

UNTERSUCHUNG PROGRAMMIERBARER SOFTWARESYSTEME
ANHAND TÄTIGKEITSBEZOGENER UND QUALIFIKATORISCHER
KRITERIEN DER SOFTWARE-ERGONOMIE

Claudia Döbele-Berger und
 Gisela Schwellach, Kassel

Zusammenfassung: Der Einsatz programmierbarer Software in Fachabteilungen ist eine Entwicklungslinie, die an Bedeutung gewinnt. Von der Software her sind Gestaltungsspielräume und die Möglichkeit einer Aufgabenintegration gegeben. Diese werden beispielhaft beschrieben.

Programmierbare Software für Anwendungsentwicklung
durch Sachbearbeiter

Heutige Softwaresysteme beinhalten arbeitsorganisatorische Gestaltungsspielräume, die die Chance einer Aufhebung stark arbeitsteiliger Arbeitsprozesse und das Zusammenfügen der bisher häufig getrennten Tätigkeitselemente von Planung, Ausführung und Kontrolle bieten. Diese Aufgabenintegration ist bei vielen Softwaresystemen festzustellen - sowohl bei Standard-Softwaresystemen als auch bei den von uns untersuchten Systemen der "Individuellen Datenverarbeitung" (IDV). Ein Beispiel dafür sind im Bereich der Personal Computer die integrierten Softwaresysteme, wobei die Integration durch Zusammenfassen verschiedener Aufgabengebiete (z. B. Textverarbeitung, Graphik, Tabellenkalkulation, Datenbank u. a.) unter einer Benutzeroberfläche gegeben ist. Bei diesen integrierten Systemen sind verschiedene qualitative Stufen zu verzeichnen. Sie reichen von dem Datenaustausch zwischen den einzelnen Modulen dieser Systeme durch eine gemeinsame Datenstruktur mit teilweise gleichem Datenformat, über Systeme mit "Programmierung" durch Makrobildung, die eine Integration der Programmsystemkommandos zu neuen "Abläufen" erlauben, bis zu umfassenden Softwaresystemen, (als programmierbare Software bezeichnet), die eine spezielle Programmiersprache, mit der universelle Programme entwickelt werden können, beinhalten.

Eine entsprechende Entwicklung im Bereich der Großrechenanlagen ist die Verfügbarmachung von Teilen bzw. des gesamten Softwaresystems der Groß-EDV auf Personal Computer unter vereinfachter Benutzeroberfläche, so daß am Arbeitsplatz wieder Funktionen zur Verfügung stehen, die bisher aufgrund der zentralen DV-Anlagen ausgelagert waren. Dadurch muß für kleinere Ad-hoc-Anwendungen nicht die zentrale DV-Anlage benutzt werden, während für große Datenmengen und/oder hohen Rechenaufwand die HOST-Kapazität zur Verfügung steht. Die Entwicklung von Anwendungsprogrammen kann dezentral am einzelnen Personal Computer erfolgen.

Um mit Hilfe dieser integrierten Softwaresysteme auch eine Verlagerung der Kompetenzen in die Fachabteilungen zu erreichen, sind Systeme notwendig, die Gestaltungsspielräume zur Problemlösung beinhalten. Damit ist im besonderen der Einsatz programmierbarer Software gemeint, die eine solche Dezentralisierung der Anwendungsentwicklung möglich macht. Dies heißt nichts anderes, als daß Sachbearbeiter, also Fachkräfte in den Fachabteilungen, Anwendungsprogramme entwickeln und somit Programmier Tätigkeiten an Endbenutzer delegiert werden (auch mit Benutzerprogrammierung bezeichnet). Anwendungsentwicklung am Arbeitsplatz bezieht sich damit auf den Arbeitsplatz des Fachgeschulten, nicht des DV-Spezialisten. Anwendungsentwicklung soll bedeuten, daß aktiv verändernd, erweiternd, entwickelnd mit Software gearbeitet, also ausdrücklich nicht mit fix und fertig vorgegebener, schlüsselfertiger, abgeschlossener, standardisierter Software umgegangen wird. Für den Sachbearbeiter könnte dies eine Höherqualifizierung bedeuten. Nach unserer Auffassung bieten nur programmierbare Softwaresysteme diese Möglichkeit einer Höherqualifizierung.

Im Rahmen unseres Beitrages wollen wir ausschließlich den Teil der Software berücksichtigen, der Anwendungsentwicklungen für die Sachbearbeitung ermöglicht. Das heißt, es muß eine Sprache für den Endbenutzer (z. B. Programmiersprache der vierten Generation) zur umfassenden Erledigung der Arbeitsaufgabe vorhanden bzw. zumindest die Möglichkeit der Makrobildung, also die Zusammenfassung von Befehlsabfolgen zu einem neuen Befehl, gegeben sein.

Experimentelle Softwareuntersuchungen - eine Methode zur Feststellung von technisch-organisatorischen Gestaltungsspielräumen

Im Rahmen des Forschungsprojektes "Softwarenutzung am Arbeitsplatz und berufliche Weiterbildung", gefördert vom Bundesministerium für Bildung und Wissenschaft und vom Hessischen Ministerium für Wirtschaft und Technik, führen wir neben betrieblichen Fallstudien experimentelle Softwareuntersuchungen durch. Die ausgewählten Softwaresysteme umfassen dabei den Bereich der industriellen Planung und Verwaltung.

Die Softwareuntersuchungen sind eine Methode zur Feststellung der arbeitsorganisatorischen Freiräume und der inhaltlichen Möglichkeiten, die die Software beinhaltet. Sie sind weiterhin eine Methode der experimentellen Analyse der Qualifikationsanforderungen. Im Vordergrund stehen dabei die zwei Hauptfragen:

- Welche arbeitsorganisatorischen Freiräume bietet die Software?
- Welche Anforderungen stellt die Software an die Benutzer?

Ausgangspunkt der Untersuchungen ist eine Typisierung der Software nach aufgabenbezogenen Bereichen. Wir unterscheiden einmal nach funktionalen Softwarebereichen (Integrierte Pakete, Datenbanksysteme, CAD, Planungssysteme) und zum anderen nach Möglichkeiten der Dialoggestaltung/-führung (Sprache, Menüsteuerung und Sprache, ICON, Makrobildung). Des weiteren treffen wir eine Unterscheidung nach hardwaremäßigen bzw. informationstechnischen Voraussetzungen (Personal Computer, HOST, Netzstrukturen/Kommunikation), da gerade die organisatorische Einbettung der programmierbaren Software (zentrale und dezentrale Nutzung) für die Möglichkeit der Anwendungsentwicklung entscheidend ist.

In den Softwareuntersuchungen wird einmal ein Vergleich der integrierten Systeme (ohne HOST-Kopplung) vorgenommen. Zum anderen werden Einzelpakete zu den Bereichen CAD, Datenbanksysteme, Planungssysteme und Textverarbeitung miteinander verglichen. Des weiteren erfolgt ein Vergleich von integrierter Software und Einzelpaketen, z. B. wird das Kalkulationsmodul - als ein Bestandteil eines integrierten Softwaresystems - mit einem Kalkulationsprogramm verglichen.

Auf der Grundlage dieser Untersuchungen können Schlußfolgerungen über das Vorhandensein von arbeitsorganisatorischen Freiräumen, die in der Software enthalten sind, sowie über allgemeine softwarebezogene Qualifikationsanforderungen gezogen werden.

Kriterien für die Softwareuntersuchungen

Wir gehen in unseren Untersuchungen von der Erkenntnis aus, daß ein Wirkungszusammenhang zwischen der Entwicklung und Anwendung neuer Technologien, der Veränderung von Arbeitsorganisation und Arbeitsinhalt sowie von Qualifikationsanforderungen und Qualifizierungsmaßnahmen besteht. Jede Ebene bietet verschiedene Gestaltungsmöglichkeiten. Von diesem Wirkungszusammenhang und den Gestaltungsmöglichkeiten ist bei der Kriterienentwicklung für die Softwareuntersuchungen auszugehen, d. h. die Kriterien müssen die verschiedenen Ebenen berücksichtigen (Döbele-Berger/Schwellach, 1986 und 1987).

Bei der Kriterienentwicklung wurden auch die Grundsätze der Software-Ergonomie (DIN-Entwurf 66234, Teil 8: Grundsätze der Dialoggestaltung) berücksichtigt, die wir im Hinblick auf arbeitsorganisatorische Gestaltungsspielräume und Qualifikationsanforderungen sowie Qualifizierungsunterstützung konkretisiert und operationalisiert haben.

Auf dieser Grundlage haben wir die nachfolgenden Kriterienbereiche festgelegt und dafür detaillierte Kriterien entwickelt (ausführlich in Döbele-Berger/Schwellach, ebd.):

A) Arbeitsorganisatorische Freiräume

- o Wie ist die Steuerbarkeit/Dialoggestaltung zu beurteilen?
- o Welche softwaremäßigen Benutzungshierarchien gibt es?
- o Für welche Aufgaben/Anwendungen ist das System einsetzbar?
- o Welche Kooperationsnotwendigkeiten bestehen?

B) Qualifikation

- o Welche Qualifikation ist beim Benutzer vorhanden?
- o Welche Qualifikationsanforderungen stellt das System?
- o Welche Qualifizierungsunterstützung bietet das System?

C) Informationstechnische Voraussetzungen

- o hardwaremäßig (Prozessor, Speicherkapazität)
- o "organisatorisch" (PC, HOST)

- o softwaremäßig (Betriebssystem)
- o kommunikationstechnisch (z. B. Datennetze)

Exemplarischer Vergleich einzelner Softwarepakete

Die Kriterien bilden die Grundlage für die Softwareuntersuchungen. Im folgenden werden wir anhand eines tabellarischen Vergleiches beispielhaft die Differenzen zwischen den Softwarepaketen Framework und KnowledgeMan-Plus als integrierte Systeme aufzeigen.

Das Universal-Softwarepaket Framework II basiert auf der Frame-Technik. Alle Daten werden in Bildschirmdarstellungselemente gegliedert, die sogenannten "Frames" (Rahmen). Diese können in beliebiger Anordnung auf dem Bildschirm plaziert werden. Der Bildschirm ist dabei die Arbeitsfläche des elektronischen Schreibtisches, auf dem mit Frames hantiert wird.

Jedes Frame kann wiederum aus Unterframes bestehen, wobei die Gliederungstiefe allein durch die Anforderungen des Benutzers bestimmt wird. Selbst dreidimensionale Datenhierarchien lassen sich leicht mit Framework darstellen, denn in sich gegliederte Frames und Unterframes können außerdem durch Übereinanderstapeln querverbunden und vernetzt werden. Den Komplexitätsgrad dieser Strukturen bestimmt der Benutzer. Ihm stehen folgende Module zur Verfügung:

- Textverarbeitung
- Datenbankverwaltung für kleine Datenbanken
- Tabellenkalkulation
- Graphik
- Telekommunikation

Das integrierte Softwaresystem KnowledgeMan-Plus umfaßt wie Framework alle Module für einen integrierten Sachbearbeiterarbeitsplatz wie Textverarbeitung, relationale Datenbank, Tabellenkalkulation, Graphik, Masken- und Reportgenerator. Es enthält für alle Module eine umfangreiche Kommandosprache sowie zahlreiche mathematisch-logische Operationen und weitgehend automatisch durchgeführte statistische Analysen, mit denen komplexe individuelle Programme entwickelt werden können.

Software Kriterien	Framework	KnowledgeMan-Plus
ARBEITSORGANISATORISCHE FREIRÄUME		
<u>Steuerbarkeit/Dialoggestaltung</u>		
Dialoggestaltung	Menüsteuerung: Anwendung, Laufwerk, Neu, Editieren, Suchen, Frames, Text, Graphik, Zahlen, Drucken Programmiersprache FRED Die Menüsteuerung in Framework unterscheidet sich von anderen dadurch, daß sie nicht hierarchisch untergliedert ist und Befehle einzelner Menüteile für andere Anwendungen verwendbar sind, so daß Umschaltungen zwischen den Menüs permanent erfolgen und auch notwendig sind.	Menüsteuerung für Maskengenerator Kommandosprache für alle anderen Module des Systems
Steuerungsvielfalt	entweder Sprache oder Menü; nicht umschaltbar	keine Wahlmöglichkeit, außer Maskengenerator über Menü oder Sprache
Sprache	eigene Sprache FRED zur Definition von Beziehungen (Funktionen); setzt auf der Frametechnik auf	eigene Sprache, die alle Programmbefehle integriert
- Wortschatzumfang	ca. 160 Befehle	ca. 200 Befehle
- Syntax	Funktionsschreibweise (angelehnt an LISP)	annähernd englisch
- deutsch/englisch	englisch	englisch mit teilweise deutschen Synonymen
- prozedural/nichtprozedural	prozedural/funktionsorientiert	strukturiert/prozedural

Software Kriterien	Framework	KnowledgeMan-Plus
- Unterprogrammtechnik/ Makro	ja Makro kann als Protokoll der Menüsteuerung angelegt werden, so daß der Benutzer die Sprache FRED nicht kennen muß, um umfassende Anwendungen entwickeln zu können.	ja
Dialogsprache (Antworten)	Menü: deutsch FRED: englisch	deutsch
Integrations- ebene	Integration unter einer Benutzeroberfläche (Desktop): - Fenstertechnik - in Form von Frames - Pulldown-Menüs Dateiformate: Text-, Graphik- und Datenframes können zu einem neuen Frame zusammengefaßt werden Makro und Programmiermöglichkeit	gleiches Dateiformat (bis auf DOS-Zugriff) für alle Module Makro und Programmiermöglichkeit unterschiedliche Benutzeroberfläche Datenschutzfunktionen: - Zugangsberechtigung - o USER-ID - o Password - Datenbankschutz - Feldschutz getrennt nach Lesen und Schreiben
inhaltliche Integration	Textverarbeitung Telekommunikation Datenbankverwaltung für kleine Datenbanken Tabellenkalkulation Graphik Konzeptentwicklung	Textverarbeitung Datenbankverwaltung Tabellenkalkulation Graphik Maskengenerator Reportgenerator
QUALIFIKATION		
<u>Qualifikations- anforderungen</u>		
Einarbeitung	Die Einarbeitung in die voreingestellte Standardoberfläche (Menüsteuerung) ist ohne EDV-	Für die interaktive Bearbeitung muß der volle Funktionsumfang nicht sofort präsent sein. Jedoch sind

Software Kriterien	Framework	KnowledgeMan-Plus
	Vorbildung möglich, aber aufgrund der Frametechnik ist eine hohe Abstraktionsleistung erforderlich. Alles, ob Text, Graphik, Datenbank, Spreadsheet, Programm, wird in Frames abgelegt, wobei Manipulationen am Frame selbst, am Inhalt und an den Formeln (Beziehungen) möglich sind. Durch die einheitliche Gestaltung der Masken und des Dialogs ist die an einer Stelle erlernte Vorgehensweise auf andere Module des Systems übertragbar.	Kenntnisse über die Benutzung von Variablen (Systemvariablen u. a.) notwendig, damit der Benutzer z. B. einen Text ausdrucken kann.
DV-Wissen	Hardwaretechnisches Detailwissen ist nicht notwendig. Allerdings muß die Frame-Philosophie verstanden sein, um Framework sinnvoll nutzen zu können. Für die Anwendungsentwicklung mit der funktionsorientierten Programmiersprache FRED ist Programmiererfahrung erforderlich. Kenntnisse in konventionellen Programmiersprachen nützen wenig.	Kenntnisse über Variablen, Systemsteuerung, Ausgabesteuerung sind notwendig. Für die Programmierung sind Kenntnisse über strukturierte und prozedurale Programmierung erforderlich.
fachliche Kenntnisse	Fachliche, wie auch arbeitsplatzübergreifende Kenntnisse sind unbedingt notwendig.	Wie bei Framework sind fachliche, arbeitsplatzbezogene sowie -übergreifende Kenntnisse erforderlich.
Abstraktionsvermögen	FRED ist sehr formelorientiert (hohe Abstraktionsleistungen auch aufgrund der Frametechnik).	-

Software Kriterien	Framework	KnowledgeMan-Plus
<u>Qualifizierungsunterstützung</u> einheitliche Struktur Hilfefunktionen Lernhilfen - Teachware - Handbuch	ist vorhanden Das Online-Hilfesystem bietet vorgehensbezogene Unterstützungen an allen Stellen von Framework (über Funktionstaste F1). In ihrem Aufbau gleichen sie den Seiten eines manuellen Handbuchs. Es gibt ein Stichwortindex. Tutorial-Disketten (für die erste Einführung am Gerät - interaktives Lernprogramm mit selbstgewählten Vorführungen) Das Handbuch umfaßt 4 Bände: - Arbeitsmappe (erste Einführung) - Einführung (Framework II-Module) - Nutzung (Hilfestellung zu fortgeschrittenen Anwendungen) - Programmierung und Datenübertragung Auf Selbstschulung ausgerichtet; anschauliche Erläuterungen in deutscher Sprache mit graphischen Hilfestellungen. Die Handbücher enthalten sehr viel Querverweise, so daß schnelles Nachschlagen z. T. schwierig ist.	nein, unterschiedliche Benutzeroberfläche, kann aber programmiert werden Das Hilfesystem enthält Informationen zu Modulen des Systems, zu allen Kontrolltasten, zu einzelnen Befehlen, Nutzvariablen, Systemvariablen; es gibt jedoch keine vorgehensbezogene Hilfestellung. Es gibt einige wenige Beispiele (auf der Diskette), die auch im Handbuch beschrieben sind (DB, Maske, Graphik, Tabelle). Es gibt ein Handbuch (keine geeignete Lernhilfe). Es ist sehr unübersichtlich aufgebaut (ohne Index); Verweise sind nicht nachvollziehbar: z. B. Konfigurationshilfe ist unter "Fehlermeldungen" zu finden. Viele Befehle stehen nicht im alphabetischen Verzeichnis.

Software Kriterien	Framework	KnowledgeMan-Plus
Zuverlässigkeit - einheitliche Kommando-sprache Zustands-meldungen Fehlermeldungen	vorhanden Funktionsschreibweise in FRED. Menüsteuerung ist sehr einheitlich. Mit Zwischenzustands- meldungen wird jeder eingeleitete Arbeits- gang kommentiert. Einzeilige Meldungen erläutern die Fehler- haftigkeit von Eingab- en. Diese Fehler- meldungen in FRED sind für den Anfänger nicht sehr hilfreich, da das Denken in "Frames" erst erlernt werden muß. Beispiel: "nichtdefinierter Be- zug" als Fehlermel- dung für eine Summen- formel Sum (a,b). Es fehlt die Ergebnis- variable.	unterschiedlich Auch der Bildschirmaufbau ist unterschiedlich. teilweise Es werden Fehlermeldungen gegeben. Sie sind aber häufig nicht selbsterklä- rend. Plausibilitätsprü- fungen sind programmier- bar.

Schlußbemerkung

In den Softwareuntersuchungen geht es uns um das Feststellen von softwaretechnischen und arbeitsorganisatorischen Gestaltungsspielräumen. Allerdings werden diese Untersuchungen unter laborähnlichen Bedingungen durchgeführt, so daß von daher die spezifischen betrieblichen Organisationsstrukturen und Aufgabenbereiche nicht berücksichtigt werden können. Die Analyse einzelner Softwaresysteme und ihre vergleichende Betrachtung anhand der von uns entwickelten Kriterien vermitteln uns aber auch eine detaillierte Kenntnis der Software, so daß eine tiefergreifende Beurteilung des Einsatzes dieser Software in den Betrieben möglich ist. Hier geht es um die Überprüfung der betrieblichen Einsatzkonzepte dahingehend, inwieweit die von uns festgestellten Gestaltungsmöglichkeiten auch tatsächlich genutzt werden, und zu ergründen, worin möglicherweise betriebliche Hemmnisse einer umfassenden Ausschöpfung bestehen. Erst auf dieser Basis wird beispielsweise eine Beurteilung möglich sein, wie arbeitsorganisatorische und qualifikatorische Veränderungen für die Fachkräfte in den Fachabteilungen aussehen, wenn sie selbst Anwendungsprogramme entwickeln.

Literatur

- BALZERT, H. (Hg.): Software-Ergonomie, Stuttgart 1983
- BEUSCHEL, W.: Qualifikationssicherung beim Einsatz von DV-Systemen. In: SCHULZ, A. (Hg.), a. a. O., S. 773 ff.
- BULLINGER, H.-J. (Hg.): Software-Ergonomie '85. Mensch-Computer-Interaktion, Stuttgart 1985
- DIN-Entwurf 66234, Teil 8: Grundsätze der Dialoggestaltung (Entwurf), Dezember 1984
- DÖBELE-BERGER, C./PINKVOHS, W./SCHWELLACH, G./ZIMMER, G. (Hg.): Softwarenutzung am Arbeitsplatz und berufliche Weiterbildung. Arbeitspapiere der Forschungsgruppe Verwaltungsautomation, 43, Kassel 1985
- DÖBELE-BERGER, C./SCHWELLACH, G./ZIMMER, G.: Softwarenutzung am Arbeitsplatz und berufliche Weiterbildung (Zwischenbericht), Kassel, November 1985
- DÖBELE-BERGER, C./SCHWELLACH, G.: Programmieren in den Fachabteilungen - Untersuchung von Softwaresystemen anhand tätig-

- keitsbezogener und qualifikatorischer Kriterien der Software-ergonomie. In: SCHULZ, A. (Hg.): a. a. O., S. 544 ff.
- dies.: Programmieren in den Fachabteilungen - Neue Anforderungen an die Beschäftigten. In: Computer und Recht (Veröffentlichung in Vorbereitung, voraussichtlich März und April 1987)
- DZIDA, W.: Kognitive Ergonomie für Bildschirmarbeitsplätze. In: Humane Produktion 2 (1980), S. 18 f.
- EIBL, G./WIMMER, K.: Architekturkonzept eines computergestützten Bürosystems. In: REICHWALD, R. (Hg.): Neue Systeme der Büro-technik, Berlin 1982, S. 255 ff.
- FLOYD, G./KEIL, R.: Integrative Systementwicklung - Ein Ansatz zur Orientierung der Softwaretechnik auf die benutzergerechte Entwicklung rechnergestützter Systeme, Eggenstein-Leopoldshafen 1984
- MÖLLER, K.-H. (Hg.): Softwaretechnik-Trends. Mitteilungen der Fachgruppe 'Software-Engineering', Heft 5-2, Juni 1985
- SCHULZ, A. (Hg.): Die Zukunft der Informationssysteme. Lehren der 80er Jahre. Berlin u. a. 1986
- SPINAS, P./TROY, N./ULICH, E.: Leitfaden zur Einführung und Gestaltung von Arbeit mit Bildschirmsystemen, München 1983
- ULICH, E.: Einige Anmerkungen zur Software-Psychologie. In: Humane Produktion 8 (1986) 1, S. 10 ff.
- VAN TREECK, W.: Programmieren durch Sachbearbeiter und Facharbeiter (Hg.: DGB-Kooperationsstelle Gewerkschaften/Hochschule Kassel, Kooperationsmaterialien Nr. 14), Kassel 1985
- WISSKIRCHEN, P. u. a.: Informationstechnik und Bürosysteme, Stuttgart 1983

Claudia Döbele-Berger
 Gisela Schwellach
 Forschungsgruppe Verwaltungsautomation
 Gesamthochschule Kassel - Universität
 Mönchebergstraße 17
 3500 Kassel