

Ergonomische Aspekte der Software-Gestaltung

Teil 8: Entwicklung einer Software unter Berücksichtigung ergonomischer Gesichtspunkte

von Gert Zülch, Volker Keller und Sascha Stowasser

Wie in Teil 1 dieser Artikelserie bereits dargelegt, ist der Nutzen der Software-Ergonomie nur indirekt und daher schwer messbar. Wird eine Software jedoch unzureichend ergonomisch gestaltet, sind die daraus resultierenden Effekte klar erkennbar. Diese Effekte können die Gesamtkosten einer Software inklusive Betreuung und Schulung der Benutzer bereits im ersten Jahr leicht verdoppeln. Aus diesem Grunde ist es wichtig, bereits bei der Entwicklung der Software ergonomische Aspekte konsequent zu berücksichtigen und somit die Kosten für eventuelle Nachbesserungen, erhöhte Betreuung sowie aufwendiger Schulung zu vermeiden.

Der vorliegende Artikel zeigt, wie software-ergonomische Gesichtspunkte in den Entstehungsphasen einer Software berücksichtigt werden sollten. Einzelne Aspekte, wie unterschiedliche Codierungsformen von Informationen, die Informationsanordnung sowie die Dialoggestaltung wurden bereits früher im Rahmen dieser Artikelserie erläutert und sollen nun exemplarisch angewendet werden.

Nach PAGEL und SIX (1994) lässt sich die Softwareentwicklung in die Phasen Pflichtenheft, Entwurf, Implementierung und Test unterteilen. Der Einsatz und Wartung zählen dann zur Nutzungsphase der Software. Nachfolgend sollen die in den Teilen 3 bis 6 der Artikelserie beschriebenen Aspekte der Software-Ergonomie den Phasen des Entwicklungsprozesses zugeordnet werden.

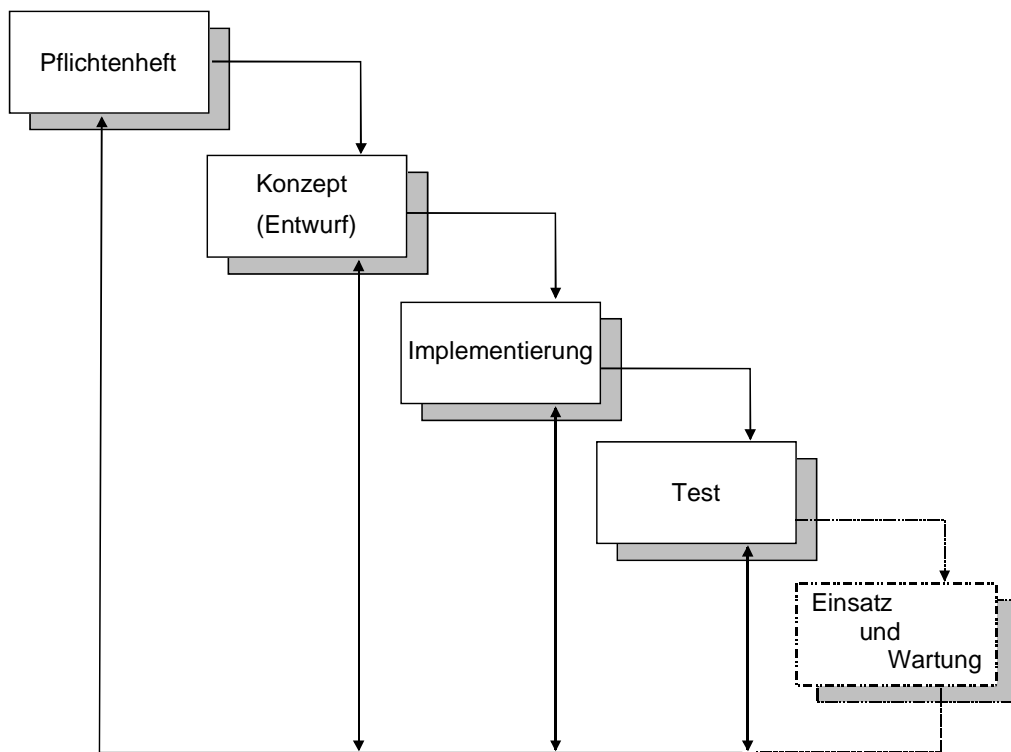


Bild 1: Iteratives Phasenmodell der Software-Entwicklung
(nach PAGEL, SIX 1994)

Analyse und Definition

In der ersten Phase der Entwicklung werden die Anforderungen an die Software bestimmt. Basierend auf den Anforderungen wird ein Pflichtenheft erstellt, in dem der Leistungsumfang der Software möglichst detailliert definiert wird. Typische Inhalte eines Pflichtenheftes sind der Funktionsumfang der Software, erste Festlegungen für die Benutzungsoberfläche, Vorgaben von Schnittstellen zur Systemumgebung, Angaben zum Leistungsverhalten, geplante Hard- und Software-Spezifikationen, Angaben zur Dokumentation der Software sowie Terminvorgaben (vgl. PAGEL, SIX 1994). Der Leistungsumfang der zu entwickelnden Software ist möglichst früh und umfassend zu definieren, da spätere Abweichungen bzw. Änderungen der Konzeption die Entwicklungskosten gravierend erhöhen. Zur Verfolgung der Entwicklungskosten können Kostenwachskurven eingesetzt werden, die Soll- und Ist-Verlauf der Entwicklung gegenüberstellen.

Wie bereits in Teil 1 der Artikelserie (ZÜLCH, KELLER, FISCHER 1997) erwähnt, müssen bei der Software-Gestaltung die Kriterien Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Fehlerrobustheit, Erwartungskonformität, Individualisierbarkeit sowie Erlernbarkeit beachtet werden. Diese Kenntnisse sollten bereits von Beginn der Softwareentwicklung an Berücksichtigung finden.

Daher es ist zunächst notwendig die mit der Software zu verrichtenden Funktionen zu analysieren. Nur auf Basis einer solchen Analyse kann eine ausreichende Funktionalität der zu erstellenden Software sichergestellt werden. Des Weiteren ist die Analyse der zukünftigen Benutzergruppe und deren Arbeitsaufgaben eine wichtige Voraussetzung für die ergonomische Gestaltung einer Software. Hierbei sind vor allem die DV-Kenntnisse, die Fachkenntnisse sowie das Abstraktionsvermögen der Benutzer von Interesse (vgl. SIEMENS 1984). Die Software muss auf die jeweilige Benutzergruppe abgestimmt sein und darf nur auf diejenigen Vorkenntnissen aufbauen, die tatsächlich in dieser Gruppe allgemein vorhanden sind.

Basierend auf diesen Analysen müssen im Pflichtenheft neben der Funktionalität auch die softwareergonomischen Gestaltungsaspekte aufgeführt werden. Solche Gestaltungsaspekte beinhalten z.B. den Leistungsumfang des Hilfesystems, die Verwendung verschiedener Dialogtechniken, eine erste Auswahl der Codierungsart (Spracheingabe oder Tastatureingabe, Sprachausgabe oder Bildschirmanzeige), Forderungen hinsichtlich selbsterklärender Abläufe oder auch eingeschränkter Individualisierbarkeit, wobei der Umfang bereits hier definiert werden sollte.

Am Ende dieser Phase steht ein erstes gedankliches Gerüst einer Software. Die zu verwendende Darstellungstechnik, beispielsweise Fenstertechnik oder Virtual Reality, ist damit festgelegt. Das geplante Einsatzgebiet der Software ist klar definiert, und es besteht eine feste Vorstellung bezüglich des Leistungsumfanges.

Entwurf

In der Entwurfsphase wird aus dem Pflichtenheft die innere Struktur der Software entwickelt (vgl. PAGEL, SIX 1994). Hierbei werden die einzelnen Komponenten der Software definiert und daraus die Softwarearchitektur bestimmt. Der Schwerpunkt der Entwurfsphase ist die Spezifikation des Funktionsumfangs der einzelnen Software-Komponenten sowie ihr Zusammenwirken untereinander. Das Ergebnis dieser Phase bezeichnet man als Softwarespezifikation; sie ist entscheidend für die spätere Qualität der Software.

Aus software-ergonomischer Sicht steht in dieser Phase die Dialoggestaltung im Vordergrund. Dabei wird die Architektur der Verzweigungen und Abhängigkeiten zwischen den einzelnen Komponenten der Software festgelegt. Die Navigation durch die einzelnen Komponenten muss logisch nachvollziehbar und übersichtlich gestaltet werden.

Beispiel hierzu kann eine Software zur Überwachung einer Maschinengruppe dienen: In einer Darstellung ist die gesamte Maschinengruppe dargestellt. Wenn eine Störung eintritt, muss der Überwachende aus dieser Übersicht in die Darstellung der Störungsanalyse sowie die Darstellung der Störungsbehebung wechseln. Nach der Behebung der Störung wechselt der Überwachende

wieder in die Übersicht. Die hier dargestellten Verzweigungen hängen unmittelbar mit der in Phase 1 erfolgten Analyse der Arbeitsaufgabe zusammen. Auch hier gilt der Grundsatz, dass man die Struktur zwar nachträglich ändern kann, aber dies häufig zu aufwendigen Änderungen im Programm führt.

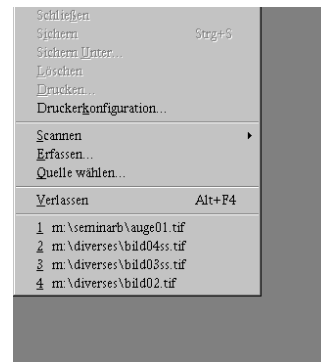
Implementierung

In dieser Phase werden die Komponenten des Entwurfs in einer Programmiersprache implementiert (vgl. PAGEL, SIX 1994). Hier erfolgt die eigentliche Ausgestaltung der Software. Aus diesem Grunde hat diese Phase auch eine zentrale Bedeutung für die Softwareergonomie.

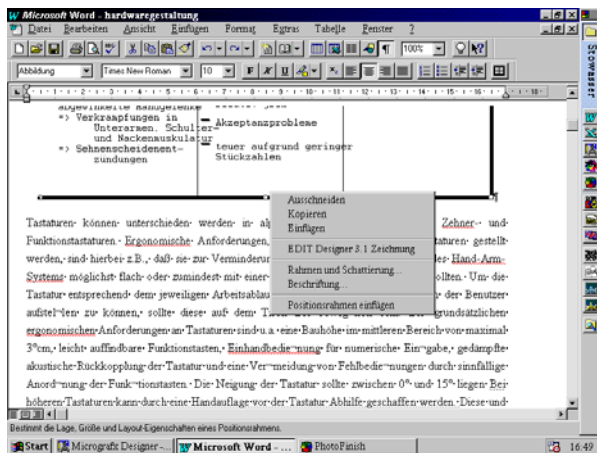
Aus softwareergonomischer Sicht steht die Dialoggestaltung im Mittelpunkt. Entsprechend der bereits im Pflichtenheft getroffenen Entscheidung wird die entsprechende Dialogtechnik eingesetzt. Grundsätzlich unterscheidet man zwischen der natürlichensprachlichen Dialogtechnik, den Kommando- und Programmiersprachen, der Formulartechnik, der direkten Manipulation sowie der Menüauswahltechnik (vgl. ZÜLCH, STOWASSER, FISCHER 1998). Nutzt man die Menüauswahltechnik, so erfolgt im weiteren Entwicklungsverlauf die Wahl der Menüart. Man hat hierbei die Auswahl zwischen Listen-, Pull-Down-, Pop-up- und Kaskaden-Menü (vgl. Bild 2).



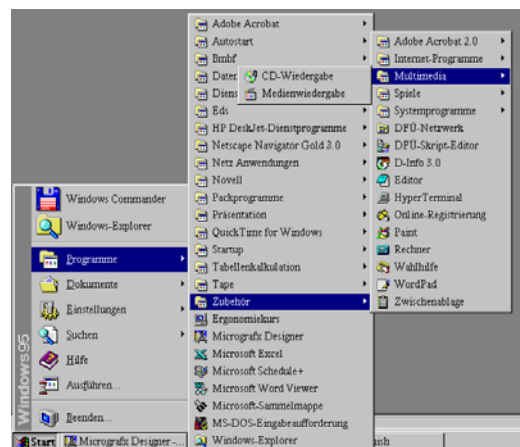
Listen-Menü



Pull-down-Menü



Pop-up-Menü



Kaskaden-Menü

Bild 2: Verschiedene Menüarten

Bei der Gestaltung der Menüs gilt es zu beachten, dass die Benennung des Menütitels kurz und prägnant sein soll. Die Benennung der Items muss verständlich und knapp sowie klar voneinander unterscheidbar erfolgen. Die Anordnung der Items kann entweder alphabetisch, nach Häufigkeit und Wichtigkeit, nach Funktionen gruppiert oder nach semantischer Ähnlichkeit gegliedert sein. Das Layout der Menüs sollte einheitlich gestaltet werden. Die Items werden linksbündig angeordnet. Wird eine Informationsart mehrfach aufgeführt, so gilt der Grundsatz, dass diese Informationsart immer an der gleichen Position erscheinen sollte.

Zur Dialoggestaltung gehört weiterhin die Handhabung von Bedienungs- und Software-Fehlern und die Konzipierung des Hilfesystems. Fehlermeldungen sind grundsätzlich so zu gestalten, dass sie auch für den ungeübten Benutzer verständlich erscheinen und ihm eine Möglichkeit zur Fehlerbehebung bieten. Hinweise zur Fehlerbehebung können entweder durch einen Verweis auf das Handbuch oder durch Anzeige entsprechender Informationen auf dem Bildschirm erfolgen (vgl. FÖRSTER, VOGEL 1994).

Bei Hilfesystemen unterscheidet man zwischen aktiven und passiven Hilfesystemen (vgl. GUNZENHÄUSER, KNOPIK 1986; BRODBECK, RUPIETTA 1994). Während aktive Hilfen in Form von Fehlermeldungen, Hinweisen oder Meldungen automatisch erfolgen, müssen passive Hilfen erst durch den Benutzer aufgerufen werden. Die Navigation in dem Hilfesystem soll sowohl bei aktiven als auch bei passiven Hilfesystemen durch gezieltes Ansteuern von Informationen möglich sein. Die Suchfunktion muss dabei berücksichtigen, dass der Benutzer nicht genau weiß, was er sucht. Das Hilfesystem muss dem Benutzer ein gefahrloses Testen von Hypothesen und die anschließende Übernahme der Arbeitsergebnisse ermöglichen. Bei der inhaltlichen Gestaltung der Hilfe ist die aktuelle Arbeitssituation des Benutzers zu berücksichtigen. Für den inhaltlichen Aufbau eines Hilfesystems kann man entweder eine funktionsorientierte oder aufgabenorientierte Darstellungsweise wählen. Die funktionsorientierte Darstellung beschreibt die Funktionen des Programms, während sich die aufgabenorientierte Darstellung an den Aufgaben orientiert, die der Benutzer mit dem Programm bearbeiten kann.

Im Anschluss an die Dialoggestaltung erfolgt die Informationsanordnung. Die wichtigsten Grundsätze der Informationsanordnung lauten (vgl. ZÜLCH, KELLER, FISCHER 1998):

- Der inhaltliche und formale Aufbau der Masken und Dialoge sollte einheitlich sein.
- Nur aufgabenbezogene Informationen sollten präsentiert werden.
- Gleichartige Datengruppen sollten mittels Formatfestlegung bzw. Masken dargestellt werden.
- Wichtige Daten sind mit Überschreibungsschutz zu versehen.

Wie im Teil 4 dieser Artikelserie (ZÜLCH, KELLER, FISCHER 1998) erläutert, sollte die Darstellung auf dem Bildschirm in die vier Klassen Informationsteil, Steuerungsteil, Verarbeitungsteil und Meldungsteil gegliedert werden. Außerdem sollten die Informationen in Informationsblöcken gruppiert sein.

Verwendet man Tabellen, so ist zu beachten, dass es eine klare Unterscheidung zwischen Kopfzeile und Inhalt der Tabelle geben muss. Texte sind in Tabellen grundsätzlich linksbündig und Zahlen rechtsbündig anzuordnen. Zahlen sollten in der Regel eine einheitliche Anzahl von Nachkommastellen aufweisen. Grundsätzlich werden Zahlen immer am Komma ausgerichtet.

Werden Diagramme eingesetzt, so sollte man sinnvollerweise für Strukturen Kreisdiagramme, für Rangfolgen Balkendiagramme, für Zeitreihen und Häufigkeiten Säulen- oder Kurvendiagramme sowie für Korrelationen Balkendiagramme oder Punktwolken verwenden.

Die Informationscodierung richtet sich danach, was dargestellt werden soll. Wie bereits in Teil 3 der Artikelserie (ZÜLCH, FISCHER, KELLER 1998) erläutert wurde, sind entsprechend der darzustellenden Informationsinhalte günstige Codierungsarten auszuwählen. Für die Präsentation von **Anweisungen und Aussagen** können sowohl alphanumerische Zeichen, abstrakte und bildhafte Symbole sowie Farbe, Blinken und Sprache als auch akustische Signale verwendet werden. Geht es um die Darstellung von **Objekten**, so nutzt man Piktogramme, abstrakte Bildzeichen oder alphanumerische Bezeichnungen. Bei **quantitativen Angaben** kann man zwischen Zahlen, Diagrammen und der Orientierung von Zeigern wählen. **Qualitativen Angaben** lassen sich hingegen durch Bildzeichen, alphanumerische Zeichen, Farbe, Helligkeit und Schraffur sowie differierender Größe darstellen. Zur Darstellung von **Hervorhebungen** werden Gruppierung, Einrückung oder Unterstreichung genutzt. Weitere Möglichkeiten bietet die Wahl unterschiedlicher Schriftarten und -größen sowie die Sperrung von Wörtern. Darüber hinaus kann man mittels Farbwahl, Variation der Helligkeit, Blinken, Einsatz von Kontrast sowie zusätzlichen Markierungszeichen Hervorhebungen verdeutlichen. Unterschiedliche **Zustände** werden durch Farbwahl, Variation der Bildzeichen, Schraffur, Punktierung oder Linienart dargestellt. Andere Möglichkeiten bietet die Darstellung durch alphanumerische Zeichen, die Variation der Helligkeit, des Kontrastes, der Schriftart oder -größe sowie der Einsatz blinkender Zeichen. Die Darstellung von **Veränderungen** kann durch Blinken oder Töne in Graphiken auch durch Anzeige des Verlaufs erfolgen.

Zum Einsatz der verschiedenen Codierungsarten gibt es jedoch einige Hinweise zu beachten. Hierzu gehört beispielsweise die Abstufung der Signalelemente einzelner Codierungsarten (vgl. Tab. 1). Hierbei ist zu beachten, daß die Unterscheidung ohne direkten Vergleich aneinandergrenzender Elemente erfolgt.

<i>Codierungsart</i>	<i>Modalität</i>	<i>Anzahl unterscheidbarer Stufen</i>	<i>Anzahl empfohlener Stufen</i>
Farbe	Helligkeit	4 – 6	2 – 3
	Farbton	9 – 11	6 – 8
	Sättigung	3 – 4	-
	Farbkontrast	20 – 25	aufgabenabhängig
	photometrischer Kontrast	4 – 6	bis 2
Gestalt	alphanumerische Zeichen	fast unbegrenzt	90 – 100
	abstrakte Bildzeichen	fast unbegrenzt	bis 20, in Sonderfällen mehr
	Piktogramme	fast unbegrenzt	bis 20, bei bekannten Piktogrammen mehr
	Größe	4 – 5	bis 3
	Orientierung	25 – 30	bis 8 (bis 24 mit Bezugssystem)
	Konfiguration	fast unbegrenzt	bis 8 gleichzeitig
Ort	absolute Lage	9 – 11	bis 9
	relative Lage	16 – 20	bis 4
Zeit	Frequenz	4 – 6	1 – 3
	Tastverhältnis	4 – 6	1 – 3

Tab. 1: Unterscheidbare Stufen bei verschiedenen Codierungsarten
(Quelle: BENZ, GROB, HAUBNER 1983)

Grundsätzlich darf man **Farbe** nur als zusätzliche Codierungsart und nur sparsam verwenden. Hierbei sollte auf die Wahl von geeigneten Farbkombinationen unter Berücksichtigung von gewohnten Bedeutungen (zum Beispiel gelb = Warnung) geachtet werden. Gesättigte Rot- oder Blau-Töne sollten nicht sparsam eingesetzt werden, da das Sehen dieser Farben eine erhöhte Beanspruchung für das menschliche Auge darstellt. Wird eine zeitliche Veränderung der Farbe eingesetzt, so sind durch Verwendung des Tastverhältnisses, d.h. dem Verhältnis von Impulsdauer zur Pausendauer, die Zeitanteile der Impuls- und Pausendauer sinnvoll festzulegen. Bei der Änderungsgeschwindigkeit kann zwischen einer kontinuierlichen oder sprunghaften Veränderung über mehrere Zwischenschritte gewählt werden. Die zeitliche Veränderung der Farbe sollte nur äußerst sparsam und nur zur Erregung von Aufmerksamkeit und bei Gefahr verwendet werden.

Zeichen sollten die Mindestzeichengröße von 3,2 mm bei einem Sehabstand von 50 und 55 cm nicht unterschreiten. Es muss eine eindeutige Unterscheidungsmöglichkeit zwischen den verschiedenen Zeichen geben. Grundsätzlich ist für Bildschirmdarstellungen eine Proportionalschrift zu verwenden.

Zur Unterscheidung unterschiedlicher **Linienzüge** sollten nicht mehr als 4 verschiedene Linien und 3 verschiedene Linienarten verwendet werden. Die Linienbreite sollte logarithmisch abgestuft sein, und es sollten maximal 3 unterschiedliche Stufen eingesetzt werden. Bei der Füllung von Flächen sollte man maximal 3 unterschiedliche Schraffuren verwenden.

Beim Einsatz von **Piktogrammen** sollten möglichst bereits bekannte Darstellungen verwendet werden. Für die zeitliche Veränderung der Piktogramme (Blinken) gelten die gleichen Maßgaben wie für die zeitliche Veränderung von Farben.

Wird die **räumliche Lage** zur Codierung genutzt, so unterscheidet man grundsätzlich zwischen der absoluten und der relativen Lage. Unter der absoluten Lage versteht man den Ort der Darstellung eines Zeichens auf der Bildschirmmaske. Im Gegensatz dazu gilt als relative Lage die Stelle eines Zeichens relativ zu einer anderen Information (z.B. Index einer Zahl). Zeitliche Veränderungen des Ortes können durch translatorische oder rotatorische Bewegung erreicht werden. Hierbei kann sich die Änderungsgeschwindigkeit entweder kontinuierlich oder sprunghaft über die Zeit ändern.

Bei Verwendung einer **akustische Codierung** sind als Parameter die Frequenz, der Schalldruck, die Schallart (Töne, Geräusche oder Sprache) sowie die Einwirkungszeit entsprechend dem jeweiligen Anwendungsfall sowie dem Umfeld zu wählen.

Bei der Verwendung von **Abkürzungen** ist auf die Verständlichkeit zu achten. Man sollte, wann immer möglich, gängige Abkürzungen verwenden. Eine Möglichkeit zum Abkürzen ist das Weglassen der Vokale. Grundsätzlich sollten aussprechbare Buchstabenfolgen verwendet werden.

Durch Nutzung des vorderen Teils eines Wortes kann ebenfalls abgekürzt werden. Eine weitere Möglichkeit bietet die Verwendung der Anfangsbuchstaben bei zusammengesetzten Wörtern. Abkürzungen müssen durchgängig verwendet werden.

Test

Nach der Implementierung und den dabei erfolgten Tests der einzelnen Komponenten erfolgt in dieser Phase die Zusammenführung der Komponenten sowie die Durchführung von Integrationstests. Dabei entstehen durch sukzessives Hinzufügen einzelner Komponenten wachsende Programmteile, die ausgiebig getestet werden (vgl. PAGEL SIX 1994). Den Abschluss dieser Phase bildet der Systemtest, der sicherstellt, dass das Softwaresystem die Produktanforderungen erfüllt.

Aus Sicht der Softwareergonomie ist die Evaluation von Software wie sie in Teil 6 der Artikelserie (ZÜLCH, FISCHER; STOWASSER 1998) beschrieben wurde, Bestandteil dieser Testphase. Durch eine Evaluation soll beispielsweise die Benutzungsfreundlichkeit beurteilt werden. Für den Entwickler der Software ist es interessant, ob die erstellte Software die geforderten Anforderungen unter funktionalen und benutzerorientierten Aspekten erfüllt. Mit Hilfe einer Evaluation kann bereits während des Entwicklungsprozesses eine diesbezügliche Qualitätsprüfung durchgeführt werden.

Einsatz und Wartung

Nach dem Abnahmetest beginnt der Praxiseinsatz der Software und umfaßt den Zeitraum von der Erstinstallation bis zum Einsatzende der Software (vgl. PAGEL, SIX 1994). Mit dem Eintritt in diese Phase ist die Software-Entwicklung abgeschlossen. Während der Installation der Software sollten Maßnahmen aufgegriffen werden, um die Software an die von den verschiedenen Benutzern gewünschte, spezifische Anwendung in der Zielumgebung anzupassen. Die Installationsprozess sollte durch ein Installationshandbuch unterstützt werden, welches den Vorgang einer vollständigen und korrekten Einrichtung der Software in allgemein verständlicher Sprache beschreibt. Nach erfolgreicher Installation der Software muß dem Benutzer die Möglichkeit gegeben werden, sich in das System einzuarbeiten. Dazu dienen betriebsinterne bzw. -externe Schulungen sowie die Verwendung einer Benutzerdokumentation. Die Erstellung einer Benutzerdokumentation sollte den Anspruch erfüllen, die Software vollständig zu beschreiben. Hierbei sind folgende Kriterien zu erfüllen (vgl. u.a. FÖRSTER, VOGEL 1994):

- **Knappheit**

Die Anzahl der Seiten, sowohl im Gesamtumfang als auch im Seitenumfang der einzelnen Kapitel, sollte begrenzt sein. Jeder Abschnitt in der Benutzerdokumentation sollte nicht mehr als 3 Seiten umfassen.

- **Verständlichkeit**

Die Benutzerdokumentation ist in einer allgemeinverständlichen Sprache zu verfassen.

- **Aufgabenbezogenheit**

Die Benutzerdokumentation sollte nicht in abstrakter Weise in die Software einführen, sondern dem Benutzer aufgabenbezogen in die Bearbeitung einführen.

- **Modularität:**

Jedes Kapitel der Benutzerdokumentation sollte in sich abgeschlossen und verständlich sein. Querverweise sollten eine Ausnahme sein.

Im Rahmen der Software-Wartung werden auftretende Fehler korrigiert, Anpassungen an die Systemumgebung vorgenommen sowie Änderungen und Weiterentwicklungen der Software angestoßen. Das Qualitätskriterium **Wartbarkeit** der Software faßt alle Eigenschaften zusammen, die das Auffinden und Beheben von Fehlern und die Verbesserung der Software unterstützen. Dieses Kriterium spielt für den Benutzer keine Rolle. Dahingegen ist die Wartbarkeit für den Betreuer und Anwender gleichermaßen wichtig. So hängt beispielsweise der Aufwand des Betreuers für Fehlersuche und -korrektur sowie die Anpassung an neue Systemsoftware fast ausschließlich von der Wartbarkeit ab (vgl. PAGEL, SIX 1994).

Nachdem in diesem Teil 8 der Artikelserie die bisher behandelten Einzelaspekte zusammengeführt wurden, erfolgt im folgenden Teil 9 ein Ausblick in die zukünftigen Trends der Softwaregestaltung. Ein Stichwort, das in diesem Zusammenhang immer wieder fällt, ist Multimedia. Aus diesem Grunde beschäftigt sich der nächste Artikel mit Multimedia-Anwendungen und ihren spezifischen softwareergonomischen Aspekten.

Literaturverzeichnis

BENZ, C.; GROB, R.; HAUBNER, P.:

Gestaltung von Bildschirm-Arbeitsplätzen.

Köln: TÜV Rheinland, 1981.

(Praxis der Ergonomie)

BRODBECK, Felix, C.; RUPIETTA, Walter:

Fehlermanagement und Hilfesysteme.

In: Einführung in die Software-Ergonomie.

Hrsg.: EBERLEH, Edmund; OBERQUELLE, Horst; OPPERMANN, Reinhard.

Berlin: de Gruyter, 1994.

FÖRSTER, Christian; VOGEL, Reinhard:

PC-Ergonomie und Ökologie.

München: Beck, 1994.

GUNZENHÄUSER, R.; KNOPIK, Th.:

Wissensbasierte Mensch-Computer-Schnittstellen in der Software-Ergonomie.

In: Software Ergonomie.

Hrsg.: BULLINGER, H.-J.; GUNZENHÄUSER, R. u.a...

Esslingen: Expert, 1986, S. 83-102.

PAGEL, Bernd-Uwe; SIX, Hans-Werner:

Software-Engineering. Band 1: Phasen der Softwareentwicklung.

Bonn u.a.: Addison-Wesley, 1994.

(Band 1: Phasen der Softwareentwicklung)

SIEMENS AG (Hrsg.):

Kommunikations-Ergonomie.

München: Eigenverlag, 1984.

ZÜLCH, Gert; FISCHER, Axel; KELLER, Volker:

Ergonomische Aspekte der Software-Gestaltung.

Teil 3: Codierung von Informationen.

In: Ergo-Med,

Heidelberg, 22(1998)1, S. 36-40.

ZÜLCH, Gert; FISCHER, Axel E.; STOWASSER, Sascha:

Ergonomische Aspekte der Software-Gestaltung.

Teil 6: Evaluation von Software.

In: Ergo-Med,

Heidelberg, 22(1998)4, S. 194-198.

ZÜLCH, Gert; KELLER, Volker; FISCHER, Axel E.:

Ergonomische Aspekte der Software-Gestaltung.

Teil 4: Anordnung von Informationen.

In: Ergo-Med,

Heidelberg, 22(1998)2, S. 105-109.

ZÜLCH, Gert; KELLER, Volker; STOWASSER, Sascha:

Ergonomische Aspekte der Software-Gestaltung.

Teil 1: Einführung in die Software-Ergonomie.

In: Ergo-Med,

Heidelberg, 21(1997)5, S. 144-147.

ZÜLCH, Gert; STOWASSER, Sascha; FISCHER, Axel:

Ergonomische Aspekte der Software-Gestaltung.

Teil 5: Gestaltung des Dialogs zwischen Benutzer und Rechner.

In: Ergo-Med,

Heidelberg, 22(1998)3, S. 154-159.

Anschrift der Verfasser

Dipl.-Ing. Volker Keller

Institut für Arbeitswissenschaft und Betriebsorganisation (ifab)

Universität Karlsruhe (TH)

Kaiserstraße 12

76128 Karlsruhe