazu beitragen, sich mit organisatorischen Aspekten wie Arbeitsinhalt und -ablauf anlaßbedingt kritisch auseinanderzusetzen und fachliche Fragen ausführlich zu diskutieren, die sonst nicht hinterfragt werden.

Ausblick

Trotz dieser ersten für das Konzept der Eigenprogrammierung aus Sicht der BenutzerInnen recht positiven Thesen bleiben darüberhinaus noch eine Reihe von grundsätzlicheren Fragen offen, die bisher in der Praxis noch nicht überprüft wurden. Wir wollen diese Fragen in einem beantragten Arbeit&Technik-Projekt überprüfen und führen sie hier als Ausblick auf derartige Diskussionen an:

- Nimmt die Handlungs-, Verfahrens- und Gestaltungsautonomie über die Eigenprogrammierung bei den BenutzerInnen langfristig zu? Entwickeln und verändern sie ihre Arbeitsmittel entsprechend ihrem sich ändernden Arbeitsstil? Nehmen sie über die Technikgestaltung hinaus auch verstärkt auf ihre Aufgabenstellung Einfluß?
- Gelingt es durch arbeitssystembezogene Qualifizierungs- und Beratungskonzepte, die Fachlichkeit im Rahmen der Eigenprogrammierung zu erhalten oder auszubauen? Oder bestätigen sich die Befürchtungen, daß sich die Arbeitszeitanteile zugunsten maschinengebundener Zeit und zuungunsten des Fachwissens verschiebt?
- Können bei der Eigenprogrammierung neue Kooperationsbeziehungen unter den Beschäftigten entstehen oder verstärkt sich die Vereinzelung?
- Nimmt die Arbeitsteilung zwischen ExpertInnen und BenutzerInnen ab? Gibt es stattdessen neue Formen der Arbeitsteilung, in der sich die technikzentrierten MitarbeiterInnen in einer Abteilung profilieren? Gibt es technikzentriertes Herangehen nur bei Männern oder auch bei Frauen?
- Lassen sich ungleiche Artikulations- und damit Durchsetzungschancen der einzelnen MitarbeiterInnen im Rahmen der Eigenprogrammierung abbauen oder werden sie verstärkt? Wie läßt sich im Gestaltungsprozeß Chancengleichheit herstellen?

Literaturverzeichnis

Baitsch, C.; Katz, C.; Spinas, P.; Ulich, E.: Computerunterstützte Büroarbeit. Ein Leit-faden für Organisation und Gestaltung. Zürich 1989

Bergdoll, K.; Friedrich, J.; Schläger, U.; Schnepel, J.: Computergestützte Sozialhilfever-waltung (Projekt PROSOZ). Bremen 1991 (in Vorbereitung)

Bonin, H.: Die Planung komplexer Vorhaben der Verwaltungsautomation. Heidelberg 1988

Brinckmann, H.; Kuhlmann, S.: Computerbürokratie. Ergebnisse von 30 Jahren öffent-licher Verwaltung mit Informationstechnik. Opladen 1990

Grimmer, K. (Hrsg.): Informationstechnik in öffentlichen Verwaltungen. Handlungs-strategien ohne Politik. Basel 1986

Holzcamp, K.: Handeln, in: Rexilius, G.; Grubitsch, S. (Hrsg.): Psychologie. Theorien, Methoden, Arbeitsfelder. Reinbek 1986

Kommunale Gemeinschaftsstelle für Verwaltungsvereinfachung (KGSt): Informations-technische Infrastruktur in Kommunalverwaltungen. Köln 1989

Krause, J.: Adaptierbarkeit, Adaptivität, Intervenierbarkeit und Hilfesysteme. Ergänzende Konzepte zu grafischen Benutzerschnittstellen auf der Grundlage erweiterter Kontextsensitivität und Wissensbasiertheit. Regensburg, Juli 1988

Oesterreich, R.; Volpert, W.: Handlungstheoretisch orientierte Arbeitsanalyse, in: Kleinbeck, U.; Rutenfranz, J.: Arbeitspsychologie (Enzyklopädie der Psychologie, Band 1) Göttingen, Toronto, Zürich 1987, S. 43-73

Riehl, H.; Schäfer, W.; Treeck, W.van: Selbstgestaltung in der Verwaltungsarbeit. Entwicklung von kommunalen DV-Anwendungssystemen durch den Sachbearbeiter in A. Kassel 1989

Schiersmann, Ch.: Computerkultur und weiblicher Lebenszusammenhang. Zugangs-weisen von Frauen und Mädchen zu neuen Technologien. Bonn 1987

Schwellach, G.; Winker, G.: Veränderungen der Arbeitssituation von Frauen in der bremischen Verwaltung durch den verstärkten Einsatz von PCs im Bereich der Text- und Sachbearbeitung, in: Schelhowe, H. (Hrsg.): Frauenwelt - Computerräume, GI-Fachtagung im September 1989 in Bremen, Proceedings. Berlin, Heidelberg 1989, S.41-47

Ulich, E.: Subjektive Tätigkeitsanalyse als Voraussetzung autonomie-orientierter Arbeitsgestaltung, in Frei, F.; Ulich, E. (Hrsg.): Beiträge zur psychologischen Arbeits-analyse. Bern 1981, S. 327-347

Arbeitskreis Software-Ergonomie
c/o Gabriele Winker
Rechenzentrum der bremischen Verwaltung
Achterstr. 30
2800 Bremen 34