

**215-450**

## DGUV Information 215-450



## Softwareergonomie

## **Impressum**

Herausgeber:  
Deutsche Gesetzliche  
Unfallversicherung e.V. (DGUV)

Glinkastraße 40  
10117 Berlin  
Tel.: 030 288763800  
Fax: 030 288763808  
E-Mail: [info@dguv.de](mailto:info@dguv.de)  
Internet: [www.dguv.de](http://www.dguv.de)

Sachgebiet „Büro“ des  
Fachbereichs „Verwaltung“ der DGUV

Titelbild: © contrastwerkstatt/Fotolia

Ausgabe: August 2016

DGUV Information 215-450  
zu beziehen bei Ihrem zuständigen Unfallversicherungsträger oder unter [www.dguv.de/publikationen](http://www.dguv.de/publikationen)

# Softwareergonomie

# Inhaltsverzeichnis

	Seite		Seite		
<b>1</b>	<b>Einführung</b> .....	<b>7</b>	<b>8.</b>	<b>Informationsdesign</b> .....	<b>30</b>
<b>2</b>	<b>Erfolgsfaktor Software</b> .....	<b>8</b>	8.1	Gestaltungsgrundsätze .....	30
2.1	Software im Arbeitsprozess .....	8	8.2	Anordnung von Elementen und Informationen .....	30
2.2	Wirtschaftliche Aspekte ergonomischer Software .....	8	8.2.1	Allgemeiner Maskenaufbau .....	30
2.3	Produktivität von Software .....	9	8.2.2	Aufgabenangemessene Anordnung von Informationen .....	32
2.4	Nutzungsqualität von Software .....	9	8.3	Die Gestaltgesetze .....	33
<b>3</b>	<b>Softwareergonomie und die Wahrnehmung, Verarbeitung und Umsetzung von Informationen durch den Menschen</b> .....	<b>10</b>	8.3.1	Gesetz der Nähe .....	33
3.1	Schnittstellen zwischen der Software und den Prozessen menschlicher Informationsverarbeitung .....	10	8.3.2	Gesetz der Ähnlichkeit .....	33
3.2	Wahrnehmung von Informationen .....	11	8.3.3	Gesetz der Geschlossenheit .....	34
3.2.1	Das visuelle Sinnessystem .....	12	8.4	Elemente von grafischen Benutzungsoberflächen (Steuerelemente) .....	34
3.2.2	Das auditive Sinnessystem .....	14	8.4.1	Auswahl der richtigen Elemente .....	34
3.2.3	Das haptische Sinnessystem .....	15	8.4.2	Einstellungen und Verhalten der Elemente .....	34
3.2.4	Ausgewählte Prozesse der Wahrnehmungsorganisation .....	15	8.5	Textgestaltung .....	35
3.3	Verarbeitung von Informationen und Wissensgenerierung .....	17	8.5.1	Verständliche Texte .....	35
3.3.1	Die begrenzte Aufnahmekapazität des Gedächtnisses .....	17	8.5.2	Positive und negative Darstellung von Text (Polarität) .....	35
3.3.2	Der Motor der Wissensgenerierung .....	17	8.5.3	Hervorheben und Darstellen von Text .....	36
3.4	Umsetzung von Handlungen .....	18	8.5.4	Texte strukturieren .....	36
<b>4.</b>	<b>Psychologische Ansätze in der Softwareergonomie</b> .....	<b>20</b>	8.6	Schrift .....	37
4.1	Psychische Belastung und Beanspruchung berücksichtigen .....	20	8.6.1	Schriftart .....	37
4.2	Software und Kognition .....	23	8.6.2	Schriftgröße .....	37
4.3	Software und Emotion .....	23	8.6.3	Mehrdeutige Zeichen .....	37
4.4	Arbeitsgestaltung im Prozess der Softwareimplementierung .....	24	8.6.4	Glättung von Schrift .....	37
<b>5.</b>	<b>Software und Bildschirm</b> .....	<b>25</b>	8.7	Farben .....	38
<b>6.</b>	<b>Softwaretypen</b> .....	<b>27</b>	8.7.1	Farbkodierung und Farbkombinationen .....	38
<b>7.</b>	<b>Rechtliche Aspekte</b> .....	<b>28</b>	8.7.2	Kritische Farbkombinationen .....	38
			8.7.3	Konventionen .....	38
			8.7.4	Kontrast .....	39
			8.7.5	Farben für Symbole .....	39
			<b>9.</b>	<b>Interaktionsdesign</b> .....	<b>40</b>
			9.1	Grundsätze der Dialoggestaltung .....	40
			9.1.1	Aufgabenangemessenheit .....	40
			9.1.2	Selbstbeschreibungsfähigkeit .....	40
			9.1.3	Erwartungskonformität .....	41
			9.1.4	Lernförderlichkeit .....	42
			9.1.5	Steuerbarkeit .....	43
			9.1.6	Fehlertoleranz .....	43
			9.1.7	Individualisierbarkeit .....	43
			9.1.8	Terminologie .....	44
			9.1.9	Meldungen .....	44
			9.1.10	Dialogverhalten und Kontrolle .....	44
			9.1.11	Häufigkeiten .....	46
			9.1.12	Fehler erkennen und vermeiden .....	47
			9.1.13	Hilfesysteme .....	47
			9.1.14	Anpassbarkeit an spezielle Bedürfnisse der Benutzerinnen und Benutzer .....	47
			9.1.15	Ein- und Ausgabemedien .....	47

	Seite		Seite
<b>10. Barrierefreie Gestaltung von Software</b> .....	<b>48</b>	<b>Anhang 1</b>	
10.1 Gesetzliche Grundlagen .....	48	Bildschirmparameter für LCD-Bildschirme .....	79
10.3 Behinderungen .....	49	<b>Anhang 2</b>	
10.3.1 Behinderungen .....	49	Vereinbarung zum Umgang mit Software .....	80
10.2.2 Technische Hilfen .....	52	<b>Anhang 3</b>	
10.3 Gestaltungsrichtlinien .....	53	Arbeitshilfen: IT-Fragenbogen aus C2Web .....	82
10.3.1 Technische Umsetzung der Gestaltungsrichtlinien .....	54	<b>Anhang 4</b>	
10.4 Qualitätssicherung .....	57	Beispielhafte Methode zur formativen Evaluation .....	85
10.4.1 Technische Formate und Plattformen .....	57	<b>Anhang 5</b>	
10.4.2 Qualifikation .....	57	Analyse der Aufgaben der Benutzerinnen und Benutzer sowie der Anforderungen und Funktionen der Software .....	88
10.4.3 Entwicklungsprozess .....	57		
<b>11. Nutzungsqualität und Überprüfung</b> .....	<b>59</b>		
11.1 Nutzungsqualität im Einsatzzyklus .....	59		
11.2 Das Arbeitssystem für die Aufgaben .....	60		
11.3 Nutzungskontexte der Software spezifizieren .....	62		
11.4 Nutzungsanforderungen spezifizieren .....	64		
11.5 Gestaltungslösungen zur Umsetzung von Nutzungsanforderungen entwickeln .....	64		
11.6 Gestaltungslösungen evaluieren .....	65		
11.7 Methoden zur Überprüfung und Bewertung von Gestaltungslösungen .....	66		
11.8 Prüfen und Bewerten von Software zu bestimmten Anlässen .....	67		
<b>12 Softwarekauf und Pflichtenheft</b> .....	<b>68</b>		
12.1 Softwarekauf .....	68		
12.1.1 Beteiligte beim Softwarekauf .....	68		
12.2 Vorgehen beim Softwarekauf .....	69		
12.2.1 Analyse – Welche Anforderungen soll die neue Software erfüllen und über welche Funktionen muss sie verfügen? .....	70		
12.2.2 Marktrecherche – Wo finden wir die geeignete Software .....	70		
12.2.3 Detaillierte Prüfung – Nach welchen Kriterien können wir Software prüfen? .....	70		
12.3. Pflichtenheft .....	71		
12.3.1 Struktur und Inhalte eines Pflichtenheftes .....	73		
12.3.2 Prüfung und Bewertung neu entwickelter Software .....	73		
<b>13 Software-Entwicklungsprozesse</b> .....	<b>75</b>		
13.1 Das Wasserfallmodell .....	75		
13.2 Extreme Programming (XP) .....	76		
13.3 Das V-Modell™ (Entwicklungsstandard) .....	77		
13.4 Das iterative Modell .....	78		



# 1 Einführung



Software spielt heute bei fast allen Arbeiten eine wichtige Rolle. Sie beeinflusst die Arbeitsleistung, die Leistungsbereitschaft, die Qualität der Wissensgenerierung und die Produktivität. Eine hohe Qualität der eingesetzten Software entscheidet mit über die Güte der Arbeitsergebnisse sowie über Zufriedenheit und Motivation der Beschäftigten. Ausfälle, Fehler, Korrekturen oder Abstürze der Software blockieren immer wieder Arbeitsabläufe und führen zu unfreiwilligen Pausen. Insofern ist Software zu einem wesentlichen Bestandteil der Wertschöpfung im Unternehmen geworden.

- Was aber bedeutet Nutzungsqualität der Software?
- Wie beeinflusst Software unsere Arbeit?
- Wie funktioniert die Informationsaufnahme und -verarbeitung beim Menschen?
- Wie funktioniert die Hardware im Zusammenspiel mit der Software?
- Wie gestaltet sich der rechtliche Rahmen?
- Was ist barrierefreie Software?
- Wie ist sinnvollerweise bei der Auswahl von Software vorzugehen?
- Wie kann Software beurteilt werden?
- Wie sollte Software entwickelt werden?

Antworten auf diese Fragen gibt die vorliegende DGUV Information 215-450 „Softwareergonomie“. Als Grundlage dient der gesetzliche Rahmen aus Arbeitsstättenverordnung und Bildschirmarbeitsverordnung in Verbindung mit aktuellen Regeln und Normen. Die DGUV Information 215-450 „Softwareergonomie“ ist somit das Referenzdokument für das Themenfeld „Softwareergonomie“ der gesetzlichen Unfallversicherung und bietet praktische Hilfen an. Sie richtet sich sowohl an Arbeitgeberinnen und Arbeitgeber sowie an Führungskräfte als auch an Entwicklerinnen und Entwickler,

Einkäuferinnen und Einkäufer und Benutzerinnen und Benutzer von Software. Ebenso sind Beraterinnen bzw. Berater in Fragen der Arbeitssicherheit und des Gesundheitsschutzes sowie alle fachlich interessierten Kreise in den Unternehmen angesprochen, bei „...Entwicklung, Auswahl, Erwerb und Änderung von Software sowie bei der Gestaltung der Tätigkeit am Bildschirm...“<sup>1)</sup> die Inhalte dieser Informationsschrift zu berücksichtigen.

<sup>1)</sup> Bildschirmarbeitsverordnung (BildscharbV) Anhang

# 2 Erfolgsfaktor Software

In diesem Kapitel geht es um den Stellenwert von Software in Unternehmen und darum, wie ergonomische Software den Arbeitsprozess beeinflusst und die Gebrauchstauglichkeit und die Nutzungsqualität sich auf die Produktivität und das wirtschaftliche Ergebnis auswirken.

## 2.1 Software im Arbeitsprozess

Der Einsatz von Informations- und Kommunikationstechnologien in den Unternehmen wächst fortlaufend. Auch die Inanspruchnahme des Internets ist mehr und mehr Bestandteil der IT-gestützten Arbeit (z. B. Warenhandel, elektronische Bezahlvorgänge und Informationsaustausch).

Von den rund 42 Millionen Erwerbstätigen in Deutschland arbeiten mittlerweile ca. 27 Millionen mindestens einmal wöchentlich an Bildschirmarbeitsplätzen<sup>2)</sup>. Dies bedeutet, dass für immer mehr Menschen Arbeit zur Computerarbeit und damit zur Arbeit mit Software wird. Die effektive Nutzung moderner Informations- und Kommunikationstechnologien ist für Unternehmen zu einem wichtigen strategischen Faktor geworden. Durch den Einsatz von Informations- und Kommunikationstechnologien lässt sich die Produktivität von Unternehmen steigern und so die Wettbewerbsfähigkeit verbessern. Informationstechnologien erleichtern den Zugang zu Daten, beschleunigen Geschäftsprozesse und eröffnen den Unternehmen neue, globale Beschaffungs- und Absatzwege.

Störungen durch fehlerhafte oder nicht ergonomisch gestaltete Software belasten die Beschäftigten, beeinträchtigen den Arbeitsablauf und verursachen Kosten. Die umfassende Berücksichtigung von softwareergonomischen Prinzipien hilft, diese Probleme bei der Erstellung, Einführung und Nutzung von Anwendungssoftware zu vermeiden.

## 2.2 Wirtschaftliche Aspekte ergonomischer Software

Aus Sicht der Unternehmensleitung ist klar: Jede Beschaffung muss mehr einbringen, als sie kostet. Wer heute in eine neue Maschine oder eine Software investiert, der rechnet exakt aus, wann sich der Kauf bezahlt gemacht hat. Oder anders ausgedrückt: wie sich die Beschaffung insgesamt als ökonomisch vorteilhaft erweist. Die ergonomische Gestaltung von Software beeinflusst direkt und indirekt den erfolgreichen Ablauf der Arbeitsprozesse – und damit auch das wirtschaftliche Ergebnis.

Die Wirtschaftlichkeit ergonomischer Software stellt sich aus Perspektive der Betriebe u. a. folgendermaßen dar:

- Produktivere und qualitativ höherwertige Aufgabenerfüllung: Beschäftigte müssen ihre Arbeit nicht damit verbringen, Fehler im Umgang mit der Software zu bewältigen. Dadurch entstehen keine kostenerzeugenden Ausfallzeiten.
- Höhere Motivation der Beschäftigten: Mit ergonomischer Software, die an menschliche Gewohnheiten und die Arbeitsaufgaben angepasst ist, bearbeiten Benutzer die Aufgaben in ihrem Unternehmen zuverlässiger, schneller und zufriedener.
- Geringere Schulungs- und Betreuungskosten: Ergonomische Software führt zu geringeren Trainingskosten z. B. durch intuitive und nutzerangepasste Bedienung.

- Vermeidung gesundheitlicher Beeinträchtigungen der Beschäftigten und damit verbundener Kosten: Höhere Zufriedenheit, geringere psychische Belastung und höhere Nutzungsqualität können gesundheitliche Beeinträchtigungen vermeiden. Dadurch verringern sich ausfallbedingte Zusatzkosten.

Weitere Aspekte, die bei der Beurteilung der Wirtschaftlichkeit zu berücksichtigen sind:

- Kosten für Überarbeitung und Weiterentwicklung der Software,
- Kosten für die System-Administration,
- Kosten durch erhöhten Bedarf an technischem Support,
- Kosten durch erhöhten Bedarf an Training,
- Kosten durch Systemabstürze,
- Kosten durch lange, Systemantwortzeiten
- Kosten durch unnötigen Aufwand bei der Dateneingabe,
- Kosten durch notwendige Fehlerkorrekturen,
- Kosten durch programmbedingte Fehlbedienungen,
- Kosten durch Unvereinbarkeit von Aufgabe, Aufgabenstruktur und Software.

Aus der Sicht der softwareherstellenden Unternehmen ist die softwareergonomische Güte ein entscheidendes Qualitätsmerkmal. Diese sogenannte Gebrauchstauglichkeit trägt zur Attraktivität einer Software und zur Zufriedenheit der Kundschaft bei und verbessert damit die Marktchancen der eigenen Produkte.

Gebrauchstauglichkeit betrieblicher Anwendungssoftware ist daher ein zunehmend wichtiger Wettbewerbsfaktor für Softwarehersteller und Anwender.

<sup>2)</sup> Statistisches Bundesamt: Nutzung von Informations- und Kommunikationstechnologien in Unternehmen, Wiesbaden 2013

### 2.3 Produktivität von Software

Der Einsatz guter Anwendungssoftware hat in den vergangenen Jahren an Bedeutung gewonnen. Wesentliche Gründe hierfür sind die

- Erreichung betriebswirtschaftlicher Ziele,
- Steigerung von Produktivität, Qualität und Zufriedenheit der Kundschaft.

Gleichzeitig sind auch die Ansprüche in Bezug auf die Gebrauchstauglichkeit von Anwendungssoftware gestiegen. Erst eine ergonomisch optimierte Anwendung eröffnet den Zugang zur effizienten Nutzung der Software-Funktionalität. Die bloße Verfügbarkeit von Funktionen ist nicht mehr ausreichend.

Verbesserungspotentiale durch höhere Gebrauchstauglichkeit werden oft unterschätzt, was dazu führt, dass die Investitionsbereitschaft in gebrauchstaugliche Softwareprodukte geringer ist, als sie sein sollte.

Die Auswirkungen guter, ergonomischer Software auf die Wirtschaftlichkeit von Projekten lassen sich mithilfe von „Usability-Metriken“ (mathematische Funktion zur Bestimmung der Eigenschaft von Software) nachweisen. Idealerweise sollten dabei Metriken gewählt werden, die zum aktuellen Projekt passen und geeignet sind, Aussagen über dessen Wirtschaftlichkeit zu machen.

Eine Metrik-Methode, mit der eine Steigerung der Produktivität einfach nachgewiesen werden kann, ist Time2Task. Der Versuchsperson wird eine Testaufgabe gestellt. Entscheidend dabei ist nicht allein, ob eine Aufgabe erfolgreich beendet wird, sondern auch die hierfür verwendete Zeit.

Weitere Metriken:

- Erfolgsrate (Success Rate)  
Die Erfolgsrate bezeichnet die Prozentzahl der von einer Testperson erfolgreich beendeten Testaufgaben in einem Usability-Test. Eine Zeitvorgabe existiert nicht. Entscheidend ist allein, ob die Testperson in der Lage war, die ihr gestellte Aufgabe zu erfüllen.
- Benutzerzufriedenheit  
Die Benutzerzufriedenheit ist eine Usability-Metrik, die die subjektiven Eindrücke der Testperson während eines durchgeführten Usability-Tests erfasst.

### 2.4 Nutzungsqualität von Software

Die Nutzungsqualität einer Software bestimmt sich hauptsächlich durch ihre tatsächliche Gebrauchstauglichkeit. „Gebrauchstauglichkeit ist das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“ (DIN EN ISO 9241-11).

Der in der DIN EN ISO 9241 definierte Begriff der Gebrauchstauglichkeit (engl. Usability) erschließt sich über die Analyse der darin eingeflossenen Kernbegriffe:

#### Nutzungskontext

Der Nutzungskontext umfasst die Benutzerin bzw. den Benutzer, die Arbeitsaufgaben, die Arbeitsmittel (Hardware, Software und Materialien) sowie die physische und soziale Umgebung, in der das Produkt genutzt wird.

#### Effektivität

Unter Effektivität ist die Genauigkeit und Vollständigkeit gemeint, mit der Benutzerinnen und Benutzer ihr Ziel erreichen. Das heißt: Ist es für die Benutzerin oder den Benutzer überhaupt möglich, den gegebenen Arbeitsauftrag mit diesem Arbeitsmittel zu bearbeiten?

#### Effizienz

Unter Effizienz versteht man das Verhältnis von Effektivität zum Aufwand, mit dem Benutzerinnen und Benutzer ein bestimmtes Ziel erreichen. Zum Aufwand zählen beispielsweise psychische oder physische Belastung, Zeit, Material oder Kosten.

#### Zufriedenheit

Mit Zufriedenheit wird die Freiheit von Beeinträchtigung und positive Einstellung der Benutzerin oder des Benutzers bei der Arbeit mit der Software beschrieben.

# 3 Softwareergonomie und die Wahrnehmung, Verarbeitung und Umsetzung von Informationen durch den Menschen

Dieses Kapitel befasst sich mit den Schnittstellen zwischen Computer und Mensch, also den Stellen, an denen Daten zu Informationen werden und umgekehrt. Sie sind das Maß für eine ergonomische Gestaltung von Software, den Prozess der Wissensgenerierung und der Aufgabebearbeitung mit Software.

## 3.1 Schnittstellen zwischen der Software und den Prozessen menschlicher Informationsverarbeitung

Häufig unterstützt Software die Beschäftigten bei der Erledigung ihrer Aufgaben. Ist die Software nach ergonomischen Prinzipien gestaltet, dann ist sie auch an den Prozess der Informationsverarbeitung des Menschen angepasst. Die Schnittstellen

zur Software betreffen dabei die Wahrnehmung, die Verarbeitung von Informationen und Wissensgenerierung und schließlich die Handlungsumsetzung.

Abbildung 1 zeigt, wie der Mensch vom Computer Daten aufnehmen, diese Daten als Informationen verarbeiten und auch an den Computer weitergeben kann. Daten, die Software über die Bildschirmanzeige

Benutzerinnen und Benutzern zur Verfügung stellt, sollen sie bei der Aufgabebearbeitung unterstützen. Die Benutzerinnen und Benutzer müssen die Daten auf der Bildschirmanzeige entziffern können (Leserlichkeit) und ihren Inhalt verstehen können (Lesbarkeit), damit die Daten mit ihren Sinnen wahrnehmbar sind (Informationsaufnahme). Können Benutzerinnen und Benutzer die Daten der Anzeige

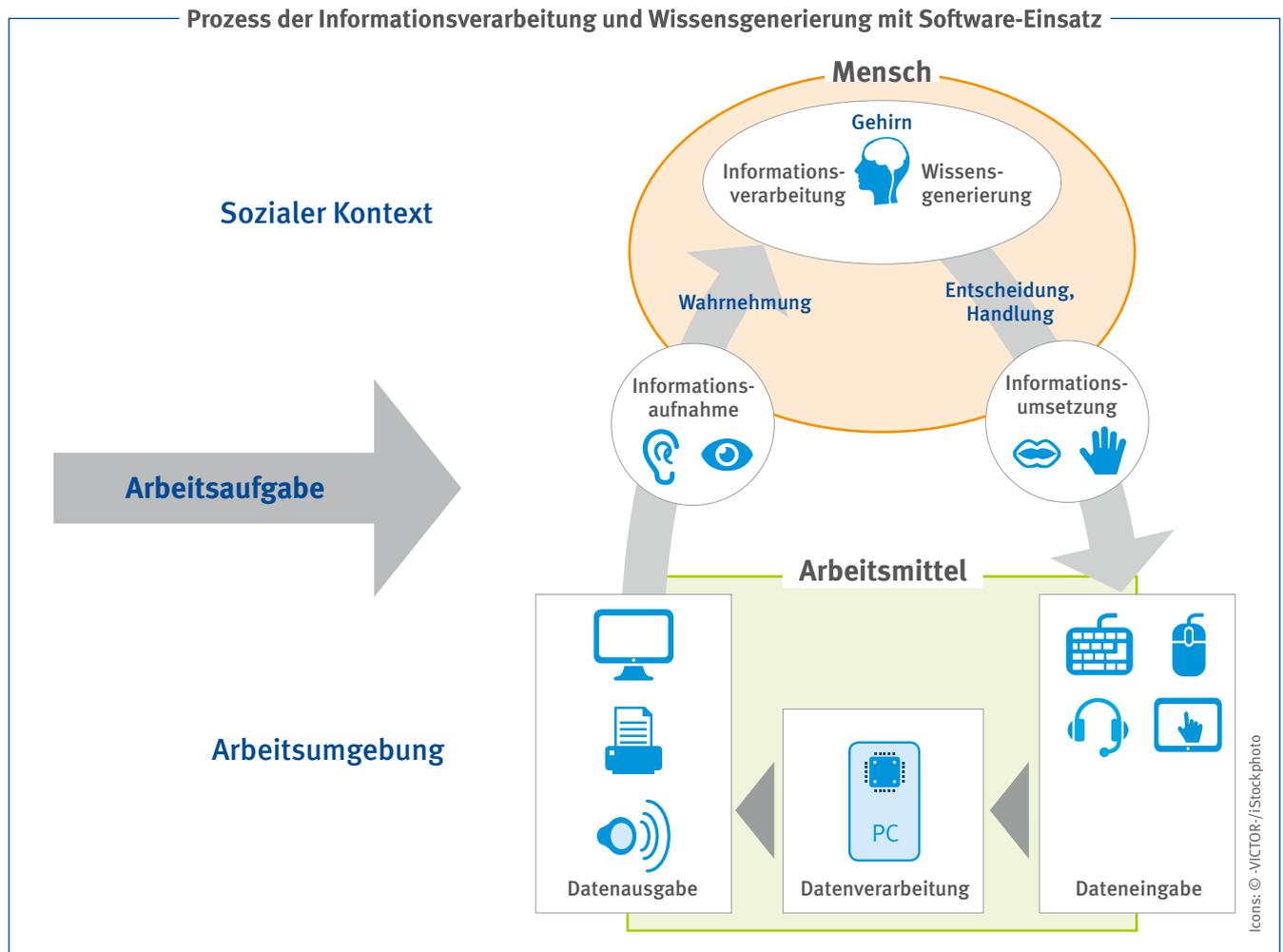


Abb. 1 Modell der Mensch-System-Interaktion in der Informationsverarbeitung

wahrnehmen, so sind sie in der Lage, diese als Informationen mit den Anforderungen aus der Arbeitsaufgabe zu verarbeiten (Informationsverarbeitung) und Wissen zu generieren. Mit diesem Wissen können sie Entscheidungen treffen, die sie in Handlungen umsetzen.

Ein Teil dieser Handlung bezieht sich auf die Nutzung der Software, indem Daten auf der Anzeige von Benutzerinnen und Benutzern mithilfe von Eingabemitteln verändert werden (Informationsumsetzung).

Die Gestaltung der Software spielt also zumindest an zwei Stellen im Prozess der Informationsverarbeitung eine wesentliche Rolle.

1. Für die Informationsaufnahme müssen die Daten so zugänglich und aufbereitet sein, dass Beschäftigte sie wahrnehmen – sehen, hören, fühlen und erkennen – können und dass sie im Umfang dem jeweiligen Teil der Aufgabebearbeitung angemessen sind.

So müssen z. B. Daten in ausreichender Zeichengröße dargestellt werden, damit Beschäftigte sie entziffern und lesen können.

Informationen sollten auf der Bildschirmanzeige strukturiert dargestellt werden, damit sich Benutzerinnen und Benutzer orientieren können. Weitere Anforderungen an eine ergonomische Gestaltung von Interaktionen mit und Informationen der Software sind z. B. in den Teilen 110 und 12 der DIN EN ISO 9241 beschrieben.

2. Für die Informationsumsetzung müssen die Daten so zugänglich und aufbereitet sein, dass Benutzerinnen und Benutzer Schritte zur Aufgabebearbeitung mithilfe der Software umsetzen können.

Deshalb sollten lediglich die Informationen in einer Struktur dargestellt

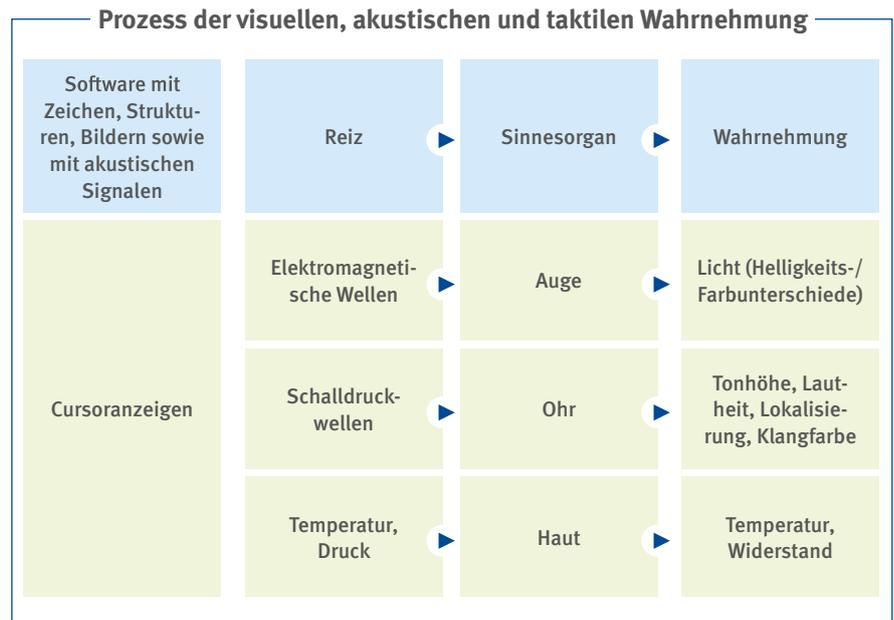


Abb. 2 Prozess der Visuellen, akustischen und taktilen Wahrnehmung

werden, die für die Bearbeitung aktuell wichtig sind. Für die Aufgabebearbeitung erforderliche Eingabemittel und Eingabefelder sollten direkt angesteuert werden können. Anforderungen an eine ergonomische Gestaltung der Eingabemittel für den Software-Einsatz sind in DIN EN ISO 9241-410 beschrieben.

Im Prozess der Informationsverarbeitung und Wissensgenerierung fällt Benutzerinnen und Benutzern die zentrale Rolle zu. Sie bearbeiten ihre Aufgaben, indem sie ihre Fähigkeiten, ihr Wissen, ihre Fertigkeiten und Erfahrungen einsetzen. Dafür steuern sie den in Abbildung 1 als Prozess beschriebenen Mensch-Software-Dialog, bei dem sie in einzelnen Aufgabenteilen, Arbeitsschritten und Teilen zur Entscheidungsfindung von der Software unterstützt werden.

Das Wissen über die Funktionsweise der Informationsverarbeitung des Menschen hilft bei der ergonomischen Gestaltung von Software. Dies wiederum hilft, die Aufgabebearbeitung zu erleichtern und Fehlbelastungen zu vermeiden. Eine Herausforderung für die ergonomische Gestaltung von Software liegt darin, alle Bedingungen, unter denen die Software genutzt wird, zu berücksichtigen. Optimale Bedingungen für den Prozess der Informationsverarbeitung können auch die Intensität psychischer Belastung senken (siehe Kapitel 4).

### 3.2 Wahrnehmung von Informationen

Damit Beschäftigte Informationen aufnehmen und umsetzen können, müssen deren Sinne diese erkennen können. Zwei Sinnessysteme sind für die Softwaregestaltung besonders wichtig:

- das visuelle System (die Augen),
- das auditive System (die Ohren).

Bei bestimmten Tätigkeiten ist auch das haptische System (die Haut) bedeutsam (siehe Kapitel 3.2.3).

Der Prozess der visuellen und der akustischen Informationswahrnehmung ist schematisch in der Abbildung 2 dargestellt. Die Sinnesorgane nehmen jeweils nur bestimmte Arten und Qualitäten von Reizen auf und leiten sie über Nervenfasern an das Gehirn weiter. Reize, die das Auge als visuelles System aufnimmt, sind Farben eines begrenzten Spektrums von Wellenlängen sowie Helligkeits- und Farbintensitäten. Mit ausreichendem Kontrast können sie im Gehirn als Zeichen, Strukturen oder Bilder wiedererkannt werden.

Das Ohr als auditives System nimmt Schalldrücke in einem begrenzten Frequenz- und Intensitätsbereich auf, die dann nach Weiterleitung des Reizes an das Gehirn als Ton, Geräusch, Laute oder Klangfolgen interpretiert werden.

### 3.2.1 Das visuelle Sinnessystem

Eine Informationsaufnahme mithilfe des visuellen Sinnessystems setzt grundsätzlich voraus, dass sich die Informationen im Gesichtsfeld des Menschen befinden. Eine zuverlässige Tiefenwahrnehmung aller Farben ist auf ca. 19° Winkelabweichung von der Sehachse in alle Richtungen begrenzt. Bei größeren Winkeln können Menschen noch Graustufen und Bewegungen erfassen, sofern nicht durch Kopfbewegungen ein größeres Blickfeld entsteht. Für die Farbgestaltung von Informationsdarstellungen sind sowohl Grenzwerte des menschlichen Sehraumes als auch seine Aufgaben zu berücksichtigen.

Über die Farbrezeptoren der Augen (Zapfen) kann der Mensch aus dem Spektrum der elektromagnetischen Strahlung Wellenlängen von ca. 380 nm bis 780 nm (nm = ein Millionstel Millimeter) als Farben von Blauviolett bis hin zu Purpurrot wahrnehmen (Abbildung 3). Wellenlängen außerhalb dieses Bereichs sind zur Informationsdarstellung nicht geeignet. Mithilfe von Hell-Dunkel-Rezeptoren (Stäbchen) auf der Netzhaut werden Helligkeitsabstufungen (Lichtintensitäten) wahrgenommen. Unterschiede zwischen Farben können Menschen nur bei ausreichender

Helligkeit sicher erkennen und wenn sie im Spektrum und/oder in ihrer Intensität ausreichend weit auseinander liegen. Das ist z. B. bei unterschiedlichen Grundfarben oder bei Schwarz-Weiß-Darstellungen der Fall. Im Spektrum zu nahe beieinander liegen z. B. schwarze Schrift auf grauem Untergrund oder gelbe Linien auf weißem Hintergrund.

Bei Dunkelheit sind Farben allenfalls als Schwarz-Weiß-Unterschiede mit Graustufungen erkennbar.

Die Softwaregestaltung sollte berücksichtigen, dass eine ausgewogene Leuchtdichte- und Helligkeitsverteilung die Seh- und Helligkeitsunterschiede, die zu erhöhten Anforderungen an die Hell-Dunkel-Adaptation führen, sollten vermieden werden. Auf der Bildschirmanzeige dargestellte Software muss in einem vorab spezifizierten Bereich von Umgebungsleuchtdichten einsetzbar sein (vgl. DGUV Information 215-442 „Beleuchtung im Büro“). Innerhalb dieser Grenzen muss die Wahrnehmbarkeit von Informationen auf der Bildschirmanzeige durch eine ergonomische Informationsdarstellung der Software erreicht werden. Da neben der Bildschirmanzeige zur

Aufgabenbearbeitung meist Vorlagen auf hellem Papier genutzt werden, sind bei ausreichender Beleuchtung die Leuchtdichten in der Umgebung und am Arbeitsplatz so, dass bei Verwendung eines hellen Hintergrundes für die Bildschirmanzeige mit weniger störenden Spiegelungen gerechnet werden kann. Das ist einer der Gründe, weshalb Positivdarstellung (dunkle Schrift auf hellem Grund) von Software grundsätzlich zu empfehlen ist.

Immer wenn ein Ausschnitt einer Bildschirmanzeige direkt angeschaut wird, fällt dieser Ausschnitt genau auf die Stelle des schärfsten Sehens auf der Netzhaut (Fovea) des Menschen (siehe Abbildung 4). Soll ein größeres Sehobjekt auf der Bildschirmanzeige betrachtet werden, so werden zum scharfen Sehen Ausschnitte von Sehobjekten nacheinander auf die Fovea fokussiert. Dazu sind Kopfbewegungen oder Bewegungen des Augapfels in Form von ruckartigen Blicksprüngen (Sakkaden) notwendig (siehe Abbildung 5). Die Wahrnehmung eines größeren Sehobjekts auf der Bildschirmanzeige, das aus mehreren Ausschnitten zusammengesetzt werden muss, benötigt mehr Zeit und führt zu einer höheren Belastung.

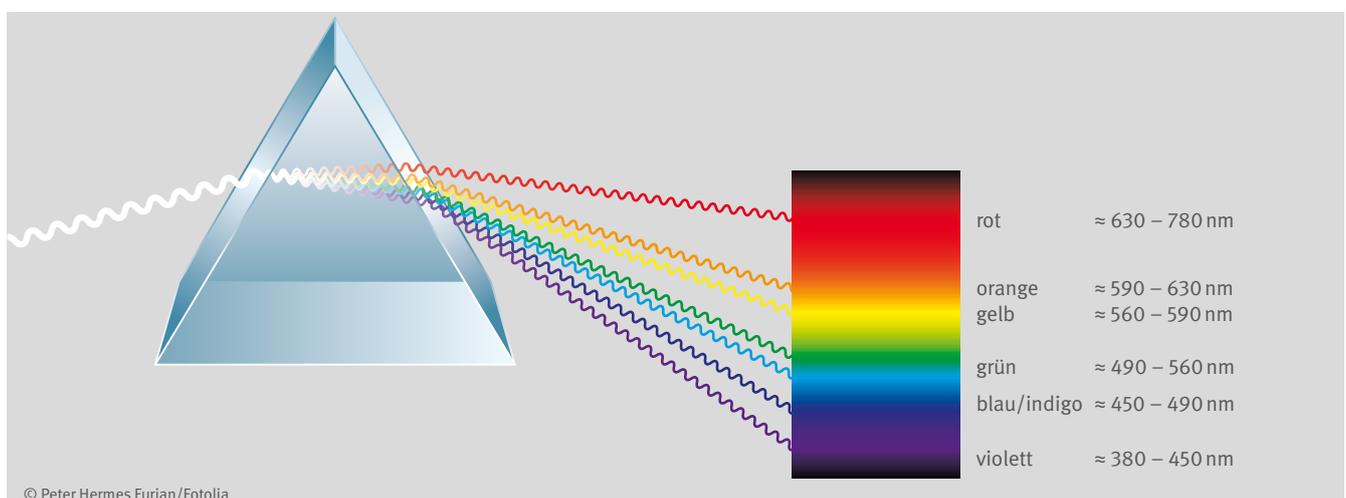


Abb. 3 Mit Hilfe eines Prismas wird „weißes“ Sonnenlicht in seine Spektralfarben zerlegt.

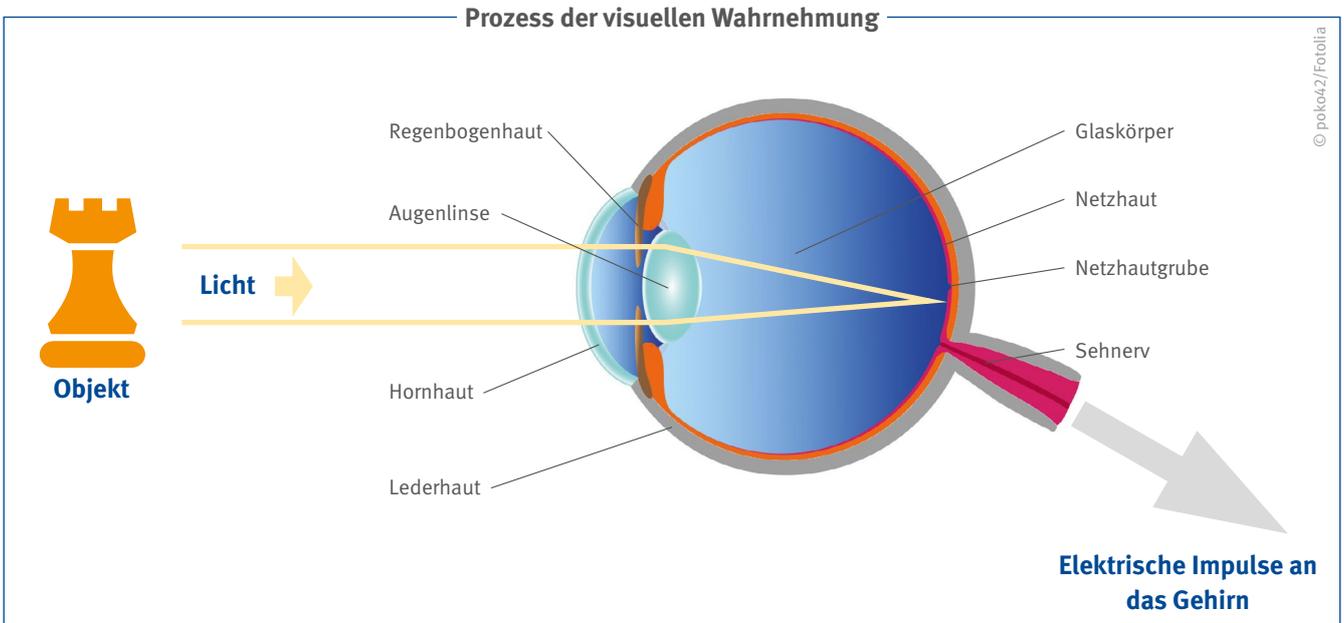


Abb. 4 Prozess der visuellen Wahrnehmung

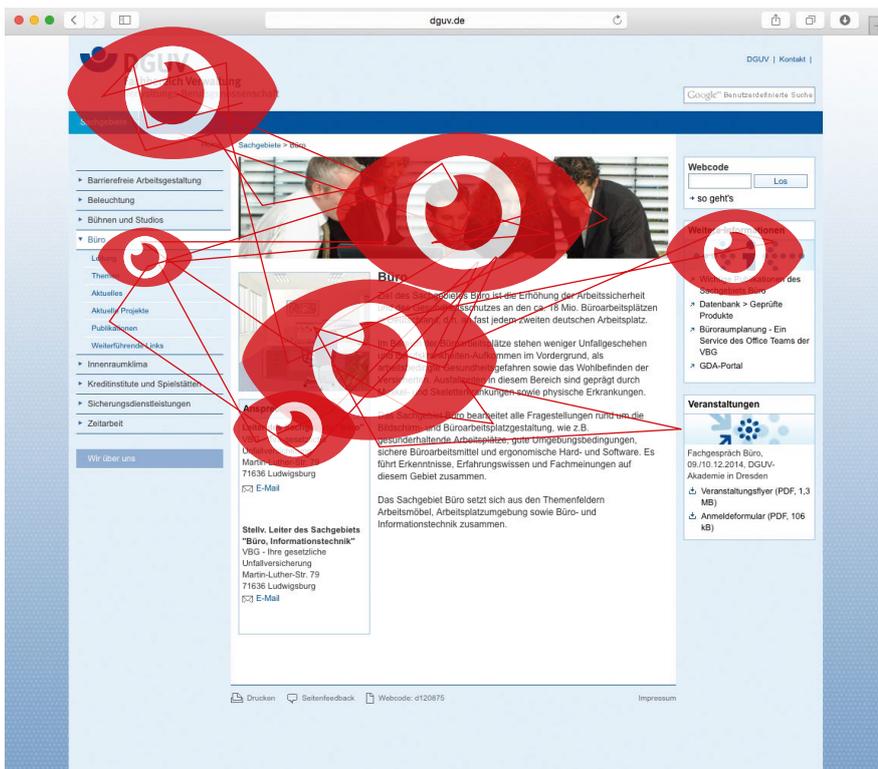


Abb. 5 Betrachtung eines Sehobjekts durch Blicksprünge auf Teile des Sehobjekts und Zusammenstellung als ganzes Sehobjekt

**Die Akkomodation - Der Vorgang der Veränderung der Augenlinse**

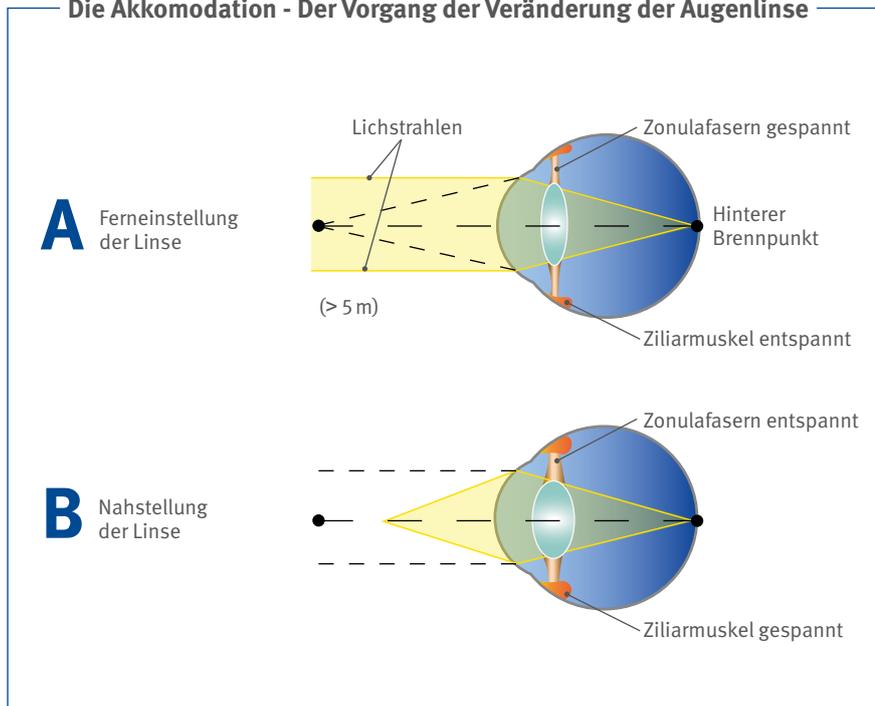


Abb. 6 Akkomodation

Neben den Blicksprüngen muss auch die Brechkraft der Augenlinse ständig so verändert werden, dass selbst ein entferntes Bild auf der Netzhaut abgebildet wird. Diesen Vorgang der Brechkraftanpassung der Augenlinse nennt man Akkomodation (Abbildung 6). Je größer der Entfernungsunterschied, desto mehr Zeit wird benötigt, um die richtige Akkomodation einzustellen und damit Gegenstände in verschiedenen Entfernungen zu erkennen. Zeiten von mehr als einer halben Sekunde sind keine Seltenheit. Bildschirmanzeige und andere Medien, die als Informationsquelle dienen, sollten daher eine ähnliche Entfernung zum Auge haben. Sind unterschiedliche Entfernungen erforderlich, sollten Informationen nach Priorität strukturiert werden. Selbstverständlich müssen Objekte auf allen Anzeigen so groß dargestellt werden, dass sie aus der Ferne leserlich sind.

Die Anpassung des Auges an unterschiedliche Beleuchtungsstärken (zum Beispiel Nachthimmel oder Sonnenschein, dunkel

oder hell) wird Adaptation genannt. Für die Adaptation sind die Regulierung des Lichteinfalls durch den Pupillenreflex sowie die Rezeptoren auf der Netzhaut des Auges (Zapfen und Stäbchen, siehe oben) verantwortlich. Die Adaptation ist ein langsamer Vorgang, den der Mensch nicht beschleunigen kann. Für die vollständige Dunkeladaptation nach einem Aufenthalt bei Tageslicht mit plötzlichem Übergang in Dunkelheit werden circa 40 Minuten benötigt. Auch wenn an Arbeitsplätzen, an denen mit Software gearbeitet wird, Übergänge nicht so extrem oder schnell auftreten, sind die Adaptationszeiten bei Darstellungen auf dem Bildschirm zu berücksichtigen.

Die Farb- und Beleuchtungsbedingungen der Bildschirmanzeige sind mit der näheren und weiteren Arbeitsumgebung abzustimmen. Blickübergänge während der Aufgabenbearbeitung vom Bildschirm zur Arbeitsumgebung machen geringe Unterschiede dieser Kontraste erforderlich. Bei tageslichtbelegten Büros sind

grundsätzlich neben einer Positivdarstellung auf dem Bildschirm auch helle und matte Bildschirmgehäuse und Schreibtischoberflächen gefordert. Damit werden geringere Anforderungen an die Hell-Dunkeladaptation der Augen gestellt und die Wahrnehmung der Informationen wird erleichtert. Diese Grundorientierung der Software (mit Positivdarstellung) ist dann auch die Vorlage für die Gestaltung der Bildschirmanzeigen innerhalb der Software. Das visuelle System verrichtet bei der Tätigkeit am Bildschirm Schwerarbeit. Dateneingabe- oder Dialogtätigkeit erfordern vom visuellen System etwa 12.000 bis 33.000 Kopf- und Blickbewegungen zwischen Bildschirm, Tastatur und Vorlage sowie 4.000 bis 17.000 Pupillenreaktionen pro Arbeitstag. Bei der Arbeit am Bildschirm kann die physische und psychische Belastung hoch sein und gesundheitliche Beschwerden können auftreten.

Darauf kann umfangreich Einfluss genommen werden durch die

- Gestaltung der Software,
- Abbildung des Dialogs zwischen Mensch und Computer sowie,
- Informationsdarstellung.

**3.2.2 Das auditive Sinnessystem**

Das Ohr nimmt Schallwellen als Druckschwankungen in der Luft wahr und setzt sie in elektrophysiologische Signale um, die dann zum Gehirn weitergeleitet werden (Abbildung 7). Dort werden sie je nach Wellenlänge und Amplitude als Töne mit Höhe und Intensität interpretiert. Schallquellen können lokalisiert werden und sind als Informationsquelle wegen der Ausbreitung von Wellen nicht richtungsgebunden.

Für das Ohr des Menschen ist nur ein Teil der Schalldruckwellen (Frequenzbereich zwischen ca. 20 Hz bis 20 kHz) als Ton hörbar. Der wahrnehmbare Frequenzbereich engt sich mit zunehmendem Alter ein. Die gleichzeitige Verarbeitung mehrerer Töne ist abhängig von den



Abb. 7 Auditive Wahrnehmung

Frequenzunterschieden schwierig (z. B. ist das Hören von Tönen bei Hintergrundgeräuschen schwierig). Der Schalldruckpegel sollte 80 dB(A) nicht überschreiten und abhängig von den Anforderungen der Aufgaben auch unterhalb von 55 dB(A) liegen. Ein wahrnehmbarer Schalldruckpegel ist von der Frequenz abhängig: Die Hörschwelle für niedrige und hohe Frequenzen des wahrnehmbaren Bereichs liegt bei ca. 60 dB, während Töne von etwa 4 kHz bereits bei 0 dB erkannt werden. Für die Verwendung akustischer Signale ergibt sich daraus Folgendes:

Akustische Signale sollten nur der Orientierung dienen und können eine weitere, z. B. visuelle Kodierung, unterstützen. Werden sie für Fehlermeldungen verwendet, kann man aus dem akustischen Signal allein noch nicht erkennen, wo der Fehler war und wie er zustande gekommen ist – es sei denn, es handelt sich um eine Sprachausgabe. Da akustische Signale meist plötzlich auftreten, kommt es sofort zu einer Ablenkung der Aufmerksamkeit, um das akustische Signal zu lokalisieren und zu interpretieren. Die Aufgabenbearbeitung und andere Wahrnehmungsprozesse werden dadurch unterbrochen. Um Störungen zu vermeiden, sollte möglichst nicht akustisch informiert werden. Für Sehbehinderte hingegen können akustische Signale essentielle Informationsquellen sein (siehe Kapitel 10 Barrierefreie Gestaltung von Software).

Informationen zur Gestaltung von akustischen Signalen werden zukünftig in der Normenreihe DIN EN ISO 9241 zusammengeführt und können im Sinne von Gefahrensignalen für Arbeitsstätten der DIN EN ISO 7731 entnommen werden.

### 3.2.3 Das haptische Sinnessystem

Mit der Haut als haptischem Sinnessystem besitzt der Mensch die Fähigkeit, Informationen zu fühlen und zu ertasten. Er kann sich dadurch mechanische Eigenschaften von Objekten zunutze machen. Beim Annähern an oder Berühren eines Gegenstandes, z. B. bei Nutzung von Eingabegeräten oder Berührungsbildschirmen (engl. Touchscreen), leiten Rezeptoren Informationen über Druck und Temperatur an das Gehirn weiter. Bei Eigenbewegung von Körperteilen (wie z. B. Arm, Hand, Finger) kommen Informationen über die aktive Kraftausübung hinzu, wenn Form, Größe und Struktur von Gegenständen ertastet werden. Eine Beschreibung der haptischen Wahrnehmung und kontrollierte Handlungssteuerung sind möglich, insbesondere, wenn diese Informationen mit bekannten Erfahrungen aus dem Gedächtnis verknüpft werden.

Weitere Informationen zur Gestaltung von Eingabemitteln und Berührungsbildschirmen bieten DIN EN ISO 9241-410 zu Gestaltungskriterien für physikalische Eingabegeräte und VDI/VDE 3850 Blatt 2 zu Merkmalen, Gestaltung und Einsatzmöglichkeiten von Benutzungsschnittstellen mit Touchscreens.

### 3.2.4 Ausgewählte Prozesse der Wahrnehmungsorganisation

Bei einem einzelnen Wahrnehmungsvorgang nehmen wir immer nur einen kleinen Ausschnitt aus einem ganzen Bild, einen Einzelton aus einer Klangfolge oder einen Buchstaben aus einer Braillezeile<sup>1)</sup> wahr. Die einzelnen Sinneseindrücke müssen also in einem zeitlichen und räumlichen Prozess organisiert werden, um den Zusammenhang zu erkennen. Bedeutende Organisationsprinzipien zur Informationsdarstellung, die unsere Wahrnehmung unterstützen sind:

- a) Unterscheidung von Figur und Hintergrund
- b) Gliederung in Bereiche
- c) Prinzipien der Gruppierung – Gestaltgesetze
- d) visuelle Bewegungswahrnehmung

- a) Unterscheidung von Figur und Hintergrund

Eine Interpretation von Gesamtbildern wird dem Menschen erleichtert, wenn er sie leicht in ihre elementaren Bestandteile „Figur“ und „Hintergrund“ zerlegen kann. Der Hinter- bzw. Untergrund ist der Bestandteil, von dem sich eine Figur abheben und wahrgenommen werden kann (z. B. dunkler Text auf und umrahmt von hellem Papier). Die Einteilung von Figur und Hintergrund wird vom Sinnessystem nicht tatsächlich in dieser Unterscheidung gesehen, sondern sie entsteht durch die Reizverarbeitung im Gehirn, bei der Wahrnehmungstäuschungen und Erfahrungswissen eine Rolle spielen (vgl. das weiße Dreieck in der Abbildung 8). Bei der Software kann Erfahrungswissen z. B. durch eine konsistente Gliederung von Bereichen (z. B. gleichbleibender Aufbau der Bildschirmmaske) gefördert werden.

Die Unterscheidbarkeit informationsrelevanter Signale vor einem Hintergrund unwichtiger Informationen kann durch

<sup>1)</sup> Braillezeile = Computer-Ausgabegerät für blinde Menschen, das Zeichen in Brailleschrift darstellt.

qualitative Veränderungen oder Mehrfachkodierung verbessert werden. Dies verringert die Anstrengung beim Filtern der Signale. Hilfreich sind Veränderung der Intensität der Signale, unterschiedliche Kodierung von Signalen durch Form, Farbe, Dauer, oder Zeitcharakteristiken, eine reduzierte Intensität des Hintergrunds (Rauschen), und ein Maskieren und Filtern mithilfe technischer Systeme (DIN EN ISO 10075-2).



Abb. 8 Prinzip der Figur-Hintergrund-Gliederung (aus: Zimbardo, Gerrig, 1999, Seite 131).

#### b) Gliederung in Bereiche

Die Strukturierung des Bildschirms durch eine konsistente (gleichbleibende) Gliederung erleichtert die Wahrnehmung als vollständiges Bild mit allen seinen Einzelteilen und dadurch auch die Orientierung auf dem Bildschirm. Eine Gliederung mithilfe einer abrupten Veränderung der Textur oder ein abrupter Wechsel der Farbqualität (Ton, Sättigung oder Helligkeit) markiert eine „Grenze“ zwischen zwei Bereichen. Diese Grenzen werden vom Sinnessystem wahrgenommen und im Gehirn als solche weiterverarbeitet, sodass der Mensch sowohl ein Bild als Ganzes erkennen als auch darin enthaltene einzelne Bereiche voneinander trennen kann. Möglichkeiten zur Gliederung

oder zur Abgrenzung von Bereichen auf dem Bildschirm werden im Kapitel zum Informationsdesign vorgestellt.

#### c) Prinzipien der Gruppierung

##### – Gestaltgesetze

Das visuelle, auditive und haptische System des Menschen folgt Prinzipien der Gruppierung, auch als Gestaltgesetze beschrieben. Danach können einzelne Reize selten korrekt wahrgenommen und verarbeitet werden, wenn sie nur als einzelne Teile betrachtet werden. Verschiedene Arten der Anordnung von einzelnen Teilen beeinflussen die Wahrnehmung der Gesamtstruktur. Die einzelnen Teile müssen daher in einer Gesamtstruktur verarbeitet werden können. Von den Gestaltgesetzen werden die Gesetze der Nähe, der Ähnlichkeit und der Geschlossenheit im Kapitel 8 „Informationsdesign“ anhand von Beispielen illustriert.

#### d) Visuelle Bewegungswahrnehmung

Die Wahrnehmung von Bewegungen oder Veränderungen von Informationen beinhaltet im Vergleich zu statischen Bildern zusätzliche Informationen und Impulse.

- Bewegungen erregen unsere Aufmerksamkeit: Eine Bewegung im Gesichtsfeld löst eine sofortige Augenbewegung zur Kontrolle aus. Dadurch wird das bewegende Objekt in der Fovea (Stelle des schärfsten Sehens) fokussiert und kann dann erkannt werden.
- Die Bewegungen eines Objektes relativ zu einer Beobachterin bzw. einem Beobachter vermitteln Hinweise zur räumlichen Tiefe und dreidimensionalen Gestalt von Objekten.
- Bewegungen vermitteln Informationen, durch die sich Figur und Untergrund unterscheiden lassen. Ein sich bewegendes Objekt wird meist als Figur auf einem festen Untergrund wahrgenommen.
- Durch Bewegungen kann die Nutzerin bzw. der Nutzer aktiv mit der Software interagieren. So kann anhand der Bewegungen des Mauszeigers auf dem Bildschirm bewertet werden, welche

Bewegungen mit der Hand an der Maus erforderlich sind, um das Ziel zu erreichen.

Die Bewegungswahrnehmung dient dem Erkennen von Veränderungen und einer räumlichen und zeitlichen Orientierung beim Schreiben eines Textes oder beim Platzieren einzelner Objekte auf dem Bildschirm. Die Darstellung von Bewegung zur Kennzeichnung eines andauernden Prozesses ist hilfreich (z. B. Fortschrittsbalken). Da blinkende, erscheinende oder sich bewegende Animationen von der eigentlichen Aufgabenbearbeitung ablenken, sollten sie nur eingesetzt werden, wenn die Aufgabe unterbrochen werden kann oder die Animationen für die Bearbeitung der Aufgaben unabdingbar erforderlich sind (z. B. Fehlermeldung).

### 3.3 Verarbeitung von Informationen und Wissensgenerierung

Bei der Gestaltung von Software sind u. a. die Funktion, die Kapazität und die Organisation des menschlichen Gedächtnisses zu berücksichtigen.

Die Wahrnehmung sowie das Vorstellen, Denken, Behalten, Bewerten, Finden von Entscheidungen und Auswählen von Handlungen gehören zur Informationsverarbeitung/Wissensgenerierung des Menschen. Das menschliche Gedächtnis verarbeitet Informationen weiter, die über die Sinnessysteme aufgenommen wurden und verknüpft sie mit dem bereits vorhandenen Wissensvorrat. Die Verarbeitung aller Informationen führt schließlich zur

- Generierung von Wissen,
- zur Lösung von Aufgabenteilen,
- zu einer Entscheidungsfindung und
- zur Umsetzung von Handlungen.

Darüber hinaus steuert das Gehirn einerseits den Vorgang der Wahrnehmung und andererseits die Umsetzung einer Handlung. So kann willentlich bestimmt werden,

- auf welches Wort einer Bildschirm-Anzeige die Augen gerichtet,
- wie die Blicke über das Wort gelenkt und
- welche Handbewegungen ausgeführt werden, sodass durch gezielte Arm-Hand-Finger-Bewegung die Maus geführt und dadurch mit dem Mauszeiger ein Eingabefeld gewählt werden kann.

#### 3.3.1 Die begrenzte Aufnahmekapazität des Gedächtnisses

Physikalische Reizmerkmale aus Daten der Software sind z. B. Farbe, Intensität, Kontrastverteilung, Position, Bewegung etc. Der Mensch nimmt diese zunächst in die Sinnes-Speicher auf. Darin werden die Daten kurze Zeit (visuell circa 0,5 Sekunden, auditiv circa 5 Sekunden) zwischengespeichert. Im Anschluss gelangen diese Informationen in das Kurzzeitgedächtnis. Dort findet ein großer Teil der bewussten

Denkarbeit statt. Erst weitere Gedächtnisprozesse interpretieren die in den sensorischen Speichern vorhandenen Sinnesdaten, messen ihnen Bedeutung bei und verknüpfen sie mit anderen Gedächtnisinhalten.

Chunks sind die größten Einheiten, die Sinnessysteme in das Gedächtnis aufnehmen können. Chunks sind z. B. mehrstellige Zahlen, Tonfolgen, Abkürzungen, Wörter, Symbole, Piktogramme oder größere begriffliche Einheiten. Die Speicherkapazität des menschlichen Kurzzeitgedächtnisses liegt bei fünf bis neun Chunks. Diese Menge kann über einen Speicherzeitraum von etwa 15 bis 30 Sekunden im Kurzzeitgedächtnis behalten werden. Je mehr Chunks gespeichert werden müssen, desto unzuverlässiger wird der Zugriff – das heißt, die korrekte Wiedergabe der einzelnen Chunks.

Eine große Anzahl von Informationseinheiten steigert die psychische Belastung und das Risiko für Fehlleistungen (vgl. auch DIN EN ISO 10075-2). Als allgemeine Empfehlungen zur Informationsverarbeitung gelten folgende Aspekte:

- Informationsmenge reduzieren und an die Erfordernisse der Aufgabe anpassen.
- Organisationsprinzipien (siehe oben) nutzen und Informationsmenge und -qualität aufgabenbezogen organisieren (Informationseinheiten bilden).
- Mithilfe künstlicher Chunks, z. B. Gruppierung von Informationen, kann das Kurzzeitgedächtnis bei der Bildschirmarbeit effektiv genutzt werden.
- Informationen priorisieren (Informationen untergeordneter Priorität oder Zusatzinformationen abrufbar bzw. für Bedarfsanfrage vorhalten).
- Verarbeitung mit einer Geschwindigkeit ermöglichen, welche die Aufnahme erlaubt.
- Schon vorhandene und gut eingeführte Chunks nutzen, z. B. Elemente der Umgang- oder Fachsprache oder bekannte grafische Darstellungen. Ein Zugriff auf

diese Chunks ist auch nach längerer Zeit möglich, da sie wegen der häufigen Verwendung im Langzeitgedächtnis gespeichert sind.

- Alle Informationen, die parallel verarbeitet werden sollen, müssen auch parallel dargestellt werden. Erfordert die Aufgabe einen Vergleich von Informationen, so sollen diese nicht seriell dargestellt, d. h. nicht auf mehrere Bildschirmanzeigen verteilt werden.
- Werden Informationen seriell dargestellt, dann sind angemessene Zeiträume vorzusehen, damit das Arbeitsgedächtnis die Auswahl und Speicherung der relevanten Daten bewältigen kann.
- Das Kurzzeitgedächtnis funktioniert umso besser, je unterschiedlicher die Chunks sind. Es ist einfacher, eine bestimmte Anzahl unterschiedlicher Chunks zu speichern als die gleiche Zahl ähnlicher Chunks. Daher ist es sinnvoller, Briefe an verschiedene Firmen unter dem jeweiligen Firmennamen zu speichern, als sie unter brief1.docx, brief2.docx etc. abzulegen.
- Zugangswege zu und Ordnung der gespeicherten Informationen des Kurzzeitgedächtnisses sind streng nacheinander angeordnet. Es ist am einfachsten, gespeicherte Informationen in der Reihenfolge wiederzugeben, in der sie auch gespeichert wurden.
- Den Benutzerinnen und Benutzern sollten mögliche Alternativen angezeigt werden, über die sie Inhalte wiedererkennen können und nicht aus dem Gedächtnis abrufen müssen.

#### 3.3.2 Der Motor der Wissensgenerierung

Das Kurzzeitgedächtnis übernimmt auch die Funktion eines Arbeitsgedächtnisses. Damit werden Wahrnehmungs-, Denk-, Bewertungs-, Entscheidungs- und Umsetzungsprozesse unterstützt und elementare kognitive Operationen (Erkennen, Vergleichen, Speichern, Abrufen, Erinnern) genutzt. Die Benutzerin bzw. der Benutzer trägt das gespeicherte Wissen über den

erreichten Zustand ihrer bzw. seiner Aufgabenbearbeitung im Langzeitgedächtnis. Um Zeichen, Berührungsmuster oder Tonfolgen zu erkennen, wird gespeichertes Wissen aus dem Langzeitgedächtnis abgerufen und kurzfristig im Kurzzeitgedächtnis mit dort vorhandenen Informationen verglichen. In diesem Prozess wird Wissen generiert. Das Arbeitsgedächtnis ist somit der Motor der Wissensgenerierung.

Langfristig wird Wissen im Langzeitgedächtnis gespeichert. Seine Kapazität scheint nach bisherigen Erkenntnissen unbegrenzt zu sein. Erinnerungsschwächen können damit zusammenhängen, dass auf vorhandenes Wissen längere Zeit nicht zurückgegriffen wurde. Neue Informationen in das Langzeitgedächtnis einzufügen, ist nur möglich, wenn sie mit vorhandenem Wissen verknüpft werden können.

Software ist daher ergonomisch so zu gestalten, dass Informationen an vorhandene Wissensseinheiten anknüpfen. Die Verknüpfungen können unter anderem Generalisierungen, Spezialisierungen, Ähnlichkeiten, Ausnahmen, Teilebeziehungen, aber auch andere inhaltlich kaum greifbare Zusammenhänge darstellen:

- Die Bekanntheit von Informationen (wie z. B. Abkürzungen oder Begriffe) ist für das Erkennen und Verstehen entscheidend.
- Werden unerwartete und seltene Wörter verwendet, dann sinkt die Lesegeschwindigkeit aufgrund des häufiger erforderlichen Vor- und Zurückspringens.
- Die Sprechbarkeit von Zeichenfolgen ist hilfreich. Beim stummen Lesen schwer sprechbarer Texte kann die Lesegeschwindigkeit auf die des lauten Lesens zurückgehen.
- Zeichenfolgen auf dem Bildschirm sollten möglichst vertraut, gut unterscheidbar und sprechbar sein.

- Als Code oder Abkürzungen sollten sprechbare Silben verwendet werden.
- Wörter und Abkürzungen sollten aus der Fachsprache der Benutzerinnen und Benutzer stammen, und aus ihrem Wissen und mit ihren Erfahrungen in Verbindung stehen (z. B. IBAN für internationale Kontonummern).
- Ein Wiedererkennen von Inhalten, die im Gedächtnis gespeichert sind, ist leichter und effektiver als das Erinnern. Daher ist das Anzeigen von Alternativen effektiver und weniger beanspruchend als relevante Informationen oder Objekte aus dem Gedächtnis abzurufen.
- Charakteristische Unterscheidungsmerkmale sollten regelhaft und konsistent am Anfang oder Ende einer Zeichenfolge platziert werden, damit sie sich auf den ersten Blick erfassen lassen (z. B. Mahnung1.docx, Mahnung2.docx etc.).
- Für die Langzeitspeicherung benötigen Menschen Zeit, bis sie Informationen in einen Kontext ähnlicher Zusammenhänge (mentaler Repräsentationen) einbetten können.
- Den Beschäftigten sollen Informationen in externen Speichern außerhalb des Gedächtnisses bereitgestellt werden, die sie abrufen können (z. B. Abrufen von Zusatzinformationen, kontext-sensitive Hilfesysteme). Dadurch kann man eine insgesamt zu hohe Beanspruchung des Langzeitgedächtnisses vermeiden.
- Erfordert es die Aufgabe, Objekte zu identifizieren oder zu wählen, dann sollte die visuelle Ähnlichkeit möglichst gering sein. Berücksichtigt die Softwaregestaltung Unterschiedsschwellen der Sinnessysteme des Menschen, wird das Erkennen von Objekten erleichtert und das Risiko für Fehler reduziert.

### 3.4 Umsetzung von Handlungen

Der Dialog mit der Software muss so zugänglich und die Darstellung von Informationen auf der Bildschirmanzeige muss so aufbereitet sein, dass Beschäftigte Entscheidungen treffen und Handlungen zur weiteren Bearbeitung von Aufgaben umsetzen können. Die Prinzipien der Interaktions- und Informationsgestaltung sind auch für die Umsetzung von Handlungen zu berücksichtigen. Angezeigt werden sollten nur die Informationen, die für die Bearbeitung aktuell relevant sind. Weitere Informationen können im Hintergrund vorgehalten werden. Optionen zur Handlungsumsetzung sollten direkt angesteuert werden können (z. B. Eingabemittel und Eingabefelder).

Die Steuerbarkeit von Handlungsschritten zur Aufgabenbearbeitung mithilfe von Eingabemitteln und Software hängt von folgenden Aspekten ab (vgl. DIN EN ISO 10075-2):

- Vom Ordnungsgrad der Steuerung (z. B. Lagesteuerung mit der Maus versus Geschwindigkeitssteuerung mit dem Joystick),
- der Dimensionalität der Steuerungsbewegungen (z. B. Pfeiltasten für vier Richtungen, Maus für Lageveränderungen auf einer Ebene),
- von Zeitverzögerungen in der Systemantwort (z. B. Antwortzeiten computer technischer Funktionen wie Datenspeicherung oder Datenfinden),
- der Anzeige von Informationen als Rückmeldung auf Steueraktivitäten der Benutzerin oder des Benutzers (z. B. Veränderung eines Optionsfeldes nach Optionsauswahl mit dem Mauszeiger),
- der Anzeigen-Stellteil-Kompatibilität (zur Bewegung der Maus proportionale Bewegung des Zeigers).

Da Handlungen meist durch eine Ansteuerung von Feldern, Optionen oder Schaltflächen auf der Bildschirmanzeige umgesetzt werden sollen, sind Systeme zur Lagesteuerung einzusetzen. Die Bewegung

der Maus führt dann proportional zu einer Mauszeigerbewegung. In diesem Fall nicht kompatibel wäre der Einsatz eines Joysticks, der als System der Geschwindigkeitssteuerung zum punktgenauen Erreichen von Anzeigepositionen ungeeignet ist.

Eine Handlungsumsetzung mithilfe von Eingabemitteln, die ergonomische Regel- und Gesetzmäßigkeiten berücksichtigt, trägt zu einer ergonomischen Gestaltung bei. Nach dem **Fitts'schen Gesetz** ist die Zeit zum Erreichen einer Steuerungsoption auf der Bildschirmanzeige eine Funktion der Entfernung zu dieser Option und ihrer Größe. Ein Ziel wird mit einem Eingabemittel also umso schneller erreicht, je größer das Ziel ist und je geringer der Abstand zum Ziel ist. Dieser Zusammenhang wurde anhand von Greifbewegungen zu Stellteilen ermittelt und konnte auch auf die Interaktions- und Informationsgestaltung übertragen werden. Kleinere Options- oder Schaltflächen und große Entfernungen von Eingaben für nacheinander folgende Arbeitsschritte sollten vermieden werden. Einerseits wird dabei das Auffinden und Fokussieren erschwert und andererseits ist nach dem **Fitts'schen Gesetz** für ein punktgenaues Erreichen relativ viel Zeit erforderlich.

Auch das **Hick'sche Gesetz** gehört zu ergonomischen Regel- und Gesetzmäßigkeiten und fördert eine ergonomische Interaktions- und Informationsgestaltung. Danach steigt die Bearbeitungszeit mit der Anzahl alternativer Optionsfelder, Schaltflächen oder Informationseinheiten zwischen denen gewählt werden soll. Daher sollten nur für die aktuellen Aufgaben Informationen dargestellt werden. Zusätzliche, aber für die Aufgabenbearbeitung nicht relevante Informationseinheiten erhöhen die erforderliche Wahrnehmungs- und Bearbeitungszeit und belegen Gedächtnisressourcen. Da sich die Wahlentscheidungen nicht auf Einzelinformationen, sondern auf Informationseinheiten beziehen, lässt sich die Informa-

tionsmenge durch eine angemessene Wahrnehmungsorganisation verringern. So kann z. B. durch eine für die Aufgaben angemessene hierarchische Strukturierung der Zugriff auf Detailinformationen bei Bedarf ermöglicht werden. Auch eine Gliederung in Haupt- und Untermenüs kann Informationseinheiten verringern und dadurch den Zugriff auf einzelne Menüoptionen verbessern.

In diesem Zusammenhang ist ein Verweis auf **Tesler's Gesetz** zur Erhaltung der Komplexität in der Softwareergonomie hilfreich. Software oder Benutzungsoberflächen als Teile davon haben eine inhärente Komplexität. Ziel eines benutzungsorientierten Designs sollte es nicht sein, diese Komplexität zu eliminieren. Die Komplexität sollte in den Gestaltungsprozess verschoben werden, indem möglichst viel Funktionalität in das System verlagert und einem Beschäftigten die Interaktion mit der Komplexität erleichtert wird.

Eine an den Aufgaben und ergonomischen Gestaltungsprinzipien orientierte Gestaltung der Interaktionsschnittstellen kann die Entwicklung von Software aufwendig machen. Dieser Aufwand rechtfertigt sich allerdings allein dadurch, dass die Nutzung dieser Software die Aufgabenbearbeitung erheblich verbessern kann. Die Nutzungsqualität verbessert sich, wenn z. B. der Grundsatz, dass Erkennen einfacher ist als Erinnern, umgesetzt wird. Informationsdarstellungen, Steuerbewegungen und Systemantworten sollen kompatibel mit den Erwartungen sein, um zusätzliche Anstrengung und widersprüchliche Information zu vermeiden (vgl. DIN EN ISO 10075-2). Erwartungskonforme Darstellungen von Informationen oder Dialogstrukturen werden durch die oben vorgestellten Organisationsprinzipien unterstützt. Auch direkte Möglichkeiten der Steuerung sollten erwartungskonform gestaltet sein. Steuerfelder, die das Blättern bzw. Vor- und Rückwärtsbewegen in einem Text

veranlassen, sollten kompatibel zur dargestellten Bezeichnung der Bewegungsrichtung und zur Bewegungsrichtung des Eingabemittels sein.

Die Gestaltung von Software unterstützt die Umsetzung von Handlungen auch, wenn

- die Bewegung des Menschen erwartungskonform und kompatibel zu einer Bewegung des Cursors führt (z. B. Maus-Anzeiger auf dem Bildschirm bewegt sich kompatibel zur Bewegung des Maus-Eingabegeräts),
- die Steuerungsflächen bei Touchscreens groß genug und für die Aufgaben strukturiert sind,
- der Widerstand bei Tastendruck ausreichend groß ist und eine wahrnehmbare Rückmeldung über erfolgte Umschaltung beim Tasten gegeben wird,
- Flächen zur Auslösung von unterschiedlichen Funktionen gekennzeichnet und voneinander abgegrenzt sind,
- weitere Aspekte beachtet sind, die für einen Einsatz von Eingabemitteln und berührungssensitiven Bildschirmen ausschlaggebend sind: Sicherheit vor Fremdauslösung, Verunreinigung durch Hautkontakt usw.

Weitere Informationen zur Gestaltung von Eingabemitteln und Berührungsbildschirmen bieten DIN EN ISO 9241-410 zu Gestaltungskriterien für physikalische Eingabegeräte und VDI/VDE 3850 Blatt 2 zu Merkmalen, Gestaltung und Einsatzmöglichkeiten von Benutzungsschnittstellen mit Touchscreens.

# 4 Psychologische Ansätze in der Softwareergonomie

In diesem Kapitel werden psychologische Ansätze bei der Arbeit am Bildschirm dargestellt, die sich insbesondere auf die Gestaltung, die Einführung und die Benutzung von Software zur Bearbeitung von Arbeitsaufträgen beziehen.



Neben der anzustrebenden Effektivität und Effizienz einer Software fordert die DIN EN ISO 9241-11 ebenso die „Zufriedenstellung des Benutzers“, um eine maximale Gebrauchstauglichkeit von Software zu erreichen. Die Freiheit von Beeinträchtigungen und die positive Einstellung gegenüber der Nutzung der Software werden als Teil der anzustrebenden User Experience verstanden. Dieses Erlebnis der Benutzerinnen und Benutzer umfasst „sämtliche Emotionen, Vorstellungen, Vorlieben, Wahrnehmungen, physiologischen und psychologischen Reaktionen, Verhaltensweisen und Leistungen“.

Psychologische Ansätze helfen dabei, all diese Aspekte in einem positiven Benutzungserlebnis zu vereinen, indem sie das Verhalten und Erleben der Benutzerin

oder des Benutzers beschreiben und messen. Die so gewonnenen Erkenntnisse liefern für die Softwareentwicklung und Implementierung bedeutsame Hinweise, um übermäßige psychische Belastung zu vermeiden. Software sollte unter Zuhilfenahme psychologischer Erkenntnisse optimal an die Benutzerin und den Benutzer, die Aufgabe und den Arbeitskontext angepasst werden, um folgende Aspekte zu unterstützen:

- ein geringes Stresserleben
- eine hohe Motivation
- Förderung psychischer Gesundheit
- menschengerechte Arbeitsgestaltung
- effektive und effiziente Aufgabenerledigung
- Steigerung des täglichen Wohlbefindens und der Zufriedenheit
- individuelle Entfaltungsmöglichkeiten

## 4.1 Psychische Belastung und Beanspruchung berücksichtigen

Ergonomische Software passt sich an die Anwenderin bzw. den Anwender, die Aufgabe und die Arbeitsumgebung an und zieht weniger psychische Belastung und Beanspruchung nach sich. Das folgende Beispiel verdeutlicht dieses Phänomen.

## Praxisbeispiel:

### Herr Altmann und seine Urlaubsvertretung am Express-Fahrkartenschalter

#### Ausgangssituation:

Herr Altmann arbeitet am Express-Fahrkartenschalter eines hoch frequentierten Bahnhofs. Sein Arbeitsauftrag ist die Beratung von eiliger Kundschaft und der zügige Verkauf von Fahrkarten. Der Geräuschpegel, komprimierte Stoßzeiten und die Kommunikation mit stark gestressten Fahrgästen sind zu bewältigende Anforderungen an seinem IT-gestützten Arbeitsplatz. Vor einer Woche wurde sein Arbeitsplatz mit einer neuen Software ausgestattet, die ergonomischen Kriterien gerecht wird und teilweise anpassbar ist.

**Die alte Software** bot Herrn Altmann keinerlei zeitliche Orientierungshilfe, da die Zeitdauer lediglich durch eine sich drehende Sanduhr symbolisiert wurde. Er fühlte sich immer wieder unter Druck, da er besonders eiligen Fahrgästen keine Auskunft darüber geben konnte, wie lange der Buchungsprozess noch dauern wird. Beim Kauf der alten Software hatte man leider auch übersehen, dass die Masken und Felder in falscher Reihenfolge angeordnet waren und die Software unverständliche Fehlermeldungen erzeugte. Wichtige Funktionen waren auf unterschiedlichen Ebenen verteilt und erleichternde Tastenkombinationen und Kurzbefehle gab es nicht. Herr Altmann merkte, dass er sich stark konzentrieren und sich viele Informationen gleichzeitig merken musste. Ihm fiel immer wieder auf, dass er seine Eingaben in Feldern eintragen musste, die an anderen Stellen platziert waren, als er erwarten würde. Der Druck der Tickets musste über ein zusätzliches Druckprogramm stattfinden. Das zusätzlich zu bedienende Druckprogramm bedeutete einen erheblichen Mehraufwand, da die Daten nicht automatisch vom Druckprogramm übernommen wurden, sondern neu eingegeben werden mussten. Dieser zeitliche Mehraufwand stand im Widerspruch zu Herrn Altmanns Ziel, seine Kundschaft möglichst schnell zu bedienen.



#### Was passiert da eigentlich?

Hat jemand das Ziel, unter Zuhilfenahme von Software und IT einen Arbeitsauftrag zu erledigen, reguliert diese Person ihr Handeln. Sie versucht, die ihr gestellten Aufgaben und Herausforderungen zu bewältigen, indem sie ihre Handlungen an die Situation und Gegebenheiten anpasst. Wenn Software sie bei dieser Handlungsregulation nicht unterstützt – sondern im Gegenteil behindert – spricht man von **Regulationsbehinderungen**. Das sind zum Beispiel:

- Variierende Informations- und Funktionsstrukturen, die eine ständige Suche und Umorientierung erfordern.
- Zeitliche Verzögerungen, weil die Software benötigte Funktionen nicht aufweist.
- Die Verteilung von Informationen eines Arbeitsschrittes auf unterschiedlichen Masken, worunter die Orientierung leidet.
- Fehlende, falsche oder unverständliche Fehlermeldungen.
- Umständliche Wechsel zwischen unterschiedlichen Anwendungen und fehlende Möglichkeiten der Zwischenspeicherung oder Korrektur von Eingaben.



Immer, wenn zusätzlicher Aufwand nötig ist oder eine Aufgabe nicht erledigt werden kann, ist die Gefahr psychischer Fehlbeanspruchung besonders groß. Typische **psychische Belastungsfaktoren** sind:

- Hohe Belastungen des Gedächtnisses und hohe Anforderungen an die Aufmerksamkeit und Konzentration.
- Zeitdruck durch Verzögerungen bei der Aufgabenerledigung, die auf die Eigenschaften der Software zurückzuführen sind.
- Orientierungslosigkeit durch fehlende Strukturierung und Übersichtlichkeit der Software.
- Unerwartete und widersprüchliche Arbeitsanforderungen durch Software, deren Funktionsweise bestimmten Arbeitsabläufen oder Handlungszielen widerspricht.

Fortsetzung >

**Die neue Software** hilft Herrn Altmann bei der zeitlichen Orientierung, indem ein roter Balken am oberen Bildschirmrand den Zeitprozess bis zum endgültigen Ausdruck der Fahrkarten symbolisiert. Sie liefert zudem sehr schnelle und genaue Informationen. Durch ein integriertes Druckprogramm können nun zügig Tickets gedruckt werden. Die Masken und Felder sind in richtiger Reihenfolge angeordnet, sodass Herr Altmann nicht zwischen unterschiedlichen Ebenen und Masken hin und her springen muss. Das führt auch dazu, dass er sich weniger Informationen merken muss.

Da Herr Altmann morgen in den Urlaub fährt, hat er zudem die Aufgabe seine **Urlaubsvertretung (Herrn Unruh)** einzuarbeiten. Die neue Software führt bei Herrn Unruh zu einem ganz anderen Beanspruchungserleben: Er ist generell eine ungeduldige Person und hat einen ausgeprägten Sinn für Ordnung und Struktur. Er gerät schnell unter Druck, wenn er mit neuen Aufgaben und Inhalten konfrontiert wird. Zudem kann er sich nicht so viele Dinge gleichzeitig merken und muss bei den Fahrgästen mehrfach nachfragen, wenn Informationen auf unterschiedlichen Masken einzugeben sind.

Daher ist Herr Altmann sehr froh, dass die neue Software einen Anfänger- und einen Expertenmodus zur Verfügung stellt. Nachdem der Expertenmodus zu negativem Beanspruchungserleben bei Herrn Unruh führte, entschied Herr Altmann sich dafür, seine Urlaubsvertretung in den Anfängermodus einzuarbeiten. Es zeigte sich schnell, dass dies eine gute Entscheidung war. Die Software unterstützte Herrn Unruh die Fahrgäste zu beraten und ermöglichte ihm, sich neben Grundfunktionen langsam in alle Optionen der Software einzuarbeiten.



#### Was passiert da eigentlich?

Je nachdem, welche individuellen Fähigkeiten und Fertigkeiten eine Benutzerin oder ein Benutzer hat und welche Ressourcen ihr bzw. ihm zur Verfügung stehen, um mit psychischer Belastung umzugehen, führt diese zu individueller Beanspruchung.

Die Folgen dieser **psychischen Beanspruchung** sind zum Beispiel:

- Ermüdung, Stress, Frustrationserleben, herabgesetzte Aufmerksamkeit und Reaktionsfähigkeit, Interessenlosigkeit, Gefühle der Langeweile
- Spaß und Freude bei der Aufgabenerledigung, Leistungssteigerung und Motivation, geistiges Wachstum, persönliche Entwicklung und Entfaltung, Qualifikationserwerb

Das Beispiel zeigt: Psychische Belastung und Regulationsbehinderungen sind für die ergonomische Softwareentwicklung zu berücksichtigende Größen.

Zusammenfassend bedeutet dies:

- Belastungen können negative und positive Beanspruchungsfolgen haben. Entweder sie motivieren die Benutzerinnen und Benutzer und führen zu einem positiven Erlebnis – oder sie behindern sie bei der Aufgabenerledigung.
- Software sollte so entwickelt werden, dass keine kognitive Überforderung der

Wahrnehmungs und Gedächtnisprozesse der Benutzerinnen und Benutzer stattfindet.

- Benutzerinnen und Benutzer sollten gezielt in die Anwendung neuer Software eingearbeitet werden und die Implementierung von neuer Software im Unternehmen sollte immer als Gestaltung von Arbeitsprozessen und Abläufen, Aufgabe (mehr dazu in Kapitel 4.4) verstanden werden.

#### 4.2 Software und Kognition

Um den kognitiven (geistigen) Eigenschaften der Benutzerin oder des Benutzers gerecht zu werden und Fehler durch Unaufmerksamkeit zu vermeiden, sollte Software zum Beispiel

- nicht zu viele Informationen gleichzeitig bereitstellen,
- die Gedächtnisspanne der Benutzerinnen und Benutzer berücksichtigen,
- konkrete Handlungshinweise geben,
- durch markante Strukturen den Blick lenken,
- in kritischen Situationen akustische Signalmeldungen liefern.

Metaphern (sprachliche Bilder) werden gezielt in der Mensch-Computer-Interaktion eingesetzt, um Softwarebenutzerinnen und Softwarebenutzer zu entlasten. Bei der Entwicklung von Metaphern werden die Erfahrungen, Erwartungen und Denkmuster der Benutzerinnen und Benutzer auf die Software übertragen. Die Vertrautheit der Benutzerinnen und Benutzer mit der Metapher erleichtert nicht nur die Orientierung und Navigation. Es werden auch unbekannte und abstrakte Inhalte und Anwendungen schneller überschaubar, erlernbar und handhabbar.

Ein Beispiel sind die Desktop-Metaphern, deren Aussehen an reale Objekte erinnern und die wie ein virtueller Schreibtisch organisiert sind. Der „Mülleimer“, „Aktenordner“, „Dokumentenstapel“ und „Drucker“ sind Beispiele, die fast jedes moderne Betriebssystem verwendet.

#### 4.3 Software und Emotion

Die Funktionalität einer Software alleine reicht nicht aus, um einen dauerhaft motivierenden und gesundheitsförderlichen Arbeitsprozess durch Software zu ermöglichen. Hinzukommen müssen anregende, motivierende und positiv-emotionale Benutzerinnen- und Benutzererlebnisse. Dem Bedürfnis der Benutzerin oder des Benutzers, sich autonom, kompetent und selbstbewusst bei der Erledigung von Aufgaben zu fühlen, sollte die Software entgegen kommen. Denn neben der Wirkung von positiv-emotionalen Erlebnissen konnte zudem belegt werden, dass negative Emotionen belastende Auswirkungen auf kognitive Prozesse haben.

Die emotionale Situation der Benutzerin oder des Benutzers entsteht unter anderem aus der Befriedigung psychischer

Bedürfnisse. Daher sollten folgende Punkte bei der Gestaltung und Implementierung von Software berücksichtigt werden:

- Jeder Mensch hat das psychische Bedürfnis nach Autonomie und Kompetenz, das durch Software nicht eingeschränkt werden sollte.
- Eine gewisse Selbststeuerung, eigene Entscheidungen treffen zu können, und das Gefühl, den Arbeitsprozess kontrollieren zu können, sollte die Software ermöglichen, um negative Emotionen zu vermeiden.
- Emotionen sind ein Teil der Beanspruchungsreaktion und sollten im Gesamtprozess der Softwareentwicklung, Implementierung und Anwendung berücksichtigt werden.

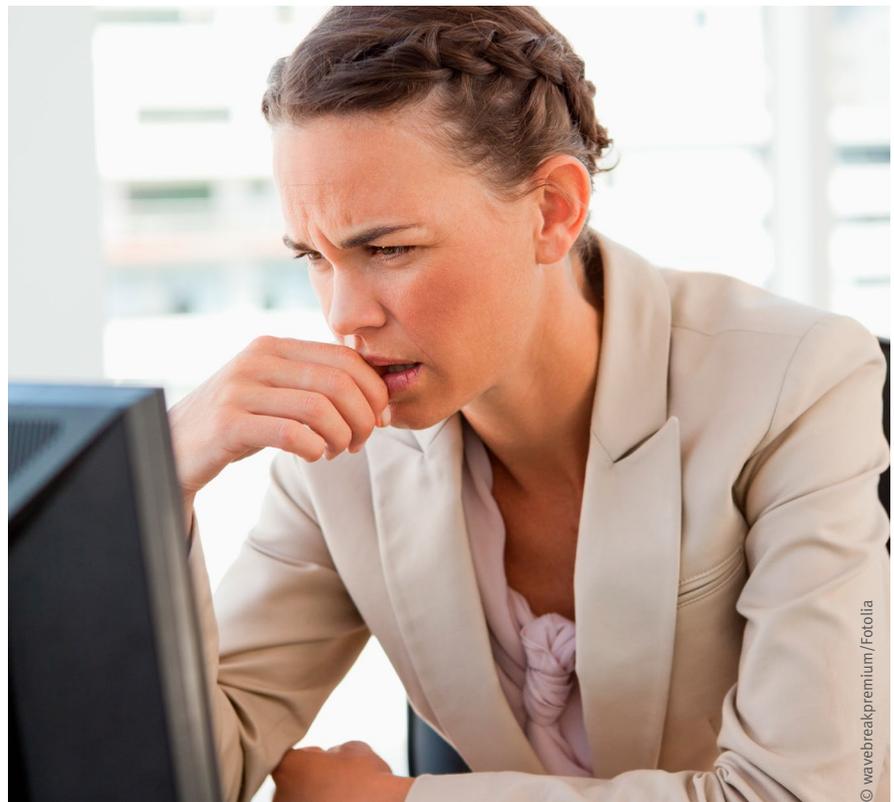


Abb. 9 Eine Benutzerin/ein Benutzer, die oder der Ärger entwickelt, ist weniger in der Lage, die Aufmerksamkeit und Konzentration aufrecht zu erhalten oder sich Dinge gut zu merken.

#### 4.4 Arbeitsgestaltung im Prozess der Softwareimplementierung

Die Zufriedenheit von Benutzerinnen und Benutzern ergibt sich auch aus den Merkmalen der Software und aus der Art und Weise, wie deren Arbeitsaufgaben gestaltet sind. Deshalb ist die Funktionsaufteilung zwischen Benutzerin oder Benutzer und Software klar zu definieren. Darüber hinaus ist bei der Einführung von Software eine bewusste Strukturierung von Aufgaben, Arbeitsprozessen und dem individuellen, sozialen und räumlichen Arbeitsumfeld nötig. Die beiden folgenden Punkte fördern somit ein erfolgreiches Arbeiten an IT-gestützten Arbeitsplätzen und bilden die Grundlage für die Gestaltung von gesundheitsförderlicher Arbeit.

##### 1. Die Gestaltung von Arbeitsaufgaben

durch softwaregestützte IT-Arbeit berücksichtigt idealerweise folgende Aspekte der Arbeitsorganisation:

- Software sollte den Benutzerinnen und Benutzern Handlungsspielräume zugestehen. Die Reihenfolge, das Tempo und die Vorgehensweise bei der Erledigung von Aufgaben sollte man individuell bestimmen können.
- Benutzerinnen und Benutzer sollten durch die Software ausreichend Rückmeldung über die jeweiligen Handlungen erhalten und Systemzustände sollten klar erkennbar sein.
- Die Ganzheitlichkeit eines Arbeitsprozesses sollte immer erkennbar sein, sodass Benutzerinnen und Benutzer der Software nie das Ziel aus den Augen verlieren und nicht durch zu kleine Einzelschritte das „große Ganze“ vernachlässigen.
- Die Fähigkeiten und Fertigkeiten der Benutzerinnen und Benutzer sollten durch Software berücksichtigt werden und eine Mischung unterschiedlicher Tätigkeiten sollte stattfinden.

##### 2. Die Gestaltung von Arbeitsabläufen und Unternehmensprozessen

hat zum Ziel, den Mensch, die Technik und die Organisation sinnvoll zu verzahnen und bedient sich dabei verschiedener Methoden des Projektmanagements, der Organisationsgestaltung und der Softwareentwicklung. Bei der Einführung neuer Software sollte man unter anderem folgende Handlungsfelder berücksichtigen (Janneck & Adelberger 2012):

- Während der Entscheidungsfindung und Planung sollten Projektmanagementstrukturen eingerichtet, Stakeholder (Projektbeteiligte) identifiziert, Projektpläne erstellt und Ressourcen bereitgestellt werden (z. B. Projektstrukturpläne, SMART-Ziele, Projektmanagement-Methoden).
- Die Softwareimplementierung sollte auf technischer und organisatorischer Ebene geplant werden. Die Partizipation (Mitbestimmung) der Benutzerinnen und Benutzer sollte bewusst in die Planung integriert werden (z. B. Usability Engineering, Anforderungsermittlungsmethoden wie Prototypen, Szenarien und Workshops).
- Der Anwendungsstart sollte durch die Bereitstellung der technischen Infrastruktur und auf organisatorischer Ebene durch Schulungen, Information, Struktur- und Prozessveränderung unterstützt werden (z. B. Testbetrieb, Informationspolitik, Change-Management).
- Den Benutzerinnen und Benutzern sollten Supportstrukturen zur Verfügung stehen (z. B. Handbücher, Leitfäden, Hotlines und Schulungen).
- Ein transparenter und effizienter Austausch durch Kommunikationsstrukturen ist zu definieren und zu ermöglichen (z. B. Analyse der Kommunikationsstrukturen, -Knoten, und Kanäle).
- Die Erhebung von Prüfdaten und deren Bewertung sollte in Form einer begleitenden Evaluation und Qualitätssicherung stattfinden (z. B. Feedbacksysteme, Qualitätsmanagement).

- Die Software sollte kontinuierlich weiterentwickelt und an die sich verändernden Arbeitsanforderungen angepasst werden (z. B. Evaluations- und Usability Engineering Methoden).

Sind diese Aspekte berücksichtigt, ist eine umfassende und solide Einführung von Software möglich. Negative Beanspruchungsfolgen können verhindert und das Erleben und Verhalten von Benutzerinnen und Benutzern im Umgang mit der Software positiv beeinflusst werden.

# 5 Software und Bildschirm

Wie hängen Software und Bildschirm zusammen und wie beeinflusst der Bildschirm die Darstellung der Software? Diese Fragen werden in diesem Kapitel beantwortet.

Die in Unternehmen eingesetzte Software hat in der Regel eine durchschnittliche Lebensdauer von etwa zehn bis zwölf Jahren. Sie wird zwar in Form neuer Versionen (Releases) immer wieder aktualisiert, bleibt aber in ihrer fundamentalen Struktur erhalten. Bildschirme und auch Rechner haben in der Regel eine durchschnittliche Lebensdauer von etwa drei bis sechs Jahren. Somit weist die Software einen deutlich längeren Lebenszyklus auf und sollte deshalb an unterschiedliche Hardwarespezifikationen anpassbar gestaltet sein. Eine Anpassbarkeit der Informationsdarstellung auf dem Bildschirm erreicht man z. B. durch eine Programmierung mit vektorbasierten oder auch skalierbaren Darstellungen. Idealerweise passt die Software die Informationsdarstellung automatisch an die jeweilige Bildschirmgröße und seine Auflösung an (sogenanntes „Responsive Design“). Vermieden werden sollten pixelorientierte oder auch absolute Darstellungen, da sie beim Wechsel auf neue, höher auflösende Bildschirme, häufig zu verkleinerten Darstellungen bzw. schlechter Lesbarkeit führen.

Dazu folgendes Beispiel (siehe hierzu Tabelle „Typische, am Markt verfügbare Auflösungen“ im Anhang 1):

Eine Software wurde vor einigen Jahren für einen „normal-formatigen“ (4:3) 15"-Bildschirm mit einer Auflösung von 1024 x 768 Pixel mit absoluten Dimensionen (z. B. Punkten und Pixeln) programmiert. Vom Unternehmen wurden neue Bildschirme angeschafft, wobei mit Hinblick auf das zukünftige papierlose Dokumentenmanagement moderne breitbildformatige (16:9) 27"-Bildschirme mit einer Auflösung von 1920 x 1200 Pixel ausgewählt wurden.

Beim Anschluss der neuen Bildschirme ergibt sich folgende Darstellung auf dem Bildschirm (siehe Abbildung 10):

Abbildung 10 zeigt, dass nur ein Teil der verfügbaren Bildschirmfläche genutzt wird. Immerhin bleibt in diesem Beispiel die absolute Zeichenhöhe erhalten bzw. die Zeichen werden sogar etwas größer auf dem neuen Bildschirm angezeigt, da die Pixelgröße von 0,294 mm x 0,294 mm auf 0,311 mm x 0,311 mm steigt.

Auszug aus der Tabelle „Typische, am Markt verfügbare Auflösungen“ (siehe Anhang 1):

Bildschirmparameter für LCD-Bildschirme

	Diagonale		Anzeigefläche mm	Physikalische Auflösung Pixel	Pixelgröße mm
	Zoll	mm			
	15	381	304 x 228	1024 x 768	0,294 x 0,294
<b>Bild 1</b>	27	686	597 x 336	1920 x 1080	0,311 x 0,311
<b>Bild 2</b>	27	686	596 x 336	2560 x 1440	0,233 x 0,233

Erweitert man das Beispiel wie folgt, so lässt sich ein weiterer Effekt veranschaulichen: Speziell der EDV-Bereich wird mit besonders hochauflösenden (16:9) 27"-Bildschirmen mit einer Auflösung von 2560 x 1440 Pixel ausgestattet.



Abb. 10 Darstellung nach Anschluss eines neuen Bildschirms

Die absolute Pixelgröße verringert sich, wodurch die Darstellung von diagonalen Linien und Kurven zwar glatter anmutet. Allerdings verkleinert sich auch – in Verbindung mit einer pixelorientierten Programmierung der Software – die Darstellung der Zeichen:



Auch dieser Effekt führt zu einer Verschlechterung der Lesbarkeit.

Vektorgrafiken basieren, anders als Rastergrafiken, nicht auf einem Pixelraster, in dem jedem Bildpunkt ein Farbwert zugeordnet ist, sondern auf einer Bildbeschreibung, die die Objekte aus denen das Bild aufgebaut ist, exakt definiert. So kann beispielsweise ein Kreis in einer Vektorgrafik über Lage des Mittelpunktes, Radius, Linienstärke und Farbe vollständig beschrieben werden und ist damit im Gegensatz zur Rastergrafik stufenlos und verlustfrei zu skalieren.

Deshalb ist es gerade beim Beschaffungsprozess von großer Wichtigkeit, dass die Darstellung der Informationen durch die Software möglichst frei skalierbar zu gestalten ist, damit sie unterschiedliche Anzeigeräte optimal unterstützen kann.

# 6 Softwaretypen

Software wird heute nicht mehr nur auf Computern, sondern in vielen anderen Geräten angewandt und verarbeitet. In diesem Kapitel werden die gängigsten Softwaretypen und deren wichtigste Vertreter vorgestellt und erklärt.

Neben Computern wird Software auch in Mobiltelefonen, Navigationsgeräten und Fernsehern usw. eingesetzt. Software steht als Begriff für ausführbare Programme und ihre zugehörigen Daten, wie z. B. Konfigurationsdateien oder Schriftartdateien. Sie kann auf verschiedenen Medien gespeichert werden, interne Berechnungen ablaufen lassen, Steuerungen von Abläufen und externen Geräten durchführen sowie auf Ausgabegeräten Grafiken und Texte anzeigen. Software steuert Geräte, berechnet Daten, erzeugt Grafiken und erledigt viele andere Aufgaben. Sie unterscheidet sich grundsätzlich von den physischen Bestandteilen der softwareverarbeitenden Geräte. Diese als Hardware bezeichneten softwareverarbeitenden Geräte, ihre Baugruppen, Einzelteile und Peripheriegeräte sind z. B. Computer, Notebook, Prozessor, Hauptspeicher, Datenträger, Drucker und vieles mehr.

Software unterscheidet man prinzipiell in Systemsoftware, systemnahe Software und Anwendungssoftware. Dabei werden Softwaretypen nach ihrer Nähe zur Hardware bzw. zur Verwendung durch Benutzerinnen und Benutzer getrennt.

Systemsoftware ist für die maßgeblichen Funktionen eines Computers erforderlich. Als Systemsoftware wird die Gesamtheit aller Programme und Dateien bezeichnet, die alle Abläufe bei Betrieb eines softwareverarbeitenden Gerätes steuern. Sie stellt eine Verbindung zur Hardware her und steuert die Verwendung der Hardwareressourcen durch die Anwendungssoftware. Sie verwaltet sowohl die internen als auch die externen Hardwarekomponenten. Die Systemsoftware kommuniziert zu diesem Zweck mit den einzelnen Hardwarekomponenten. Arten von Systemsoftware sind z. B. Betriebssysteme, Firmware und Gerätetreiber.

Systemnahe Software wird in Bereichen zwischen dem Betriebssystem und der Anwendungssoftware eingesetzt. Als systemnahe Software bezeichnet man z. B. Dienstprogramme, Datenbank-Verwaltungswerkzeuge und Programmierwerkzeuge.

Anwendungssoftware sind ausführbare Programme, die nützliche oder gewünschte Funktionen unterstützen, die zur Bearbeitung von Aufgaben dienen. Anwendungssoftware untergliedert sich in Standardsoftware, Spezialsoftware und Individualsoftware.

Standardsoftware deckt einen klar definierten Anwendungsbereich ab. Sie kann als fertiges Produkt erworben werden. Bekannte Beispiele für Standardsoftware sind Office-Pakete. In diesen findet sich eine Zusammenstellung von Standardsoftware, die Arbeiten im Büro (z. B. Textverarbeitung, Tabellenkalkulation) unterstützt.

Spezialsoftware wird für bestimmte Aufgaben wie z. B. Patientenverwaltung, Lohnbuchhaltung, Maschinenkonstruktion konzipiert und berücksichtigt somit Erfordernisse und Wünsche einzelner Anwendergruppen.

Individualsoftware wird den Anforderungen der Kundschaft entsprechend maßgeschneidert erstellt oder wesentlich an deren Bedürfnisse angepasst. Im Vergleich zur Standardsoftware entstehen verhältnismäßig hohe Kosten für die Entwicklung, Pflege und Wartung von Individualsoftware. Gleichzeitig ist es möglich, alle Anforderungen an die Software umzusetzen und somit auch die Erfordernisse der Anwenderin bzw. des Anwenders an die Softwareergonomie in das Lastenheft zu überführen.

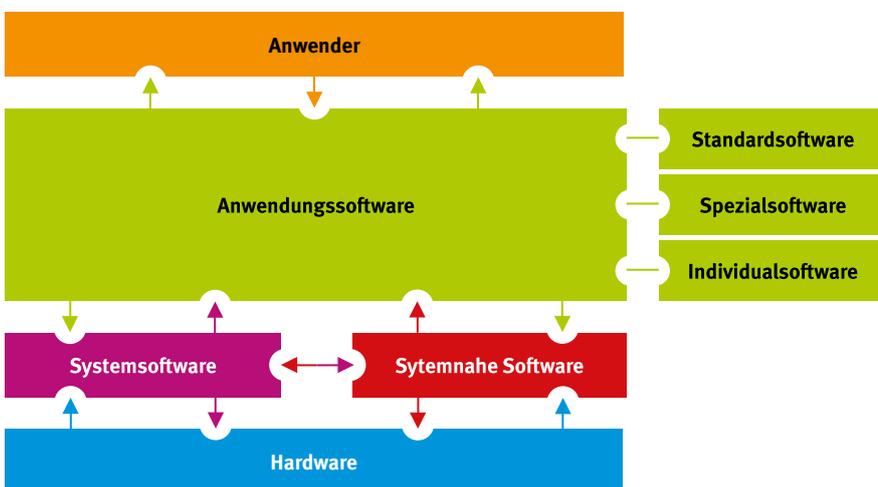


Abb. 11 Zusammenhang zwischen Anwenderin bzw. Anwender, Soft- und Hardware

# 7 Rechtliche Aspekte

In diesem Kapitel werden rechtliche Vorgaben vorgestellt, die zur Gestaltung von Software nach ergonomischen Grundsätzen verpflichten. Zudem werden konkrete Hinweise auf rechtlich relevante ergonomische Grundsätze der Gestaltung von Software gegeben und erläutert, wann und wie sie zu berücksichtigen sind.

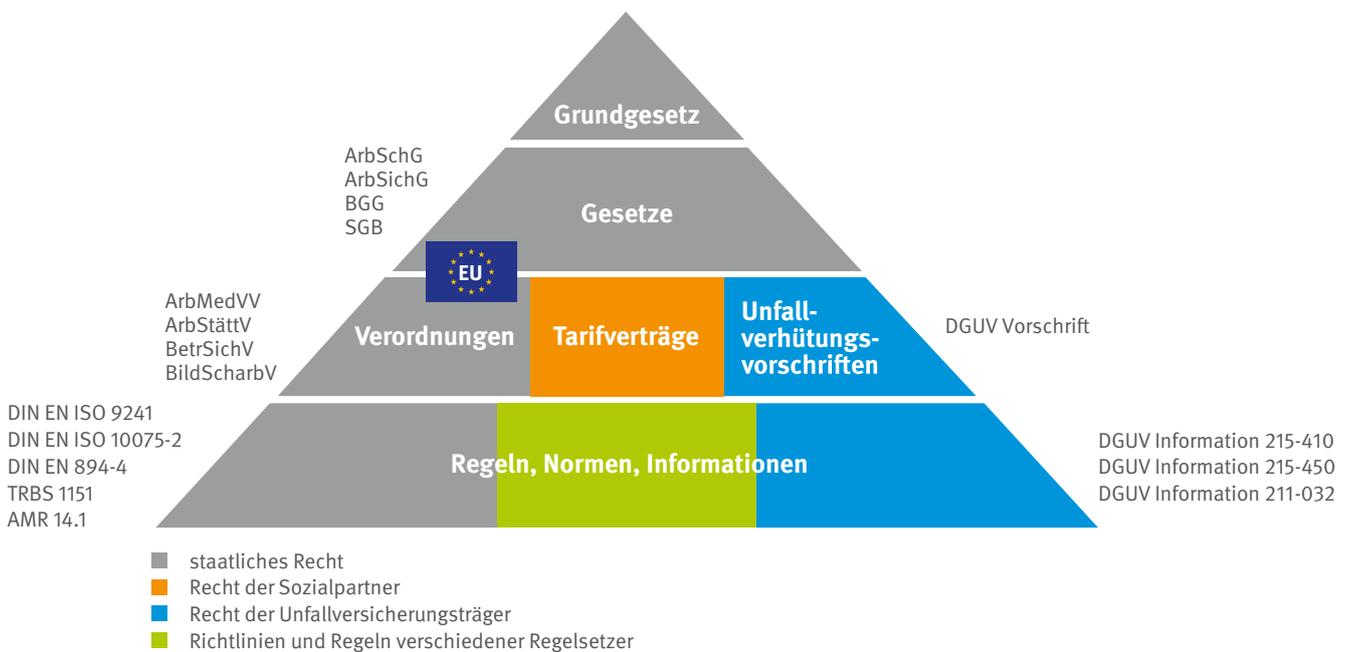


Abb. 12 Pyramide des Arbeitsschutzrechts mit Bezug zur Softwareergonomie

Softwareergonomie und die Anforderungen an eine ergonomische Gestaltung sind im Arbeitsschutzrecht an vielen verschiedenen Stellen verankert. Vorgaben aus Gesetzen und Verordnungen (oberer Teil der Abbildung 12) sind rechtlich bindend und zu berücksichtigen. Die Inhalte sind meist als Schutzziele formuliert. Um diese Ziele erreichen zu können, leiten Regeln, Normen, Richtlinien, Informationen und Handlungshilfen an und zeigen Lösungswege zur Umsetzung und Einhaltung von rechtlichen Vorgaben auf (unterer Teil der Abbildung 12). Einige dieser Dokumente werden zur inhaltlichen Auslegung von Gesetzen, Verordnungen und Vorschriften herangezogen. So sind z. B. nach der Bildschirmarbeitsverordnung

(BildscharbV), Anhang 20 die „Grundsätze der Ergonomie [...] insbesondere auf die Verarbeitung von Informationen durch den Menschen anzuwenden.“

Daher werden z. B.

- Anforderungen an die Informationsdarstellung von Software aus DIN EN ISO 9241-12,
- Gefährdungen an der Schnittstelle Mensch-Arbeitsmittel der TRBS 1151 in Verbindung mit der Arbeitsstättenverordnung (ArbStättV) rechtlich relevant. Bei Bedarf kann alternativ ein vergleichbares Schutzniveau durch eine dokumentierte Gefährdungsbeurteilung nachgewiesen werden.

Der rechtliche Rahmen für Arbeitsschutzaktivitäten im Bereich von Büro und Verwaltung ist in der DGUV Information 215-410 als Leitfaden für die Gestaltung von Bildschirm- und Büroarbeitsplätzen dargestellt und erläutert. Folgende, für die Büroarbeit relevante Gesetze, Verordnungen und Vorschriften des Arbeitsschutzrechts können ebenfalls im Internet gesichtet werden:

- Arbeitsschutzgesetz (ArbSchG)
- Gesetz über Betriebsärzte, Sicherheitsingenieure und andere Fachkräfte für Arbeitssicherheit (ASiG)
- Arbeitsstättenverordnung (ArbStättV)
- Bildschirmarbeitsverordnung (BildscharbV)

- Behindertengleichstellungsgesetz (BGG)
- Betriebssicherheitsverordnung (BetrSichV)
- Barrierefreie-Informationstechnik-Verordnung (BITV 2.0)
- Unfallverhütungsvorschriften (z. B. DGUV-Vorschrift 1)
- Verordnung zur arbeitsmedizinischen Vorsorge (ArbMedVV)

Bei der Entwicklung, Beschaffung, Nutzung oder Änderung von Software (Anhang 20 der BildscharbV, DGUV Information 215-410) sind verschiedene rechtliche Vorgaben und Empfehlungen zu beachten.

So sind etwa nach § 4 des ArbSchG bei der Entwicklung von Software z. B. der Stand der Technik und Arbeitsmedizin sowie arbeitswissenschaftliche Erkenntnisse der Gestaltung zu berücksichtigen. Konkretisiert wird dies für den Prozess der Softwareentwicklung z. B. in DIN EN ISO 9241-210. Bereits bei der Entwicklung ist darauf zu achten, dass vom Einsatz des Arbeitsmittels Software nach § 4 des ArbSchG oder § 4 der BetrSichV keine Gefährdungen und Beeinträchtigungen ausgehen. Sie können z. B. durch nicht leserliche Zeichenhöhen auftreten oder durch unvollständige Programmierung sowie durch Lage und Anordnung von Bildschirmanzeigen mit dem Risiko für Fehlfunktionen und Sicherheitslücken (TRBS 1151, DIN EN 894-4) entstehen.

Bei der Beschaffung von Software als Arbeitsmittel ist aus rechtlicher Sicht darauf zu achten, dass der Lieferant schriftlich darauf hingewiesen wird, einschlägige Anforderungen für Sicherheit und Gesundheitsschutz einzuhalten (§ 5 der DGUV Vorschrift 1). Weiterhin sollten bei der Beschaffung von Software als Arbeitsmittel nach §§ 3 und 6 des ASiG die Betriebsärztin oder der Betriebsarzt bzw. die Fachkraft für Arbeitssicherheit einbezogen werden. Die zu beschaffende Software muss an die auszuführende

Arbeitsaufgabe angepasst sein (Anhang der BildscharbV i. V. m. DIN EN 29241-2) und macht eine regelmäßige Unterweisung der Beschäftigten zum Arbeitsmittel Software erforderlich (§ 12 des ArbSchG).

Bei der Nutzung von Software an Bildschirmgeräten sind Vorsorgeuntersuchungen anzubieten (Anhang der ArbMedVV; AMR 14.1). Das Arbeitsmittel Software eignet sich nach § 3 der BetrSichV erst dann zur Nutzung, wenn „auch die ergonomischen Zusammenhänge zwischen Arbeitsplatz, Arbeitsmittel, Arbeitsorganisation, Arbeitsablauf und Arbeitsaufgabe“ berücksichtigt sind, die z. B. in DIN EN ISO 6385 beschrieben und mit verschiedenen Teilen aus der DIN EN ISO 9241 umgesetzt werden können. Anforderungen, Arbeitsmittel auch für Beschäftigte mit Behinderungen zu gestalten, leiten sich aus § 11 des BGG i. V. m. der BITV 2.0 ab und werden z. B. durch die Leitlinien für die Zugänglichkeit von Software (DIN EN ISO 9241-171) konkretisiert.

Schließlich fordern § 3 des ArbSchG und § 3 BetrSichV nach jeder Veränderung am Arbeitsmittel Software eine erneute Gefährdungsbeurteilung, in die auch die Grundsätze der Ergonomie auf die Verarbeitung von Informationen durch den Menschen einbezogen werden müssen (Anhang 20 der BildscharbV i. V. m. DIN EN ISO 10075-2). Die veränderte Software muss den jeweils aktuell auszuführenden Arbeitsaufgaben angepasst sein (Anhang 20 der BildscharbV i. V. m. DIN EN 29241-2).

Aus dem Arbeitsschutzrecht leitet sich ab, dass Software ergonomisch gestaltet sein muss, um Gefährdungen und Beeinträchtigungen der Benutzerinnen und Benutzer bei der Bearbeitung ihrer Aufgaben möglichst zu vermeiden. Die vorliegende DGUV Information 215-450 zeigt Lösungswege zur Umsetzung und Einhaltung von rechtlichen Vorgaben auf und hilft, einen Prozess der Gestaltung zu

organisieren, Anforderungen an die Gestaltung aufzustellen und eine ergonomische Gestaltung umzusetzen.

# 8 Informationsdesign

Dieses Kapitel beschäftigt sich mit der Darstellung von Informationen auf dem Bildschirm und berücksichtigt dabei gleichzeitig die Informationsverarbeitung durch den Menschen. Themen sind z. B. Farbwahl, Zeichensätze, Informationsstruktur, Textgestaltung, Zeichendarstellung.

## 8.1 Gestaltungsgrundsätze

Gestaltungsgrundsätze können verwendet werden, um das Informationsdesign in einem ersten Schritt allgemein zu prüfen:

- Information **akkurat** (genau) darstellen.  
Beispiel: Ein Drucker-Icon ist so gestaltet, dass es auf Anhieb als Drucker erkannt werden kann.
- Information **konsistent** darstellen, Texte konsistent formulieren.  
Beispiel: Die Reihenfolge der Adressfelder ist immer gleich in der gesamten Anwendung. Die Funktion „Speichern und Fenster schließen“ ist immer gleich benannt. Meldungen werden in einer konsequenten Form präsentiert.
- Information **verständlich** und unzweideutig darstellen.  
Beispiel: Die Beschriftung der Felder entspricht der Schreibweise und den Benennungen der Benutzerin bzw. des Benutzers und der Aufgabe und enthält keine zweideutigen Abkürzungen.
- Information für die Benutzerinnen und Benutzer **auffindbar** und **erkennbar** gestalten.  
Beispiel: Wichtige zusammengehörige Felder sind als Gruppe in einer Maske aufzufinden und nicht an verschiedenen Stellen in einer Maske oder auf andere Masken verteilt.
- Information voneinander **unterscheidbar** gestalten.  
Beispiel: Symbole sind deutlich voneinander unterscheidbar gestaltet und unterscheiden sich nicht nur in wenigen Pixeln.
- Textinformationen **leicht lesbar darstellen**.  
Beispiel: Die Schrift ist groß genug, sodass sie für die Anwenderin oder den Anwender ohne Probleme lesbar ist

(Schrift-Beispiel). Auf kleine Serifenschriften wird verzichtet (Schrift-Beispiel).

- Information **erwartungskonform** und den Nutzergewohnheiten entsprechend präsentieren.  
Beispiel: Adressfelder werden immer in einer der Arbeitsaufgabe üblichen Reihenfolge angeordnet. Das Datumsformat entspricht den Erwartungen der Benutzerinnen und Benutzer.
- Informationen sind so anzuordnen und darzustellen, dass sie visuelle **Suchprozesse** unterstützt.  
Beispiel: Informationen werden entsprechend der Arbeitsaufgabe gruppiert.

## 8.2 Anordnung von Elementen und Informationen

### 8.2.1 Allgemeiner Maskenaufbau

Bildschirmmasken sollten so gestaltet werden, dass sie den Anzeigebereich optimal nutzen. Es sollte kein zusätzlicher Navigations- oder Scrollaufwand bei ausreichend großer Schrift entstehen. Die benötigten Informationen sollten aufgabenangemessen und konsistent in Masken- oder Seitenbereichen platziert werden:

Die Umsetzung von Gestaltungsregeln führen zu einem Maskenbild, in dem es einfach ist, sich zu orientieren und Informationen schnell visuell aufzufinden (siehe Abbildung 13):

- die Elemente besitzen immer die gleiche Höhe,
- Abstände in und zwischen den Elementen sind konsistent,
- die Anzahl der vertikalen und horizontalen Fluchtlinien ist minimiert.

Das (negative) Beispiel (Abbildung 14) zeigt, dass die Orientierung bei unterschiedlichen Abständen und fehlenden Fluchtlinien schwer fällt. Die Maske könnte besser so aussehen (Abbildung 15):

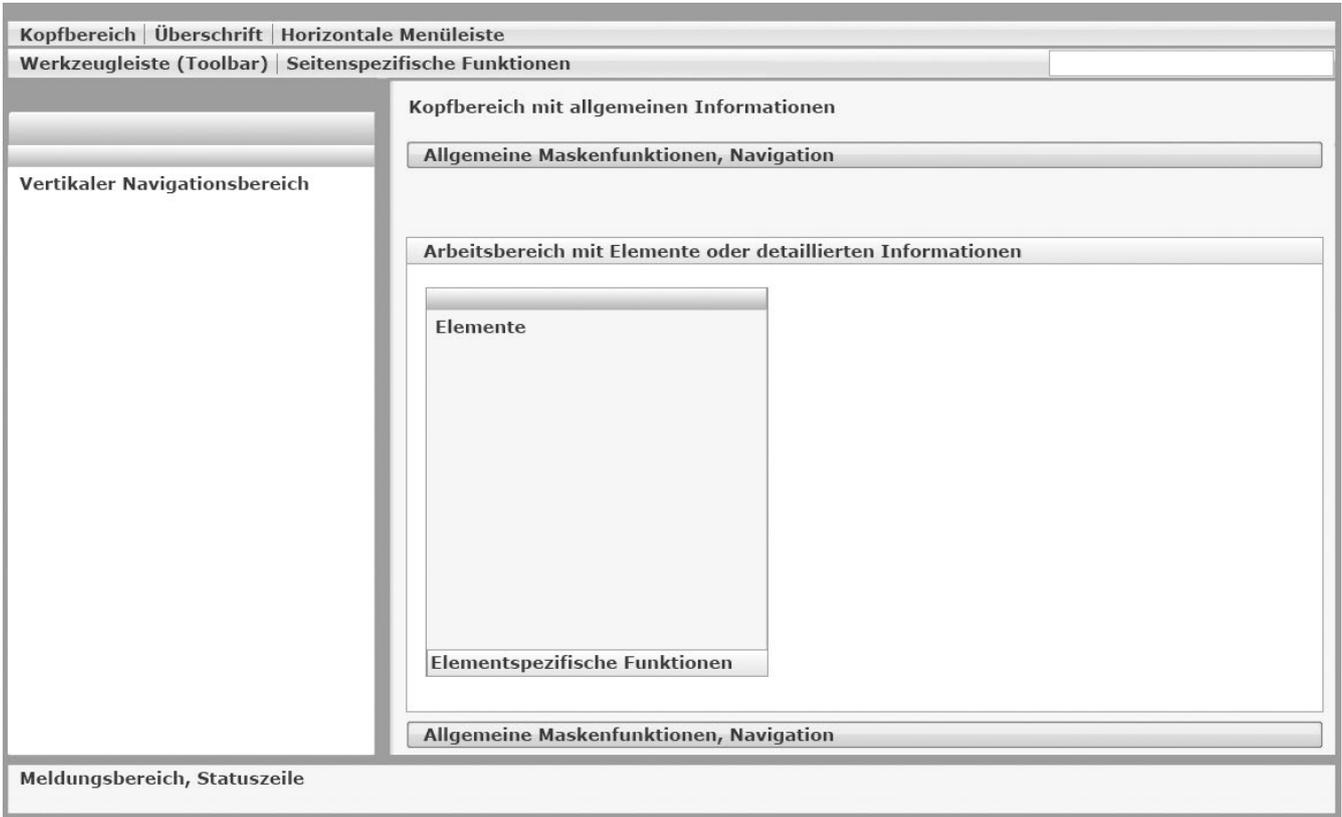


Abb. 13 Typischer Aufbau des Anzeigebereichs einer Anwendung mit verschiedenen Unterbereichen

Das Formular zeigt verschiedene Eingabefelder und Kontrollen:

- Name:** Schneider
- Straße:** Hauptstraße 12
- Vorname:** (leeres Feld)
- PLZ u. Ort:** 12345 | Großstadt
- Führerschein vorhanden
- Geschäftsadresse
- Personalausweis vorhanden
- Reisepass vorhanden

Abb. 14 Unterschiedliche Abstände und Höhen

Das Formular zeigt die gleiche Struktur wie Abb. 14, jedoch mit roten Hilfslinien zur Visualisierung der Konsistenz:

- Vorname:** (leeres Feld)
- Name:** Schneider
- Straße:** Hauptstraße 12
- PLZ u. Ort:** 12345 | Großstadt
- Führerschein vorhanden
- Geschäftsadresse
- Personalausweis vorhanden
- Reisepass vorhanden

Abb. 15 Konsistente Anordnung und Ausrichtung der Elemente (mit Visualisierung durch Hilfslinien)

**Kundendaten**

Name:  Mobil:

Vorname:  Name Kind 1:

Straße:  Name Kind 2:

Führerschein vorhanden Vermögen:

Reisepass vorhanden PLZ u. Ort:

Personalausweis vorhanden

Netiz 1:   Bemerkung:

**Kundendaten**

Name:  Name Kind 1:

Vorname:  Name Kind 2:

Straße:

PLZ u. Ort:

Tel.:

Mobil:  Vermögen:

Führerschein vorhanden Bemerkung:

Reisepass vorhanden

Personalausweis vorhanden

Abb. 16 Unpassende Anordnung der vier angekreuzten Felder oben und passendere Platzierung der Felder unten.

### 8.2.2 Aufgabenangemessene Anordnung von Informationen

In einer Maske sind alle Elemente, die für eine Aufgabe notwendig sind zusammengefasst und aufgabenangemessen angeordnet.

Im Beispiel (Abb. 16) ist das „PLZ/Ort“-Feld abseits anderer Adressfelder wie „Straße“ platziert. Auch das „Mobil“-Feld liegt entfernt vom „Tel.“-Feld.

Die Positionierung des „PLZ/Ort“-Feldes ist für eine Aufgabe, in der man nur Adressdaten eingibt, ungeeignet. Die Anordnung würde beim Navigieren mit der Tab-Taste von Feld zu Feld einen deutlich erhöhten Navigationsaufwand erzeugen. Das visuelle Auffinden des „PLZ/Ort“-Feldes ist erschwert.

Ein weiteres Beispiel für die Wichtigkeit einer sinnvollen Ausrichtung sind vertikal am Komma ausgerichtete Zahlen, die visuell miteinander verglichen werden (Abbildung 17).

Auf der rechten Seite gewährleistet eine vertikale Ausrichtung am Komma die visuelle Vergleichbarkeit; links ist der Vergleich der Zahlen deutlich erschwert.

102,00	102,00
92,60	92,60
45,07	45,07
1345,99	1345,99
0,80	0,80

Abb. 17 Vertikale Ausrichtung von Zahlen am Komma

### 8.3 Die Gestaltgesetze

"Gestaltgesetze" sind aus visuellen Wahrnehmungsprozessen abgeleitet. Aus diesem Grund sind sie bei der Gestaltung von Benutzungsoberflächen zu berücksichtigen.

#### 8.3.1 Gesetz der Nähe

Das Gesetz der Nähe besagt, dass Elemente, die sich in räumlicher Nähe zueinander befinden, als zusammengehörig wahrgenommen werden. Um z. B. eine Beschriftung als zugehörig zu einem bestimmten Feld wahrzunehmen, muss diese in räumlicher Nähe platziert werden (Abbildung 18).

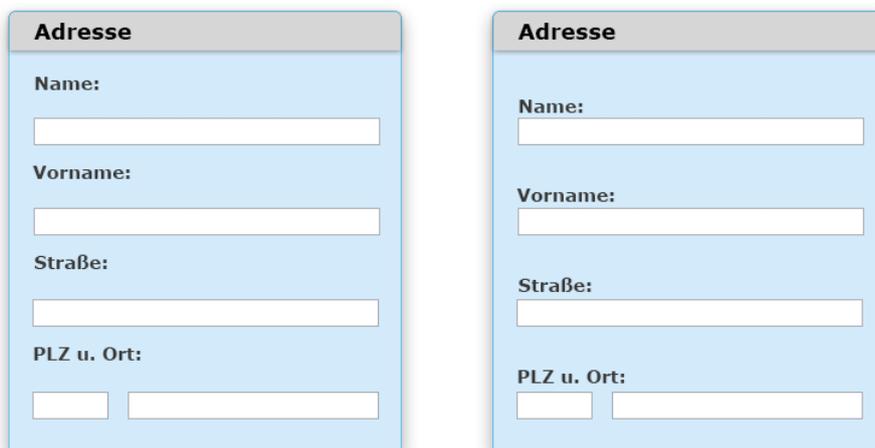


Abb. 18 Ungünstige Anordnung links, rechts Einhaltung des Gestaltgesetzes der Nähe.

Das Gesetz der Nähe wirkt hier durch den vertikalen Abstand zwischen Beschriftung und Feld. So werden die Feldbezeichnungen (auch Label genannt) „Straße“ und „PLZ u. Ort“ im linken Teil eher dem darüber liegenden Feld zugeordnet, statt – wie beabsichtigt – dem darunter liegenden. Auf der rechten Seite fällt die richtige Zuordnung leichter.

#### 8.3.2 Gesetz der Ähnlichkeit

Das Gesetz der Ähnlichkeit besagt, dass ähnliche Elemente als zusammengehörig wahrgenommen werden. Bei der Gestaltung von Masken ist somit auf die visuelle Trennung von ähnlichen, aber funktional unterschiedlichen Elementen zu achten:

Gehören die Checkboxes rechts zu den Feldern, oder bilden sie eine eigene Gruppe? Die Checkbox „Geschäftsadresse“ wird als Teil der Checkboxgruppe wahrgenommen, weil alle Elemente ähnlich sind, obwohl sie funktional nicht dazu gehört.



Abb. 19 Ungünstige Gruppierung in Bezug auf die Checkbox „Geschäftsadresse“ oben und verbesserte Anordnung unten.

Durch die verbesserte Anordnung wird visuell deutlich, dass die Checkbox „Geschäftsadresse“ funktional zu den Adressfeldern gehört, die anderen jedoch nicht (siehe Abbildung 19).

PKW-Daten Hersteller: VAW Ausstattung: Basis Polster: Leder Hubraum: 2997 cm Höchstgeschw.: 254 km/h Durchschn. verbr.: 11,5 L		Funktion: Ausk. Bearbeiter Bond Fahrzeugtyp: Perican Lackierung: Silber Glas: Getönt Leistung: 250 kW Tankinhalt: 80 L Länge: 4700 cm	
Baujahr	Preis S Benzin	Preis S Diesel	Preis Turbo
2012:	58 000 €	58 000 €	80 000 €
2013:	60 000 €	61 500 €	84 000 €
2014:	63 000 €	64 000 €	87 000 €
2015:	66 500 €	67 500 €	90 000 €

PKW-Daten: Hersteller: VAW Fahrzeugtyp: Perican Ausstattung: Basis Lackierung: Silber Polster: Leder Glas: Getönt	Technik: Hubraum: 2997 cm Leistung: 250 kW Höchstgeschw.: 254 km/h Tankinhalt: 80 L Durchschn. verbr.: 11,5 L Länge: 4700 cm		
Baujahr	Preis S Benzin	Preis S Diesel	Preis Turbo
2012:	58 000 €	58 000 €	80 000 €
2013:	60 000 €	61 500 €	84 000 €
2014:	63 000 €	64 000 €	87 000 €
2015:	66 500 €	67 500 €	90 000 €

Abb. 20  
Oben: Feldgruppe ohne geschlossene Gestalt  
Unten: Feldgruppe mit geschlossener Gestalt

### 8.3.3 Gesetz der Geschlossenheit

Das Gesetz der Geschlossenheit besagt, dass fehlende Bestandteile einer „geschlossenen“ Gestalt von der Betrachterin bzw. vom Betrachter ergänzt werden. Häufig erzeugen Elemente in Masken jedoch keine „ergänzbare“ Gestalt. Eine Gruppierung ist nur schwer wahrzunehmen, das Auffinden von Informationen fällt schwerer (Abbildung 20).

Die Elemente sind im oberen Teil der Abbildung so angeordnet, dass sie visuell nicht zu einer geschlossenen Gestalt ergänzt werden können. Die Elemente scheinen deswegen funktional nicht zusammen zu gehören. Im unteren Teil kann dagegen eine geschlossene Gestalt (Rechteck) erzeugt werden. Die Zusammengehörigkeit ist hergestellt.

## 8.4 Elemente von grafischen Benutzeroberflächen (Steuerelemente)

Es gibt eine Vielzahl von Elementen, mit denen man eine Anwendung bedient, z. B. Eingabefelder, Auswahllisten, Radio-buttons (Optionsfelder), Checkboxes (Kontrollkästchen, Markierungsfelder) und Abwandlungen davon. Um die Effizienz der Bedienung zu unterstützen, müssen die Elemente sorgfältig entsprechend der beabsichtigten Verwendung (Arbeitsaufgabe) und der Art der Informationen ausgewählt werden.

### 8.4.1 Auswahl der richtigen Elemente

Jedes Element hat eine Reihe von Auswahlkriterien, unter welchen Bedingungen es optimal einsetzbar ist.

Die Verwendung von Radiobuttons ist hier die richtige Wahl, da Radiobuttons als Alternativen verstanden werden. Zwei Checkboxes für die Auswahl Männlich / Weiblich zu verwenden, ist falsch, da mit Checkboxes üblicherweise Optionen dargestellt werden. Eine Dropdown-Auswahlliste kann zwar prinzipiell verwendet werden, ist jedoch umständlich zu bedienen und mit Hinblick auf die hier gegebene Auswahl von zwei Elementen ein ungünstiges Element.



Abb. 21 Richtige, falsche und ungünstige Elementverwendung

### 8.4.2 Einstellungen und Verhalten der Elemente

Ergonomische Eigenschaften der Elemente können wesentlich zur Effizienz der Bedienung beitragen. Das folgende Beispiel soll demonstrieren, wie wichtig an die Aufgabe angepasste Einstellungen sind.



Abb. 22 Listenelement mit zu kurzer Listenlänge

Die ausgeklappte Liste in der Abbildung 22 ist viel zu kurz und würde unnötiges Scrollen erforderlich machen. Dies ist gerade dann von Bedeutung, wenn das Element häufig zu bedienen ist.

Eine weitere wichtige Forderung in Bezug auf das Verhalten von Elementen ist die Möglichkeit der Tastaturbedienung. Vor allem bei tastaturintensiven Arbeiten wie Dateneingaben ist sicherzustellen, dass man (alle) Elemente nicht nur mit der Maus, sondern auch mit der Tastatur auswählen und bedienen kann. Funktionen wie Texte kopieren, aus einer Liste auswählen und Schaltflächen aktivieren sind auch mit der Tastatur, zum Beispiel durch sogenannte Tastaturkürzel (Shortcuts), zu ermöglichen.

Der Status der Elemente sollte jederzeit visuell unterschieden werden können und erkennbar sein. Die Auswahl in einer Liste oder die Veränderung eines Links, wenn sich der Mauszeiger darüber befindet (Mouseover), sind unterscheidbar zu gestalten (z. B. Farben, Rahmen).

Für jede Anwendung sind die folgenden Einstellungsbeispiele zu prüfen:

- **Eingabefelder** sind immer so zu dimensionieren, dass die meisten Einträge vollständig angezeigt werden. Unnötig

lange Eingabefelder sollte man vermeiden. Alternativ kann der komplette Text in einem separaten Fenster angezeigt werden.

- **Listen** sind immer so groß anzuzeigen oder aufzuklappen, dass eine zweckmäßige Anzahl von Einträgen sichtbar ist. Dies reduziert die Notwendigkeit zu scrollen und unterstützt bei der Suche nach Einträgen.
- **Tabellen** sind anpassbar zu gestalten, z. B. die Sortierichtung innerhalb einer Spalte, die Spaltenreihenfolge, die Spaltenbreite. Alle Anpassungen sollten gespeichert werden können.
- **Schaltflächen** sind entsprechend der Dialogsituation so zu belegen, dass die benötigte Funktion mit der „Enter“-Taste ausführbar ist (Standard-Schaltfläche), z. B. „Speichern“.
- **Datumsfelder** sind mit einer Kalenderfunktion oder mit Kürzeln bedienbar zu gestalten, z. B. wenn durch die verkürzte Eingabe eines Datums von Tag/Monat das aktuelle Jahr automatisch ergänzt wird.

## 8.5 Textgestaltung

### 8.5.1 Verständliche Texte

Texte bzw. Informationen sind verständlich, nicht kompromittierend oder beleidigend und entsprechend dem Vokabular der Benutzerinnen und Benutzer zu formulieren. Fachbegriffe oder kryptische Codes sind zu vermeiden.

Im Unterschied zur Leserlichkeit, die sich auf die visuelle Erkennbarkeit bezieht, ist mit Lesbarkeit die Verständlichkeit von Texten und Informationen generell gemeint. Wie verständlich ein Text ist, wird durch die sprachliche Gestaltung beeinflusst. Je verständlicher Beschriftungen oder Meldungstexte sind, desto leichter ist es, die aktuelle Dialogsituation zu verstehen, die nächsten Arbeitsschritte zu erledigen und das System zu erlernen.

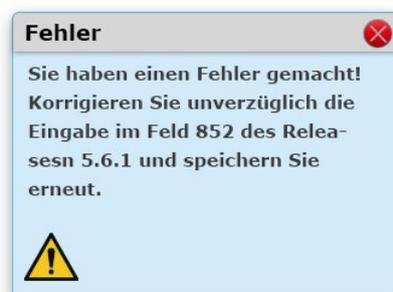


Abb. 23 Ungünstiger Meldungstext

In der abgebildeten Meldung wird die Benutzerin bzw. der Benutzer gewissermaßen „beschuldigt“, einen Fehler gemacht zu haben (Abb. 23). Auch das Wort „unverzüglich“ ist nicht angemessen. Es werden keine verständlichen Angaben gemacht, welches Feld eigentlich betroffen ist, sondern es wird nur ein kryptischer Code angezeigt. Mit dieser Meldung hat die Benutzerin oder der Benutzer Schwierigkeiten, das Feld zu finden und den Fehler zu korrigieren.

Innerhalb der Anwendung ist auf eine einheitliche Terminologie und Formulierung zu achten. So kann z. B. die Funktion

zur Übermittlung von Daten immer „Datei übertragen“ genannt werden, und nicht an einigen Stellen zusätzlich „Senden“ oder „Upload“. Funktionen sollten mit Verben wie Speichern, Schließen bezeichnet werden. Für das Aufrufen von Masken sollten hingegen Substantive verwendet werden, z. B. „Einstellungen...“.



Abb. 24 Kryptische, nicht verständliche Beschriftung

Abkürzungen in Beschriftungen können zu einer verschlechterten Verständlichkeit beitragen. In der Abbildung sieht man die etwas ungewöhnliche Abkürzung der Postleitzahl und des Ortes (Abb. 24).

### 8.5.2 Positive und negative Darstellung von Text (Polarität)

Texte sollten in dunkler oder schwarzer Schrift vor einem hellen Hintergrund dargestellt werden (positive Polarität, Positivdarstellung). Ein angemessener Wert für einen Kontrast ist dabei nicht zu unterschreiten. Softwareseitig sollte hier ein theoretischer Kontrast von mindestens 4,5:1 programmiert sein (siehe Abschnitt 8.7 Farben/Kontrast). Das Gleiche gilt für Symbole. Sie sollten ebenfalls vor einem hellen Hintergrund in dunkleren Farben dargestellt werden.

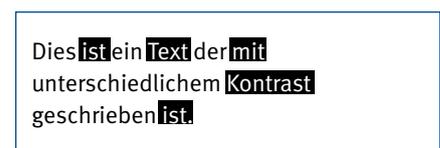


Abb. 25 Inverse Darstellung von Textabschnitten

Grundsätzlich sind inverse Darstellungen von Textabschnitten zu vermeiden. In Bezug auf die Aufgabenstellung kann dies aber im Einzelfall von Vorteil sein, z. B. Auswahl in einer Liste.

Geeignet	Ungünstig	
	Bereits genutzt	schlecht lesbar
Hervorhebung Hervorhebung Hervorhebung	<u>Hervorhebung</u> <b>Hervorhebung</b>	<i>Hervorhebung</i> HERVORHEBUNG

Abb. 26 Verschiedene Hervorhebungen von Text

EIN TEXT, DER PLÖTZLICH NUR GROß IST, IST DEUTLICH SCHWIERIGER ZU LESEN, ALS EIN NORMALER TEXT MIT GROß- UND KLEINSCHREIBUNG.  
 Ein Text, der plötzlich nur groß ist, ist deutlich schwieriger zu lesen, als ein normaler Text mit Groß- und Kleinschreibung.

Abb. 27 Unterschied zwischen Großschreibung und Groß- und Kleinschreibung

Dies ist ein Beispiel für einen Text, der ca. 70–80 Zeichen pro Zeile hat. Der Text hat eine gute Lesbarkeit.

Dies ist ein Beispiel für einen Text, der deutlich mehr als 80 Zeichen pro Zeile hat. Der Text verliert dadurch seine gute Lesbarkeit.

Dies ist ein Beispiel für einen Textverlauf  
 Mit zu wenigen Buchstaben pro Zeile. Er ist weniger gut lesbar.

Abb. 28 Unterschiedliche Textverläufe

DE3450000000234057380

DE34 5000 0000 0234 0573 80

Abb. 29 IBAN-Nummer (links ungruppiert, rechts gruppiert)

### 8.5.3 Hervorheben und Darstellen von Text

Hervorhebungen innerhalb von Fließtexten sparsam verwenden. Mögliche geeignete Hervorhebungen sind z. B. Fettdruck, eine andere Textfarbe oder eine andere Hintergrundfarbe, ungeeignet sind z. B. inverse Darstellung oder Großschreibung.

Unterstreichen sollte in der Regel für die Kennzeichnung von Links vorbehalten sein. Die inverse Darstellung ist für Textmarkierung und Auswahl reserviert. Kurve Zeichen sind wegen des sogenannten Treppeneffekts (siehe Abschnitt 8.6.4 Schrift/Glättung) bei kleineren Schriften zu vermeiden.

Für Benutzerinnen und Benutzer wichtige Informationen wie z. B. der Name einer Person in einem Eingabefeld sind ohne Ausnahme in normaler Groß- und Kleinschreibung zu schreiben.

Bei der Darstellung von Text ist darauf zu achten, wie viele Buchstaben eine Zeile hat. Ab ca. 80 Buchstaben pro Zeile wird der Text weniger gut erfassbar; dies gilt ebenso für zu kurze Zeilen.

### 8.5.4 Texte strukturieren

Gruppierung ist ein wichtiges Hilfsmittel, Informationen verständlicher zu gestalten. Wenn Listeneinträge, Texte oder Zahlenfolgen ohne Gruppierung in größerer Zahl (z. B. größer 8) angezeigt werden, so ist es ungleich schwieriger, bestimmte Informationen zu erfassen.

Ein Beispiel für die Gruppierung von Zahlen ist die IBAN-Nummer. Die Gruppierung erleichtert das Lesen, Eingeben und Prüfen einer solchen Nummer.

## 8.6 Schrift

### 8.6.1 Schriftart

Bei der Wahl der Schriftart kommt es darauf an, wie gut ein Text visuell erkannt und gelesen werden kann (Leserlichkeit). Entscheidend dabei ist nicht so sehr, einzelne Buchstaben zu erkennen, sondern vielmehr, Wörter als Ganzes zu erfassen. Aus diesem Grund ist es wichtig, die Schrift so zu wählen, dass man das Wortbild gut wahrnehmen kann.

Folgende proportionale Schriften ohne Serifen sind z. B. zu empfehlen:

- Verdana
- Calibri
- Arial
- MS Sans Serif
- Tahoma

Diese Schriften ergeben auf dem Bildschirm ein gutes Schriftbild. Dies heißt, es gibt bei hinreichender Zeichengröße gleichmäßige und ausreichend große Abstände zwischen den Buchstaben; sie laufen nicht ineinander. Die Schriften haben ein akzeptables Verhältnis zwischen Zeichenbreite und -höhe (ca. 7:10 für Großbuchstaben).

Die stufenweise Vergrößerung der Schrift, z. B. durch eine Zoomfunktion, bewirkt meist keine Verschlechterung der Darstellung. Eine Verkleinerung ist dagegen unter Umständen problematisch und kann das Schriftbild verschlechtern (z. B. variierende Abstände zwischen den Zeichen).

Innerhalb einer Anwendung sollte man nur eine Schriftart verwenden, damit sich die Benutzerin oder der Benutzer an das Schriftbild gewöhnen kann.

### 8.6.2 Schriftgröße

Für jedes Display und jede Anwendung ist im Einzelfall zu prüfen, ob die Mindestanforderung für Leserlichkeit bezüglich der absoluten Zeichenhöhe auf dem Bildschirm (mit 20 – 22 Bogenminuten, siehe Abbildung 31) mit der verwendeten

Empfohlene Zeichengröße in Anhängigkeit vom Sehabstand	
Sehabstand (mm)	Empfohlene Zeichenhöhe (mm)
500	3,2 bis 4,5
600	3,9 bis 5,5
700	4,5 bis 6,4
800	5,2 bis 7,3

Abb. 30 Empfohlene Zeichenhöhe von Großbuchstaben (bzw. Kleinbuchstaben mit Oberlänge) in Abhängigkeit vom Sehabstand

Schriftgröße eingehalten wird. Die absolute Zeichenhöhe kann z. B. mit einer Messfolie direkt am Bildschirm einfach nachgemessen werden.

Die Schriftgröße muss in Bezug auf die speziellen Bedürfnisse der Benutzerin bzw. des Benutzers und die Arbeitsaufgabe angepasst werden können. Der Benutzerin oder dem Benutzer sollte man ermöglichen, Masken oder Seiten insgesamt und fehlerfrei zu vergrößern (zu skalieren), wenn Schriften für sie oder für ihn zu klein sind – unabhängig vom Endgerät oder der Art des Displays und der Anwendung.

Wie können Schriften vergrößert werden?

1. Der Anzeigebereich einer Anwendung wird über eine Zoomfunktion vergrößert. Nachteil ist, dass man eventuell scrollen muss, um an wichtige Details und Funktionen zu gelangen; d. h., der Bearbeitungsaufwand kann entsprechend erhöht sein. Neben erhöhten Navigationsanforderungen kann es zudem vorkommen, dass nur Texte vergrößert werden, jedoch keine Elemente wie Eingabefelder oder Listen.
2. Die Vergrößerung wird über eine spezielle Funktion der Anwendung eingestellt.
3. Man bietet eine alternative Benutzungsoberfläche an, die komplett für eine größere Schrift und größere Elemente gestaltet ist.

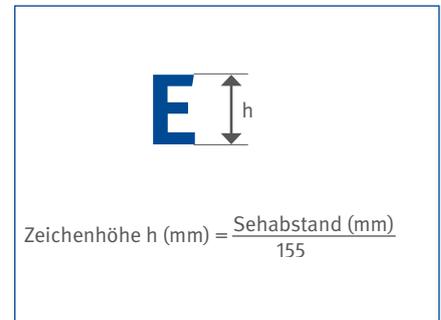


Abb. 31 Faustformel zur Berechnung der Zeichenhöhe

### 8.6.3 Mehrdeutige Zeichen

Wenn es erforderlich ist, einzelnen Buchstaben zu identifizieren, müssen Zeichensätze verwendet werden, die dies gut ermöglichen. Die Buchstaben l ("großes i") und „l“ (kleines „L“) sind nicht in jeder Schriftart eindeutig voneinander unterscheidbar, z. B.:

Schriftart	kleines „l“ und großes „i“	
Verdana	l I	Gute Unterscheidbarkeit
Calibri	l l	Großes „i“ ist nur etwas kleiner
Arial	l l	Kein Unterschied
MS Sans Serif	l l	Kein Unterschied
Tahoma	l I	Gute Unterscheidbarkeit

Abb. 32: Mehrdeutige Zeichen

### 8.6.4 Glättung von Schrift

Die Glättung ist ein Hilfsmittel, um die „gezackten“ Schrägen und Kurven einer Schrift optisch „glatter“ darzustellen.



Abb. 33 Vergrößerte Darstellung einer geglätteten Schrift (unten)

Nicht nur um die Kanten der Buchstaben werden zusätzliche Bildpunkte eingefügt, es werden sowohl die Linien der Zeichen selbst, als auch die Abstände zwischen den Buchstaben ausgeglichen (z. B. zwischen dem „M“ und dem „e“) (siehe Abbildung 33).

Ein optischer Ausgleich des „Treppeneffekts“ (engl. Antialiasing) kann die Lesbarkeit durch ein verbessertes Schriftbild erhöhen. Doch je kleiner die Schrift ist, desto „verschwommener“ oder unschärfer kann sie werden.

Bilder werden ebenfalls geglättet, wenn sie verkleinert oder vergrößert in einer Anwendung eingebunden sind. Bei einer Glättung können wichtige Pixel, die zur Erkennbarkeit und Unterscheidbarkeit von Symbolen beitragen, verloren gehen.

## 8.7 Farben

### 8.7.1 Farbkodierung und Farbkombinationen

Farben sollten grundsätzlich sparsam verwendet und nicht als einzige Form der Kodierung genutzt werden. Als Faustregel gilt, dass nicht mehr als fünf bis sechs Farben für die Gestaltung von Informationen zu verwenden sind. Bei der Farbwahl soll die Erkennbarkeit und die Unterscheidbarkeit der Farben gewährleistet sein. Die Merkfähigkeit der Benutzerinnen und Benutzer darf nicht zu sehr beansprucht werden, wenn Informationen zusätzlich mithilfe von Farben kodiert werden. Neben Farben für Text und Hintergrund spielt die richtige Farbwahl auch für Symbole, Links, die Kennzeichnung von Pflichtfeldern oder von Elementen mit fehlerhaften Eingaben eine große Rolle. Dabei muss unterschieden werden zwischen Farben, die eine bestimmte Information übermitteln, und Farben, die ausschließlich der Ästhetik dienen.

### 8.7.2 Kritische Farbkombinationen

Farbkombinationen, die aufgrund stark unterschiedlicher Wellenlängen der Farben nicht gleichzeitig scharf gesehen werden können, sollten vermieden werden. Deshalb sollten Farben von den Rändern des Spektralbereichs wie Rot und Blau nicht kombiniert werden. Insbesondere nicht als Kombination von Text- und Hintergrundfarbe. Bei solchen Farbkombinationen versucht das Auge ständig entweder die eine Farbe oder die andere Farbe scharf zu stellen, wodurch ein Tiefeneffekt entsteht, der das Lesen enorm erschweren kann.

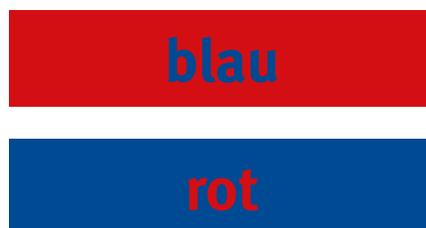


Abb.34 Chromostereopsis-Effekt

Auch bei Farbkombinationen mit grauem Hintergrund ist Vorsicht geboten. Ein weiterer physiologischer Aspekt ist die sogenannte Rot-Grün-Farbfehlsichtigkeit. Neben Rot und Grün können kombinierte Farbfehlsichtigkeiten noch bei Blau-Gelb oder bei einzelnen Farben Rot, Grün oder Blau vorkommen. Texte in solchen Farbkombinationen sind für Betroffene schlecht bis gar nicht zu lesen.

### 8.7.3 Konventionen

Farbkonventionen beziehen sich hier auf die Verwendung von Farben im westlichen Kulturkreis. Die Farben Rot, Gelb und Grün haben eine besondere Bedeutung, die man bei der Gestaltung von Oberflächen berücksichtigen sollte.



Abb.35 Verwendungsbeispiele der Farben Rot, Gelb und Grün nach europäischen Konventionen

Rot soll als Hervorhebung wirklich nur dann verwendet werden, wenn es sich um eine wichtige Meldung handelt und die Benutzerin oder der Benutzer z. B. Gefahr läuft, Daten zu verlieren, falls sie oder er eine bestimmte Funktion ausführt. Es ist zu vermeiden, den Text selbst in Rot hervorzuheben, ebenso wie den Hintergrund. Stattdessen kann man Rot z. B. für eine Fläche, einen Rahmen oder ein Symbol verwenden, um die Dringlichkeit zu verdeutlichen.

Auf keinen Fall ist Rot z. B. für häufig genutzte Maskenbereiche zu verwenden, nur um eine bestimmte Information hervorzuheben. Auch andere gesättigte Farben sind mit großer Vorsicht zu verwenden.

**8.7.4 Kontrast**

Für die Leserlichkeit (visuelle Erkennbarkeit) von Text spielen Kontrast und Helligkeit eine entscheidende Rolle. Neben der Einstellbarkeit am Bildschirm oder über das Betriebssystem kann die Helligkeit der Vorder- und Hintergrundfarbe den Kontrast beeinflussen.



Abb. 36 Verschlechterung der Leserlichkeit durch Kontrastreduzierung

Die Abbildung 36 zeigt eine häufige Dialogsituation, in der Felder inaktiv dargestellt sind (unten). Durch die Grautöne mit fast derselben Helligkeit für Schrift und Hintergrund wird zwar der inaktive Zustand angezeigt, was aber auch den Kontrast und damit die Leserlichkeit deutlich reduziert. Dies ist problematisch, wenn die Informationen trotz Deaktivierung gelesen werden müssen.



Abb. 37 Verbesserter Kontrast durch Wahl einer geeigneten Farbe

Eine geeignete Farbwahl kann den Kontrast und die Leserlichkeit von Text erhöhen, wie in Abbildung 37 zu sehen ist.

Eine andere Vordergrundfarbe (in der Abbildung 37 ein Gelb mit höherer Farbintensität bzw. Helligkeit) kann die visuelle Erkennbarkeit (Leserlichkeit) erhöhen.

Der Kontrast kann auch durch eine Software, einem sog. Farbkontrast-Analyzer theoretisch bestimmt werden. Dieser Farbkontrast-Analyzer ermittelt den softwareseitig vorgegebenen Kontrast zwischen der programmierten Vordergrund- und der programmierten Hintergrundfarbe. Der Kontrast sollte einen Wert von 4,5:1 nicht unterschreiten. Für häufig gelesene oder wichtige Texte ist ein Verhältnis von 7:1 zu bevorzugen.

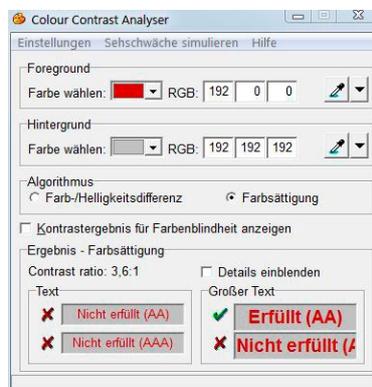


Abb. 38 Screenshot des Farbkontrast-Analyzers (Download unter [www.C2Web.de](http://www.C2Web.de))

**8.7.5 Farben für Symbole**

Farben oder Farbflächen sind nicht allein für die Kodierung von Information zu verwenden. Durch einen (zusätzlichen) Text, eine grafische Kontur oder ein prägnantes Symbol kann leichter erkannt werden, dass eine Information übermittelt wird.

In der Abbildung sind oben nur Farben zur Kennzeichnung des Status eines Netzwerkdruckers verwendet, unten dagegen auch weitere Zeichen (Häkchen und Kreuz). Dies verbessert Unterscheidbarkeit und Erkennbarkeit. Zusätzlich ist es sinnvoll, einen erklärenden Text auf der rechten Seite des Symbols anzuzeigen. (Abbildung 39).



Abb. 39 Kennzeichnung des Status eines Netzwerkdruckers

# 9 Interaktionsdesign

In diesem Kapitel werden die sieben „Grundsätze der Dialoggestaltung“ aus der Norm DIN EN ISO 9241-110 und deren Empfehlungen dargestellt und erläutert. Die Empfehlungen liefern konkrete Details für die Umsetzung der Grundsätze. Die Anforderungen der Norm können sowohl beim Bewerten von Oberflächen einer aktuellen oder zu entwickelnden Software als auch beim Einkauf von Software verwendet werden.

Zunächst werden die Grundsätze der Norm vorgestellt. Danach werden die wichtigsten Empfehlungen der Grundsätze erläutert. Die Empfehlungen und Grundsätze beziehen sich nur auf einen Ausschnitt aller ergonomischen Anforderungen. Für eine umfassendere ergonomische Beurteilung einer Oberfläche ist es ratsam, weitere softwareergonomische Normen der Reihe DIN EN ISO 9241 heranzuziehen.

## 9.1 Grundsätze der Dialoggestaltung

### 9.1.1 Aufgabenangemessenheit

#### DIN EN ISO 9241 - 110

„Ein interaktives System ist aufgabenangemessen, wenn es den Benutzer unterstützt, seine Arbeitsaufgabe zu erledigen, d. h., wenn Funktionalität und Dialog auf den charakteristischen Eigenschaften der Arbeitsaufgabe basieren, anstatt auf der zur Aufgabenerledigung eingesetzten Technologie.“

Das heißt, die Benutzungsschnittstelle (Funktionen, Masken, Dialoge) ist so gestaltet, dass sie die Erledigung der Arbeitsaufgabe (effizient und effektiv) unterstützt.

Die aufgabenangemessene Gestaltung der Benutzungsoberfläche ist ein wichtiger Grundsatz und betrifft eine Vielzahl an Gestaltungsaspekten. Ebenso sind Informationen aufgabenangemessen anzuordnen, zu strukturieren und darzustellen. Finden Benutzerinnen und Benutzer die von ihnen benötigten Informationen nicht oder sind sie z. B. auf mehrere Masken verteilt, erhöht sich der Dialogaufwand. Die Anzahl der Klicks oder der Dialogschritte steigt und es besteht Verbesserungspotenzial.

Im Folgenden sind ein paar Beispiele angeführt, die verdeutlichen, wie Informationen aufgabenangemessen präsentiert und gestaltet werden können. Z. B. sollten Spalten in einer Tabelle so sortiert werden, dass der für die Arbeitsaufgabe benötigte Wert sofort sichtbar ist. Alle Felder und Informationen, die zu einer Arbeitsaufgabe gehören, sollten möglichst gruppiert werden und gut erreichbar sein. Wird der Dialog für mehrere unterschiedliche Aufgaben verwendet, so ist abzuwägen, welche Anordnung für die häufigste Aufgabe am sinnvollsten ist. Darüber hinaus sollten angezeigte Informationen mit Vorlagen oder Dokumenten (Quelldokumente, z. B. Unfallanzeige) übereinstimmen, von denen Daten manuell übertragen werden.

Häufig ist die Strukturierung der Informationen in Dialogen bzw. auf Masken nicht aufgabenangemessen, sodass es sehr aufwendig ist, zu diesen zu navigieren. Dies trifft auch zu, wenn Informationen verteilt auf einer Maske angeordnet sind, sodass die relevanten Felder einer Arbeitsaufgabe nur über mehrere Dialogschritte (oder Stationen mit der Tab-Taste) erreichbar sind.

Natürlich gilt die aufgabenangemessene Bereitstellung und Anordnung auch für Funktionen. Insbesondere sind solche Funktionen zu erwähnen, die das Arbeiten mit dem System effizienter machen. So kann z. B. eine Information, die im

Zuge der Aufgabenerledigung benötigt wird, die sich aber an einer anderen Stelle befindet, mithilfe eines direkten Links gut erreichbar gemacht werden.

Aufgabenangemessene Voreinstellungen (Vorbelegungen) helfen, Aufgaben zu erledigen, ohne dass Einstellungen aufwendig geändert werden müssen. So wird beispielsweise beim Drucken automatisch ein Standarddrucker in der Nähe angeboten, ohne dass einer gesucht und ausgewählt werden muss.

#### Wenn Aufgabenangemessenheit nicht eingehalten wird ...

Auf einer Maske nicht aufgabenangemessen verteilte Felder erschweren die Suche, was zu Lasten der Leistungsfähigkeit und der Produktivität geht. Es können vermehrt Benutzungsfehler auftreten. Durch den erhöhten Navigationsaufwand benötigen Benutzerinnen und Benutzer mehr Zeit zur Bearbeitung der Aufgabe. Sie müssen sich mehr anstrengen, die benötigten Informationen zusammen zu tragen, deren Struktur und Anordnung zu verstehen bzw. sich zu merken.

### 9.1.2 Selbstbeschreibungsfähigkeit

#### DIN EN ISO 9241 - 110

„Ein Dialog ist in dem Maße selbstbeschreibungsfähig, in dem für den Benutzer zu jeder Zeit offensichtlich ist, in welchem Dialog, an welcher Stelle im Dialog er sich befindet, welche Handlungen unternommen werden können und wie diese ausgeführt werden können.“

Das heißt, für Benutzerinnen und Benutzer ist zu jedem Zeitpunkt offensichtlich, wo sie sich im System befinden, welche Eingaben sie vornehmen und welche Funktionen sie ausführen können, um die Arbeitsaufgabe zu erledigen.

Dieser Grundsatz ist sowohl für Laien als auch für Fachleute zu berücksichtigen. Durch die angezeigten Informationen sollte ihnen schnell klar sein, welche Funktionen zur Verfügung stehen, wo sie sich im System befinden und wie sie die nächsten Arbeitsschritte oder Arbeitsaufgaben erledigen können.

Vor allem Beschriftungen von Eingabefeldern und Schaltflächen, zusätzliche Informationen z. B. zum erwarteten Eingabeformat, sind handlungsleitend und damit

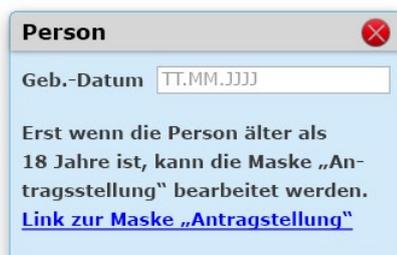


Abb. 40 Zusätzlicher Text zu einer Feldabhängigkeit zur Erhöhung der Selbstbeschreibungsfähigkeit

selbstbeschreibungsfähig zu formulieren (siehe Abbildung 40). Informationen z. B. über den aktuellen Systemzustand, bei längeren Download- oder Suchprozessen oder über Abhängigkeiten zwischen Feldern gehören ebenfalls dazu und machen verständlich, wie das System funktioniert.

In Anwendungen wird gerne eine Vielzahl an Funktionen mithilfe von Symbolen dargestellt (siehe Abbildung 41). Die Anforderung für Benutzerinnen und Benutzer besteht darin, die Funktion zu erkennen und sich möglichst viele Symbole zu merken. Symbole mit Beschriftung erhöhen die Selbstbeschreibungsfähigkeit der Anwendung.

#### ! Wenn Selbstbeschreibungsfähigkeit nicht eingehalten wird ...

Geringe Selbstbeschreibungsfähigkeit kann einen erhöhten Bedienungs- und Schulungsaufwand erzeugen. Hinzu kommt eine erhöhte Beanspruchung des Kurzzeitgedächtnisses.

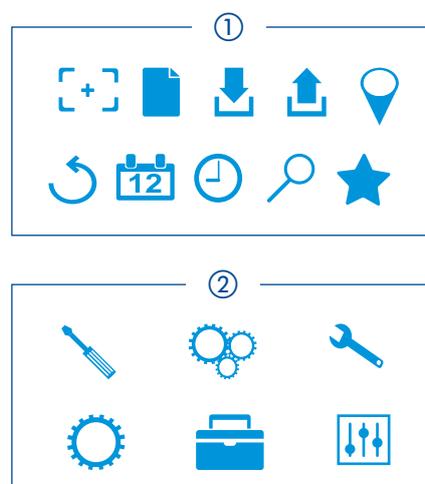


Abb. 41

- ① In Anwendungen gibt es eine Vielzahl an Symbolen, die man sich merken muss.
- ② Unterschiedliche Zahnradsymbole stehen für „Einstellen oder Verwalten“, der „Schraubenschlüssel“ steht ebenfalls für „Ändern“, „Einstellen“.

### 9.1.3 Erwartungskonformität

#### DIN EN ISO 9241 - 110

„Ein Dialog ist erwartungskonform, wenn er den aus dem Nutzungskontext heraus vorhersehbaren Benutzerbelangen sowie allgemein anerkannten Konventionen entspricht.“

#### „ANMERKUNG 1

Allgemein anerkannten Konventionen zu entsprechen, ist lediglich ein Aspekt von Erwartungskonformität.“

#### „ANMERKUNG 2

Konsistenz erhöht grundsätzlich die Vorhersehbarkeit eines Dialoges.“

Das heißt, die Benutzungsschnittstelle und deren Verhalten entsprechen den Erwartungen der Benutzerinnen und Benutzer, dem eigenen geistigen Modell vom System, sowie den eigenen Belangen und allgemeinen Konventionen.

Dieser Grundsatz soll sicherstellen, dass Erfahrungen, Gewohnheiten und die geistigen Modelle der Benutzerin oder des Benutzers von der Funktionsweise einer Anwendung berücksichtigt werden. Informationen sind erwartungskonform zu strukturieren, konsistent anzuordnen und darzustellen.

Erwartungen und Gewohnheiten haben sich während der Benutzung von Anwendungen entwickelt. Je weiter die Bedienung einer neuen Anwendung von diesen Erfahrungen entfernt ist, desto größer ist der Lernaufwand.

Sprachliche oder gestalterische Konventionen, wie das Platzieren der Menüleiste oberhalb oder links vom Informationsbereich, sind zu berücksichtigen. Gleiches gilt für die Erwartung an eine spezielle Strukturierung und Darstellung von Informationen. Adressfelder sind z. B. immer in der erwarteten Reihenfolge bzw. auf die gleiche Art und Weise anzuzeigen. Eine

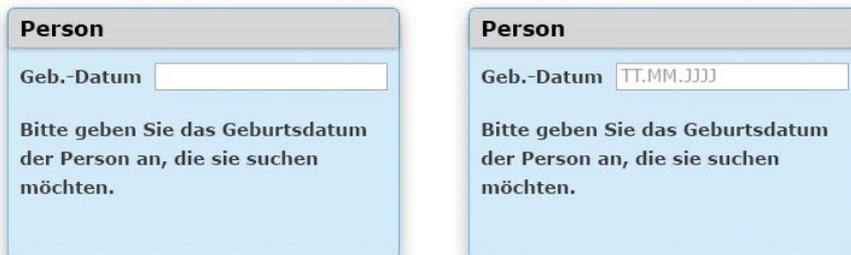


Abb. 42 links: Fehlende Formatangabe beim Geburtsdatum, rechts: Vorgabe des Formats



Abb. 43 Unerwartete Menüoption

widerspruchsfreie, konsistente Anordnung, z. B. von Überschriften, Schaltflächen oder Fehlermeldungen, erleichtert das Auffinden der benötigten Informationen und Funktionen.

Die Anwendung in Abbildung 42 – links lässt offen, welches Datumsformat benötigt wird.

Das Beenden der Anwendung in Abbildung 43 wird sicher nicht hier gesucht.

**! Wenn Erwartungskonformität nicht eingehalten wird:**

Die Bedienung der Anwendung erfordert eine erhöhte Konzentration. Das Arbeitstempo wird geringer, der Lernaufwand höher und die Fehlerwahrscheinlichkeit steigt.

**9.1.4 Lernförderlichkeit**

**DIN EN ISO 9241 - 110**

„Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen der Nutzung des interaktiven Systems unterstützt und anleitet“.

Das heißt, die Gestaltung des Systems erlaubt es, die Bedienung der Benutzungsschnittstelle mit vertretbarem Aufwand zu erlernen. Hierzu werden Bedienungskonzepte lernförderlich gestaltet.

Dieser Grundsatz bezieht sich auf das Entwickeln von Bedienungskonzepten und die Anzeige von Informationen, die das Erlernen der Bedienung unterstützen.

Zusätzliche erklärende Informationen z. B. in Meldungen oder in einem Assistent können dabei helfen, das System und dessen Funktionsweise schneller zu verstehen und zu erlernen. Informationen sind verständlich, hilfreich und zielgruppenorientiert zu formulieren.

**! Wenn Lernförderlichkeit nicht eingehalten wird ...**

Die Benutzerinnen und Benutzer müssen deutlich mehr Zeit aufwenden, um die Bedienung der Software zu erlernen und das Arbeitsergebnis zu erreichen. Hinzu kommt eine sinkende Motivation beim Umgang mit kompliziert zu bedienender Software.

**9.1.5 Steuerbarkeit**

**DIN EN ISO 9241 - 110**

„Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.“

Das heißt, der Benutzer oder die Benutzerin kann den Dialogablauf nach eigenen Belangen starten und zur Erledigung der Aufgabe vollständig steuern. Die Steuerung umfasst auch Unterbrechungen.

Es sollte möglich sein, die Bearbeitung zu unterbrechen und die Arbeit ohne Datenverlust an anderer Stelle oder zu einem späteren Zeitpunkt fortzusetzen.

Auf welche Weise Steuerung oder Kontrolle bereitzustellen ist, ergibt sich durch die Anforderungen der Arbeitsaufgabe bzw. durch die Bedürfnisse der Benutzerinnen und Benutzer. Ist ein Wechsel zwischen verschiedenen Aufgaben oder Masken nötig, sollte die Anwendung dies ermöglichen. Dauert der Download einer Datei sehr lange, ist eine Abbruchmöglichkeit zu realisieren.

**! Wenn Steuerbarkeit nicht eingehalten wird ...**

Werden Benutzerinnen bzw. Benutzer daran gehindert, den Dialog nach ihren Bedürfnissen zu steuern, kann die Arbeit mit der Anwendung umständlicher sein und mehr Arbeitszeit benötigen. Eine Anpassung des Arbeitsablaufs an die Anwendung kann nötig werden.

### 9.1.6 Fehlertoleranz

#### DIN EN ISO 9241 - 110

„Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.“

Fehlertoleranz wird mit den Mitteln erreicht:

- Fehlererkennung und -vermeidung (Schadensbegrenzung);
- Fehlerkorrektur oder
- Fehlermanagement, um mit Fehlern umzugehen, die sich ereignen.“

Das heißt, die Software unterstützt Benutzerinnen und Benutzer konstruktiv bei der Behebung erkannter Fehler. Darüber hinaus führen fehlerhafte Eingaben nicht zu Systemabstürzen.



Abb. 44 Unpassende Fehlermeldung (links), informative und unterstützende Meldung (rechts)



Abb. 45 Anpassbare Anzeige von Maskeninformationen durch „verbergen“ bzw. „anzeigen“.

Fehlertoleranz bezieht sich auf zwei Aspekte: Einerseits erkennt die Anwendung fehlerhafte Eingaben und unterstützt bei deren Korrektur. Andererseits toleriert die Anwendung Fehler bis zu einem gewissen Grad und überlässt den Benutzerinnen und Benutzern die Entscheidung, wann sie diese korrigieren. Es kann an anderen Stellen im Dialog weiter gearbeitet werden, bevor man einen Fehler korrigiert. Die Anwendung kann auch einen Vorschlag für die Behebung des Fehlers anbieten und diesen nach Bestätigung automatisch korrigieren.

#### ! Wenn Fehlertoleranz nicht eingehalten wird ...

Ein nicht fehlertolerantes System erschwert den Umgang mit Fehlersituationen, Fehler werden nicht rechtzeitig erkannt und korrigiert. Den Fehler zu verstehen und anschließend zu korrigieren, benötigt zusätzliche Arbeitszeit.

### 9.1.7 Individualisierbarkeit

#### DIN EN ISO 9241 - 110

„Ein Dialog ist individualisierbar, wenn Benutzer die Mensch-System-Interaktion und die Darstellung von Informationen ändern können, um diese an ihre individuellen Fähigkeiten und Bedürfnisse anzupassen.“

#### „ANMERKUNG 1

Obwohl es oft wünschenswert ist, dem Benutzer Möglichkeiten zu individuellen Einstellungen zur Verfügung zu stellen, ist dies kein Ersatz für ergonomisch gestaltete Dialoge. Außerdem sollen die Möglichkeiten zu individuellen Einstellungen nur innerhalb bestimmter Grenzen möglich sein, sodass Änderungen keine vorhersehbaren Beeinträchtigungen oder Schädigungen des Benutzers hervorrufen können.“

#### „ANMERKUNG 2

Individualisierbarkeit kann ein Mittel sein, die Barrieren für die Zugänglichkeit zu senken, indem man den besonderen Erfordernissen und/oder Fähigkeiten eines möglichst großen Benutzerkreises entgegenkommt“.

Das heißt, Dialogverhalten und die Darstellung von Informationen können Benutzerinnen und Benutzer nach ihren Bedürfnissen und Fähigkeiten anpassen.

Dieser Grundsatz fordert, dass das Verhalten von Elementen und die Darstellung von Informationen bis zu einem bestimmten Grad anpassbar gestaltet werden. Welche Anpassungsmöglichkeiten eingerichtet werden, ergibt sich aus den Bedürfnissen der Benutzerinnen und Benutzer, den Arbeitsaufgaben und daraus abgeleiteten ergonomischen Anforderungen (siehe Abbildung 45).

Anpassungsmöglichkeiten sind z. B. die vergrößerte Darstellung einer kompletten Maske (Skalierbarkeit) oder die Farbgebung und Kontrastgestaltung. Anpassungen sollten bei Bedarf dauerhaft verfügbar sein. Ebenso sollte bei einer Software der Unterstützungsgrad (Hilfe, geführte Dialoge mit Assistenten) anpassbar gestaltet sein.

**! Wenn Individualisierbarkeit nicht eingehalten wird ...**

Fehlen Möglichkeiten zu erforderlichen Anpassungen, kann es zu erhöhtem Arbeits- bzw. Dialogaufwand kommen.

**Umsetzung der Empfehlungen aus DIN EN ISO 9241-110**

**9.1.8 Terminologie**

Alle Texte sind mit einem Vokabular zu formulieren, das die Benutzerinnen und Benutzer sofort verstehen und mit dem sie vertraut sind. Dies gilt für Beschriftungen (Labels) sowie Menü- und Listeneinträge gleichermaßen. Es sollten keine technischen Formulierungen oder Abkürzungen verwendet werden, die schwer verständlich sind und bei denen es schwer fällt, diese zu erlernen.

Im erweiterten Sinne fallen Symbole ebenso in dieses Themengebiet. Symbole sind verständlich zu gestalten. Es sind solche Symbole zu verwenden, die für eine bestimmte Funktion erwartet werden (weil man z. B. ein ähnliches Symbol aus anderen Anwendungen kennt). Ein Beispiel wäre das Symbol einer Diskette für die Funktionalität „Speichern“. Obwohl es kaum noch Disketten gibt, hat die Benutzerin oder der Benutzer eine entsprechende Erwartung aufgebaut. Es empfiehlt sich, Beschriftungen zu den Symbolen anzuzeigen.

**9.1.9 Meldungen**

Meldungen als Reaktionen der Software sollen unmittelbar auf die entsprechende Aktion der Benutzerin oder des Benutzers angezeigt werden.

Meldungen sollen konstruktiv und objektiv formuliert sein, die Formulierungen sollen die Kenntnisse und Fähigkeiten von Benutzerinnen und Benutzern berücksichtigen. Handelt es sich z. B. um seltene Dialogsituationen, so sind ausführlichere Meldungen anzuzeigen und auf Anfrage bereitzustellen.

Meldungen sollten nicht nur informativ, sondern auch lernförderlich sein und helfen, Wissen darüber aufzubauen, wie das System oder die Anwendung funktionieren. Ein informativer Meldungstext (gerade bei Fehlern) macht deutlich, welches Problem entstanden ist bzw. wie es dazu gekommen ist, und welche weiteren Schritte zur Behebung unternommen werden müssen.

Bei länger andauernden Reaktionen des Systems ist die Anzeige einer Sanduhr ungeeignet. Die Benutzerin oder der Benutzer sollte vielmehr mit Textmeldungen oder auch Fortschrittsbalken informiert werden, ggf. mit der Möglichkeit, Prozesse abzubrechen. Ab wann und in welcher Dialogsituation eine ausführliche Meldung notwendig wird, hängt von den Erfahrungen und Erwartungen der Benutzerin oder des Benutzers ab.

**9.1.10 Dialogverhalten und Kontrolle**

Benutzerinnen und Benutzern sollte es möglich sein, Start und Verlauf des Dialogs mit der Software zu kontrollieren. Was bedeutet das?

1. Benutzerinnen und Benutzer können die Geschwindigkeit der Arbeit mit dem System über Anpassungen einzelner Funktionen kontrollieren. Meldungen in Popup-Fenstern werden z. B. solange lesbar angezeigt, wie dies nötig ist. Sie kontrollieren

ebenso die Anzeige der Datenmenge z. B. in einer Tabelle oder Liste. Einstellbar wäre dann die Anzahl der pro Seite angezeigten Zeilen.

2. Die Arbeit mit Dialogen kann unterbrochen und zu einem späteren Zeitpunkt ohne Datenverluste wieder aufgenommen werden, wenn es die Arbeitsaufgabe erfordert. Eine Eingabemaske kann z. B. geöffnet bleiben, um andere Aufgaben in einer Anwendung zu erledigen. Auf der anderen Seite können Daten zwischengespeichert werden, ohne dass der Dialog bzw. das bearbeitete Fenster geschlossen und dieses erneut geöffnet werden muss.
3. Eingaben und Änderungen in Masken können zurückgenommen werden, sodass die ursprünglichen Daten wieder angezeigt werden.

Fehlertolerant ist die Software, wenn Benutzerinnen bzw. Benutzer bei dem von ihnen oder vom System erkannten Fehlern die Korrektur zurückstellen können. Auch die automatische Korrektur von Fehlern zusammen mit Meldungen zur Korrektur unterstützt die Benutzerin und den Benutzer.

Das Erkunden und Erlernen des Systems wird unterstützt, wenn die Möglichkeit eingerichtet wird, Schritte zurückzunehmen und die Originaldaten wieder anzuzeigen. Bei selten verwendeten oder bei komplexeren Dialogen kann ein Assistent („Wizard“) Benutzerinnen und Benutzer in lernförderlicher Weise unterstützen, indem er sie anleitet, z. B. ein Formular auszufüllen und ihnen die einzelnen Schritte genau erklärt. Der geführte Dialog hilft so beim Erlernen der Schritte.

Das Verhalten der Dialoge (z. B. Eingabemasken) und speziell der Elemente ist innerhalb einer Anwendung konsistent (durchgängig/einheitlich) zu gestalten. Durch Konsistenz kann sich die Benutzerin oder der Benutzer an ein bestimmtes Verhalten gewöhnen und die Erwartung

**Person bearbeiten**

Name:

Vorname:

Straße:

PLZ u. Ort:

**Fehler im Feld „PLZ“:**  
 Sie haben die Postleitzahl mit einem Leerzeichen geschrieben. Bitte korrigieren Sie die PLZ jetzt oder später.

**Personen:** Anzahl der angezeigten Zeilen einstellen

Name	Vorname	Straße	PLZ
Maier	Gerhard	Maierstr. 124	51203
Müller	Wilfried	Müllerstr. 4	51203
Schmidt	Kerstin	Schmidtstr. 89	52008
Schmitt	Claudia	Schmittallee 90	65201
Schneider	Wolfgang	Schneiderplatz 1	65201
Schubert	Susanne	Schuberstr. 15	20856
Treiner	Kristina	Treinerweg 9	48950

Abb. 46 Kontrolle und Steuerbarkeit in einer Maske

aufbauen, dass die Elemente überall gleichermaßen funktionieren. Eine erwartungskonforme Bedienung erleichtert die Aufgabenerledigung und reduziert die Notwendigkeit, sich auf unterschiedliche Arten der Bedienung zu konzentrieren. Hierzu gehören z. B. Tastaturbelegungen wie Funktionstasten und Kurzwahl-tasten (z. B. Shortcuts wie Strg+P für „Drucken“). Konsistenz gilt ebenso für alle verwendeten Begriffe und Texte. Beschriftungen von Elementen sind einheitlich zu formulieren, auch die Benennung von Funktionen oder Meldungstexte.

Neben der Konsistenz innerhalb einer Anwendung ist auch auf die externe Konsistenz mit anderen Anwendungen innerhalb eines Arbeitsumfeldes (Nutzungskontextes) zu achten.

In der Abbildung 47 ist ein Speichern im untergeordneten Dialog „Person bearbeiten – Details“ möglich, doch was ist mit dem Speichern im Hauptdialog? Die Bedienung ist an dieser Stelle inkonsistent und nicht erwartungskonform.

Generell ist sicherzustellen, dass das Dialogverhalten und die Funktionalität der Anwendung zu jedem Zeitpunkt offensichtlich sind. Das bedeutet, dass ohne große Mühe verständlich ist, wo man sich im System befindet und welche Dialogschritte an diesem Punkt ausgeführt werden können. Durch eine unübersichtliche Gestaltung einer Maske, ein verschachteltes Menü, unbekannte Abkürzungen oder zu viele Symbole ohne Beschriftung werden mögliche Dialogschritte nicht offensichtlich.

Die Benutzerin oder der Benutzer kann in dieser Maske den Dialog in unterschiedlicher Weise steuern. So kann er oder sie

- 1 ... die Bearbeitung der Maske jederzeit abbrechen und die Änderungen verwerfen,
- 2 ... die Daten zwischenspeichern, ohne die Maske schließen zu müssen,
- 3 ... den letzten Eingabeschritt rückgängig machen,
- 4 ... wieder zu den Originaldaten vor der Änderung zurückkehren (oder Standardeinstellungen verwenden),
- 5 ... das Fenster geöffnet lassen (oder minimieren), um zwischendurch ein weiteres Fenster zum Bearbeiten einer Person zu öffnen,
- 6 ... die Anzahl der dargestellten Zeilen einstellen,
- 7 ... die Fehler erst später beheben und zu anderen Feldern gehen.

Darüber hinaus sind Dialogzustände zu vermeiden, in denen die Anwendung nicht mehr auf Eingaben reagiert und auch keine weitere Aktion mehr möglich ist, außer die Anwendung zu beenden oder den Computer auszuschalten.

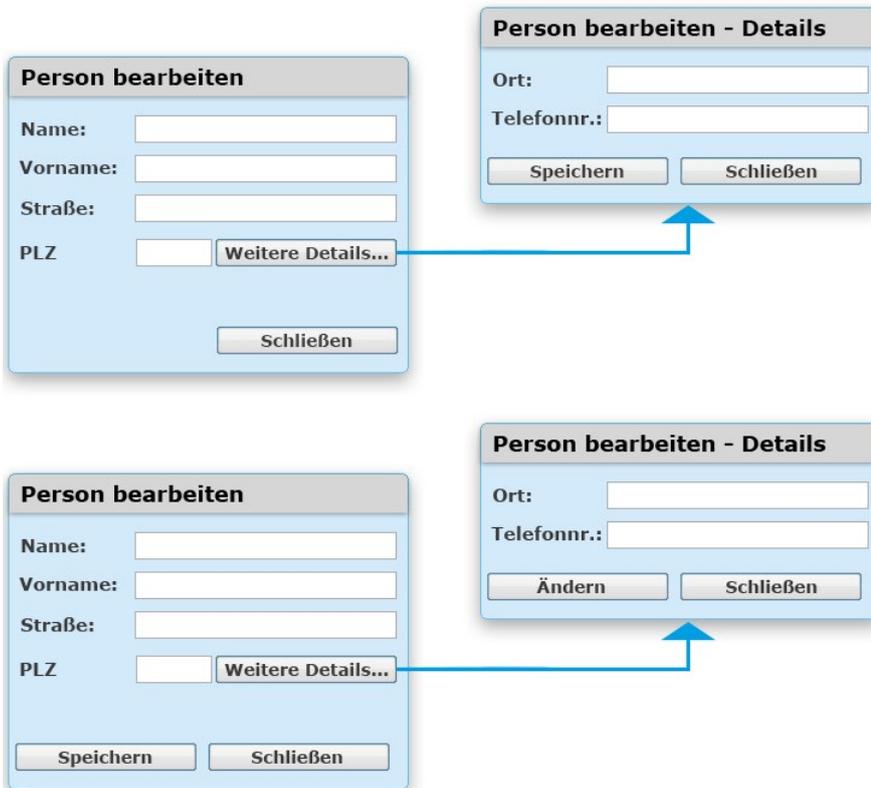


Abb. 47 Unklares Speicherverhalten (oben) und klare „Ändern“ und „Speichern“ Funktionen (unten)



Abb. 48 Berücksichtigung von häufig genutzten Daten

### 9.1.11 Häufigkeiten

Der Dialog kann vereinfacht werden z. B. durch die Vorbelegung von Eingabe- und Auswahlfeldern bei häufig verwendeten Eingaben und Auswahlen. Vorbelegungen, Gruppierungen, Anordnung und weitere Hilfestellungen können den Aufwand beim Bearbeiten oder Eingeben von häufig genutzten Daten deutlich reduzieren. Vorbelegungen betreffen nicht nur Eingaben in Eingabefelder, sondern auch Auswahlen in Listen oder Checkboxes. Sie können geändert oder überschrieben werden. Welche Vorbelegung oder voreingestellte Auswahl in den Feldern erscheint, sollte die Benutzerin oder der Benutzer definieren können.

Typische Auswahlen in Listen oder Menüs sind entsprechend der Häufigkeit der Benutzung zu gruppieren oder zu sortieren. Häufige Eingaben kann die Anwendung automatisch ergänzen, sofern dies im Einzelfall sinnvoll und gewünscht ist.

Um das lange Navigieren in einer umfangreichen Kundenliste zu vermeiden, sind in der Abbildung 48 im oberen Teil der Liste häufige und zuletzt bearbeitete Kundinnen und Kunden platziert.

### 9.1.12 Fehler erkennen und vermeiden

Die Software sollte das Erkennen und Vermeiden von Fehlern ermöglichen. Das Erkennen von Fehlern wird unterstützt, indem die Eingaben und Felder markiert werden, die einen Fehler enthalten. Meldungen weisen zusätzlich präzise auf den Fehler hin. Sie sind so zu platzieren und zu formulieren, dass sie wahrgenommen und verstanden werden.

Fehler kann man vermeiden, wenn verständliche Informationen auf Dialogschritte und Eingaben hinweisen. Nicht zuletzt ist es wichtig, gerade solche Dialogsituationen herauszufinden, in denen typischerweise Fehler auftreten, um an diesen Stellen Fehler, z. B. durch entsprechende Gestaltung oder Informationen,

zu vermeiden. Plausibilitätsprüfungen sind ein Mittel, Fehler früh zu erkennen, wobei die Korrektur der Fehler gegebenenfalls später erfolgen kann. Die Anwendung sollte auch in der Lage sein, Fehler selbst zu korrigieren oder eine Korrektur vorzuschlagen.

Im Zusammenhang mit der Vermeidung von Fehlern sind Pflichtfelder eindeutig zu kennzeichnen. Verteilte Pflichtfelder können z. B. mit einer speziellen Navigation direkt aufeinanderfolgend bearbeitet werden.

### 9.1.13 Hilfesysteme

Es sollte grundsätzlich vermieden werden, dass Benutzerinnen und Benutzer das Hilfesystem in Anspruch nehmen müssen. Zu besonders schwierigen Dialogsituationen sollten schon im Dialog verständliche Informationen gegeben werden, um die Arbeitsaufgabe mit dem System bewältigen zu können. Beschriftungen sollten ausführlicher gehalten sein und z. B. die Eingabe oder Abhängigkeiten zwischen Feldern erklären.

Wenn ein Hilfesystem benötigt wird, sind Informationen für die aktuelle Dialogsituation anzubieten (sogenannte „kontext-sensitive Hilfe“).

### 9.1.14 Anpassbarkeit an spezielle Bedürfnisse der Benutzerinnen und Benutzer

Durch eine Analyse der Benutzerbedürfnisse bzw. des Nutzungskontexts werden individuelle Anforderungen an die Software erhoben. Dies kann bedeuten, dass z. B. eine vergrößerte Schrift einzustellen ist, weil häufiger eine Sehschwäche vorliegt. Spezielle Eigenschaften der Benutzerin oder des Benutzers können es gegebenenfalls erlauben, Tastaturkürzel in einem Menü zu verwenden, statt die Einträge mit der Maus anzuklicken.

Der Benutzerin oder dem Benutzer kann man die Möglichkeit anbieten, eigene Begriffe, Beschriftungen und Abkürzungen

Abb. 49 Ergänzende Fehlermeldung

Im Feld „Straße“ ist eine zu hohe Hausnummer eingegeben worden. Bitte korrigieren Sie die Hausnummer.



Abb. 50 Anpassbarkeit der Scrollfunktion (links) und Auswahlmöglichkeiten für die Darstellung am Bildschirm (rechts)

zu verwenden, z. B. eigene Tastaturkürzel. Es besteht jedoch die Gefahr, dass benutzerdefinierte Begriffe oder Abkürzungen nicht ergonomisch sind (oder vergessen werden), sodass Andere mit diesen individuellen Anpassungen nicht arbeiten können.

Insgesamt kann die Bedienungseffizienz erhöht werden, wenn die Oberfläche anpassbar ist und Interaktionsformen auswählbar sind. Die Anpassbarkeit der Oberfläche sollte jedoch das genaue Ermitteln der Bedürfnisse der Benutzerinnen und Benutzer nicht ersetzen. Möglichkeiten der Anpassung sind sinnvollerweise dann vorzunehmen, wenn in der Designphase keine eindeutigen Designentscheidungen getroffen werden konnten. Die Ergonomie einer Software wird deutlich erhöht sein, wenn man für die Nutzung z. B. eine bis drei unterschiedliche Schriftgrößen anbietet.

### 9.1.15 Ein- und Ausgabemedien

Ein- und Ausgabemedien sollten der Arbeitsaufgabe entsprechen und für die Benutzerinnen und Benutzer wählbar sein. Je nach Art der Arbeitsaufgabe können Eingaben mit der Tastatur, der Maus, als Spracheingaben oder auf andere Weise sinnvoll sein. Bei der Bearbeitung einer Grafik sollten sämtliche Funktionen mit der Maus erreichbar sein, bei der Eingabe von Adressen sollen alle Funktionen und Felder mit der Tastatur erreichbar sein. Das Blättern in der Tabelle einer Datenbankanwendung kann über einen Touchscreen erledigt werden oder alternativ über die Maus oder ein anderes Zeigeinstrument.

# 10 Barrierefreie Gestaltung von Software

Barrierefrei nennen wir Einrichtungen, Produkte und auch Software, die Menschen mit Behinderungen selbstständig nutzen können. Dabei schwingt im Begriff „barrierefrei“ stets auch die Bedeutung von „universell nutzbar“ mit. Es geht in diesem Kapitel also nicht um Lösungen speziell für Behinderte, sondern im Gegenteil um ein erweitertes Nutzungskonzept, das möglichst alle Zielgruppen einbezieht.

## 10.1 Gesetzliche Grundlagen

Die Barrierefreiheit des öffentlichen Raums, einschließlich des Internets, ist weltweit als eine gesellschaftliche Aufgabe anerkannt. Die Europäische Union unterhält seit 1999 Aktionsprogramme für eine „Informationsgesellschaft für alle“, in denen die Barrierefreiheit eine wichtige Rolle spielt. In Deutschland wurde im Jahr 2002 das Behindertengleichstellungsgesetz (BGG) erlassen, dazu eine Ausführungsverordnung „Barrierefreie Informationstechnik-Verordnung (BITV)“. Demnach müssen die Bundesbehörden und andere Einrichtungen des öffentlichen Rechts auf Bundesebene seit Ende 2005 ihre Internetseiten und Informations-Software barrierefrei gestalten. Die technischen Standards übernimmt die BITV weitgehend von den internationalen, vom World Wide Web Consortium (W3C) herausgegebenen Richtlinien zur barrierefreien Gestaltung von Web-Inhalten „Web Content Accessibility Guidelines (WCAG)“, die seit 1999 bestehen und kontinuierlich aktualisiert und ausgebaut werden.

Auch im Bereich der Arbeitsplätze setzt sich die Idee der Barrierefreiheit immer mehr durch. In den USA gibt es seit 1998 die Richtlinie „Section 508“, nach der Regierungsbehörden bei der Beschaffung ihrer gesamten IT-Ausstattung, Hardware, Software und Telekommunikation darauf achten müssen, dass auch Menschen mit Behinderungen sie nutzen können. In der Europäischen Union ist eine vergleichbare Regelung für die Vergabe öffentlicher Aufträge im Frühjahr 2014 verabschiedet worden. Die technischen Standards sind als Europäische Norm EN 301 549

„Accessibility requirements suitable for public procurement of ICT products and services in Europe“ niedergelegt. Die EU-Vergaberichtlinie 2014/24, die auch die Barrierefreiheit als Vergabekriterium nennt, ist im April 2016 in nationales Recht umgesetzt worden (Vergaberechtsmodernisierungsverordnung - VergRModVO).

In Deutschland besteht durch das Sozialgesetzbuch schon lange eine generelle Beschäftigungspflicht für Schwerbehinderte (§ 71 SGB IX). Arbeitgeberinnen und Arbeitgeber, sowohl im öffentlichen Dienst als auch in der Wirtschaft, müssen 6 % ihrer Arbeitsplätze an Schwerbehinderte vergeben oder alternativ eine Ausgleichsabgabe zahlen. Sie haben Anspruch auf öffentliche Förderung zur behindertengerechten Gestaltung der bereitgestellten Arbeitsplätze. Diese Maßnahmen, die auf dem Schutzrecht des Einzelnen beruhen, haben zu zahlreichen individuellen Anpassungen geführt, jedoch bisher nur wenig dazu beigetragen, dass eine barrierefreie betriebliche IT-Infrastruktur entstehen kann. So ist zu meist die in Betrieben genutzte Anwendungssoftware nicht barrierefrei – mit Ausnahme von Produkten, die auch an US-Regierungsbehörden vertrieben werden und dort die Richtlinie Section 508 erfüllen müssen. Mit der europaweiten Verpflichtung des öffentlichen Dienstes auf barrierefreie IT-Ausstattung wird sich auch in Deutschland der Markt entsprechend anpassen.

Ein gesellschaftliches Umdenken ist vor allem seit der Verabschiedung der UN-Behindertenrechtskonvention im Jahr 2008

zu bemerken. Die Umsetzung der UN-Konvention in nationales Recht betrifft alle Lebensbereiche und genießt eine große mediale Aufmerksamkeit. Seitdem setzt sich ein Wandel in der gesellschaftlichen Einstellung zu Menschen mit Behinderungen durch. Die Priorität wird nicht mehr in der Integration, der Fürsorge für den einzelnen Menschen gesehen, sondern in der Inklusion, der barrierefreien Gestaltung der Gesellschaft, in der die oder der Einzelne trotz ihrer oder seiner persönlichen Einschränkung selbstbestimmt leben kann.

### Definition „Barrierefreiheit“ nach BGG § 4

„Barrierefrei sind ... Systeme der Informationsverarbeitung, akustische und visuelle Informationsquellen und Kommunikationseinrichtungen ..., wenn sie für behinderte Menschen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind.“

## 10.2 Universelles Design

Barrierefreiheit ist universelles Design. Dies sagt auch die Definition des BGG aus (siehe Kasten), nach der Einrichtungen barrierefrei sind, wenn sie von Menschen mit Behinderungen „in der allgemein üblichen Weise“ genutzt werden können. Erst wenn ein Dienst nicht barrierefrei gemacht werden kann, fordert das BGG behindertengerechte Spezial-Lösungen, wie etwa einen Gebärdensprachdolmetscher zur Unterstützung gehörloser Menschen im Verwaltungsverfahren.

Universelles Design wendet die Prinzipien der Softwareergonomie und der Gebrauchstauglichkeit auf die größtmögliche Zielgruppe an. Während die allgemeine Ergonomie sich an gut 90 % der Zielgruppe wendet, will universelles Design möglichst 100 % erreichen. Einbezogen werden Menschen mit eingeschränkten körperlichen, sensorischen und mentalen Fähigkeiten, ebenso aber einschränkende Umgebungsbedingungen wie Blendung oder Lärm, und situative Einschränkungen wie z. B. das Verbot für die Autofahrerin bzw. den Autofahrer, das Telefon während der Fahrt mit Hand und Auge zu bedienen.

Barrierefreie Produkte sind universell nutzbar. Das Musterbeispiel für universelles Design ist die abgesenkte Bordsteinkante, die gesetzlich verordnet wird, damit Rollstuhlfahrerinnen und Rollstuhlfahrer sich im Straßenverkehr ohne fremde Hilfe bewegen können (siehe Abbildung 51). Darüber hinaus aber wird diese Einrichtung auch von Müttern mit Kinderwagen, Radfahrenden und Reisenden mit Rollenkoffer sehr geschätzt. Was für die einen zwingend erforderlich ist, um teilhaben zu können, bringt anderen mehr Komfort oder mehr Effizienz.

Auch die Wirtschaftlichkeit ist ein Argument für universelles Design. Offensichtlich ist Barrierefreiheit in unserer alternativen Gesellschaft volkswirtschaftlich effizient, denn je besser alle Lebensbereiche barrierefrei gestaltet sind, desto weniger Aufwand entsteht für persönliche Assistenz oder für technische Anpassungen für einzelne Individuen. Auf betriebswirtschaftlicher Ebene rechnet sich Barrierefreiheit besonders dann, wenn es um nachhaltige Lösungen für eine möglichst große Zielgruppe geht.



Abb. 51 Abgesenkte Bordsteinkanten sind für Rollstuhl- und Radfahrende gleichermaßen gut geeignet.

## 10.3 Behinderungen

Wie soll man eine Software für alle nutzbar machen? Schließen sich die verschiedenen Bedürfnisse nicht gegenseitig aus? Die Lösung des Rätsels besteht darin, die grundlegenden menschlichen Fähigkeiten des Wahrnehmens, Verstehens und Handhabens als Kontinuum zu betrachten, von stark bis zu schwach ausgeprägten Formen, und sie in möglichst großer Bandbreite zu unterstützen. Hierzu gehören auch technische Hilfen, die den Verlust einer Fähigkeit ausgleichen.

### Sehen

Wir Menschen sind Augentiere, nehmen Information bevorzugt über den Gesichtssinn auf. Bilder sind in mancher Hinsicht leistungsfähiger als Sprache – ein Bild sagt mehr als tausend Worte. In der Datenverarbeitung nutzen wir diese Fähigkeit, indem wir Informationen durch flächige Anordnung, Farben und Icons grafisch aufbereiten und hierdurch Bedeutungszusammenhänge sichtbar machen.

Sehbehinderungen sind Einschränkungen des Sehvermögens, die nicht durch eine Brille ausgeglichen werden können.

- Eine verminderte Sehschärfe kann durch die **Vergrößerung** der Zeichen ausgeglichen werden. Individuelle Anforderungen reichen von einem geringen Zoomfaktor, der die Übersichtlichkeit nur wenig beeinträchtigt, bis hin zu großen Zeichen.
- Unterschiede in der Lichtempfindlichkeit äußern sich in verschiedenen Anforderungen an den **Helligkeitskontrast**. Es gibt beide Ausprägungen: Ältere Menschen brauchen zumeist mehr Licht bzw. einen starken Kontrast. Bei manchen Augenkrankheiten ist zu viel Licht schmerzhaft, dann werden schwache Kontraste oder eine Negativdarstellung (helle Zeichen auf dunklem Hintergrund) bevorzugt.

- Die **Farbwahrnehmung** ist bei manchen Menschen durch genetische Disposition eingeschränkt. Verbreitet ist die Rot-Grün-Farbfehlsichtigkeit, von der überwiegend Männer betroffen sind (ca. 6 bis 8 % in Deutschland).

Sehbehinderungen sind so vielfältig, dass es keine konkrete Gestaltung gibt, die für alle passt. Die Richtlinien verlangen vor allem die individuelle Einstellung von Schriftgrößen und -farben sowie einen Mindestkontrast. Bedeutungsunterschiede sollen nicht durch Farbe allein, sondern z. B. auch durch verschiedene Symbole dargestellt werden (siehe Kap. 8 „Informationsdesign“).

### Blindheit

Fällt der Gesichtssinn aus, finden sich andere Wege, um Informationen aufzunehmen und Zusammenhänge zu verstehen. Bei Blindheit spielt die sprachliche Vermittlung eine besondere Rolle. Zur Nutzung eines Computers sind Blinde auf spezielle Hilfstechniken angewiesen.

- Information muss als Text kodiert sein, damit sie von der Hilfstechnik vorgelesen werden kann.
- Bilder und Grafiken müssen beschriftet und sprachlich erläutert werden.
- Grafisch übermittelte Bedeutungszusammenhänge müssen auch sprachlich oder durch semantische Strukturen (siehe Kapitel 10.5) dargestellt werden.

Dieser Aufwand zahlt sich vielfältig aus, speziell im Internet, wo dieselben Maßnahmen u. a. zur Aufbereitung für Suchmaschinen dienen.

### Hören

Einschränkungen des Hörvermögens spielen in der Datenverarbeitung nur eine begrenzte Rolle, gewinnen aber durch die Verbreitung von Multimedia-Angeboten an Bedeutung.

- Signale sollen nicht nur akustisch, sondern auch optisch übermittelt werden.

- Gesprochene Sprache muss in Schrift umgesetzt werden.

Die textuelle Aufbereitung von Videos und Sprachdokumenten trägt, ähnlich wie die Maßnahmen für Blinde, zu einer Aufwertung des Informationsangebots bei.

Untertitel in Filmen sind gut für Gehörlose, dienen aber auch zum Auffinden durch Suchmaschinen, helfen beim Erlernen einer Fremdsprache und sind in lauten Umgebungen (Flughafen) nützlich.

### Gehörlosigkeit

Menschen, die vor ihrer Geburt oder in der frühen Kindheit das Gehör verloren haben, konnten die Lautsprache nicht erlernen und beherrschen infolgedessen die Schriftsprache nur begrenzt. Für ein genaues Verständnis komplexer sprachlicher Inhalte sind sie auf die Gebärdensprache angewiesen. Videoaufnahmen von Gebärdensprache können eingesetzt werden, um Software zu erläutern und Informationsangebote zu übersetzen.

Videos in Gebärdensprache sind ein spezielles Angebot für gehörlose Menschen und müssen nicht generell, sondern nur aufgrund einer Bedarfsanalyse eingesetzt werden.

Verpflichtet sind z. B. deutsche Behörden nach BITV § 3. Sie müssen auf der Startseite ihres Internetangebots Videos in Gebärdensprache bereitstellen, die Informationen zum Inhalt, Hinweise zur Navigation und zu weiteren Informationen in Gebärdensprache enthalten.

### Verstehen

Das Verstehen von Inhalten, Strukturen und Funktionen von Software ist eine kognitive Leistung, die jeden Menschen je nach den spezifischen Umständen herausfordern kann.

- Unerfahrene Benutzerinnen oder Benutzer erkennen ungewohnte Bedienelemente nicht.
- Bei starker Sehbehinderung oder im Smartphone können größere Informationsblöcke nicht im Zusammenhang wahrgenommen werden.



Abb. 52 Film mit Untertiteln und Gebärdensprache

- Bei Blindheit erkennt man neue Programmsituationen nicht immer sofort.
- Nach einer Störung muss man sich neu orientieren.
- Bei Zeitmangel, Stress oder Ablenkung kann man komplexe Strukturen nicht erfassen.
- Menschen mit geringem Bildungsstand, fremder Muttersprache oder geringer Fachkenntnis verstehen sprachlich komplex dargestellte Inhalte nicht.

Menschen mit Sinnesbehinderungen haben es schwerer, Informationen aufzunehmen und sich einen Überblick zu verschaffen. Sie profitieren besonders von den Regeln der allgemeinen Softwareergonomie, wie übersichtliche Navigation und konsistente Gestaltung, die in die Gestaltungsregeln für barrierefreie Webinhalte aufgenommen wurden.

Einfache Sprache unterstützt viele Zielgruppen.

Eine **einfache Sprache** fördert die Verständlichkeit von Inhalten. Wichtige Regeln sind einfache Sätze, die Vermeidung von Fremdwörtern und die Erläuterung notwendiger Fachbegriffe. Einfache Sprache entspricht laut WCAG dem Niveau der Sekundarstufe I (mittlere Schulbildung, 9./10. Klasse). Hiervon profitieren nicht nur Lernbehinderte, sondern jedermann. Einfache Sprache kann man in der Journalistenausbildung lernen. Auch vertiefende Fachinformationen sollen eine Einführung auf einfacherem Sprachniveau haben und damit für eine größere Zielgruppe leichter zugänglich sein.

#### Kognitive Behinderung

Unter dem Begriff „Lernbehinderung“ werden verschiedene kognitive Störungen zusammengefasst, die Menschen in der Verarbeitung von Informationen beeinträchtigen. Eine geeignete Informationsaufbereitung und die Vermeidung von

Ablenkungen kann die Nutzbarkeit von Software und Internet auch für diese Zielgruppe erweitern.

Leichte Sprache richtet sich an Menschen mit einer Lernbehinderung.

**Leichte Sprache** richtet sich an Menschen, deren Sprachverständnis in etwa dem Grundschulniveau entspricht. Eine stark vereinfachte Grammatik, konkrete Begriffe und die Reduzierung der Inhalte auf das Wesentliche machen Informationen für diese Zielgruppe zugänglich. Übersetzungen in leichte Sprache fertigen spezielle Dienstleister an, u. a. das „Netzwerk Leichte Sprache“. Ebenso wie Gebärdensprache gilt leichte Sprache als behinderungsspezifisches Angebot, zu dem nur Behörden verpflichtet sind.

Die **optische Aufbereitung** von Informationen, von der leicht lesbaren Schrift über sprechende Zwischenüberschriften bis zur Akzentuierung durch Grafiken, ist für viele Menschen hilfreich. In besonderem Maße profitieren Menschen mit Lese-/Rechtschreibschwäche davon, die sich nur schwer einen Überblick über einen Text verschaffen können. Für diese Zielgruppe ist auch das Vorlesen von Texten geeignet, vor allem wenn das Mitlesen durch eine optische Hervorhebung des gesprochenen Textes unterstützt wird.

Bei verminderter Konzentrationsfähigkeit ist es wichtig, Ablenkung durch **Animationen** zu vermeiden bzw. Animationen abschaltbar zu machen. Blinken in einer Frequenz ab drei Bildern je Sekunde kann sogar epileptische Anfälle auslösen und ist unbedingt zu vermeiden.

#### Handhaben

Informationstechnik ist immer auf irgendeine Weise interaktiv. Zur Bedienung ist es erforderlich, Optionen aus Menüs auszuwählen, Schaltflächen (sog. Buttons)

zu drücken oder Daten in Eingabefelder einzutippen. Die klassischen Eingabegeräte sind Tastatur und Maus, neuerdings ergänzt um Touchscreens, die mit Gesten bedient werden. Die Handhabung dieser Eingabegeräte erfordert motorische Fähigkeiten wie Stetigkeit, Zielgenauigkeit, Schnelligkeit, Kraft und einen gewissen Aktionsradius. Gerade die Mausbedienung erfordert eine Feinmotorik, die schon bei leichter Einschränkung der Hand nicht mehr aufzubringen ist.

Maus und Touchscreen sind zur Bedienung auf eine gute Feinmotorik angewiesen.



Abb. 53 Maus in der Hand, Geste auf Touchscreen

#### Motorische Behinderung

Für körperlich behinderte Menschen, die in der Steuerung ihrer Hände beeinträchtigt sind, existieren vielfältige technische Hilfen zur Computerbedienung.

Die **Tastaturbedienung** wird von Menschen mit motorischer Behinderung ebenso wie von Blinden und von erfahrenen Benutzern eingesetzt. Barrierefreie Software muss lückenlos mit der Tastatur bedienbar sein.

Menschen mit Behinderungen benötigen oftmals **mehr Zeit** und eine höhere **Fehlertoleranz**, um eine Aufgabe am Computer zu erledigen. Auch diese Anforderungen stehen in engem Bezug zur Softwareergonomie.

### Ältere Menschen

Die Wahrscheinlichkeit einer Behinderung steigt mit dem Alter. Altersbedingte Einschränkungen gibt es in allen genannten Funktionen:

- im Sehen und Hören
- im Kurzzeitgedächtnis und in anderen kognitiven Funktionen
- in der Kraft und Genauigkeit von Bewegungen

Zwar tritt die einzelne Funktionseinschränkung bei älteren Menschen oft nur in leichter Form auf, doch in der Kombination kann sich eine erhebliche Beeinträchtigung ergeben.

## 10.4 Technische Hilfen

Einige Behinderungen erfordern spezielle Ein-/Ausgabegeräte, um den Computer benutzen zu können. Software-Entwicklerinnen und -Entwickler sollten sich mit der generellen Funktionsweise von assistiven Technologien vertraut machen, denn sie bestimmen wesentlich die Arbeitsweise von Menschen mit Behinderungen.

Technische Hilfen zur Computernutzung gibt es seit dem Aufkommen der Personal-Computer Anfang der 1980er-Jahre. Viele wurden mit der Zeit in die Betriebssysteme integriert, einige fanden ihren Markt als Standardsoftware. Scanner mit Texterkennungssoftware waren anfangs ein hochpreisiges Hilfsmittel an Blindenarbeitsplätzen, bis die Texterkennung so gut war, dass sie in jedem Büro von Nutzen sein konnte. Die Wortansage bei der Texteingabe in Smartphones wurde zuerst als Hilfsmittel für motorisch Behinderte entwickelt. Die Entwicklung der Sprachsteuerung für Smartphones genießt große mediale Aufmerksamkeit. Der Hilfsmittelmarkt ist also kein exotischer Randbereich, sondern ein Testlabor für innovative Ideen in der Computerindustrie.

### Eingabehilfen des Betriebssystems

In modernen Betriebssystemen sind verschiedene Eingabehilfen implementiert. Unter der Bezeichnung „Erleichterte Bedienung“ oder „Bedienungshilfen“ sind aus der Systemsteuerung u. a. abrufbar:

- der Kontrastmodus zur Definition eigener Farbschemata und Schriftgrößen
- die Bildschirmlupe für die Ausschnittvergrößerung
- eine Einrastfunktion für Tastaturkürzel (sog. Short-cuts) und Umschalttasten für die einhändige Bedienung der Tastatur
- eine Bildschirmtastatur für die Texteingabe mit Zeigergeräten
- Einstellung der Reaktionszeit der Maus
- Steuerung der Maus mit der Tastatur
- Sprachaus- und -eingabe

Innovative Wege ging das iPhone®, das erstmals die Sprachausgabe VoiceOver® in das Betriebssystem integrierte. Die Verbindung von Touchscreen und Sprachausgabe ist für Blinde besonders einfach zu bedienen.

### Screenreader

Screenreader sind Programme zur Computernutzung für Blinde und Sehbehinderte. Der Screenreader interpretiert den Bildschirminhalt und gibt die Informationen über Braillezeile (Abbildung 54)<sup>1)</sup>, Sprachausgabe und Großschrift aus.

Der Screenreader stellt Informationen über die Funktion der Bedienelemente, die Bedeutung der grafischen Symbole, die aktuelle Eingabeposition und den Aufbau des gesamten Bildschirms zur Verfügung. Diese Information muss die Software liefern. Barrierefreie Software nutzt soweit möglich die Standardfunktionen des Betriebssystems und kommuniziert darüber hinaus über eine spezielle Accessibility-Schnittstelle mit dem Screenreader. In der Entwicklung von browserbasierten Anwendungen stellt die



Abb. 54 Braillezeile

1) Braillezeile = Computer-Ausgabegerät für blinde Menschen, das Zeichen in Brailleschrift darstellt.

Konformität mit den Webstandards sicher, dass der Screenreader ausreichende Informationen erhält.

Blinde steuern den Computer vorwiegend mit der Tastatur, da die Nutzung der Maus mit Screenreader zwar möglich, aber sehr umständlich ist. Der Screenreader hat eigene Shortcuts, die neben den Shortcuts der Anwendung zur Navigation auf dem Bildschirm genutzt werden.

**Vergrößerungssoftware**

Bei hochgradiger Sehbehinderung wird üblicherweise eine Vergrößerungssoftware eingesetzt. Diese erfüllt als externes System die gleiche Funktionalität wie die Bildschirmlupe und der Kontrastmodus des Betriebssystems, jedoch mit größeren Einstellungsbereichen, besserer grafischer Qualität und genauerer Anpassbarkeit.

**Spezialtastaturen, Zeigegeräte**

Für Menschen mit verminderten motorischen Fähigkeiten existieren vielfältige alternative Geräte zu Tastatur und Maus. Je nach den verbleibenden Fähigkeiten gibt es besonders große, besonders kleine, speziell angeordnete Tastaturen, robuste Trackballs, Kopf-, Augen- oder

Mundsteuerung. Solange nur ein Schalter mit einem beliebigen Körperteil gesteuert werden kann, ist die Computerbedienung im Prinzip möglich.

**Sprachsteuerung**

Die Steuerung des Computers über Sprachbefehle erfordert eine Anpassung und Eingewöhnung, ist dann aber effizient als Hilfsmittel für Menschen mit motorischer Behinderung einsetzbar. Darüber hinaus wird die Spracheingabe erfolgreich zum Diktieren für Fachsprachen, z. B. von Ärztinnen und Ärzten oder Juristinnen und Juristen, verwendet. Die Implementierung der Sprachsteuerung im Betriebssystem ist noch relativ neu und wird neue Verwendungsmöglichkeiten hervorbringen.

**Technische Beratung**

Die Auswahl der passenden technischen Hilfen zur Computernutzung erfordert eine genaue Analyse der individuellen Fähigkeiten und der zu erledigenden Aufgaben, außerdem eine gute Kenntnis der infrage kommenden Produkte. Eine herstellerunabhängige Beratung bieten u. a. Integrationsämter, Arbeitsagenturen und Unfallversicherungsträger.

**10.5 Gestaltungsrichtlinien**

Die Barrierefreiheit von Software wird in internationalen Richtlinien und Standards definiert. Maßgebend wurden die Web Content Accessibility Guidelines (WCAG) des World Wide Web Consortiums (W3C) für die Barrierefreiheit von Inhalten im Internet. WCAG 2.0 basiert auf generellen Gestaltungsprinzipien, die in jeder technischen Umgebung einzuhalten sind (Abbildung 56). Hieran orientieren sich auch neuere Richtlinien zur barrierefreien Gestaltung von Dokumenten und von Software. Die WCAG sind Grundlage nationaler Gesetzgebung, so auch der deutschen Barrierefreie-Informationstechnik-Verordnung – BITV 2.0.

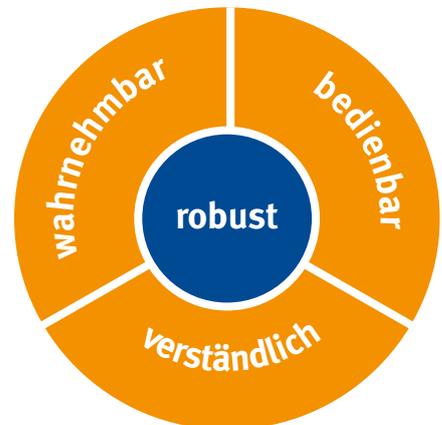


Abb. 56 Die vier Gestaltungsprinzipien nach WCAG 2.0: wahrnehmbar, bedienbar, verständlich, robust



Abb. 55 Spezialtastatur

**Die Richtlinien des W3C**

Das W3C erarbeitet die Standards für das Internet, u. a. Hypertext Markup Language (HTML) für die Kodierung von Inhalten, Cascading Style Sheets (CSS) für Stilvorlagen, Scalable Vector Graphics (SVG) für Vektorgrafiken. Das W3C hat eine Web Accessibility Initiative (WAI), die auf die Barrierefreiheit der Webstandards achtet und Richtlinien zur barrierefreien Gestaltung herausgibt. Zentrale Richtlinien der

WAI sind:

- die Web Content Accessibility Guidelines (WCAG) zur barrierefreien Gestaltung der Inhalte
- die Authoring Tools Accessibility Guidelines (ATAG), die u. a. Online-Editoren betreffen
- die User Agent Accessibility Guidelines (UAAG) für die barrierefreie Funktionalität des Browsers bzw. des Anzeigegerätes
- Für die Anpassung von Web-Anwendungen an assistive Technologien gibt es den Standard Accessible Rich Internet Applications (WAI-ARIA).

### WCAG 2.0

Die Web Content Accessibility Guidelines (WCAG) bestehen seit 1999. Die zweite Version WCAG 2.0, verabschiedet im Dezember 2008, wurde grundlegend neu strukturiert, um der raschen technischen Entwicklung im Internet standhalten zu können.

Das hierarchisch aufgebaute Regelwerk basiert auf vier Gestaltungsprinzipien – wahrnehmbar, bedienbar, verständlich und auf einem technischen Prinzip: robust. Es ist in 12 Richtlinien und 61 Erfolgskriterien aufgefächert. Die Erfolgskriterien sind testbar und in die drei Konformitätsstufen A, AA und AAA differenziert. Darunter stehen technische Dokumente, die nicht mehr zu den verbindlichen Richtlinien gehören, sondern informativen Charakter haben. Jedem Erfolgskriterium sind ausreichende und empfohlene Techniken zugeordnet.

Die Richtlinien gelten für alle im Internet gebräuchlichen Inhaltsformate. Während WCAG 1.0 die W3C-eigenen Formate HTML und CSS voraussetzte und den Gebrauch von Fremdformaten und Programmiersprachen unter strenge Auflagen stellte, sind seit WCAG 2.0 auf barrierefreien Websites auch Dokumente in PDF und Anwendungen in JavaScript und Flash® erlaubt, sofern sie barrierefrei gestaltet sind.

## 12 Richtlinien nach WCAG 2.0

### 1. Wahrnehmbar

- 1.1. Textalternativen für Nicht-Text-Inhalte
- 1.2. Alternativen für zeitbasierte Medien
- 1.3. Anpassbar an verschiedene Darstellungsmodi
- 1.4. Unterscheidbare Darstellung von Inhalten

### 2. Bedienbar

- 2.1. Vollständig per Tastatur zugänglich
- 2.2. Ausreichend Zeit
- 2.3. Anfälle vermeiden (kein Blitzen)
- 2.4. Übersichtliche Navigation

### 3. Verständlich

- 3.1. Leseverständnis fördern
- 3.2. Vorhersehbar (Konsistente Darstellung)
- 3.3. Hilfestellung bei der Eingabe

### 4. Robust

- 4.1. Kompatibel mit allen Browsern und mit Hilfstechniken

### Die deutsche Barrierefreie-Informationstechnik-Verordnung – BITV 2.0

Die Novelle der BITV vom September 2011 (BITV 2.0) entspricht in den technischen Regeln im Wesentlichen den WCAG 2.0. Einige Regelungen der BITV weichen von der internationalen Richtlinie ab:

- Es gilt ein höheres Anforderungsniveau. Die Erfolgskriterien werden nicht wie in WCAG zu den drei Konformitätsstufen A, AA und AAA zusammengefasst, sondern zu zwei Stufen, Priorität I und II. BITV Priorität I umfasst WCAG-Level A und AA, Priorität II umfasst zusätzlich Level AAA. Alle Angebote im Bereich der BITV müssen Priorität I erfüllen, „zentrale Navigations- und Einstiegsangebote“, z. B. Portalseiten, müssen Priorität II erfüllen. Das Einstiegsniveau nach WCAG Level A ist im Bereich der BITV nicht vorgesehen.

- Einfache Sprache: Anstelle der Richtlinie 3.1.5 „reading level“ wird die aus WCAG 1.0 bekannte Formulierung „Einfache Sprache“ eingesetzt. Die Zuordnung des Sprachniveaus zu einem deutschen Schulabschluss wird hierdurch für die weitere Diskussion offen gehalten.
- Ergänzend zu den Bestimmungen der WCAG wird die Bereitstellung von Informationen in Leichter Sprache und in Deutscher Gebärdensprache verlangt, um den besonderen Belangen gehörloser, hör-, lern- und geistig behinderter Menschen Rechnung zu tragen.

### Weitere Gestaltungsrichtlinien

WCAG 2.0 war wegweisend für weitere Richtlinien zur barrierefreien Gestaltung von Informationstechnik. Die neue Beschaffungsrichtlinie EN 301549 „Accessibility requirements suitable for public procurement of ICT products and services in Europe“ übernimmt WCAG 2.0 Level AA für Internetseiten und, in angepasster Form, ebenfalls für nicht-webbasierte Dokumente und Software.

Für barrierefreie PDF-Dokumente gibt es den ISO-Standard 14289-1 PDF/UA (Universal Accessibility), der die Konzepte von WCAG 2.0 auf PDF anwendet.

Für barrierefreie Software gibt es in der Normenreihe DIN EN ISO 9241 „Ergonomie der Mensch-System-Interaktion“ den Teil 171 „Leitlinien für die Zugänglichkeit von Software“.



Abb. 57 Eingabeformular bei 200 % Textvergrößerung. 1) Überlagerung – das darf nicht passieren. 2) Besser: fluides Layout mit Silbentrennung. 3) Im Smartphone ist die einspaltige Darstellung von Formularen üblich.

## 10.6 Technische Umsetzung der Gestaltungsrichtlinien

WCAG 2.0 ist eine Richtlinie an der Schnittstelle von Ergonomie und Technik. In den vier Prinzipien spiegeln sich die Anforderungen von Menschen mit Behinderungen, die mit eingeschränkten Fähigkeiten und assistiven Technologien das Internet auf vollwertige Weise nutzen wollen.

Barrierefreiheit gilt als Ausweis für qualitativ hochwertiges Webdesign, vor allem, weil die technische Umsetzung der Richtlinien eine korrekte Anwendung der Webstandards verlangt. Die ergonomische Zielsetzung und der Anwendungskontext sind jedoch bei den Webentwicklerinnen und -entwicklern noch nicht allgemein bekannt, wie die häufigen Fehler in der Implementierung der empfohlenen Techniken zeigen. Im Folgenden sollen der technische Aspekt der Barrierefreiheit und das Zusammenspiel von Technik und Ergonomie an einigen Beispielen verdeutlicht werden.

### Alternativtexte

„Alternativtexte für Bilder“ ist die bekannteste Anforderung an barrierefreies Webdesign. Für alle Bildelemente, seien es Symbole, Infografiken oder Fotos, müssen Alternativtexte vorhanden sein. Die Alternativtexte sollen funktional äquivalent sein, also den Inhalt oder die Funktion des Bildes wiedergeben, sodass die Information auch dann zur Verfügung steht,

wenn das Bild ausfällt, etwa weil es bei geringer Bandbreite nicht geladen wurde oder weil Blinde es nicht wahrnehmen können.

Alternativtexte müssen sinnvoll und angemessen sein. Der genaue Wortlaut ist eine Aufgabe der User-Interface-Designerin bzw. des User-Interface-Designers oder der Autorin bzw. des Autors. Der Alternativtext soll den Inhalt des Bildes knapp benennen. Bei verlinkten Bildern (Icons, Logos, Schaltflächen) soll der Alternativtext das Ziel des Links bzw. die ausgelöste Aktion benennen. Rein dekorative Bilder mit nur unbedeutendem Inhalt werden durch ein leeres ALT-Attribut vor dem Screenreader verborgen. Informative Bilder (Diagramme, Fotos) benötigen eine ausführliche Beschreibung, die über die knappe Benennung im ALT-Attribut hinausgeht. Sinnvolle Alternativtexte trifft man also auf Internetseiten vor allem dann an, wenn die Beschäftigten der gesamten Produktionskette, vom User-Interface-Design über die HTML-Entwicklung bis zur Redaktion, in den Anforderungen der barrierefreien Gestaltung geschult ist.

### Semantische Strukturierung

Damit Inhalte auf verschiedene Weise dargestellt werden können, ohne dass Information oder Struktur verloren geht, ist eine semantisch korrekte Kodierung in den passenden HTML-Elementen erforderlich. Diese Anforderung gilt für die

Inhalte im Detail wie auch für den Aufbau der ganzen Seite. Texte werden als Absätze <p>, Listen <li>, Überschriften <h1> bis <h6> ausgezeichnet (getaggt). So können Screenreaderbenutzerinnen oder -benutzer mit einem Tastaturshortcut von einer Überschrift zur nächsten springen und sich einen Überblick über die Inhalte verschaffen. Sie erfahren die Anzahl von Elementen in einer Liste und die Art der Eingabefelder in einem Formular. Für die Funktionsbereiche einer Seite, z. B. Navigation, Hauptinhalt und Randspalte, gibt es seit HTML5 ebenfalls die passenden semantischen Elemente.

Semantisch korrektes HTML ist ein Hauptmerkmal der standardkonformen Programmierung, ist aber in der Praxis des Webdesigns noch nicht in voller Konsequenz angekommen. Weithin durchgesetzt hat sich bisher die Regel, Tabellen <table> nur für Datentabellen zu verwenden und nicht zu Layout-Zwecken zu missbrauchen. Vielfach wird bereits die Verwendung von Stylesheets (softwaremäßige Formatvorlagen) als „standardkonform“ bezeichnet, was im Sinne der Barrierefreiheit zu kurz gegriffen ist.

### Vergrößerung

Die oft gestellte Frage nach einer ausreichenden Schriftgröße wird aus Sicht der Barrierefreiheit nicht beantwortet, hierfür ist auf die allgemeine Softwareergonomie (vgl. Kapitel 5) zurückzugreifen. WCAG 2.0 fordert stattdessen, dass alle Inhalte bis

zu 200 % vergrößert werden können. Diese Regel unterstützt Menschen mit einer leichten Sehbehinderung, die noch nicht auf eine Bildschirmlupe angewiesen sind.

Die Vergrößerung kann auf verschiedene Weise implementiert werden, mit mehr oder weniger Komfort bei der Benutzung. Die einfachste Form ist der Seitenzoom, eine Funktion des Browsers, die keine besonderen Eigenschaften der Website voraussetzt. Hierbei wird die ganze Seite proportional vergrößert. Wenn die verfügbare Fläche ausgeschöpft ist, überragt die Seite den Bildschirm und man muss seitwärts scrollen, was die Orientierung und das Leseverständnis erschwert. Ein weitergehendes Erfolgskriterium für Level AAA / Priorität II ist daher, dass die Inhalte bei einer Vergrößerung von 200 % ohne Seitwärtsscrollen gelesen werden können. Dieses Verhalten ist mit der Textvergrößerung erreichbar, die nur die Schrift vergrößert, während die Inhalte nach unten umbrechen und nicht breiter als der Bildschirm werden. Dieses fluide Layout basiert auf einer Vielzahl von CSS-Techniken, u. a. auf dem Einsatz relativer Schriftgrößen und der relativen Bemaßung von Abständen.

Eine konsequente Fortsetzung findet das fluide Layout im Responsive Design, der Anpassung von Inhalten an sehr verschiedene Anzeigegeräte wie Smartphones, Tablets und Breitbildmonitore. Hierbei werden in Abhängigkeit von der Bildschirmgröße verschiedene Layouts definiert, u. a. eine einspaltige Darstellung für Smartphones. Responsive Design ist eines der zentralen Themen des aktuellen Webdesigns und ein Paradebeispiel dafür, wie barrierefreie Praktiken sich für die Mehrheit der Benutzerinnen und Benutzer auszahlen.

### Tastaturbedienung

Motorisch eingeschränkte Menschen und Blinde sind zur Bedienung von Software auf die Tastatur angewiesen, daher muss alle Funktionalität vollständig mit der

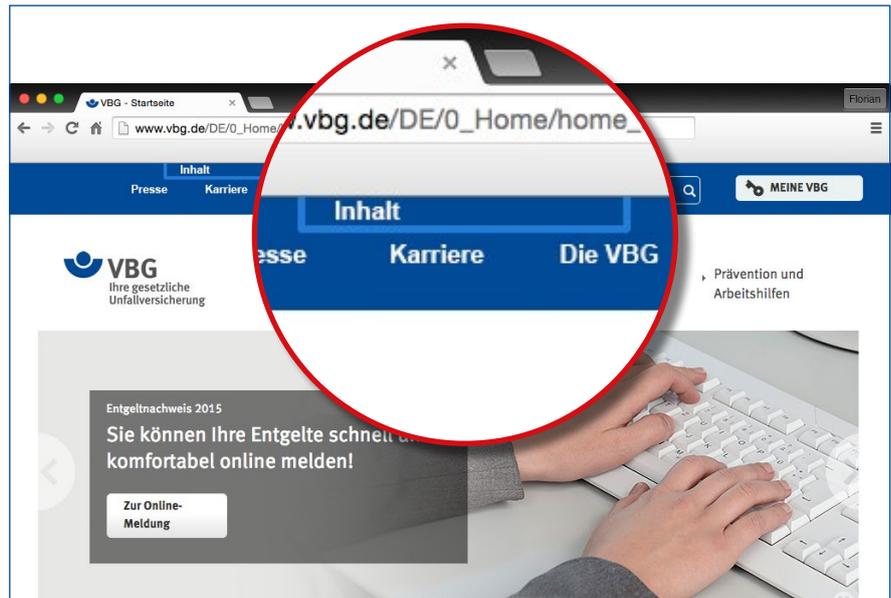


Abb. 58 Sprungmarke „Zum Inhalt“ wird sichtbar bei Aktivierung mit der Tastatur.

Tastatur erreichbar sein. HTML-Seiten sind für Tastaturbedienung eingerichtet. Links und Formularelemente können mit der Tab-Taste ausgewählt (fokussiert) und mit Enter oder Space aktiviert werden. Ergänzend können Sprungmarken eingesetzt werden, um Gruppen von Links zu überspringen und direkt zum Ziel zu kommen (Abbildung 58). Shortcuts sind auf HTML-Seiten nur begrenzt einsetzbar, da das dafür vorgesehene access-key-Attribut mit den Shortcuts der diversen Browser konkurrieren muss.

Die HTML-Ausstattung für Tastaturbedienung genügt nicht mehr, wenn große Portalsysteme oder komplexe Web-Anwendungen zu bewältigen sind. Mit Javascript und WAI-ARIA kann eine differenzierte Tastatursteuerung aufgebaut werden, wie sie aus Desktop-Anwendungen bekannt ist. Entsprechende Ansätze gibt es in den bekannten Javascript-Bibliotheken.

Eine effiziente Tastaturbedienung erfordert, dass

- eine sinnvolle Tab-Reihenfolge besteht,
- die Auswahl gut sichtbar ist,
- die Auswahl bei allen Bedienschritten mitgeführt wird,
- die Tastaturbefehle gut merkbar und dokumentiert sind.

### Technische Robustheit

Internetseiten sind robust, wenn man sie mit allen Browsern und Anzeigegegeräten

und insbesondere auch mit assistiven Technologien nutzen kann, ohne dass Inhalte oder Funktionen verloren gehen.

Eine Voraussetzung für die Kompatibilität mit einer Vielzahl von Technologien ist die formale Richtigkeit der Kodierung. **Valides HTML**, das den Syntaxcheck des W3C besteht, ist also ein Erfolgskriterium für barrierefreie Websites.

Die Bedeutung von Bedienelementen muss in den Dimensionen **Name, Rolle, Wert** programmatisch erkennbar sein, damit sie von assistiven Technologien ausgewertet werden kann. HTML bringt diese Qualität von Haus aus mit, wie oben unter „Semantische Strukturierung“ ausgeführt wurde. In gleicher Weise müssen sich auch andere Formate zu erkennen geben. Auch Dokumente in PDF und E Books in E PUB müssen „getaggt“ sein, also auf programmtechnisch erkennbare Weise in Überschriften, Absätze, Listen etc. gegliedert sein. Anwendungen in Javascript, Flash® oder anderen Programmiersprachen müssen, ähnlich wie HTML-Formulare, ihre Bedienelemente programmatisch erkennbar machen. Von Vorteil ist z.B., wenn ein Javascript-Widget wie die Baumstruktur oder der Schieberegler mit den semantisch passenden HTML-Elementen aufgebaut wird. Ergänzend stehen WAI-ARIA-Attribute zur Verfügung, um weitere Semantik und Verhalten hinzuzufügen.

## 10.7 Qualitätssicherung

Wie kann man sicherstellen, dass Software in barrierefreier Qualität zur Verfügung steht? Die Weichen werden bereits bei der Auswahl der geeigneten technischen Plattform gestellt. Ressourcen wie Best Practices, Tutorials und Prüftools sind eine wichtige Voraussetzung dafür, dass qualifiziertes Personal für Entwicklung und Qualitätssicherung aufgebaut werden kann.

### 10.7.1 Technische Formate und Plattformen

Während die Gestaltungsrichtlinien für barrierefreie Software technikübergreifend formuliert werden können, sind die verschiedenen technischen Formate und Plattformen sehr unterschiedlich auf die Barrierefreiheit vorbereitet. Wichtig ist, dass leistungsfähige Schnittstellen für assistive Technologien zur Verfügung stehen. Am weitesten entwickelt sind die Voraussetzungen im Internet.

#### Websites

Informationsangebote im Internet in Text, Bild, Ton und Film können heute unter Anwendung der Gestaltungsrichtlinien von WCAG 2.0 nahezu vollständig barrierefrei gestaltet werden. Die Basis hierfür ist semantisch korrektes HTML, das für assistive Technologien zugänglich ist. Bei der Umsetzung hilft eine Vielzahl von Internetressourcen, u.a. das Portal „Einfach für Alle“ der Aktion Mensch, das die im Web verfügbaren Anleitungen, Musterlösungen und Prüftools sammelt. Ein praxistaugliches Verfahren für die Qualitätssicherung steht u.a. mit dem BITV-Test zur Verfügung.

#### Web-Anwendungen

Moderne browserbasierte Software verwendet im User-Interface die Formate HTML5, CSS3, JavaScript und AJAX. Barrierefreiheit wird durch die Anwendung der WCAG sichergestellt, sowie zusätzlich durch Einsatz von WAI-ARIA zur Beschreibung der dynamischen Features für

assistive Technologien. Während die statischen Aspekte von Web-Anwendungen mit Websites vergleichbar sind, ist in den dynamischen Aspekten der Stand der Barrierefreiheit heute noch relativ experimentell. Sowohl HTML5 als auch WAI-ARIA sind im Fluss, mit bisher uneinheitlicher Implementierung in Browsern und Screenreadern. Die bekannten JavaScript-Frameworks, u.a. jQuery, MooTools und Sencha Ext JS, verbessern laufend die Barrierefreiheit ihrer User-Interface-Elemente. Barrierefreie Web-Anwendungen sind zurzeit noch ein Pionierthema für Accessibility-Expertinnen und -Experten, die Konzepte werden in den einschlägigen Blogs diskutiert.

#### Andere technische Plattformen

Außerhalb der Browserwelt basiert die Entwicklung von barrierefreier Software auf den Accessibility-Features des jeweiligen Betriebssystems. Die Betriebssysteme unterstützen die Barrierefreiheit im Prinzip auf ähnliche Weise. Sie stellen barrierefreie Standardroutinen für Bedienoberflächen zur Verfügung. Darüber hinaus gibt es eine Accessibility-Schnittstelle, mit der die speziellen Bedienelemente von Anwendungssoftware für assistive Technologien beschrieben werden können.

Viele Programmiersprachen unterstützen die Accessibility-Schnittstelle des Betriebssystems. Konkrete technische Hinweise erhält die Entwicklung durch die von den Herstellungsfirmen der Programmiersprache herauszugebenden Gestaltungsregeln. Es gibt jedoch kaum produktspezifische Ratgeberliteratur oder Fachforen. Relativ gut unterstützt ist die Barrierefreiheit z. B. mit Java® und SAP®. Ein positives neueres Beispiel geben die für iOS® verfügbaren Richtlinien für barrierefreie Smartphone-Apps.

Wer assistive Technologien herstellt, hat sich in der Vergangenheit auf Microsoft Windows® konzentriert und liefert Skripte vorzugsweise für Standardsoftware in diesem Betriebssystem.

### 10.7.2 Qualifikation

Barrierefreie Gestaltung gehört bisher nicht zu den regulären Ausbildungsgängen der IT- und Designberufe und wird an den Hochschulen nur sporadisch gelehrt. Entsprechend unsicher ist die Qualität von Dienstleistungsangeboten zur Barrierefreiheit. Im Bereich des BITV-Tests gibt es ein Verzeichnis der Agenturen, die bereits als gut zugänglich bewertete Internetauftritte erstellt haben. Im März 2014 hat sich mit der International Association of Accessibility Professionals (IAAP) ein internationaler Berufsverband der Accessibility-Expertinnen und -Experten gegründet.

### 10.7.3 Entwicklungsprozess

Barrierefreiheit ist so grundlegend, dass sie in einer Software zumeist nicht oder nur mit großem Aufwand nachgerüstet werden kann. Ein gutes Ergebnis kann man nur erwarten, wenn bereits in der Konzeptionsphase die Anforderungen von Menschen mit Behinderungen berücksichtigt werden.

#### Design-Prozess

Für die Entwicklung barrierefreier Software empfiehlt sich besonders das Vorgehen nach dem Human Centered Design Prozess, ergänzt um die Zielgruppe der Menschen mit Behinderungen:

- Usability-Richtlinien werden um die Anforderungen der Barrierefreiheit erweitert.
- Arbeitsabläufe berücksichtigen auch den Gebrauch von assistiven Technologien, z. B. Vergrößerungssoftware und Screenreader.

Verschiedene Nutzungsstrategien werden berücksichtigt, z. B. ausschließliche Tastaturbedienung.

- Usability-Tests werden auch mit Menschen mit Behinderungen durchgeführt.

#### *Entwicklungsbegleitende Tests*

Zur Erfolgskontrolle während der Entwicklung bietet sich der Test von Mustervorlagen und Teilprodukten an.

Für Webseiten steht eine Vielfalt von Prüf-Tools zur Verfügung. Web-Anwendungen sollten darüber hinaus mit assistiven Technologien getestet werden.

Der Accessibility-Test von Desktop-Software besteht im Wesentlichen in der Bedienung der Software mit der Tastatur, mit den Accessibility-Optionen des Betriebssystems und mit assistiven Technologien. Für die Fehleranalyse gibt es Tools zur Abbildung der Accessibility-Schnittstelle. Eine praktische Handreichung für die Entwicklung liegt in der technikübergreifenden IBM-Checkliste für barrierefreie Software vor.

#### 10.7.4 Zertifizierung

Prüfsiegel sind bisher ausschließlich für barrierefreie Websites etabliert. Auf nationaler Ebene sind u.a. der deutsche BITV-Test und das schweizerische Zertifikat der Stiftung „Zugang für alle“ zu nennen.

Der BITV-Test ist ein Prüfverfahren für Websites zur Feststellung der Konformität mit der Barrierefreie-Informationstechnik-Verordnung (BITV). Das Verfahren besteht seit 2002 und wurde im Auftrag der Bundesregierung vom BIK-Projekt entwickelt, einem Konsortium der deutschen Blindenverbände mit einem privaten Testinstitut. Die Prüfschritte verwenden frei verfügbare Prüf-Tools und sind vollständig dokumentiert. Die Bewertung erfolgt nach einem Punktesystem, insgesamt können 100 Punkte erreicht werden.

Ab 90 Punkten wird ein Webauftritt als „gut zugänglich“ bewertet, ab 95 Punkten als „sehr gut zugänglich“. Der BITV-Test ist für entwicklungsbegleitende Tests frei verfügbar. Das Prüfsiegel „90plus“ wird von lizenzierten Testinstituten erteilt, die im BITV-Testkreis zusammengeschlossen sind.

Eine internationale Harmonisierung von Zertifikaten wird durch Richtlinien zum Aufbau von Testverfahren eingeleitet. Vom W3C wurden entsprechende Empfehlungen unter dem Titel „Website Accessibility Conformance Evaluation Methodology (WCAG-EM)“ im Juli 2014 veröffentlicht. Eine Harmonisierung der Testverfahren in Europa wird im Rahmen der Beschaffungsrichtlinie EN 301 549 erwartet.



#### Empfehlenswerte Prüf-Tools

für den Test von Internetseiten

- Web Accessibility Toolbar - ► [www.visionaustralia.org/digital-access-wat](http://www.visionaustralia.org/digital-access-wat)
- W3C-Validator - ► <http://validator.w3.org>
- Color Contrast Analyzer - ► [www.visionaustralia.org/digital-access-cca](http://www.visionaustralia.org/digital-access-cca)

für den Test von PDF-Dokumenten

- PDF Accessibility Checker (PAC) - ► [www.access-for-all.ch/ch/pdf-werkstatt/pdf-accessibility-checker-pac.html](http://www.access-for-all.ch/ch/pdf-werkstatt/pdf-accessibility-checker-pac.html)

für den Test von Software unter MS Windows®

- Inspect - ► [msdn.microsoft.com/en-us/library/windows/desktop/dd318521%28v=vs85%29.aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd318521%28v=vs85%29.aspx)

# 11 Nutzungsqualität und Überprüfung



Abb. 59  
Einsatzzyklus von Software

## 11.1 Nutzungsqualität im Einsatzzyklus

Software ist ein Arbeitsmittel. Ihre Nutzungsqualität ist hoch, wenn sie die Beschäftigten bei der Bearbeitung ihrer Aufgaben in allen Phasen des Einsatzzyklus (Abbildung 59) optimal unterstützt.

Eine hohe Nutzungsqualität von Software bezieht sich auf Merkmale, die eine ergonomische Gestaltung von Software auszeichnen. Zur Überprüfung und Bewertung von Nutzungsqualität gehören die folgenden Prozesse:

- Arbeitssysteme nach ergonomischen Grundsätzen gestalten (DIN EN ISO 6385)
- Nutzeraufgaben ableiten (DIN EN 29241-2, DIN EN 614-2, DIN EN ISO 11064-1)
- Benutzungsschnittstellen gestalten (DIN EN ISO 9241-210)

Das Prüfen und Bewerten der Nutzungsqualität soll gravierende Mängel einer ergonomischen Gestaltung, gravierende Gefährdungen und Beeinträchtigungen der Benutzerinnen bzw. der Benutzer bereits in frühen Phasen des Einsatzzyklus aufdecken. Maßnahmen und

Lösungsvorschläge zur Mängelbeseitigung müssen wie bei Risiko- und Gefährdungsbeurteilungen entwickelt und dann erneut überprüft und bewertet werden.

Es gibt verschiedene Gründe, weshalb die Nutzungsqualität von Software nicht pauschal beurteilt werden kann:

- Geprüft und bewertet werden kann nur für einen spezifizierten Nutzungskontext.
- Einzelkomponenten in Nutzungskontexten ändern sich und dadurch verändert sich ein vormals spezifizierter Nutzungskontext.
- Makellose Qualität von Software für jeden beliebigen Einsatzzweck ist nicht erreichbar.
- Rechtliche Vorgaben benennen Mindestanforderungen und Methoden nur unvollständig.
- Nicht alle Anforderungen sind bekannt und daher können nie alle erfüllt werden.
- Die ausreichende Benutzerzahl für eine korrekte Methoden- bzw. Testauswertung wird meist nicht erreicht.

Jede Phase des Einsatzzyklus von Software beinhaltet unterschiedliche Aufträge (siehe Tabelle 1), die im Projektmanagement abgearbeitet werden sollten. Die Projektleitung sollte zumindest sehr

vertraut mit der Fachabteilung sein, in der die Software genutzt werden soll. Die Fachabteilung kann beschreiben, wie sich Veränderungen von Arbeitsprozessen auf Veränderungen bei der Software auswirken sollen und umgekehrt. Die Projektgruppe setzt sich je nach Phase im Einsatzzyklus der Software anders zusammen. In der Projektgruppe sollten Benutzerinnen und Benutzer, Beschäftigte von Fachabteilung und IT-Abteilung, Mitglieder des Betriebs- und Personalrates, Fachkraft für Arbeitssicherheit (Sifa), Betriebsärztin oder Betriebsarzt, Softwareentwicklerinnen und -entwickler, Arbeits- und Organisationsgestalterinnen oder -gestalter und weitere Expertinnen bzw. Experten nach Bedarf mitarbeiten.

**Tabelle 1** Phasen des Einsatzzyklus von Software

Phase	Aufträge	Beteiligte
Planung/ Konzeption	<ul style="list-style-type: none"> <li>• Projektmanagement aufsetzen</li> <li>• Planungsprozess organisieren</li> <li>• Ziele, Grenzen, Kriterien zur Analyse und Gestaltung des Arbeitssystems und der Arbeitsaufgaben entwickeln</li> <li>• Nutzungskontext spezifizieren</li> <li>• zwischen Softwarekauf, -neuentwicklung, -anpassung wählen</li> <li>• Erfordernisse aus den Aufgaben ableiten</li> <li>• Lastenheft erstellen</li> <li>• Anforderungen an die Software spezifizieren</li> <li>• Pflichtenheft erstellen</li> <li>• Ergebnisse dokumentieren</li> </ul>	Fachabteilung, Benutzerinnen/Benutzer, Betriebsratsmitglieder, Betriebsärztin/Betriebsarzt, Sifa, Arbeits-/Organisationsgestalterin/-gestalter, weitere Expertinnen/Experten (z. B. IT-Abteilung, Projektmanagement)
Entwicklung und Überprüfung	<ul style="list-style-type: none"> <li>• Entwicklungsprozess dokumentieren</li> <li>• Anforderungen an Prototypen und Produkttests spezifizieren</li> <li>• Prüfkriterien spezifizieren</li> <li>• Prüfkonzepte, Methoden, Verfahren, Tests vorbereiten, durchführen und dokumentieren</li> <li>• Ergebnisse dokumentieren</li> </ul>	Fachabteilung, Benutzerinnen/Benutzer, Betriebsärztin/Betriebsarzt, Sifa, Arbeits-/Organisationsgestalterinnen und -gestalter, Software-Programmiererinnen/-Programmierer, weitere Expertinnen/Experten
Einführung	<ul style="list-style-type: none"> <li>• Einführungsprozess organisieren</li> <li>• Rollout begleiten</li> <li>• Benutzerbezogene Einweisungen und Schulungen organisieren</li> <li>• Benutzerinnen/Benutzer beraten (Hotline, Software-Support)</li> <li>• Beschwerdemanagement bezogen auf Aufgabe und Software (Fachabteilung, IT-Abteilung) umsetzen</li> <li>• Ergebnisse dokumentieren</li> </ul>	Fachabteilung, Benutzerinnen/Benutzer, Betriebsärztin/Betriebsarzt, Sifa, Arbeits-/Organisationsgestalterinnen und -gestalter, IT-Abteilung, weitere Expertinnen/Experten (für z. B. Schulung, Software-Administration)
Nutzung und Pflege	<ul style="list-style-type: none"> <li>• Änderungsmanagement umsetzen</li> <li>• Mittel- und langfristigen Änderungsbedarf sammeln</li> <li>• Lösungen umsetzen</li> <li>• Ergebnisse dokumentieren</li> </ul>	Fachabteilung, Benutzerinnen/Benutzer, Betriebsärztin/Betriebsarzt, Sifa
Ablösung	<ul style="list-style-type: none"> <li>• Ablöseprozess organisieren</li> <li>• Koordination mit parallel startenden Arbeitsstrukturen</li> <li>• Ergebnisse dokumentieren</li> </ul>	Fachabteilung, Benutzerinnen/Benutzer, Sifa, Arbeits-/Organisationsgestalterinnen und -gestalter

**11.2 Das Arbeitssystem für die Aufgaben**

Zum Prüfen und Bewerten von Gestaltungslösungen in einem Arbeitssystem werden arbeitswissenschaftliche Kriterien herangezogen (Abbildung 60), die auch für die Softwareergonomie gültig sind:

- Ausführbarkeit bewertet, ob die Aufgaben immer vollständig mithilfe des Werkzeugs Software ausgeführt oder bearbeitet werden können.
- Schädigungslosigkeit bewertet, ob eine Benutzerin oder ein Benutzer während der Aufgabenbearbeitung mithilfe der Software geschädigt oder behindert wird.
- Beeinträchtigungsfreiheit bewertet, ob eine Benutzerin bzw. ein Benutzer

während der Aufgabenbearbeitung mithilfe der Software behindert oder die Informationsverarbeitung während des Arbeitsprozesses erschwert oder gestört wird.

- Gesundheitsförderlichkeit bewertet, ob eine Benutzerin oder ein Benutzer während der Aufgabenbearbeitung mithilfe der Software gesundheitsförderliche Bedingungen erfährt.

Diese Kriterien haben den Vorteil, dass sie in den Arbeitswissenschaften traditionell angewendet werden. Sie werden auch zum Ableiten von Nutzeraufgaben und zum Gestalten von Benutzungsschnittstellen genutzt und kennzeichnen die Gebrauchstauglichkeit.

Bei Gebrauchstauglichkeit handelt es sich um das Ausmaß, in dem ein Produkt in einem spezifizierten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.

Im Zentrum der Gestaltung eines Arbeitssystems steht die Aufgabe. Eine Aufgabe beschreibt, was Nutzerinnen und Nutzer mit Arbeitsmitteln in ihrer Umgebung erreichen sollen (z. B. Beschäftigte nehmen eine Schadensmeldung mithilfe von Software bei Kundinnen und Kunden auf). Aus den Ausführungsbedingungen für die Aufgaben ergeben sich Anforderungen an die Arbeitsmittelgestaltung (z. B. Software muss lesbare Informationsdarstellung auf dem Bildschirm eines Notebooks

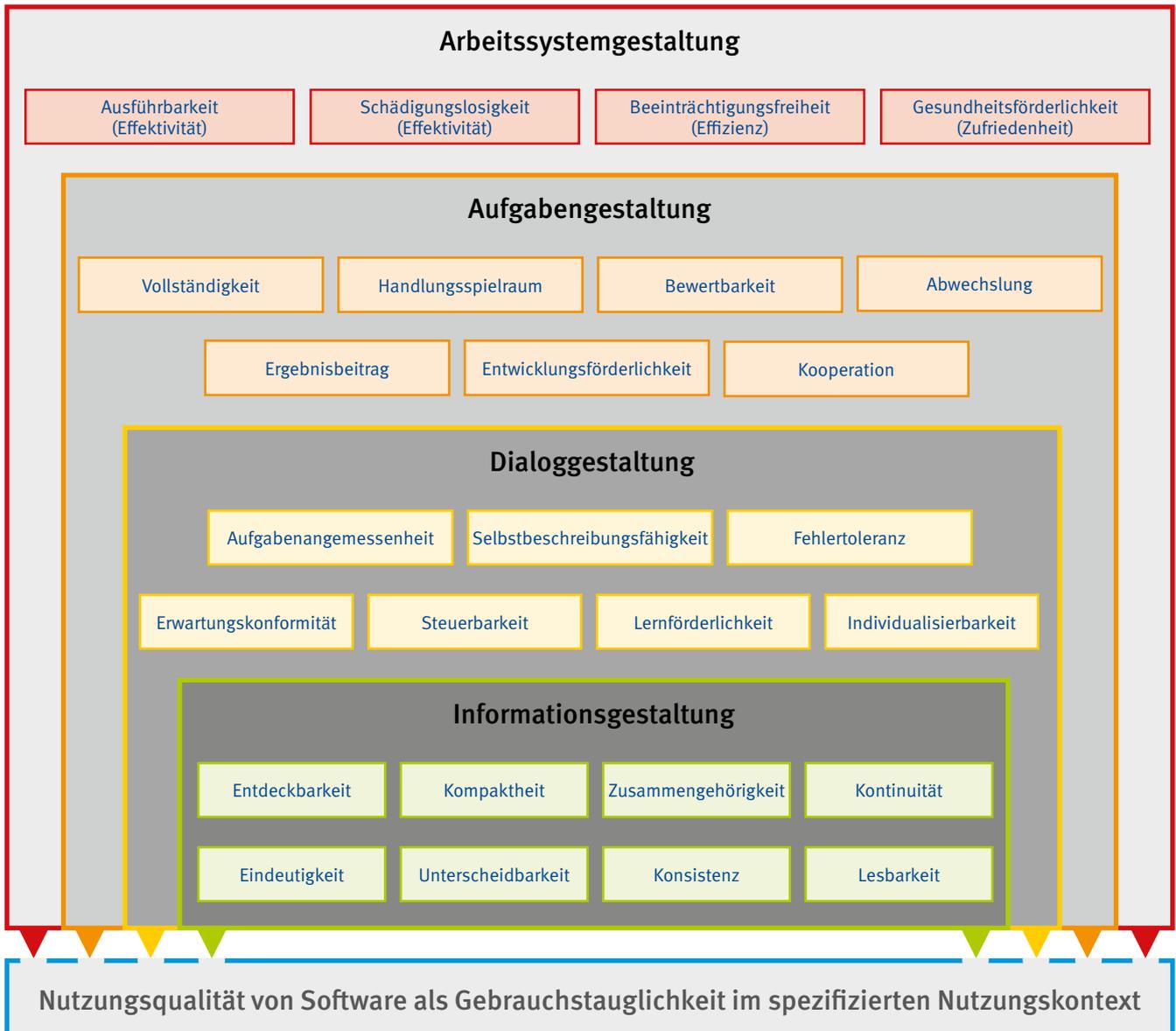


Abb. 60 Gestaltungshierarchie in der Softwareergonomie

ermöglichen). Zur Aufgabenbearbeitung sind auch Qualifikationen der Beschäftigten erforderlich (z. B. alle relevanten Informationen zur Schadensbewertung erfragen und dokumentieren können). Durch die Aufgaben werden die Bedingungen für die Aufgabenbearbeitung mithilfe der Software mit den Eigenschaften und Fähigkeiten der Benutzerin bzw. des Benutzers verbunden.

Mithilfe der Prinzipien der Aufgabengestaltung werden Aufgaben entwickelt, die an die physischen und psychischen Anforderungen des Menschen angepasst sind (Abbildung 60). Für einen Teil dieser Aufgaben sind Benutzungsschnittstellen so zu gestalten, dass die Aufgaben nachträglich nicht eingeschränkt werden. Dafür werden Prinzipien der Dialog- bzw. Interaktionsgestaltung und Prinzipien der Informationsgestaltung berücksichtigt

(siehe Kapitel 8 „Informationsdesign“ und Kapitel 9 „Interaktionsdesign“). Diese Prozesse der ergonomischen Gestaltung sind hierarchisch aufgebaut (Abbildung 60). Bereits durch das Gestalten von Aufgaben werden Arbeitsinhalte und -abläufe vorbestimmt. Die Interaktions- und Informationsgestaltung können Mängel der Aufgabengestaltung nicht mehr kompensieren.

Tabelle 2 Merkmale des Nutzungskontextes: Beispiele in Anlehnung an DIN EN ISO 9241-11

Benutzer	Arbeitsaufgabe	Arbeitsmittel
<ul style="list-style-type: none"> <li>• <b>Benutzertyp</b></li> <li>• repräsentative Benutzer</li> <li>• nicht repräsentative und indirekte Benutzer</li> <li>• <b>Fertigkeit und Wissen</b></li> <li>• Erfahrung mit dem Produkt</li> <li>• Erfahrung mit dem System</li> <li>• Erfahrung mit der Arbeitsaufgabe</li> <li>• Erfahrung mit der Organisation</li> <li>• Übungsgrad</li> <li>• Fertigkeiten mit dem Eingabemittel</li> <li>• Qualifikationen</li> <li>• Sprachfertigkeiten</li> <li>• allgemeine Kenntnisse</li> <li>• <b>Persönliche Merkmale</b></li> <li>• Alter</li> <li>• Geschlecht</li> <li>• physische Grenzen und Behinderungen</li> <li>• intellektuelle Fähigkeiten</li> <li>• Einstellungen</li> <li>• Motivation</li> </ul>	<ul style="list-style-type: none"> <li>• Aufgabenzerlegung</li> <li>• Aufgabenbezeichnung</li> <li>• Aufgabenhäufigkeit</li> <li>• Aufgabendauer</li> <li>• Häufigkeit von Ereignissen</li> <li>• Handlungsspielraum</li> <li>• physische und psychische Anforderungen</li> <li>• Aufgabenabhängigkeiten</li> <li>• Aufgabenergebnisse</li> <li>• gefährliche Auswirkungen von Fehlern</li> <li>• sicherheitskritische Erfordernisse</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Allgemeine Beschreibung</b></li> <li>• Produktbezeichnung</li> <li>• Produktbeschreibung</li> <li>• Hauptanwendungsbereiche</li> <li>• wichtige Funktionen</li> <li>• <b>Spezifikation</b></li> <li>• Hardware</li> <li>• Software</li> <li>• Materialien</li> <li>• Dienstleistungen</li> <li>• Weiteres</li> </ul>

**11.3 Nutzungskontexte der Software spezifizieren**

Konkrete Anforderungen an die Gestaltung des Arbeitsmittels Software werden aus einem spezifizierten Nutzungskontext abgeleitet. Ein Nutzungskontext umfasst die Benutzerinnen und Benutzer, deren Arbeitsaufgaben, die Ausrüstung am Arbeitsplatz (Hardware, Software, weitere Arbeitsmittel und Materialien), die physische, informatorische und soziale Arbeitsumgebung, in der das Produkt Software genutzt wird. Zur Analyse, Spezifikation und Dokumentation des Nutzungskontextes für die Software ist ein systematisches Vorgehen hilfreich:

1. Alle Arbeitsszenarien für jede Phase im Einsatzzyklus von Software beschreiben.
2. Alle Ziele und Aufgaben im Arbeitssystem analysieren. Die Funktionen identifizieren, die ausgeführt werden sollen und solche, die von der Software unterstützt werden sollen. Die Prinzipien der Aufgabengestaltung anwenden.
3. Ein systematisches Vorgehen zur Aufgabengestaltung umsetzen und Nutzeraufgaben ermitteln. Anhand der Ergebnisse der Aufgabengestaltung, Nutzergruppen zusammenstellen und Qualifikationsanforderungen entwickeln.
4. Mit Ergebnissen der Aufgabengestaltung und ihrer Ausführungsbedingungen

gen die Nutzungskontexte für die Phasen des Einsatzzyklus spezifizieren und dokumentieren.

**1. Arbeitsszenarien im Einsatzzyklus**

Zunächst werden Arbeitsszenarien im Einsatzzyklus der Software gesammelt und später werden damit Nutzungskontexte spezifiziert. Eine Sammlung sollte zunächst anhand vorhandener Dokumente wie z. B. Organigrammen oder Betriebsabläufen begonnen werden. Anschließend sollte die Sammlung von moderierten Arbeitsgruppen mit den beteiligten Personengruppen (Abteilungsleitung, Benutzerinnen und Benutzer, Betriebsärztin oder Betriebsarzt, Sifa; siehe Kapitel 12 „Softwarekauf und Pflichtenheft“) vervollständigt werden. Das Ergebnis sind Arbeitsszenarien für jede Phase des Einsatzzyklus. Arbeitsszenarien aus der Nutzungsphase der Software sollten sich auf die jeweils kleinste organisatorische Einheit eines Betriebs (z. B. eine Unterabteilung, Arbeitsgruppe) beziehen, in der die Software Beschäftigte unterstützen soll.

**2. Vorgehen bei der Aufgabengestaltung**

Die Aufgabengestaltung ist der wichtigste Teil der Arbeitssystemgestaltung. Zunächst werden die Ziele des Arbeitssystems z. B. als Abteilung oder Arbeitsgruppe beschrieben, in der die Software zukünftig eingesetzt werden soll. Für eine Abteilung werden Arbeitsprozesse,

erwünschte und unerwünschte Arbeitsergebnisse sowie rechtliche, wirtschaftliche, ergonomische, personale, organisationale Grenzen und Rahmenbedingungen benannt. Vorhandene Dokumente wie z. B. Organigramme, Betriebsabläufe, Unternehmensziele oder Aufgabenszenarien sollten berücksichtigt werden. Zusammen mit Informationen der Geschäftsleitung und aus den Abteilungen, in denen die Software eingeführt und genutzt wird, sollten vollständige Ziele formuliert werden.

Aufgaben werden als Arbeitsprozesse in Funktionen/Teilaufgaben bzw. logische Einheiten von Verrichtungen aufgegliedert. Dabei kommen Funktionsorganigramme, Flussdiagramme, Zeitverlaufsanalysen und Netzwerk- oder Verknüpfungsanalyse zum Einsatz. Die Funktionen werden weiter verfeinert bis schließlich Funktionen der Arbeitsabschnitte den Benutzerinnen und Benutzern, der Software oder wahlweise beiden zugewiesen werden können. In diesem Arbeitsschritt ist eine Zuarbeit durch einzelne Abteilungen zu moderierten Arbeitsgruppen erforderlich.

Funktionen, die den Benutzerinnen und Benutzern zugeordnet wurden, werden so zu Aufgaben zusammengestellt, dass sie die Prinzipien der Aufgabengestaltung erfüllen (Tabelle 2). Die so vorgenommene Mensch-Software-Funktionsteilung

Umgebung		
Organisatorische Umgebung	Technische Umgebung	Physische Umgebung
<ul style="list-style-type: none"> <li>• <b>Struktur</b></li> <li>• Arbeitsstunden</li> <li>• Gruppenarbeit</li> <li>• Funktion</li> <li>• Arbeitspraxis</li> <li>• Hilfestellung</li> <li>• Unterbrechungen</li> <li>• Führungsstruktur</li> <li>• Kommunikationsstruktur</li> <li>• <b>Einstellung und Kultur</b></li> <li>• Vorschriften für den Umgang mit Computern</li> <li>• organisatorische Ziele</li> <li>• Geschäftsbeziehungen</li> <li>• <b>Arbeitsgestaltung</b></li> <li>• Mischarbeit</li> <li>• Leistungsmessung</li> <li>• Ergebnisrückmeldung</li> <li>• Selbständigkeit</li> <li>• Entscheidungsfreiheit</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Konfiguration</b></li> <li>• Hardware</li> <li>• Software</li> <li>• Referenzmaterial</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Arbeitsplatzbedingungen</b></li> <li>• atmosphärische Bedingungen</li> <li>• akustische Bedingungen</li> <li>• Wärmebedingungen</li> <li>• Wahrnehmungsbedingungen</li> <li>• Umgebungsstabilität</li> <li>• <b>Arbeitsplatzgestaltung</b></li> <li>• Größe und Einrichtung</li> <li>• Körperhaltung</li> <li>• Arbeitsplatz</li> <li>• <b>Arbeitsplatzsicherheit</b></li> <li>• Gesundheitsgefährdung</li> <li>• Schutzkleidung und -vorrichtungen</li> </ul>

geht in die weitere Analyse und Spezifikation der Nutzungskontexte ein. Mit den Aufgabenbeschreibungen kann man bereits Gefährdungen und Beeinträchtigungen beurteilen. Mit den Nutzeraufgaben können Aufgabenbeschreibungen der Arbeitsplätze erstellt werden. Auf der Basis der Nutzeraufgaben werden Qualifikationsanforderungen abgeleitet. Auf der Basis der Aufgabenbestandteile, die durch Software unterstützt werden, werden Anforderungen an die Gestaltung der Benutzungsschnittstellen abgeleitet. In diesem Arbeitsschritt ist eine Zuarbeit durch einzelne Abteilungen, Betriebsärztin oder Betriebsarzt und Fachkraft für Arbeitssicherheit sowie auch weiteren im Rahmen der Software-Einführung beauftragten Fachleuten und Beschäftigten zu moderierten Arbeitsgruppen hilfreich.

### 3. Nutzergruppen und Qualifikationsanforderungen

Mithilfe der Nutzeraufgaben werden Nutzergruppen zusammengestellt und Qualifikationsanforderungen entwickelt. Zukünftig erforderliche Qualifikationen werden mit vorhandenen verglichen und Anforderungen sowie Ziele für Schulungsmaßnahmen zur Einführung und Nutzung der Software abgeleitet. Die Zusammenstellung einer Nutzergruppe sollte eine moderierte Arbeitsgruppe entwerfen, die auch bei der Aufgabengestaltung mitarbeitet. Das Personalwesen kann unterstützend einbezogen werden.

### 4. Analyse, Spezifikation und Dokumentation der Nutzungskontexte

Je nach Phase des Einsatzzyklus (z. B. Entwickeln, Überprüfen, Einführen, Nutzen, Pflegen) wird die Software für unterschiedliche Zwecke eingesetzt (z. B. als Entwicklungsplattform, Testkörper, Schulungsgegenstand, Werkzeug zur Bearbeitung von Nutzeraufgaben). Von Nutzungskontexten sind unterschiedliche Nutzergruppen betroffen (z. B. Entwicklerinnen und Entwickler, Einkäuferinnen und Einkäufer, Benutzerinnen und Benutzer). Ein Teil der unterschiedlichen Aufgaben ist schließlich mithilfe der Software zu bearbeiten (z. B. Gestaltungsmängel identifizieren und verbessern, Benutzerinnen und Benutzer am Arbeitsplatz unterstützen).

Eine Software wird meist an mehreren Arbeitsplätzen eingesetzt, die sich einerseits unterscheiden und andererseits über die Zeit verändern. Der Nutzungskontext und die Anforderungen an eine ergonomische Gestaltung von Software ändern sich, sobald

- sie an verschiedenen Arbeitsplätzen eingesetzt wird,
- sie verschiedene Aufgaben unterstützen soll (Einzelbrief schreiben, Serienbrief erstellen),
- verschiedene Hardware eingesetzt wird (Desktop-PC, Notebook) oder
- Einsatzorte wechseln (Büro, Fertigungshalle, Beratung beim Kunden).

Inwieweit diese Einsatzbedingungen ergonomischen Grundsätzen der Gestaltung entsprechen oder andere Gefährdungen damit verbunden sind, muss für jeden neuen Kontext neu ermittelt werden. Durch jede Veränderung an den Aufgaben und Arbeitsbedingungen entsteht ein neuer Nutzungskontext. Die Gebrauchstauglichkeit der Software kann für einen spezifizierten Nutzungskontext geprüft und bewertet werden. Das kann eine erneute Anpassung der Software an leicht veränderte Nutzungskontexte erforderlich machen.

Folgende Aktivitäten tragen zur Beschreibung von Nutzungskontexten bei:

- Konkrete Ergebnisse beschreiben, die bei der Aufgabenbearbeitung erzielt werden,
- Nutzergruppen beschreiben und zusammenstellen,
- Nutzungskontexte beschreiben und zusammenstellen.

Die Zusammenstellung der Nutzungskontexte sollte in moderierten Arbeitsgruppen mit den Betroffenen aller Phasen des Einsatzzyklus vervollständigt werden. Als Ergebnis sollten Beschreibungen der Nutzungskontexte getrennt nach Nutzergruppen dokumentiert sein.

#### 11.4 Nutzungsanforderungen spezifizieren

Aus jedem Nutzungskontext (mit allen Komponenten des Arbeitssystems) werden Nutzungserfordernisse und -anforderungen abgeleitet.

1. Identifizieren von Erfordernissen aus Nutzungskontext, Wünschen und Zielen von Kundinnen/Kunden und Benutzerinnen/Benutzern,
2. Ableiten von Nutzungsanforderungen aus Erfordernissen,
3. Abgleich von Anforderungen mit Erfordernissen durch Bewerten von Unterschieden und Ableiten von Maßnahmen.

Erforderlich ist das, was für eine ergonomische Aufgabenbearbeitung notwendig ist, und das, was notwendig ist, die Aufgaben im Nutzungskontext angemessen bearbeiten zu können (Was ist wie und wann mit welchem Ergebnis zu tun?). Darauf bezieht sich eine Anforderung als das technisch Machbare (Was kann wie und wann mit welchem Ergebnis getan werden?).

Anforderungen beschreiben Funktionen, Abläufe und Vorgaben, die technisch umgesetzt werden können. Damit sollen Erfordernisse ausgefüllt und Aufgaben der Nutzergruppen im Nutzungskontext tatsächlich unterstützt werden. Anforderungen sollen aus der Perspektive der Nutzeraufgabe und nicht aus der der Softwarefunktionalität beschrieben werden.

Aus der Aufgabe „Zusammenfassen von Einzelgutachten zu einer Stellungnahme und formale und inhaltliche Endkontrolle durch die Benutzerin bzw. den Benutzer“ ergeben sich verschiedene Erfordernisse und Anforderungen. Es kann erforderlich sein, Textpassagen aus Einzelgutachten zusammenzufassen oder Teile direkt in die Stellungnahme zu übernehmen. Dies erfordert Vergleiche zwischen den Einzelgutachten und der Stellungnahme, bei

der eine parallele Informationsdarstellung zu empfehlen ist. Als Anforderung kann eine parallele Darstellung auf nebeneinander liegenden Bildschirmen ausreichender Größe und Auflösung abgeleitet werden. Eine Darstellung in unterschiedlichen Fenstern eines Bildschirms könnte dazu führen, dass zwischen unterschiedlichen Ansichten seriell gewechselt werden muss und/oder Informationen zu klein oder nur unvollständig dargestellt werden können.

Nutzungsanforderungen sind so weit zu spezifizieren und zu dokumentieren, dass eine Unterstützung der Benutzerinnen und Benutzer bei der Informationsaufnahme, -verarbeitung und -umsetzung durch die Software nachvollzogen werden kann. Als Ergebnis werden Interaktionsmöglichkeiten für Eingaben, Auswahlen und Ausgaben und die Darstellung von Informationen vor, während und nach der Interaktion beschrieben. Über Anforderungen hinweg werden Konflikte oder Widersprüche ermittelt. Abschließend wird überprüft und bewertet, inwieweit Anforderungen mithilfe von Benutzungsschnittstellen erfüllt werden können.

Da in dieser Prozessstufe viele Detailinformationen anfallen, ist eine systematische Analyse und Dokumentation von Ergebnissen wichtig. An der Spezifizierung von Nutzungsanforderungen sind auch die Benutzerinnen und Benutzer beteiligt.

#### 11.5 Gestaltungslösungen zur Umsetzung von Nutzungsanforderungen entwickeln

Ziel der Gestaltungslösungen ist es, konkrete Benutzungsschnittstellen der Software zu entwickeln, die bei der Bearbeitung von Aufgaben im Nutzungskontext unterstützen. Die Benutzungsschnittstelle soll die Beschäftigten nicht von der Aufgabenbearbeitung ablenken. Folgende Aktivitäten tragen zur Entwicklung von Gestaltungslösungen bei:

- Prozessmodell für vollständige Tätigkeiten zusammenstellen
- Interaktion entwickeln und modellieren
- Benutzungsschnittstelle entwerfen und gestalten

Aus den bereits durchgeführten Aufgabenanalysen (vgl. Anhang 5) werden vollständige Tätigkeiten (Aufgaben) mit Prozessen der Planung, Vorbereitung, Umsetzung und Kontrolle anhand des Prozessmodells zusammengestellt. Erfordernisse und Anforderungen wurden bereits ermittelt. Werden Nutzungsanforderungen als Abfolge von Schritten aufgelöst, dann sollte die Interaktion der Benutzerin oder des Benutzers mit der zukünftigen Benutzungsschnittstelle eine Abarbeitung einer Aufgabe ermöglichen. Diese Interaktion mit der Software ergibt auch, welche Informationen die Benutzerin oder der Benutzer und welche die Software wann, warum, wie und wofür benötigt, selbst verarbeiten soll oder ggf. weitergibt.

Als Ergebnis entstehen erste Prototypen für Benutzungsschnittstellen, die immer ausgereifere Nutzungsanforderungen abbilden können. Benutzungsschnittstellen bilden in frühen Stadien eher rudimentäre Funktionen ab. Sobald sich damit parallel und interaktiv Aufgabenbearbeitungen simulieren lassen, werden Nutzergruppen in eine formative Evaluation der Gestaltungslösungen einbezogen.

## 11.6 Gestaltungslösungen evaluieren

Die Nutzungsqualität der Software prüft und bewertet man für jeden spezifizierten Nutzungskontext. Dies kann gravierende Mängel bei der Unterstützung der Aufgabenbearbeitung vermeiden. Wenn sich einzelne Merkmale des Nutzungskontexts ändern, können sich unterschiedliche Ergebnisse der Prüfung und Bewertung ergeben. Das kann z. B. dann auftreten, wenn eine Software dieselbe Arbeitsaufgabe an einem Desktop-PC und an einem Notebook unterstützen soll. Das Vorgehen ist ähnlich einer Risiko- oder Gefährdungsbeurteilung, die bei verändertem Nutzungszweck oder veränderten Rahmenbedingungen ebenso wieder anzupassen ist.

Aus Prototypen für Benutzungsschnittstellen stellt man Prototypen der Software zusammen. Prototypen sind bereits in sehr frühen Entwicklungsphasen hilfreich. Je nach Entwicklungsreife und Einsatzzweck können Prototypen der Visualisierung von Schnittstellen, der Funktionalität der Software oder der Interaktion mit Nutzergruppen dienen (z. B. Skizzen auf Papier, programmierte Prototypen, Funktionstests, Use-Szenarien). Sobald mit Prototypen Aufgabenbearbeitungen interaktiv simuliert werden können, werden Nutzergruppen in formative Evaluationen der Gestaltungslösungen einbezogen. Diese Evaluationen sollen aktuelle Gestaltungslösungen verbessern. Sie richten sich an den Aufgaben aus, die mithilfe der Software unterstützt werden sollen.

Formative Evaluationen ab einem frühen Stadium der Entwicklung

- erlauben einen laufenden Abgleich mit Nutzungsanforderungen,
- tragen zur Gebrauchstauglichkeit der Software im Nutzungskontext bei,
- reduzieren den Aufwand für Schulungsmaßnahmen,
- vermeiden aufwendige Korrekturen in späten Entwicklungsphasen,
- tragen zur Kostenreduktion über den Einsatzzyklus der Software bei,
- ermöglichen es, die Nutzungsqualität der Software zu optimieren.

In jeder Entwicklungsphase prüft und bewertet man zunächst mithilfe von Funktionstests, GUI-Tests und Inspektionen, inwieweit die Schnittstellen Anforderungen abdecken können (vgl. Tabelle. 3). Dabei sind die spezifizierten Nutzungsanforderungen für jeden Nutzungskontext und Anforderungen an eine ergonomische Gestaltung aus Normen oder anderen Gestaltungsrichtlinien zu berücksichtigen.

Fachabteilungen, Benutzerinnen und Benutzer in den Evaluationsprozess einzubinden, ist bei grundsätzlichen Entscheidungen notwendig (z. B. formal korrekte Umsetzung von Aufgabenprozessen, Informationsstruktur zur Visualisierung). Erforderlich ist die Einbindung auch, wenn Benutzungsschnittstellen bereits verbesserte Gestaltungslösungen beinhalten und die Schnittstellen mit ersten Funktionen Interaktionen zur Aufgabenunterstützung erlauben.

Es wird zwar empfohlen, Evaluationen einzelner Prototypen mithilfe von Usability-Methoden auf fünf Benutzerinnen und Benutzer zu begrenzen, da jeder weitere immer weniger neue Mängel aufdecken wird. Allerdings sinkt die Aussagekraft und Validität einer Evaluation zur Softwarequalität mit sinkender oder geringer Anzahl von Nutzerinnen und Nutzern. Eine Generalisierung für ähnliche Situationen ist dann nicht möglich.

Formative Evaluationen können die Gebrauchstauglichkeit von Gestaltungslösungen verbessern, aber nicht fehlerfrei stellen. Es kann keine fehlerfreie Software geben, da die Abwesenheit von Fehlern durch die Prüfungen mithilfe der Methoden bzw. Tests nicht nachgewiesen werden kann. Lediglich das Vorhandensein von Fehlern kann man nachweisen. Die Qualität der Software kann sich durch folgende Regeln verbessern:

- Veränderte oder verbesserte Gestaltungslösungen immer wieder erneut überprüfen und bewerten.
- Das Prüfen und Bewerten von Gebrauchstauglichkeit von Software auch im fortgeschrittenen Entwicklungsstadium nicht nur auf ausgewählte Use-Szenarien beschränken, sondern alle Nutzungskontexte abdecken.
- Auch mit Mängeln nach der Fertigstellung von Software und während der Nutzungsphase der Software rechnen und Nachbesserungen umsetzen.
- Bei Entwicklung von Software auf einen Prozess der ergonomischen Analyse, Bewertung und Gestaltung setzen und alle Nutzungskontexte berücksichtigen.

**Tabelle 3** Formative Evaluationen mit Evaluationsziel, -gegenstand, -beteiligten und -adressaten

Teststufe (Was?)	Gegen was ?	Fokus auf ... /Schicht	Beispiel	Wer
Funktionstest	DV-Konzept	Datenhaltung	Wird der Antrag mit der korrekten Prämie korrekt gespeichert?	Entwicklung
GUI-Test	Oberflächenkonzept	Präsentationsebene/GUI	Funktioniert die Auswahlliste?	Qualitätssicherung
Fachtest	Fachkonzept	Funktionen	Wird die Prämie korrekt berechnet?	Fachabteilung
Usability Test	Benutzer-Objekt-Modell	Use-Szenarien	Kann man erwartungskonform arbeiten?	Benutzerin und Benutzer

### 11.7 Methoden zur Überprüfung und Bewertung von Gestaltungslösungen

Software wird über die verschiedenen Phasen des Einsatzzyklus häufiger überprüft und bewertet, d. h. ein Ist-Zustand wird immer mit einem Soll-Zustand verglichen. Soll-Ist-Unterschiede sind Softwaremängel oder -fehler. Sie zeigen sich dadurch, dass mit einer aktuellen Softwarelösung die allgemeinen und spezifizierten Anforderungen für die Nutzung (noch) nicht erreicht wurden und die Aufgabenbearbeitung (noch) nicht angemessen unterstützt wird. (Teil-)Aufgaben oder Anforderungen, für die noch keine (vollständige) Gestaltungslösung entwickelt wurde, bleiben zunächst unberücksichtigt.

Soll-Ist-Unterschiede sollten ebenso wie ihre Beseitigung im fortgeschrittenen Entwicklungsstadium dokumentiert werden. Anhand eines Prüfkonzepts sollte dokumentiert werden, was zum aktuellen Entwicklungsstand der Software erreicht ist und welche Ergebnisse noch nicht erzielt werden können. Jeder Stand sollte immer in mehreren W-Fragen konkretisiert werden:

- Was (aktueller Zustand eines Prüfgegenstandes) soll?
- Gegen was (Anforderungen und daraus abgeleitete Prüfkriterien)?
- Mit welchem Kriterium?
- Wann (Zeitpunkt im Entwicklungsprozess oder in der Phase des Einsatzzyklus)?
- Von wem (Entwicklerinnen und Entwickler, Einkäuferinnen und Einkäufer, Benutzerinnen und Benutzer)?
- Wie (Funktionstest, Benutzungstest)?
- Womit (Prüfablauf und -methoden) geprüft und bewertet werden?

Als Verfahren zur Überprüfung und Bewertung kann man grundsätzlich Dokumenten- und Produktanalyse, Beobachtung sowie schriftliche und mündliche Befragung nutzen. Einige hilfreiche Methoden

sind im Anhang 4 dargestellt. Es gibt nicht die richtige, immer passende und gültige Methode oder Kombination von Methoden. Die Auswahl ist abhängig von der Problem- und Zielstellung im Entwicklungsprozess zu treffen. Die Wahl sollte begründet und die Begründung dokumentiert werden. So reicht die Wahl irgendeines Verfahrens zur formativen Evaluation von Gestaltungslösungen nicht aus, um eine hohe Nutzungsqualität von Software sicherzustellen. Ein Verfahren kann immer nur als ein Baustein in einem Prozess der Prüfung und Bewertung eingesetzt werden.

Allen Verfahren gemeinsam ist, dass mit ihrer Hilfe einige, aber nie alle Mängel der ergonomischen Gestaltung der Software aufgedeckt werden können. Wie man aufgedeckte Mängel beheben kann, geht aus den Verfahrensanleitungen meist nicht hervor. Die Verfahren eignen sich zur formativen Evaluation (Nachweis von Mängeln), nicht aber zur summativen Evaluation (Nachweis von Mängelfreiheit). Bei einem **Funktionstest** wird eine Testperson gebeten, eine Funktion anzuwenden. Er wird während des Ausprobierens beobachtet und man prüft, ob ein vorab definierter Effekt erzielt wird. Da eine Funktion in ihren Möglichkeiten begrenzt ist, wird es wenige alternative Wege des Ausprobierens geben. Da ein einmaliger Funktionstest auch zufällig effektiv sein kann, sollte die Zuverlässigkeit eines Testergebnisses in mehrfachen Tests durch unterschiedliche Testpersonen ermittelt werden. Eine getestete Funktionalität reicht für Aussagen zur Nutzungsqualität noch nicht aus.

Bei einem **Benutzungstest** wird eine Testperson angeleitet, im Nutzungskontext Aufgaben mithilfe eines Prototyps der Software zu bearbeiten. Dabei kann die Testperson während des Bearbeitens beobachtet werden, oder ihre Arbeitsschritte und Interaktionen mit der Software zur Bearbeitung der Aufgabe und die Teilergebnisse der Aufgabe werden

dokumentiert. Auch kann die Testperson eine freie Bewertung der Software bzw. nach vorgegebenen Merkmalen abgeben. So wird geprüft und bewertet, ob ein vorab definierter Effekt erzielt und unerwünschte Nebeneffekte vermieden werden können. Die Nutzung der Software ist bei einem Benutzungstest durch alternative Wege z. B. der Reihenfolge des Bearbeitens vielfältiger. Zufällige Ergebnisse können nur durch mehrfach wiederholte Tests in zuverlässigere Ergebnisse überführt werden, wenn auch unterschiedliche Bearbeitungswege und unterschiedliche Testpersonen einbezogen sind. Die Unsicherheit über ein Testergebnis nimmt mit steigendem Entwicklungsgrad der Software und mit umfangreicher Berücksichtigung von Nutzungskontexten zu.

Das Prüfen und Bewerten von Software-Prototypen ist ein wesentlicher Schritt in Richtung Gebrauchstauglichkeit und Nutzungsqualität. Mängel oder Fehler der Software bei der Unterstützung der Aufgabenbearbeitung müssen identifiziert, Ursachen aufgedeckt und alternative Lösungsvarianten entwickelt werden. Alternative Lösungsvarianten müssen in neue Prototypen umgesetzt werden, die wiederum mit Funktions- und Benutzungstests auf Mängel zu prüfen und zu bewerten sind.

Während der Entwicklung von Prototypen bis hin zur finalen Version der Software sollen immer mehr Nutzungskontexte abgedeckt werden. Daher beziehen sich auch formative Evaluationen auf einen wachsenden Anwendungsbereich. Formative Evaluationen sollten solange durchgeführt werden, bis eine Software frei von gravierenden Mängeln der Gestaltung ist.

### 11.8 Prüfen und Bewerten von Software zu bestimmten Anlässen

Die Bezeichnung von Software (z. B. Office-Paket zur Unterstützung von Büroaufgaben) deutet häufig bereits Einsatzzwecke an. Daran lässt sich allerdings nicht die Qualität der ergonomischen Gestaltung der Software ablesen. Eine pauschale Bewertung von Nutzungsqualität der Software unabhängig vom spezifizierten Nutzungskontext ist nicht möglich. Eine Bürosoftware kann nicht allgemein für die Nutzung im Büro geprüft und bewertet werden. Das kann nur für vorab spezifizierte Nutzungskontexte gelingen.

Ablauf und Inhalt des Prüfens und Bewertens von Software unterscheiden sich je nach Anlass. Daraus können jeweils besondere Anforderungen entstehen:

- Lebenszyklus berücksichtigen: Bewertung des aktuellen Zustands, Beschaffung, Erweiterung, Neuentwicklung.
  - Von einer alten auf eine neue Version (z. B. Update, Upgrade) umstellen.
  - Identifikation des Ist-Zustands und Abgleich mit einem Soll-Zustand bei Erwägungen zur Modifizierung vorhandener Software oder anstehendem Neukauf.
  - Eine Veränderung des Aufgabengebiets zieht Anpassung des Arbeitsmittels Software für neue oder geänderte Aufgaben nach sich.
  - Prüfen und Bewerten von Standardsoftware auf Eignung für das geplante Einsatzgebiet oder zur Identifikation von möglichem Änderungs- oder Anpassungsbedarf.
  - Individualisierung von Standardsoftware.
  - Prüfen und Bewerten von Prototypen während des Entwicklungsprozesses einer Software (Wer muss was wann wie mit welchem Ziel tun?).
  - Rollout (Einführung von Software im Betrieb bzw. am Arbeitsplatz: Wer muss was wann wie mit welchem Ziel tun?).
- Ermittlung und Umgang mit aufgedeckten Nutzungsmängeln während des Rollouts.
  - Neue Erkenntnisse aus der Gefährdungsbeurteilung.
  - Beschwerden über die Software während der Nutzungsphase.
  - Veränderungs-/ Aktualisierungsmanagement.
  - Umgang mit Nutzungswünschen bzw. Veränderungen.
  - Dokumentation der Nutzungsqualität von Software für spezifizierte Nutzungskontexte.
  - Softwarenutzung auf unterschiedlichen Endgeräten (Desktop-PC bis Smartphone).

# 12 Softwarekauf und Pflichtenheft

Für die Anschaffung von Software benötigt man Beurteilungskriterien und Hilfen. In diesem Kapitel wird der Beschaffungsprozess einer ergonomisch gestalteten Software mit deren wichtigsten Beurteilungskriterien vorgestellt.

## 12.1 Softwarekauf

Für die Anschaffung von Software benötigen Unternehmerinnen bzw. Unternehmer, Führungskräfte, Systemadministratorinnen bzw. Systemadministratoren und Einkäuferinnen bzw. Einkäufer Beurteilungskriterien und Hilfen. Natürlich ist bei der Beschaffung neuer Software auch die Wirtschaftlichkeit zu berücksichtigen. Neben den direkten Anschaffungs- bzw. Entwicklungskosten sind aber auch die Nutzungskosten und Folgekosten, die erst nach dem Kauf oder der Entwicklung einer Software auflaufen, einzubeziehen. Diese Folgekosten sind bei ergonomisch gestalteter Software deutlich geringer, auch wenn diese in der Anschaffung etwas teurer sein kann. Folgekosten können sich durch Systemadministration, technischem Support, Training der Beschäftigten, Systemabstürze, lange Systemantwortzeiten, unnötigen Aufwand bei der Dateneingabe, notwendige Fehlerkorrekturen und programmbedingte Fehlbedienungen ergeben.

Eine hohe Gebrauchstauglichkeit der Software ermöglicht eine gesundheitsgerechte Aufgabenbewältigung, regt einen kontinuierlichen Verbesserungsprozess an und fördert die Motivation der Beschäftigten, mit der Software Arbeitsaufgaben zu bewältigen. Sie beeinflusst zudem maßgeblich die Qualität der Arbeit, der Produkte und der Leistungen.

### 12.1.1 Beteiligte beim Softwarekauf

Wer neue Software anschaffen oder vorhandene weiterentwickeln will, macht diese Aufgabe oft zur Angelegenheit von Systemadministratorinnen bzw. Systemadministratoren oder von Software-Expertinnen bzw. -Experten im Unternehmen. Die Kenntnis dieser Fachleute ist beim



Abb. 61 Beteiligte beim Kauf und der Weiterentwicklung von Software

Kauf neuer Software unersetzlich. Mit der Aufgabe, die Qualität der Arbeitsanforderungen und -aufgaben einzuschätzen und zu beschreiben, sind Systemadministratorinnen oder -administratoren allein überfordert. Die Führungskräfte, die für die Qualität der Produkte und Leistungen verantwortlich sind und auch die Perspektiven und zukünftigen Planungen im Kopf haben, sind beim Kauf oder der Weiterentwicklung einer Software frühzeitig zu beteiligen. Unersetzlich bei der Entscheidung sind die Benutzerinnen und Benutzer mit ihren Kenntnissen der zu bewältigenden Aufgaben. Sie wissen am besten, wie die Aufgaben bearbeitet werden sollten, und kennen die alltäglichen Probleme und Anforderungen. Außerdem sind sie es, die mit der Software zukünftig arbeiten müssen. Mitglieder des Betriebsrats/Personalrats sind entsprechend ihrer Mitbestimmungsrechte zu beteiligen (siehe auch Anhang 2 „Vereinbarung zum

Umgang mit Software“). Auch die Kenntnisse der Fachkräfte für Arbeitssicherheit und der Betriebsärztinnen und Betriebsärzte mit ihren Kompetenzen zur Softwareergonomie sowie zur Belastung und Beanspruchung durch die Arbeiten mit der Software müssen berücksichtigt werden. Schließlich empfiehlt es sich, eine für das Budget verantwortliche Person aus den Bereichen Beschaffung und Einkauf von Beginn an am Entscheidungsprozess zu beteiligen. Je mehr diese Person die konkreten Probleme und Anforderungen kennt, umso eher werden bei ihren Entscheidungen auch die qualitativen Aspekte sowie die Aspekte Belastung und Beanspruchung als Grundlage für produktives und gesundes Arbeiten eine Rolle spielen.

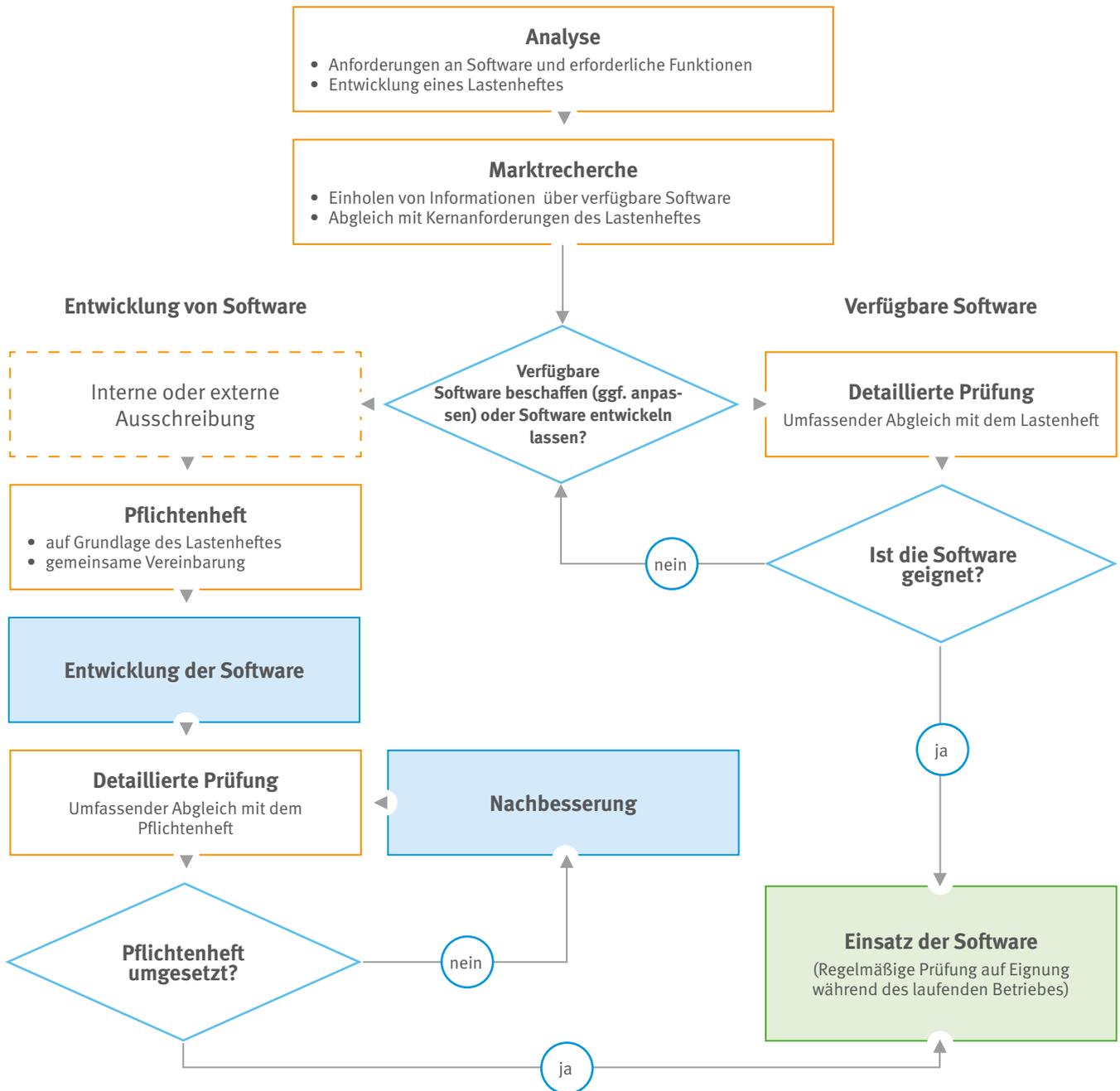


Abb. 62 Systematisches Vorgehen bei der Anschaffung von Software

### 12.2 Vorgehen beim Softwarekauf

Wird neue Software benötigt, empfiehlt sich ein systematisches Vorgehen mit einer gezielten Analyse einschließlich einer Prüfung der ergonomischen Gestaltung der anzuschaffenden Software. Die Ergebnisse werden in einem Lastenheft dokumentiert. Erst nach einer gewissenhaften Prüfung und Bewertung, gestützt auf nachvollziehbare Prüfkriterien, kann man eine richtige Kaufentscheidung treffen.

Eine sorgfältige Prüfung und Bewertung von Software vor dem Kauf gibt auch Aufschluss über die zu erwartenden Nutzungskosten. Diese sind niedriger, wenn die Software ergonomisch gestaltet und somit an die Fähigkeiten der Benutzerinnen und Benutzer und die zu bearbeitenden Aufgaben angepasst ist.

Folgende Grundsystematik ist bei der Anschaffung neuer Software zu empfehlen:

- **Analyse** – Welche Anforderungen und Funktionen soll die neue Software erfüllen?

- **Marktrecherche** – Welche fertige Software gibt es, die den Anforderungen gerecht wird?
- **Entscheidung** – Gibt es eine fertige Software, die die Kernanforderungen aus dem Lastenheft hinreichend erfüllt oder muss eine individuelle Software entwickelt werden?

Falls es eine fertige Software gibt:

- **Detaillierte Prüfung** – Ist die Software geeignet und gebrauchstauglich für die zu bearbeitenden Aufgaben (Durchführung eines umfassenden Praxistests)?

- **Entscheidung** – Soll die Software gekauft werden oder ist doch keine geeignete fertige Software auf dem Markt und es muss eine individuelle Software entwickelt werden? (Eventuell sind auch Teile für eine fertige Software neu zu entwickeln.)

Falls eine Software programmiert werden muss:

- **Interne oder externe Ausschreibung** – sofern eine Ausschreibung erforderlich ist, sollte das Lastenheft die Grundlage bilden
- **Pflichtenheft erstellen** – Welche Aufgaben und Funktionen soll die Software detailliert erfüllen?
- **Detaillierte Prüfung** – Ist die Software geeignet und gebrauchstauglich für die zu bearbeitenden Arbeitsaufgaben (Durchführung eines umfassenden Praxistests)?

### 12.2.1 Analyse – Welche Anforderungen soll die neue Software erfüllen und über welche Funktionen muss sie verfügen?

Zunächst wird analysiert, welche Aufgaben mit der neuen Software durchgeführt werden sollen. Das konkrete Ziel dieser Analyse ist die Ermittlung aller Anforderungen, denen die neue Software entsprechen soll, sowie aller Softwarefunktionen, die dafür notwendig sind. Die Beschreibung der Aufgaben führt zu Kriterien, mit denen die Software geprüft und bewertet werden kann. Das Ergebnis der Analyse wird in einem Lastenheft niedergelegt, auf dessen Basis die detaillierte Prüfung der in Frage kommenden Software vorgenommen werden kann.

Die Beantwortung der nachfolgend aufgeführten Fragen zur Analyse der Aufgaben sowie der Anforderungen und Funktionen der Software unterstützt eine systematische Ermittlung der einzelnen Aufgaben und die Bewertung der Anforderungen an die benötigte Software.

- Welche Kernaufgaben müssen mit der Software bearbeitet werden?

- Welche weiteren Aufgaben sollen mit der Software bearbeitet werden?
- Aus welchen Teilarbeitsschritten setzen sich die Arbeitsaufgaben zusammen?
- Was soll die Software innerhalb der einzelnen Teilaufgaben leisten?
- Welche grundlegenden Funktionen muss die Software erfüllen? (z. B.: Drucken, Kopieren, Bearbeiten, Blättern, Auflisten, Einstellungen)
- Welche speziellen Funktionen in einzelnen Bereichen soll die Software erfüllen? (z. B.: Einlesen von Daten, Rückmeldungen, Statistik)
- Wie soll die Software für die Arbeitsaufgabe gestaltet sein? (z. B.: Nutzerführung, Oberfläche, Anpassung an bestehende Programme, Orientierungssysteme, Farben)
- Welche Bedienanforderungen soll die Software erfüllen? (z. B.: Dialoggestaltung, Menüstruktur, Benutzungsschnittstellen, Barrierefreiheit)
- Mit welchen Programmen muss die Software kompatibel sein? (z. B.: Betriebssystem, andere Anwendungsprogramme, Internet, Intranet, Extranet)
- Wie soll die Software Fehler behandeln? (z. B.: Sicherheitsabfragen, automatische Fehlerkorrekturen, Fehlerbehandlung durch die Benutzerin oder den Benutzer)

Alle Beteiligten sollten die o. g. Fragen in einer Besprechung gemeinsam beantworten. Zur Vorbereitung empfiehlt es sich, die Fragen vorab von allen Beteiligten einzeln beantworten zu lassen.

### 12.2.2 Marktrecherche – Wo finden wir die geeignete Software

Nach der Analyse erfolgt eine Recherche auf der Grundlage des erstellten Lastenheftes, ob eine geeignete Software erworben werden kann. Dazu wird ermittelt, welche Software-Produkte verfügbar sind. Diese vergleicht man hinsichtlich der Anwendungsbereiche mit den benötigten Anforderungen und Funktionen.

Prinzipiell sollte man die Software auswählen, die möglichst viele Anforderungen erfüllt und alle benötigten Funktionen aufweist. Gibt es keine geeignete Einzel-Software, ist zu prüfen, ob die Anforderungen durch Nutzung mehrerer Programme zu erfüllen sind.

Gibt es eine Software, die den gestellten Anforderungen entspricht, sollte diese in jedem Fall gründlich geprüft und bewertet werden, bevor sie gekauft wird. Die dafür benötigten Testversionen können in der Regel im Internet heruntergeladen oder vom Handel zur Verfügung gestellt werden.

Ergibt die Recherche, dass keine Software auf dem Markt vorhanden ist, die allen Anforderungen genügt, ist die Entwicklung einer individuellen Software anzustreben.

### 12.2.3 Detaillierte Prüfung – Nach welchen Kriterien können wir Software prüfen?

Um die Eignung der Software zu prüfen, sollte man in zwei Schritten vorgehen:

- Prüfung der Software im Hinblick auf die Erfüllung der Anforderungen (Vergleich mit den Antworten auf die Fragen aus Abschnitt 12.2.1 mit dem Lastenheft)
- Prüfung der Software auf Übereinstimmung mit den Qualitätskriterien der Gestaltung (Standards für die ergonomische Gestaltung von Software)

Die Qualitätskriterien der Softwareergonomie ergeben sich u. a. aus Teilen der DIN EN ISO 9241. Informationen hierzu liefern auch die Kapitel 7 „Rechtliche Aspekte“, 8 „Informationsdesign“ und 9 „Interaktionsdesign“.

Es ist nicht selbstverständlich, dass die Software auch tatsächlich die Qualitätskriterien der Softwareergonomie berücksichtigt. Daher muss mithilfe konkreter Aufgaben aus den einzelnen Arbeitsbereichen überprüft und bewertet werden, ob die Software für die auszuführenden Aufgaben ergonomisch gestaltet ist. So kann man wesentliche Gestaltungsmängel erkennen.

**12.3. Pflichtenheft**

Wer eine Software komplett neu erstellen oder eine vorhandene Software ändern lassen will, sollte auf der Grundlage eines Lastenheftes ein Pflichtenheft erstellen. In einem Lastenheft werden die Anforderungen systematisch und detailliert zusammengetragen, die von der zu

programmierenden Software in Bezug auf Güte und Funktion überprüfbar erfüllt werden sollen. Es empfiehlt sich, die Inhalte des Lastenheftes schrittweise zu verfeinern. So kann ein Lastenheft zur Markterkundung schrittweise bis hin zu einer Software-Spezifikation der Programmiererin oder des Programmierers verfeinert werden, woraus sich das Pflichten-

heft, unter Berücksichtigung der betrieblichen Zielsetzungen, entwickeln lässt.

Das Lastenheft ist Grundlage für die Entscheidung darüber, welches Angebot die Anforderungen am besten erfüllen wird. Auf der Basis des Lastenheftes wird mit dem Auftragnehmer das Pflichtenheft entwickelt und vereinbart. Das Pflichten-

**Tabelle 4** Prozessphasen und Aufgaben bei der Beschaffung von Individualsoftware

Prozessphase	Aufgaben	Informationen über
Situation analysieren	<ul style="list-style-type: none"> <li>• Projektteam bilden</li> <li>• Ist-Zustand erfassen</li> <li>• Probleme aufzeigen</li> <li>• Bedarfe ermitteln</li> </ul>	Arbeitsaufgaben, Zusammenhänge von Arbeitsaufgaben, vor- und nachgelagerte Arbeitsaufgaben, bisherige Probleme der Aufgabenbearbeitung, bisherige Lösungswege etc.
Lösung spezifizieren	<ul style="list-style-type: none"> <li>• Ziele formulieren</li> <li>• Anforderungen formulieren</li> <li>• Wünsche formulieren</li> <li>• (Analysehilfe verwenden)</li> </ul>	Funktionsteilung Mensch/Arbeitsmittel, Zuschneiden von Arbeitsaufgaben, Zusammenhänge von Arbeitsaufgaben, vor- und nachgelagerte Aufgaben, Flexibilität der Aufgabenbearbeitung, Qualifikationsanforderungen an Beschäftigte, Arbeitsergebnisse etc.
Ausschreibung vorbereiten	<ul style="list-style-type: none"> <li>• Lastenheft erstellen</li> </ul>	Bestandteile/Gliederung des Pflichtenheftes, Bedeutung von Anforderungen (Basis-, Detail-Anforderungen), Aufgaben als Anforderungen, ergonomische Gestaltung als Anforderungen, Bestimmung der Überprüfung und der Bewertungskriterien etc.
Angebote einholen	<ul style="list-style-type: none"> <li>• Mehrere mögliche Anbieterinnen und Anbieter zur Abgabe eines Angebotes auffordern (Grundlage: Lastenheft mit Anforderungskatalog)</li> </ul>	Anbieterinnen und Anbieter, Erfahrungen von bekannten Unternehmen, Kammern, Verbände, Berufsgenossenschaften, Referenzen etc.
Angebote prüfen und bewerten (Evaluation)	<ul style="list-style-type: none"> <li>• Angebote bewerten (Grundlage: Lastenheft mit Anforderungskatalog)</li> <li>• Angebote auswählen: Welches Angebot erfüllt alle Anforderungen am besten?</li> <li>• Vertragsverhandlungen Softwarespezifikationen abstimmen (Grundlage: Lastenheft mit Anforderungskatalog)</li> <li>• Anforderungen mit Programmierern/Programmierern konkretisieren (Auftragnehmerin/Auftragnehmer) und Pflichtenheft entwickeln</li> </ul>	Bedeutung der Bewertung, Pflichtenheft, Lastenheft, Anforderungskatalog etc.
Vertrag abschließen	<ul style="list-style-type: none"> <li>• Vertrag abschließen (Pflichtenheft mit Anforderungskatalog und Software-Spezifikation sind Bestandteil)</li> </ul>	
Projektumsetzung planen und realisieren	<ul style="list-style-type: none"> <li>• Einführung planen</li> <li>• Einführung realisieren</li> </ul>	Auswirkungen auf Tagesgeschäft, betroffene Arbeitsplätze, betroffene Beschäftigte etc.
Arbeitsmittel prüfen und bewerten	<ul style="list-style-type: none"> <li>• jede Realisierungsstufe bewerten</li> <li>• (Grundlage: Pflichtenheft mit Anforderungskatalog)</li> <li>• Eventuell vertragsgemäße Ausführung des Arbeitsmittels einfordern</li> </ul>	Bedeutung der Bewertung, Pflichtenheft, Lastenheft, Anforderungskatalog; Prüfprotokoll etc. Grad der Zielerreichung anhand des Pflichtenheftes
Arbeitsmittel bereitstellen und abnehmen	<ul style="list-style-type: none"> <li>• Arbeitsmittel bereitstellen, das die dokumentierten Anforderungen vollständig und wie vertraglich vereinbart erfüllt (Grundlage: Pflichtenheft mit Anforderungskatalog)</li> <li>• Arbeitsmittel abnehmen und (Rest-)Zahlung veranlassen</li> </ul>	Zuverlässigkeit und Zweckmäßigkeit des Arbeitsmittels, Pflichtenheft, Anforderungskatalog; Prüfprotokoll, Gewährleistung etc.

Tabelle 5 Gliederung eines Lastenheftes – Beispiel

Inhalt	Informationen über
Ausgangslage	bezogen auf das Unternehmen: Charakteristik des Unternehmens, Organisation, Struktur und Umfang des EDV-Einsatzes, Anstoß für die Beschaffung, Projektorganisation etc.
IST-Zustand	bezogen auf den Einsatzbereich des zukünftigen Arbeitsmittels: Aufbau der Organisation (Zuständigkeiten, Kompetenzen, Funktionen etc.), Arbeitsabläufe (Welche Aufgaben gehören zusammen oder folgen aufeinander?) und Arbeitsgegenstände (Arbeitsergebnisse), Aufteilung der Arbeitsaufgaben auf die Beschäftigten und auf die Arbeitsmittel, Qualifikation der Beschäftigten, Güte und Menge der Arbeitsmittel etc.
Ziele	Beschreibung der Ziele, die durch den Einsatz eines (geänderten/neuen) Arbeitsmittels erreicht werden sollen, jeweils bezogen auf das Beschaffungsprojekt insgesamt und bezogen auf die einzelnen Aufgaben, die im betroffenen Unternehmensbereich von den Beschäftigten bearbeitet werden; für Ziele können bereits Kriterien und Gewichtungen gesetzt werden (Muss-Kriterien, Kann-Kriterien) etc.
Anforderungen	Dieser Teil des Lastenheftes ist der wichtigste. Hier kann die Liste „Analyse der Aufgaben der Nutzerinnen und Nutzer sowie der Anforderungen und Funktionen der Software“ (Anhang 5) zur Hilfe genommen werden. Präzise Beschreibung der Anforderungen, die an das Arbeitsmittel gestellt werden, das heißt Aufgaben mit Aufgabenzielen und Aufgabenteilen, die durch das Arbeitsmittel unterstützt werden sollen sind notwendig. Die Anforderungen sind für beide oben genannten Zielebenen (Beschaffungsprojekt und Einzelaufgaben im Unternehmensbereich) ausführlich zu dokumentieren. Hinweise über zum Beispiel Aufgabeninhalt, benötigte Eingaben und Ausgaben, Datenbasis und Schnittstellen sollten zusätzlich gegeben werden. Bereits hier müssen allgemeine Basis- und Detail-Anforderungen an eine ergonomische Gestaltung der Software formuliert werden, sowie die Beschreibung der Anforderungen durch eine unter Umständen vorhandene Systemplattform etc.
Mengengerüst	Menge eingehender und weitergegebener Daten, Umfang des Datenbestandes, Menge anzuzeigender und einzugebender Daten und Abfragen etc. Hier kann die Liste „Analyse der Aufgaben der Nutzerinnen und Nutzer sowie der Anforderungen und Funktionen der Software“ zur Hilfe genommen werden.
Qualitäts- und Zielbestimmung	Festlegung von Qualitätsmerkmalen mit zugehörigen Qualitätsgraden des Arbeitsmittels. Die Qualitätsmerkmale müssen messbar sein und die Form ihrer Überprüfung muss festgelegt werden. Bei der Überprüfung der Software wird ermittelt, inwieweit die Anforderungen erfüllt wurden. Qualitätsmerkmale und Prüfkriterien (bisher dokumentierte Anforderungen) sind detailliert und vollständig anzugeben und es ist unter Umständen auch auf allgemeine Richtlinien (Standards, Werksnormen, insbesondere Prüfprotokolle) zu verweisen. Die Überprüfung bildet die Grundlage zur Entscheidung, ob sich mit der Software die formulierten Ziele erreichen lassen etc. Hier kann die Liste „Analyse der Aufgaben der Nutzerinnen und Nutzer sowie der Anforderungen und Funktionen der Software“ (siehe Anhang 5) zur Hilfe genommen werden.
Test-Aufgaben	Zusammenstellung benutzerbezogener Testfälle, die sich auf umfangreiche Arbeiten mit dem Arbeitsmittel am Arbeitsplatz im Unternehmen unter realen Arbeitsbedingungen mit realen Aufgaben beziehen etc.
Ergänzungen	Beschreibung von Ergänzungen oder speziellen Anforderungen und weiterer Leistungen (zum Beispiel Installation, Wartung, Garantie, bauliche und räumliche Voraussetzungen, Testdaten, Hilfspersonal, Normen, Lizenzen, Glossar, Dokumentationen zum Arbeitsmittel, Einführung und Schulung der Beschäftigten) Beschreibung von Anforderungen an anbietende Unternehmen/Entwicklerinnen bzw. Entwickler (zum Beispiel generelle Anbietermerkmale, Referenzen, Projektorganisation, Unterstützung vor, während, nach Projektrealisierung, technischer Support, Leistungsabsicherung, Gewährleistung etc.).
Angebots-Aufbau	Vorgaben zur Selbstdarstellung der Anbieterin bzw. des Anbieters, ihrer/seiner kurz gefassten Stellungnahme zum Pflichtenheft, der detaillierten Projektumsetzung (siehe vorherige Punkte), des Zeitplanes, der Kosten, des Supports, der Leistungsabsicherung etc.
Administratives	Vorgaben zur Vertraulichkeit, Informationen zur Bewertung des Angebotes und der Projektrealisierung, Rückfragen, Termine etc.
Anhang	Anlagen zum Lastenheft wie zum Beispiel Vorgaben zur Qualitätsbestimmung, Prüfung und Bewertung der Software etc.

heft hingegen ist ein wesentlicher Vertragsbestandteil bei der Beauftragung zur Erstellung der Software und dient bei der Abnahme der Software zur Bewertung, ob die gestellten Anforderungen tatsächlich erfüllt bzw. welche Nacharbeiten notwendig sind.

Die im Pflichtenheft dokumentierten Anforderungen an die Software bilden die gemeinsame Basis für Beschafferinnen/ Beschaffer und Programmiererinnen/ Programmierer während eines Projektes.

Die zentrale Bedeutung des Pflichtenheftes im Prozess der Beschaffung von Arbeitsmitteln wie Software verdeutlicht die schematische Darstellung „Prozessphasen und Aufgaben bei der Beschaffung von Individualsoftware“.

### 12.3.1 Struktur und Inhalte eines Pflichtenheftes

Der inhaltliche Aufbau eines Pflichtenheftes muss in seinen Bestandteilen den Zielen entsprechen, die mit dem Softwarekauf verfolgt werden. Da jeder Beschaffungsprozess andere Anforderungen an ein Arbeitsmittel stellt, gibt es nicht das allgemein gültige Standard-Pflichtenheft.

Die Softwareergonomie muss sich möglichst in allen Gliederungsabschnitten des Pflichtenheftes, wie auch bereits im Lastenheft, widerspiegeln. Eine softwareergonomische Gestaltung führt dazu, dass die Beschäftigten an ihrem Arbeitsplatz die anstehenden Arbeitsaufgaben effektiv, effizient und zufriedenstellend bearbeiten können. Daher ist im Pflichtenheft zu dokumentieren, dass diesem Thema ein hoher Stellenwert beigemessen wird. Ein Software-Anbieter sollte auf diese Anforderungen eingehen, um eine qualitativ hochwertige Arbeit abzuliefern.

Die Ziele der softwareergonomischen Gestaltung sind so zu konkretisieren, dass deren Anforderungen entsprechend der Qualitäts- und Zielbestimmung direkt überprüfbar sind.

Die Anforderungen können als Aussagesatz (zum Beispiel „Die Software muss Hilfsfunktionen zur Verfügung stellen.“) oder als Frage (zum Beispiel „Stellt die Software Hilfsfunktionen immer zur Verfügung?“) formuliert werden, wobei die betroffene Arbeitsaufgabe genannt sein sollte.

Die Basis-Anforderungen aus den Kriterien zur Prüfung und Bewertung von Softwareergonomie sind wichtige Mindestkriterien der software-ergonomischen Gestaltung. Eine Software sollte diese Basis-Anforderungen erfüllen, um wesentliche Aspekte der ergonomischen Gestaltung zu berücksichtigen.

### 12.3.2 Prüfung und Bewertung neu entwickelter Software

Das Pflichtenheft dient als Bewertungs- und Entscheidungsgrundlage zur Prüfung und Bewertung. Das Projektteam der Beschaffung und das der Programmierung sollten sehr früh und mehrmals wiederholt überprüfen, welche der im Pflichtenheft dokumentierten Anforderungen an die Software bereits erfüllt sind. Dabei sollten die im Folgenden beschriebenen „Tipps zur Durchführung von Prüfung und Bewertung von in Auftrag gegebener Software“ beachtet werden.

Die Abnahme der Software erfolgt erst dann, wenn Benutzerinnen und Benutzer und Projektteam die neue Software nach den Anforderungen des Pflichtenheftes bewertet haben. Das Ergebnis muss zeigen, dass alle dokumentierten Anforderungen an die Software erfüllt sind und die Software so eingesetzt werden kann, dass die anfangs formulierten Zielsetzungen effektiv und effizient erfüllt werden.

Grundsätzlich gilt: Eine systematische und sorgfältige Herangehensweise bei der Beschaffung von Software trägt zu guten Arbeitsbedingungen bei und vermeidet mittel- und langfristig überflüssige Kosten.

## i

## Tipps zur Prüfung und Bewertung von in Auftrag gegebener Software

- ! Bei jeder Prüfung muss jede einzelne (Prüf-)Aufgabe mit der Software von den zukünftigen Benutzerinnen und Benutzern so weit wie möglich ausgeführt werden. Die Prüfung darf nicht nach der ersten Abweichung von einer Muss-Anforderung abgebrochen werden, da das Ziel einer Prüfung immer eine möglichst umfassende Dokumentation aller aktueller Mängel einer Software-Version sein sollte.
- ! Bei der softwareergonomischen Prüfung sind unter anderem folgende Methoden üblich: lautes Denken der Benutzerinnen oder Benutzer, Videoaufzeichnung aus verschiedenen Blickwinkeln, Usability-Labs.
- ! Im Verlauf der Prüfung darf die Software nicht verändert werden. Erst nach dem Abschluss der vollständigen Prüfung darf auch „experimentiert“ werden (zum Beispiel Suche nach (Um-)Wegen, wie es vielleicht doch gehen könnte, Ausprobieren nach Veränderungen von Einstellungen etc.).
- ! Das Ergebnis einer Prüfung ist ein Prüfprotokoll, in dem die Prüferinnen und Prüfer und weitere anwesende Personen die Prüfbedingungen, das Prüfverfahren und die entdeckten Mängel (z. B. Bug-Liste) dokumentieren müssen. Dafür kann die folgende Checkliste verwendet werden.
- ! Software auch nach folgenden Aspekten prüfen und bewerten:

  - Alle Funktionen der Software mindestens einmal ausprobieren
  - Alle möglichen Ausgabeformen mindestens einmal erzeugen
  - Alle verschiedenen Eingabekombinationen nutzen
  - Alle Anweisungen mindestens einmal ausführen
  - Alle Prozeduren mindestens einmal aufrufen
  - Alle Grenzfälle (zum Beispiel leere Verarbeitungsschritte) einmal erreichen
  - Alle Fehlermeldungen mindestens einmal erzeugen (Liste bei Auftragnehmerinnen bzw. Auftragnehmern anfordern!)
  - Alle möglichen unerwarteten (Fehl-) Eingaben erzeugen (zum Beispiel „Sich-auf-die-Tastatur-Setzen“)
  - Unerwartete Ereignisse erzeugen (zum Beispiel Druckerkabel herausziehen, Rechner ausschalten)
  - Leistungsgrenzen ausloten (zum Beispiel Antwortzeiten, verarbeitete Transaktionen pro Zeiteinheit)
  - Mit Maximal- und Minimalwerten bei spezifizierten Mengen arbeiten
  - Definierte Fehlersituationen provozieren
  - Installation der Software anhand von dokumentierten Instruktionen
  - Inbetriebnahme der Software anhand von dokumentierten Instruktionen
  - Dauer einer störungsfreien Laufzeit der Software testen
  - Neuinstallation, Neu-Inbetriebnahme der Software nach Unterbrechung

# 13 Software-Entwicklungsprozesse

Um größere Softwareprojekte erfolgreich umzusetzen, ist es unabdingbar, sich über das passende Softwareentwicklungsmodell (software engineering) klar zu werden und mit ihm die Nutzungsanforderungen umzusetzen. Nur so gibt es ein Ergebnis, das gut anwendbar und technisch robust ist. In diesem Kapitel werden drei verschiedene Software-Entwicklungsmodelle vorgestellt.

Softwareentwicklungsprozesse dienen zur Steuerung einer Softwareentwicklung von der Konzeption bis zum Einsatz im Echtbetrieb und beinhalten alle anfallenden Änderungen der Software. In der Regel sind sie durch spezielle Vorgehensmodelle (Prozessmodelle) gekennzeichnet und dienen dazu, die Softwareentwicklung übersichtlich und beherrschbar zu gestalten.

Es gibt eine große Vielfalt an unterschiedlichen Softwareentwicklungsprozessen, jedoch keinen universell gültigen. An dieser Stelle werden exemplarisch einige gängige Modelle nebeneinander gestellt:

## 13.1 Das Wasserfallmodell

Das Wasserfallmodell ist eines der ältesten Modelle der Softwareentwicklung.

Es unterteilt die Softwareentwicklung in möglichst klar zueinander abgegrenzte Phasen, die genau beschrieben und vollständig zu durchlaufen sind. Die Ergebnisse einer jeden Phase werden an ihrem Ende dokumentiert und verabschiedet und sind bindend für die nächste Phase. Sie werden also wie bei einem Wasserfall an die nächste Stufe weiter gegeben.

Das Wasserfallmodell kann folglich als „dokumentgetrieben“ charakterisiert werden.

Das Wasserfallmodell ist sehr effizient für Softwareprojekte, bei denen die Anforderungen, Leistungen und Abläufe im Vorfeld bereits präzise beschrieben werden können und diese sich während des Entwicklungsprozesses nicht oder nur wenig ändern.

Gleichzeitig zeigt sich hier auch die Schwäche des Modells. Durch die starren, aufeinander aufbauenden Phasen gestattet dieses Modell nur wenig Flexibilität. Flexibilität ist beispielsweise gefordert, wenn sich Anforderungen an die zu entwickelnde Software ändern oder man während der Entwicklung feststellt, dass in einer der frühen Phasen von falschen Annahmen ausgegangen wurde. Dies kann dazu führen, dass die fertig gestellte Software zwar den beschriebenen Anforderungen entspricht, nicht aber den

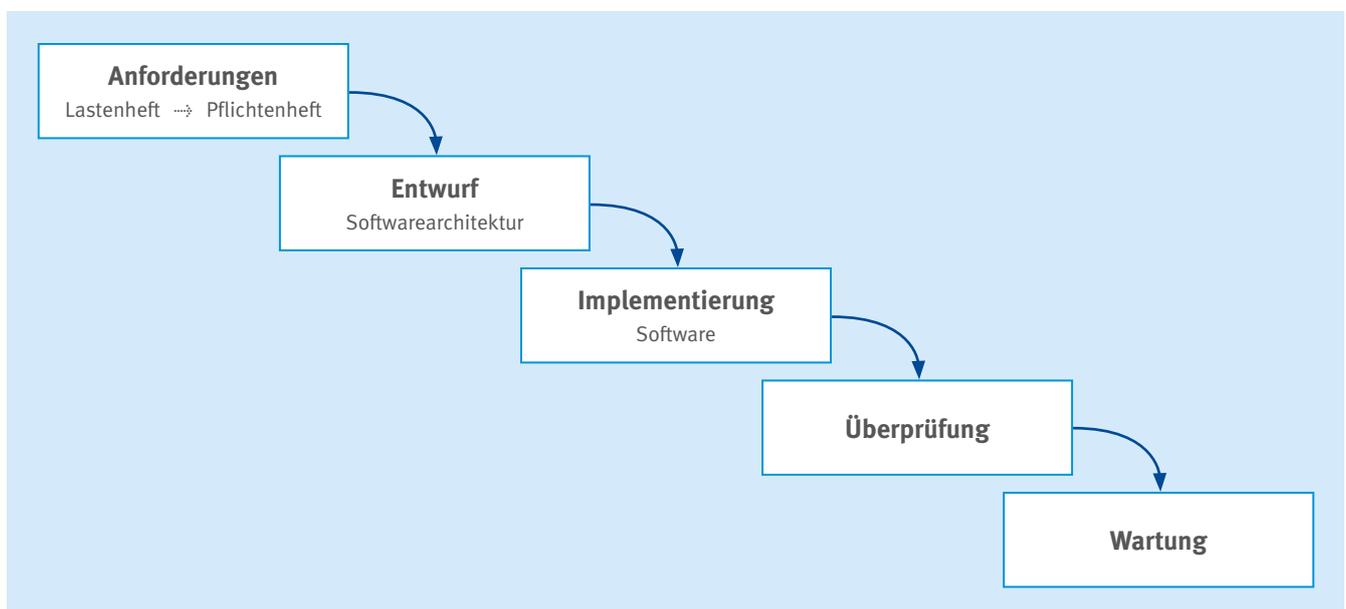


Abb. 63 Das Wasserfallmodell

aktuellen Erfordernissen. Auch aus diesem Grund wurden vom Wasserfallmodell viele weitere Prozessmodelle abgeleitet, z. B. das **Spiralmodell**, das den starren Ablauf des Wasserfallmodells aufbricht, indem derselbe Arbeitsschritt mehrmals durchlaufen wird und dabei die Ergebnisse verfeinert oder verändert werden (Iteration).

**13.2 Extreme Programming (XP)**

Extreme Programming ist eine „agile“ Methode des Softwareentwicklungsprozesses. Diese begreift sich als Gegenentwurf zu traditionellen Softwareentwicklungsprozessen, welche Entwicklerinnen und Entwickler oft als zu schwerfällig und bürokratisch empfinden. Formellen Abstimmungen und intensiver Dokumentation wird folglich wenig Bedeutung beigemessen.

bekannt sind bzw. sich mit fortschreitender Entwicklungszeit ändern oder ganz entfallen können. Anders als bei den traditionellen Methoden, bei denen das fertige Produkt erst zum Ende des Entwicklungsprozesses zur Verfügung steht, versucht Extreme Programming möglichst schnell einen funktionsfähigen Prototypen zur Verfügung zu stellen und diesen dann sukzessive, gemeinsam mit dem Kunden um die gewünschten Funktionalitäten zu erweitern.

Extreme Programming geht davon aus, dass die Anforderungen an die zu erstellende Software zu Beginn nicht vollständig

Extreme Programming kann folglich als „modellgetrieben“ charakterisiert werden.

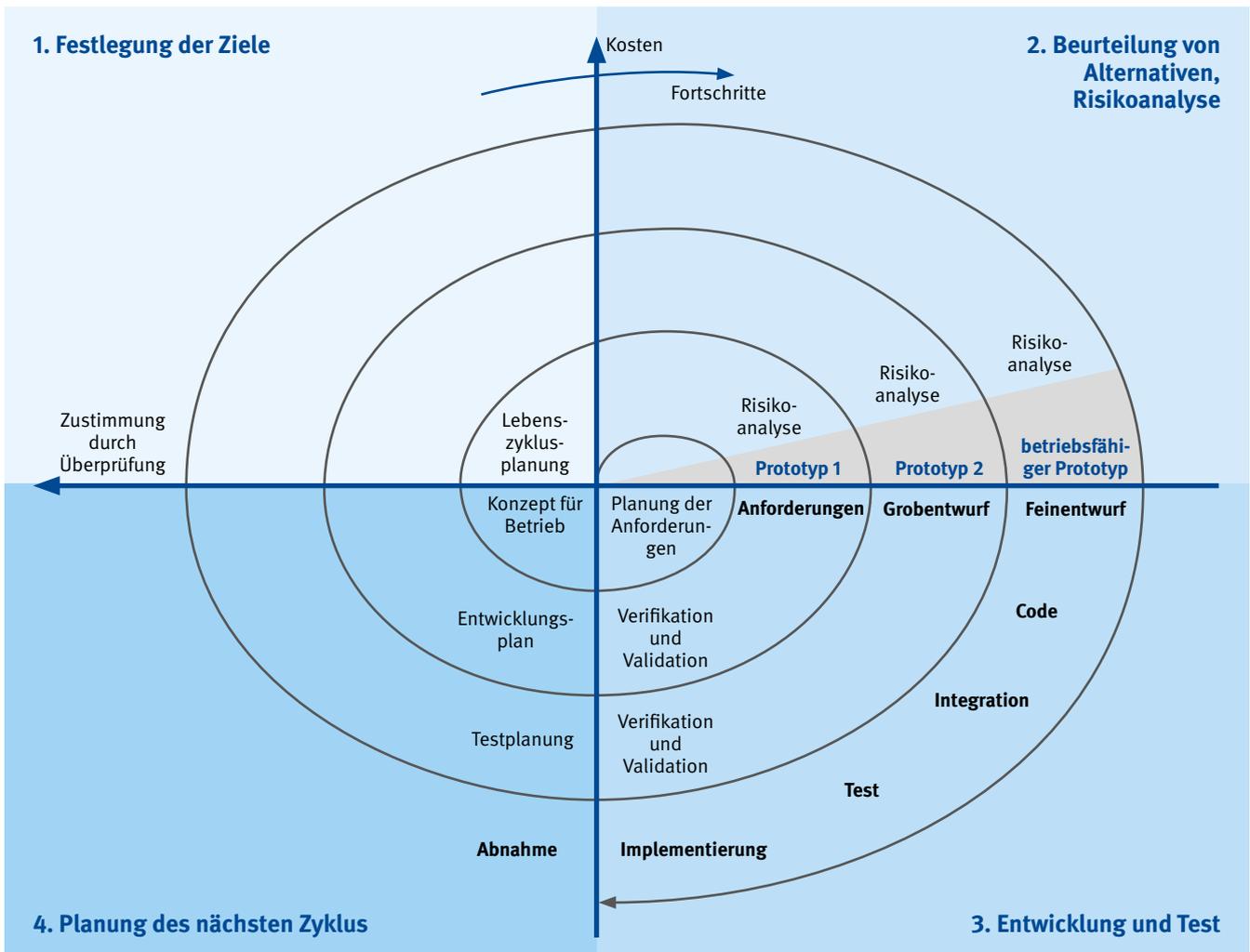


Abb. 64 Spiralmodell nach Boehm

Die ständige Rückkopplung der Kundin bzw. des Kunden während des Entwicklungsprozesses soll verhindern, dass ihr/ ihm nach einer langen Entwicklungszeit ein fertiges Produkt übergeben wird, das sie/er nicht in der gewünschten Weise nutzen kann.

So schlüssig diese Vorgehensweise theoretisch erscheint, so schwierig gestaltet sich häufig die praktische Umsetzung. Insbesondere die Kundin bzw. der Kunde muss für die Möglichkeit, die Software jederzeit im Entwicklungsprozess an ihre/ seine aktuellen Anforderungen anpassen zu können, Einschränkungen und Forderungen auf sich nehmen. So kann zu Beginn des Projektes kein genauer Leistungsumfang festgelegt werden und somit auch kein fester Preis oder Auslieferungstermin für das „fertige“ Produkt. Zudem muss die Kundin bzw. der Kunde den entwickelnden Personen zutrauen, in einem derart dynamischen Prozess nicht den roten Faden zu verlieren. Und letztlich muss sie/er selbst erhebliche, hochqualifizierte eigene Ressourcen in das Projekt einbringen.

### 13.3 Das V-Modell™ (Entwicklungsstandard)

Das V-Modell erweiterte anfangs lediglich die jeweiligen Entwurfsstufen des Wasserfallmodells um zeitlich nachfolgende Teststufen.

Mittlerweile unterstützt das V-Modell XT™ auch viele andere Ansätze (z. B. Agile Systementwicklung) und wird auch in der Privatwirtschaft eingesetzt.

Dem V-Modell XT™ liegt der Ansatz zugrunde, einerseits durch die ingenieurmäßige, stark strukturierte und dokumentierte Vorgehensweise über das Gesamtprojekt nicht den roten Faden zu verlieren. Es regelt dazu unter anderem die Kommunikation zwischen Auftraggeberin bzw. Auftraggeber und Auftragnehmerin bzw. Auftragnehmer, um typische Missverständnisse auszuschließen.

Andererseits bricht es die starren Strukturen der klassischen phasenorientierten Vorgehensmodelle auf, indem lediglich Aktivitäten und Ergebnisse definiert wer-

den und keine strikte zeitliche Abfolge. Insbesondere fehlen die typischen Abnahmen am Ende einer Phase.

Auf Basis der Erfahrungen aus der Anwendung im Projektalltag wurden auch iterative Elemente in das V-Modell™ aufgenommen.

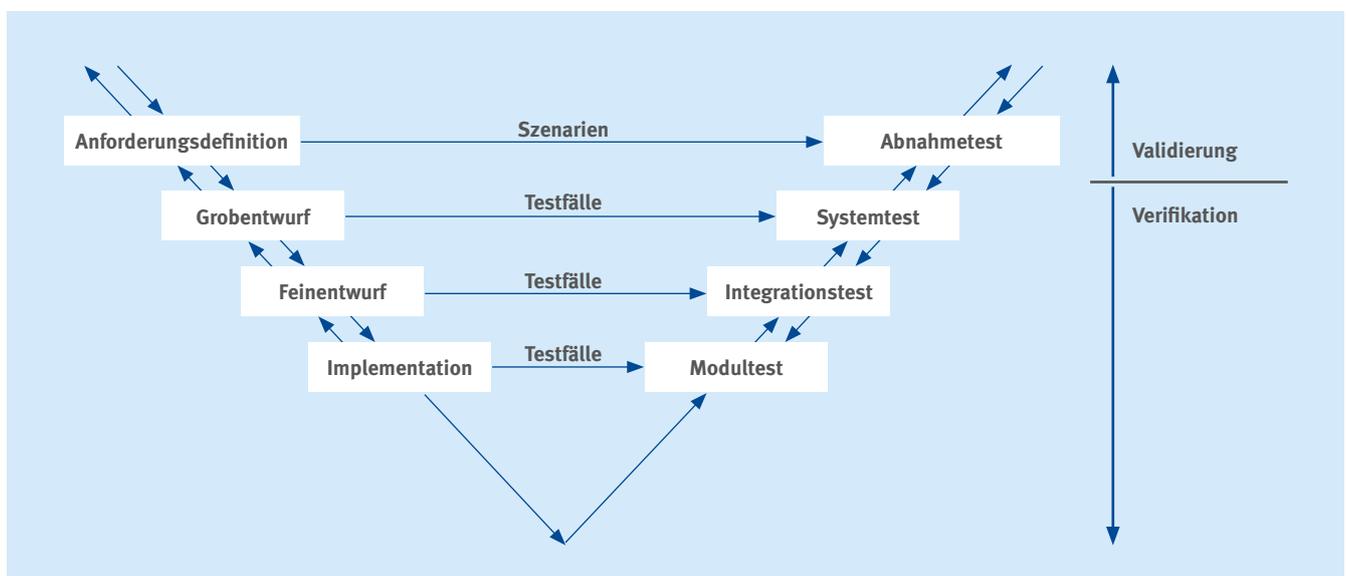


Abb. 65 Das V-Modell™

### 13.4 Das iterative Modell

Die DIN EN ISO 9241 Teil 210 beschreibt ein benutzerinnen-/benutzerorientiertes Vorgehen bei der Softwareentwicklung.

Im Unterschied zu den klassischen Softwareentwicklungsprozessen der vorhergehenden Abschnitte liegt der Fokus nicht auf der technischen Qualität der zu entwickelnden Software. DIN EN ISO 9241 Teil 210 ist ein iteratives Modell zur Entwicklung einer gebrauchstauglichen, menschenzentrierten Software (Usability Engineering, UE). Bei sogenannten iterativen Entwicklungsmodellen ist die Aufteilung des Projektes in Iterationen (dt. Wiederholungen) bzw. Kleinst-Projekte vorgesehen.

In der Praxis stehen die Modelle der Softwareentwicklung (SE) und des Usability Engineering (UE) nebeneinander, auch, was die handelnden Akteurinnen und Akteure betrifft. Spätestens seit dem großen Erfolg von Handheldprodukten setzt sich jedoch auch in Kreisen der

Entwicklerinnen und Entwickler die Erkenntnis durch, dass sich das rein technische System der Gebrauchstauglichkeit unterordnen muss, um Erfolg zu haben.

Am Anfang eines neuen Entwicklungsprozesses müssen sich Anwendende und Entwickelnde mit der Frage beschäftigen, wie sich die beiden Sichtweisen und Prozesse miteinander verbinden lassen. Ein pragmatischer Ansatz besteht darin, vor dem eigentlichen Entwicklungsprozess einen systematischen Prozess des Usability Engineering durchzuführen und so qualitativ hochwertige und präzise formulierte Nutzungsanforderungen zu entwickeln. Diese bilden die grundsätzlichen Leitplanken der Softwareentwicklung. Durch die weiteren Iterationen besteht zu jedem Zeitpunkt der Entwicklung die Möglichkeit zum Abgleich der Software mit den ursprünglichen Anforderungen und zu ggf. erforderlichen Nachbesserungen, die hier noch mit relativ vertretbarem Aufwand realisiert werden können.

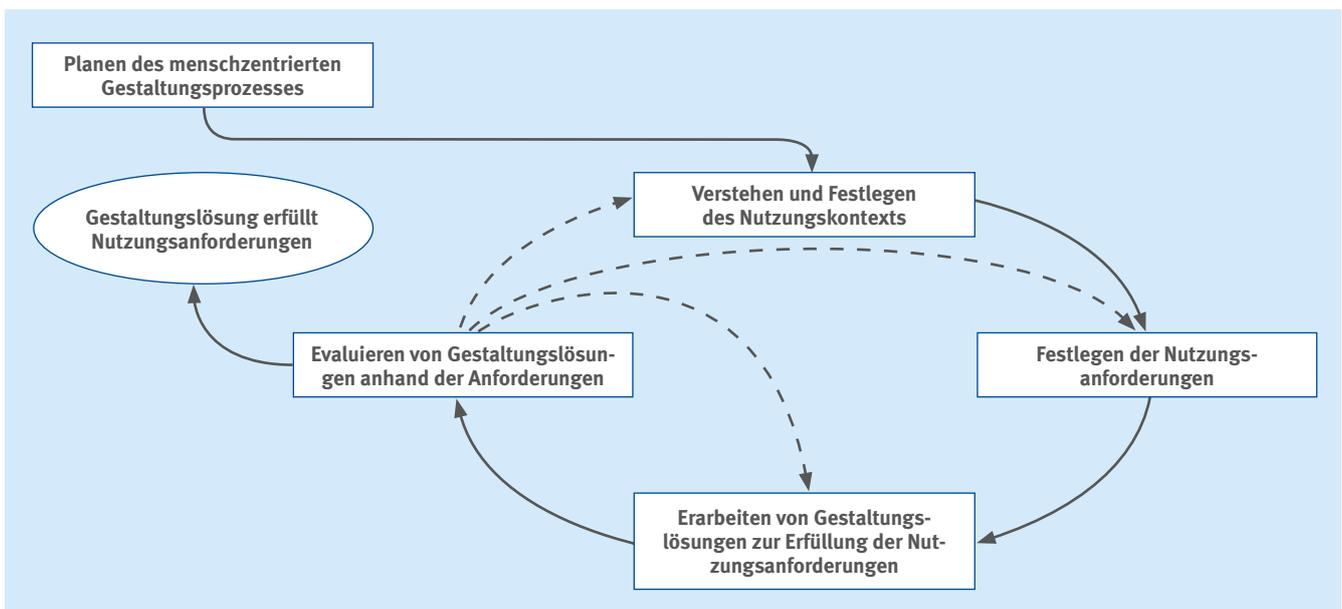


Abb. 66 Wechselseitige Abhängigkeit menschenzentrierter Gestaltungsaktivität (Quelle: DIN EN ISO 9241-210)

# Anhang 1

## Bildschirmparameter für LCD-Bildschirme

### Bildschirmparameter für LCD-Bildschirme

Format	Diagonale		Anzeigefläche mm	Physik. Auflösung Pixel	Pixelgröße mm
	Zoll	mm			
Normalformat	15	381	304 x 228	1024 x 768	0,294 x 0,294
4:3 bzw. 5:4	17	432	338 x 270	1280 x 1024	0,264 x 0,264
	19	483	376 x 301	1280 x 1024	0,294 x 0,294
	20	508	408 x 306	1600 x 1200	0,255 x 0,255
	21,3	541	432 x 324	1600 x 1200	0,270 x 0,270
	20	508	433 x 271	1680 x 1050	0,258 x 0,258
	Breitbildformat	21,1	541	454 x 284	1680 x 1050
16:10 bzw. 16:9	22	559	474 x 296	1680 x 1050	0,282 x 0,282
	23	584	509 x 286	1980 x 1080	0,265 x 0,265
	24	610	518 x 324	1920 x 1200	0,270 x 0,270
	26	660	551 x 344	1920 x 1200	0,287 x 0,287
	27	686	597 x 336	1920 x 1080	0,311 x 0,311
	27	686	582 x 364	1920 x 1200	0,303 x 0,303
	27	686	596 x 336	2560 x 1440	0,233 x 0,233
	27,5	699	593 x 371	1920 x 1200	0,309 x 0,309
	30	762	641 x 401	2560 x 1600	0,250 x 0,250

# Anhang 2

## Vereinbarung zum Umgang mit Software

### Vereinbarung zum Umgang mit Software

Diese Vereinbarung gilt für die Nutzungsqualität der Software in unserem Unternehmen. Sie beschreibt den Rahmen, der als Mindeststandard zur Gestaltung des Arbeitssystems bei Arbeiten mit Software am Bildschirm gilt. Die Nutzungsqualität der Software beschreibt die Gebrauchstauglichkeit der Software in einem hochwertigen Nutzungskontext (Effektivität, Effizienz, Zufriedenheit, Qualität der sozialen Beziehungen).

#### Ziel

Ziel der Vereinbarung ist es, über die Gestaltung des Arbeitssystems die Realisierung von möglichst hochwertigen Produkten und Leistungen mit Hilfe der Software zu erreichen sowie gleichzeitig die Sicherheit und Gesundheit der Beschäftigten zu schützen und zu fördern.

#### Beurteilung der Arbeitsbedingungen, Maßnahmen

Gemäß § 3 Bildschirmarbeitsverordnung wird die Gefährdungs- und Belastungsbeurteilung (entsprechend § 5 ArbSchG) durchgeführt, insbesondere unter Berücksichtigung der Arbeitsbedingungen, Anforderungen an das Sehvermögen und psychischer Belastung. Nach der Beurteilung der Arbeitsbedingungen wird festgelegt, welche Maßnahmen einzuleiten sind, einschließlich Verantwortlichkeiten, Terminierung und Kontrolle. Bei der Festlegung der Maßnahmen werden die Erfahrungen der Beschäftigten berücksichtigt.

Die Ergebnisse der Gefährdungs- und Belastungsbeurteilung einschließlich der entwickelten Maßnahmen werden dokumentiert (gemäß § 6 ArbSchG).

#### Qualität des Arbeitssystems

Es werden nur zertifizierte (GS-Zeichen) Arbeitsmittel angeschafft. Die Arbeitsmittel werden, sofern vorgeschrieben, regelmäßig geprüft und die Beschäftigten verwenden nur geprüfte Arbeitsmittel. Es sind alle Voraussetzungen zu schaffen, damit die Arbeitsumgebung (wie Klima, Beleuchtung, Lärm, Raum) mindestens den vorgeschriebenen Bedingungen entspricht (zum Beispiel Arbeitsstättenverordnung, Bildschirmarbeitsverordnung). Die Arbeitsplätze sind bzw. werden ergonomisch gestaltet, wofür die materiellen Voraussetzungen zu schaffen sind. Die Beschäftigten sind verpflichtet, die zur Verfügung gestellten Arbeitsmittel ergonomisch zu benutzen.

#### Gebrauchstauglichkeit der Software

Die eingesetzte Software im Unternehmen muss gebrauchstauglich sein. Dazu müssen folgende Grundsätze der Dialoggestaltung berücksichtigt werden:

- Aufgabenangemessenheit
- Selbstbeschreibungsfähigkeit
- Steuerbarkeit
- Fehlertoleranz
- Erwartungskonformität
- Individualisierbarkeit
- Lernförderlichkeit

Zusätzlich ist bei der Darstellung von Informationen folgendes zu berücksichtigen:

- Organisation von Informationen
- Verwendung grafischer Objekte
- Gebrauch von Kodierverfahren

### Information und Unterweisung der Beschäftigten

Die Beschäftigten erhalten alle Informationen, die für ein fachlich hochwertiges, sicheres und gesundes Arbeiten am Bildschirm erforderlich sind. Mindestens jährlich beziehungsweise bei Neuanschaffung von Arbeitsmitteln und neuen Arbeitsaufgaben erhalten die Beschäftigten eine Unterweisung über Sicherheit und Gesundheit bei der Bildschirmarbeit.

### Verbesserungsprozess

Alle Beschäftigten besprechen Probleme und Fehler im Arbeitssystem sowie Abweichungen der Software von den Kriterien der Gebrauchstauglichkeit mit den zuständigen Vorgesetzten. Die Vorgesetzten werden Maßnahmen gemäß dem Grundsatz der Verhältnismäßigkeit ergreifen, um die Arbeitsbedingungen entsprechend den staatlichen und unfallversicherungsträgerseitigen Anforderungen gerecht zu werden.

Beschäftigte und Vorgesetzte besprechen regelmäßig, gemeinsam und partnerschaftlich, wie das Arbeitssystem zu verbessern ist.

### Mischarbeit und Pausen

Im Grundsatz wird für alle DV-gestützten Tätigkeiten die Arbeitsorganisation so gestaltet, dass Bildschirmarbeit regelmäßig durch andere Tätigkeiten unterbrochen wird. Sollte das nach eingehender Prüfung im Einzelfall nicht möglich sein, wird eine auf den Einzelfall abgestellte Pausenregelung vereinbart.

### Arbeitsmedizinische Vorsorge

Allen Beschäftigten an Bildschirmarbeitsplätzen ist während der Arbeitszeit eine arbeitsmedizinische Vorsorge durch die Betriebsärztin oder den Betriebsarzt anzubieten (Angebot alle 5 Jahre, bei über 40 jährigen Personen alle 3 Jahre). Diese umfasst eine ärztliche Beratung und das Angebot einer Untersuchung der Augen und Sehvermögens.

Wenn die arbeitsmedizinische Vorsorge ergibt, dass Beschäftigte eine spezielle Sehhilfe für die Arbeit am Bildschirm benötigen, trägt die Unternehmerin bzw. der Unternehmer hierfür in erforderlichem Umfang die Kosten.

### Qualifikationsmaßnahmen

Um eine Überforderung der Beschäftigten zu vermeiden und einen produktiven Einsatz der Software zu gewährleisten, erhalten alle Beschäftigten ein Training, das mindestens enthält:

- Vermittlung und Erweiterung der Grundkenntnisse des Umgangs mit der Software
- Vermittlung und Erweiterung der Grundkenntnisse zur Lösung der spezifischen Arbeitsaufgaben
- Informationen zum sachgerechten Umgang mit den Geräten (Hardware)

\_\_\_\_\_  
Unterschrift Unternehmer

\_\_\_\_\_  
Unterschrift Betriebs-/Personalrat

\_\_\_\_\_  
Datum

\_\_\_\_\_  
Datum

# Anhang 3

## Arbeitshilfen: IT-Erhebungsbogen aus C2Web

### 1 Erhebungsbogen für Teilnehmerinnen und Teilnehmer

#### 1.1 Einleitung

Mögliche Probleme mit einer Software kennen Sie, als Nutzerinnen und Nutzer am Besten. Der Erhebungsbogen dient dazu, ein erstes Stimmungsbild zu erhalten, um möglichen Handlungsbedarf für Verbesserungen rasch zu erkennen. Das Ausfüllen des Bogens nimmt maximal 10 Minuten in Anspruch.

#### 1.2 Datenschutzerklärung

Im Rahmen der Erhebung werden keine personenbezogenen Daten verwendet, dies ist lediglich optional vorgesehen. Daher ist eine Zuordnung der Auskünfte zu einer bestimmten Person nicht möglich.

#### 1.3 Informationen zur Befragung

Zu bewertende Software (Bezeichnung, Version):	z. B. Microsoft Word 2013
Nutzergruppe, der die teilnehmende Person angehört:	z. B. „Sachbearbeiter im Mahnwesen“ oder „Mitarbeiter am Check In“
Nachname, Vorname der befragten Person (Nur ausfüllen, wenn Sie dies für Rückfragen anbieten bzw. wünschen!)	

#### 1.4 Erhebung

	Benotung durch Nutzerinnen und Nutzer (Schulnotensystem)					
	1 Trifft voll zu	2 Trifft weitgehend zu	3 Trifft überwiegend zu	4 Trifft selten zu	5 Trifft sehr selten zu	6 Trifft gar nicht zu
1. Die Zeichen auf dem Bildschirm (z. B. in Menüs und Formularen) sind immer gut lesbar und unterscheidbar.						
Bemerkungen:						
2. Bei der Software sind Farben und Symbole immer eindeutig erkennbar, unterscheidbar und verständlich.						
Bemerkungen:						
3. Die Software erfordert keine überflüssigen Schritte und Eingaben, um zum Ziel zu kommen.						
Bemerkungen:						
4. Die Software zeigt dem Nutzer nur Informationen, die wirklich bei der Aufgabe benötigt werden oder dafür hilfreich sind.						
Bemerkungen:						

	Benotung durch Nutzerinnen und Nutzer (Schulnotensystem)					
	1 Trifft voll zu	2 Trifft weitge- hend zu	3 Trifft überwie- gend zu	4 Trifft selten zu	5 Trifft sehr selten zu	6 Trifft gar nicht zu
5. Die Software ist so voreingestellt, dass die Nutzerin bzw. der Nutzer selbst keine weiteren Voreinstellungen mehr vornehmen muss.						
Bemerkungen:						
6. Bei der Software ist die Bedeutung der angezeigten Begriffe, Bezeichner, Auswahlmöglichkeiten, Hilfen etc. verständlich und führt nicht zu Fehlinterpretationen.						
Bemerkungen:						
7. Bei der Software ist immer erkennbar, wie man das, was man tun will, auch tun kann.						
Bemerkungen:						
8. Die Software liefert immer Rückmeldungen, die verständlich und hilfreich sind.						
Bemerkungen:						
9. Die Software liefert aufgrund von Eingaben immer die benötigte Information, um einen Schritt weiter zu kommen.						
Bemerkungen:						
10. Die Software lässt sich ohne Einarbeitungsaufwand nutzen, wenn man die zu bearbeitende/auszuführende Aufgabe selbst gut kennt.						
Bemerkungen:						
11. Man kann jederzeit so fortfahren wie erforderlich, ohne Umwege gehen zu müssen.						
Bemerkungen:						

	Benotung durch Nutzerinnen und Nutzer (Schulnotensystem)					
	1 Trifft voll zu	2 Trifft weitge- hend zu	3 Trifft überwie- gend zu	4 Trifft selten zu	5 Trifft sehr selten zu	6 Trifft gar nicht zu
12. Der Nutzer wird bei fehlerhaften Eingaben vor gravierenden Aktionen durch die Software (z. B. durch Sicherheitsabfragen) geschützt.						
Bemerkungen:						
13. Bei fehlerhaften Eingaben ist der Fehler immer erkennbar und es ist klar, wie der Fehler behoben werden kann.						
Bemerkungen:						
14. Bei Bedarf kann die Nutzerin oder der Nutzer für sie bzw. für ihn ungünstige Voreinstellungen ändern.						
Bemerkungen:						

# Anhang 4

## Beispielhafte Methode zur formativen Evaluation

### Beispielhafte Methoden zur formativen Evaluation

Methode	Beschreibung	Ziel	Testgegenstand	Einsatzphase	Ergebnis	Hinweise zum Einsatz
Affinity Diagramm	Strukturieren und Systematisieren von vielen einzelnen ungeordneten Einzelinformationen (Ideen, Lösungsvorschläge, Aktivitäten, Funktionen) nach ihrer Nähe	Systematische Darstellung von zeitlichen, räumlichen oder sachlogischen Abhängigkeiten	Einzelinformationen	Planung und Konzeption, Entwicklung und Prüfung	Ablaufdiagramme, Themenschwerpunkte, Maßnahmenbündel, Alternativen	
Alternativen-Vergleichstest	Zwei Versionen der Software werden anhand von Merkmalen direkt miteinander verglichen	vergleichende Bewertung	Nutzung durch Benutzerinnen/ Benutzer	Entwicklung und Prüfung	relative Präferenzen oder Vergleiche einzelner Merkmale	Gefahr, dass eine bessere Alternative als gute Lösung deklariert wird, selbst wenn sie erhebliche Gestaltungsmängel hat
Aufgabenanalyse	Nutzungsszenarien erstellen	Beschreibung von Arbeitsprozessen in einer organisatorischen Einheit	Arbeitssystem	Planung und Konzeption, Entwicklung und Prüfung	Aufgaben und Erfordernisse	
Befragung (schriftlich, mündlich; standardisiert oder nicht)	Formen: Interview, Fragebogen. Personen antworten auf offene oder geschlossene und vorab festgelegte Fragen. Standardisierte schriftliche Befragungen beziehen sich auf festgelegte Merkmale	Anregungen erhalten, Meinungen abfragen, Bedürfnisse ermitteln, Einstellungen erheben, Bewertungen in angeleiteter Form ermitteln	Einschätzungen durch Expertinnen/Experten oder Benutzerinnen/Benutzer	Planung und Konzeption, Entwicklung und Prüfung, Einführung, Nutzung und Pflege, Ablösung	Spektrum an Meinungen, Einschätzungen	dient bei nicht standardisierten Befragungen der Exploration
Beobachtung des Nutzungskontexts	Beobachtung eines zukünftigen Nutzungskontexts und möglichst systematische und detaillierte Dokumentation	Nutzungskontext beschreiben	Arbeitssystem	Analyse von Aufgaben und Erfordernissen	spezifizierter Nutzungskontext	durch Beobachtungsschema systematisieren, wiederholte Durchführung auch von anderen Personen
Card Sorting	Einzelinformationen auf Karten werden strukturiert und systematisiert (kann die Erstellung von Affinity-Diagrammen unterstützen)	Darstellung von Einzelinformationen mithilfe von Karten flexibel und allmählich entwickeln	Inhalte und Struktur von Einzelinformationen	Planung und Konzeption, Entwicklung und Prüfung	Ablaufdiagramme, Themenschwerpunkte, Maßnahmenbündel, Alternativen	
Cognitive Walkthrough	Expertin/Experte handelt als Benutzerin/Benutzer während der Inspektion eines Software-Prototypen	Entwicklung alternativer Lösungsmöglichkeiten	Ausprobieren und Benutzung durch Expertinnen/Experten	Planung und Konzeption, Entwicklung und Prüfung, Einführung	Identifikation von Gestaltungsmängeln	
Dokumentenanalyse	gestaltungsrelevante Informationen in vorhandenen Dokumenten finden	Beschreibung von Erfordernissen und Anforderungen und ermitteln von Begründungen	vorhandene Dokumente	Planung und Konzeption	Erfordernisse, Anforderungen	Phase im Einsatzzyklus, Testtyp, Screeningdetail, Einsatzvoraussetzungen, Durchführungsaufwand, Aussagekraft

Methode	Beschreibung	Ziel	Testgegenstand	Einsatzphase	Ergebnis	Hinweise zum Einsatz
Eye Tracking	Erfassung und Dokumentation von Blickverläufen und Verweildauern der Benutzerin/des Benutzers während der Benutzung von Software	Blickpunkte, -verläufe und Verweildauern im Kontext	Nutzung durch Benutzerinnen/ Benutzer	Entwicklung und Prüfung	mit Aufgaben- und Nutzungskontext interpretierbar	
Fokusgruppe, Gruppendiskussion	moderierte themenzentrierte Gruppendiskussion zu einem vorab festgelegten Thema (Fokus)	Entwicklung von Lösungsmöglichkeiten für aufgedeckte Mängel; Abstimmung des weiteren Vorgehens für die Entwicklung	Einschätzungen von Benutzerinnen/ Benutzern und Expertinnen/ Experten	Entwicklungsprozess, Analyse von Aufgaben und Erfordernissen, Entwicklung von Prototypen, Nutzung im Kontext	Zwischenergebnisse bewerten, Maßnahmen ableiten, Dokumentation	Fokusgruppen werden für sehr viele verschiedene Themen angewendet. Detaillierte Anleitungen können recherchiert werden.
Gebrauchstauglichkeit testen / Usabilitytest	unterschiedliche Formen: (1) jedes einzelne der hier genannten Verfahren zur formativen Evaluation, (2) ein Testverfahren als Fragebogen, mit dem sich die Gebrauchstauglichkeit für ein Produkt ermitteln lassen soll, (3) Untersuchung am Arbeitsplatz der Benutzerin/des Benutzers oder im Labor zur Ermittlung von Gebrauchstauglichkeit	Gebrauchstauglichkeit des Produkts/ der Software	Nutzung durch Benutzerinnen/ Benutzer	Planung und Konzeption, Entwicklung und Prüfung, Einführung, Nutzung und Pflege, Ablösung	Ergebnis der Prüfung und Bewertung von Gebrauchstauglichkeit und Hinweise zur Ableitung von Maßnahmen zur Umgestaltung	
Heuristische Evaluation	Fachleute analysieren Software anhand von Heuristiken und typischen bzw. spezifizierten Nutzungsszenarien		Ausprobieren und Benutzung durch Expertinnen/ Experten	Nutzung im Kontext		
Lautes Denken	Während der Interaktion mit einem Software-Prototypen spricht eine Benutzerin/ein Benutzer alle Überlegungen, Gedanken und Handlungsabläufe laut aus	Unklarheiten während Benutzung identifizieren und Anregungen für Verbesserungsmaßnahmen gewinnen	Nutzung durch Benutzerinnen/ Benutzer	Entwicklung von Prototypen	Mängel, Fehler, Unklarheiten, fehlende Informationen	
onsite Befragungen	automatisierte Abfrage von Benutzerinnen/ Benutzern während des Software-Einsatzes durch aufpoppende Fragebogen		Nutzung durch Benutzerinnen/ Benutzer	Nutzung im Kontext		

Methode	Beschreibung	Ziel	Testgegenstand	Einsatzphase	Ergebnis	Hinweise zum Einsatz
Personas	Benutzerinnen/Benutzer werden anhand ihrer sogenannten Charaktereigenschaften und ihrer Verhaltensweisen mit Software durch Experten typisiert	typische Benutzer/Benutzerinnen identifizieren	Benutzer-/Benutzerinneneigenschaften	Planung und Konzeption	Nutzertypen	Softwaregestaltung orientiert sich an erfassten Nutzereigenschaften und nicht an ergonomischen Prinzipien der Gestaltung
Prototyping	Benutzerinnen/Benutzern werden Entwürfe der Software vorgelegt, die benutzt und nach Merkmalen bewertet werden sollen	Sammlung von Verbesserungsvorschlägen und Identifikation von Mängeln	Benutzung von Prototypen	Entwicklung und Prüfung	Gestaltungsmängel, Verbesserungsvorschläge	
Sequenzmodell	Analyse und systematische Dokumentation von Arbeitsschritten zur Aufgabenbearbeitung mit Vokabeln von Benutzungsschnittstellen	Ablaufmodelle für Aufgabenbearbeitung mit der Software entwickeln	Nutzeraufgaben, Software-Funktionsmodelle	Planung und Konzeption	Modell für eine Sequenz von Arbeitsschritten	
Storyboard	Der Arbeits- und Interaktionsprozess wird als Kurzgeschichte illustriert.	Arbeits- und Interaktionsabläufe lebendig, aber strukturiert beschreiben	Arbeits- und Interaktionsabläufe	Planung und Konzeption, Entwicklung und Prüfung	Wesentliche Schritte in den Arbeits- und Interaktionsabläufen	
SUMI (Software Usability Measurement Inventory)	Standardisierter Fragebogen mit 50 Fragen, der Beeinträchtigungen durch die Arbeit mit Software erhebt.	Beeinträchtigungen werden allgemein und spezifisch bewertet auf den fünf Dimensionen Affekt, Effizienz, Hilfestellung, Steuerbarkeit und Lernförderlichkeit.	Nutzung durch Benutzerinnen/Benutzer	Entwicklung und Prüfung, Nutzung und Pflege	Identifikation von Beeinträchtigungen durch die Arbeit mit Software	mindestens zehn hinreichend qualifizierte Nutzerinnen/Nutzer
SUS (System Usability Scale)	Mithilfe von zehn Fragen wird eine sehr allgemeine Einschätzung zur Gebrauchstauglichkeit eines Produkts ermittelt.	vermittelt sehr grobe Einschätzung zur Gebrauchstauglichkeit der Software	Nutzung durch Benutzerinnen/Benutzer	Entwicklung und Prüfung	unspezifische und vage Aussage zur Gebrauchstauglichkeit	
Szenarien	Beschreibung von Situationen wann Benutzerinnen/Benutzer wie warum mit welchem Ergebnis mit Software interagieren.	Einsatz- und Test-szenarien entwickeln; Nutzungskontexte ableiten	Nutzung durch Benutzerinnen/Benutzer	Planung und Konzeption, Entwicklung und Prüfung, Einführung, Nutzung und Pflege, Ablösung	Nutzungskontexte, Einsatz- und Test-szenarien	
Use Cases	Nutzungsablauf verschiedener Benutzerinnen/Benutzer ermitteln und dokumentieren	Konzeption für die Nutzerführung	Abläufe für Aufgabenbearbeitung	Planung und Konzeption	Ablaufmodell für einzelne Benutzungsgruppen	

# Anhang 5

Analyse der Aufgaben der Benutzerinnen und Benutzer sowie der Anforderungen und Funktionen der Software

Analyse der Aufgaben der Benutzerinnen und Benutzer sowie der Anforderungen und Funktionen der Software				
Nr.	Funktionen und Arbeitsaufgaben	Prioritäten		
		1	2	3
1	Welche grundlegenden Arbeitsaufgaben sollen mit der Software bearbeitet werden?			
1.1				
2	Aus welchen Teilarbeitsschritten setzt sich die Arbeitsaufgabe zusammen?			
2.1				
2.2				
2.3				
2.4				
2.5				
2.6				
2.7				
3	Was soll die Software innerhalb der einzelnen Teilaufgaben leisten?			
3.1				
3.2				
3.3				
3.4				
3.5				
3.6				
4	Welche speziellen Anforderungen der Software bei der Realisierung der Aufgaben wären wünschenswert?			
4.1				
5	Welche grundlegenden Funktionen soll die Software erfüllen? (Zum Beispiel: Drucken, Kopieren, Bearbeiten, Blättern, Listen, Tabellen, Formatieren, Sprache, Schriftgröße)			
5.1				
5.2				
5.3				
5.4				
5.5				
5.6				
5.7				
5.8				
5.9				

Analyse der Aufgaben der Benutzerinnen und Benutzer sowie der Anforderungen und Funktionen der Software				
Nr.	Funktionen und Arbeitsaufgaben	Prioritäten		
		1	2	3
6	Welche speziellen Funktionen in einzelnen Bereichen soll die Software erfüllen? (Zum Beispiel: Einlesen von Daten, aus welchen Programmen; Eingabeanforderungen, Rückmeldungen, Rechenfunktionen, Statistik, Grafikgestaltung, Videoverbindung)			
6.1				
6.2				
6.3				
6.4				
6.5				
6.6				
6.7				
6.8				
7	Welche Verknüpfungen von Daten soll die Software vornehmen? (Zum Beispiel: Gruppen bilden, Verknüpfung von Datenfeldern, Bild/Text-Kombination)			
7.1				
7.2				
7.3				
7.4				
7.5				
8	Wie soll die Software für die Arbeitsaufgabe gestaltet sein? (Zum Beispiel: Benutzerführung, Windowsoberfläche, Anpassung an bestehende Programme, Orientierungssysteme, Farben)			
8.1				
8.2				
8.3				
8.4				
8.5				
8.6				
8.7				
8.8				
9	Welche Bedienanforderungen soll die Software erfüllen? (Zum Beispiel: Dialoggestaltung, Menüausführung, Menüstruktur, Benutzungsschnittstellen, Barrierefreiheit)			
9.1				
9.2				
9.3				
9.4				
9.5				

Analyse der Aufgaben der Benutzerinnen und Benutzer sowie der Anforderungen und Funktionen der Software				
Nr.	Funktionen und Arbeitsaufgaben	Prioritäten		
		1	2	3
10	Mit welchen Programmen muss die Software kompatibel sein? (Zum Beispiel: Betriebssystem, andere Anwendungsprogramme, Internet/Intranet/Extranet)			
10.1				
10.2				
10.3				
10.4				
10.5				
11	Wie soll die Software Fehler behandeln? (Zum Beispiel: Sicherheitsabfragen, automatische Fehlerkorrekturen, Fehlerbehandlung durch Benutzerinnen und Benutzer)			
11.1				
11.2				
11.3				
11.4				
11.5				
12	Welche Voraussetzungen muss die Software erfüllen? (Zum Beispiel: Installation, Online-Nutzung)			
12.1				
12.2				
12.3				
12.4				
12.5				
12.6				
12.7				
12.8				

# Anhang 6

## Verzeichnis der verwendeten Normen

DIN EN ISO 6385:2004-05	„Grundsätze der Ergonomie für die Gestaltung von Arbeitssystemen (ISO 6385:2004-05)“
DIN EN ISO 7731:2008-12	„Ergonomie – Gefahrensignale für öffentliche Bereiche und Arbeitsstätten – Akustische Gefahrensignale (ISO 7731:2003)“
DIN EN ISO 9241-11:1999-01	„Ergonomische Anforderungen mit Bildschirmgeräten – Teil 11: Anforderungen an die Gebrauchstauglichkeit; Leitsätze (ISO 9241-11:1998)“
DIN EN ISO 9241-12:2000-08	„Ergonomische Anforderungen mit Bildschirmgeräten – Teil 12: Informationsdarstellung (ISO 9241-12:1998)“
DIN EN ISO 9241-110:2008-09	„Ergonomie der Mensch–System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung (ISO 9241-110:2006)“
DIN EN ISO 9241-171:2008-10	„Ergonomie der Mensch–System-Interaktion – Teil 171: Leitlinien für die Zugänglichkeit von Software (ISO 9241-171:2008)“
DIN EN ISO 9241-210:2011-01	„Ergonomie der Mensch–System-Interaktion – Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme (ISO 9241-210:2010)“
DIN EN ISO 9241-410:2012-12	„Ergonomie der Mensch–System-Interaktion – Teil 410: Gestaltungskriterien für physikalische Eingabegeräte (ISO 9241-410:2008 + Amd.1:2012)“
DIN EN ISO 10075-2:2000-06	„Ergonomische Grundlagen bezüglich psychischer Arbeitsbelastung – Teil 2: Gestaltungsgrundsätze (ISO 10075-2:1996)“
DIN EN ISO 11064-1:2001-08	„Ergonomische Gestaltung von Leitzentralen – Teil 1: Grundsätze für die Gestaltung von Leitzentralen (ISO 11064-1:2000)“
DIN EN 614-2:2008-12	„Sicherheit von Maschinen – Ergonomische Gestaltungsgrundsätze – Teil 2: Wechselwirkungen zwischen der Gestaltung von Maschinen und den Arbeitsaufgaben“
DIN EN 894-4:2010-11	„Sicherheit von Maschinen – Ergonomische Anforderungen an die Gestaltung von Anzeigen und Stellteilen – Teil 4: Lage und Anordnung von Anzeigen und Stellteilen“

Deutsche Gesetzliche  
Unfallversicherung e.V. (DGUV)

Glinkastraße 40  
10117 Berlin  
Tel.: 030 288763800  
Fax: 030 288763808  
E-Mail: [info@dguv.de](mailto:info@dguv.de)  
Internet: [www.dguv.de](http://www.dguv.de)