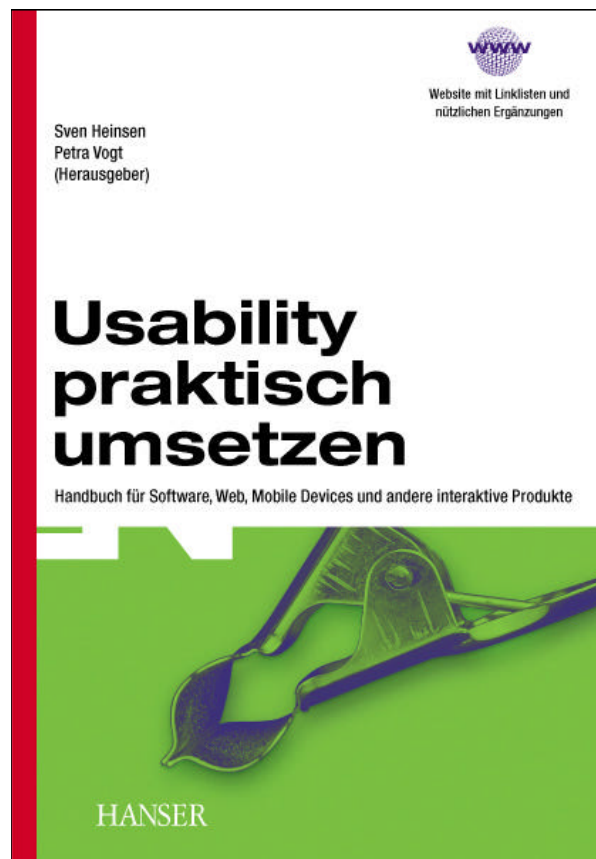


Kapitel 4 aus

# »Usability praktisch umsetzen«

Herausgegeben von Sven Heinsen und Petra Vogt  
© Carl Hanser Verlag 2003  
ISBN 3-446-22272-2



Zu bestellen über  
<http://www.hanser.de/buch/2003/3-446-22272-3.htm>



# 4

## Usability im Unternehmen



von Matthias Müller-Prove

## 4 Usability im Unternehmen

*Dieses Kapitel zeigt, in welchem komplexen Rollengeflecht Usability-Spezialisten im Unternehmen agieren und welchem Spannungsfeld sie dabei ausgesetzt sind. Es geht auch darauf ein, welche Rollen Usability-Experten in einem Team wahrnehmen und wie sie als Team organisiert sein können. Schließlich gibt es Tipps zum erfolgreichen „Evangelisieren“ des Themas Usability im Unternehmen.*

### 4.1 Der Usability-Experte in der Produktentwicklung

Benutzer sehen Programme als Hilfsmittel, die ihnen bei der täglichen Arbeit nützlich sein sollen. Das Spektrum ihrer Einschätzungen reicht dabei vom notwendigen Übel bis hin zum unverzichtbaren Werkzeug. Gute Software zeichnet sich nicht nur durch möglichst fehlerfreie Funktionen aus, sondern auch dadurch, dass die Benutzungsoberfläche ergonomisch gestaltet ist. Eine Software mit weniger Ecken und Kanten kann so zum selbstverständlichen Instrument werden, das die Kreativität der Anwender unterstützt. Für die Funktionalität sind die Entwickler zuständig – für das Interface die Usability-Experten.

Als Usability-Experte arbeitet man nicht alleine. Man ist Teil einer komplexen Organisation, die aus den beiden Hauptsäulen Engineering und Marketing besteht. Letztgenanntes ist nicht nur für die Vermarktung des Produktes zuständig, sondern auch in der entgegengesetzten Richtung für die Erhebung der Kundenanforderungen, aus denen die Spezifikationen für die nächste Version der Software zusammengestellt werden. Die Koordination dieser Bereiche wird von dem Produktmanagement übernommen, das in der Entwicklung oder im Marketing angesiedelt sein kann.

Unter Software sind im Folgenden Programme zu verstehen, die für einen großen Anwenderkreis entwickelt werden. Im Gegensatz zu Individualsoftware, die man in enger Zusammenarbeit mit dem Auftraggeber entwickelt, muss Standard-Software die Anforderungen von vielen Kunden abdecken. Die Situation im Webdesign ist hier grundsätzlich vergleichbar mit den Erstellungsprozessen von Software. Die Zahl der Besucher einer Website soll möglichst groß sein, sodass der einzelne Nutzer des Online-Angebots zumeist unbekannt bleibt. Im Unterschied zur Software-Erstellung sind aber die Produktzyklen im Web-Bereich deutlich kürzer. Man benötigt für den Launch einer professionellen Website Monate – die Projektzeit einer Software-Version beträgt hingegen zumeist mehr als ein Jahr.

Die Aufgabe des Usability-Experten besteht insbesondere darin, die Perspektive des Anwenders im Entwicklungsprozess zu vertreten. Er ist quasi Anwalt des späteren Benutzers und vertritt konsequent dessen Interessen gegenüber allen anderen Bereichen im Software-Entwicklungsprozess. In diesem dichten Geflecht von Verantwortlichkeiten hat er dafür zu sorgen, dass am Ende ein benutzbares und im Sinne der Software-Ergonomie gutes Produkt entwickelt wird.

Die folgenden Abschnitte beschreiben im Einzelnen die Zusammenarbeit mit den jeweiligen Teams und die Zielkonflikte, die dabei auftreten können. Neben Engineering und Marketing bilden das Qualitätsmanagement, die Dokumentation für die Benutzerhandbücher sowie die Support-Abteilung weitere wichtige Bereiche. Anschließend wird die Beziehung zwischen Usability und dem Upper-Management behandelt, dem die Aufgabe zukommt, langfristige und strategische Ziele festzulegen.

### 4.1.1 Engineering

Entwickler leben in einer eigenen Welt. Von wenigen Ausnahmen abgesehen, existieren für sie nur der Computer und sie selbst. Ein Problem gilt als erledigt, wenn der Algorithmus korrekt ist und der Compiler keine Syntaxfehler mehr meldet. In Sachen guter Bedienbarkeit sind sie bestenfalls ratlos, da dieser Aspekt häufig nicht Teil ihrer Ausbildung war. Viele erliegen dann leider allzu oft der Verlockung, sich selbst als Maßstab zu nehmen nach der Devise, „Wenn ich es bedienen kann, kann es auch jeder andere normale Mensch bedienen.“ – Entwickler sind nicht normal. Programmieren ist eine hochspezialisierte Tätigkeit, die viel Praxis erfordert und mit Fug und Recht in den Rang einer Ingenieurswissenschaft erhoben werden darf. Manchmal reicht sie sogar in ihrem Abstraktionsgrad an die Mathematik heran. Das Engineering bildet damit den Kern der Produktentwicklung, ohne den es nicht geht. Niemand sonst im Team ist so schwer zu ersetzen wie ein Entwickler. Er hat im Laufe der Zeit in hohem Maße Detailwissen über das Programm gesammelt, sodass es für einen Kollegen nur mit sehr viel Engagement möglich wäre, die Arbeit an dieser Stelle fortzuführen.

Es kommt häufiger vor, dass Entwickler diese Stellung ausnutzen, um den gesamten Entwicklungsprozess in die Richtung zu bewegen, die ihnen als die angenehmste erscheint. Dabei muss nicht einmal böse Absicht im Spiel sein. Es ist nur zu leicht, sich in diesem selbstreferenziellen Kreis zu verfangen. Alan Cooper zollt der dominanten Rolle des Entwicklers denn auch Tribut, indem er sie mit Irren vergleicht, die ihre Anstalt selber leiten: „The Inmates Are Running the Asylum“ (Cooper, 1999). Mit dieser Haltung haben die Entwickler es weit gebracht. Es besteht folglich aus ihrer Perspektive auch kein Anlass, sich mit den Ratschlägen eines Interaktionsdesigners zu befassen, die im Zweifel nur zusätzliche Arbeit bedeuten. Sie blocken ab, indem sie behaupten, dass der Vorschlag technisch nicht umsetzbar wäre.

Für den Usability-Experten ist es wichtig, diese Haltung zu verstehen, um angemessen damit umgehen zu können. Einem aus der Informatik stammenden Software-Ergonomen fällt es nicht schwer, sich in die Denk- und Arbeitsweise eines Entwicklers hineinzusetzen. Viel schwieriger haben es Usability-Experten, die aus der Psychologie oder dem Grafikdesign kommen. Sie sprechen per se nicht die Sprache der Engineers. Daher ist die Gefahr besonders groß, dass sie in der Entwicklungsabteilung nicht anerkannt werden. Wenn sie sich mit einer Idee auf dünnem Eis vortasten, werfen die Entwickler ihnen mit Vergnügen noch Steine zu.

Jeder Usability-Experte braucht Kompetenz und Standfestigkeit. Er muss damit rechnen, dass sein Gegenüber von der Methodik des Usability-Engineering nichts versteht, trotzdem aber auf seinen Lösungen beharrt. In dieser Situation ist der Usability-Experte gefordert, für die bessere Bedienbarkeit der Software zu werben und seine Vorschläge durch das Erklären des eigenen Vorgehens zu untermauern. Teils durch diplomatisches Geschick, teils durch permanentes Vortragen eines Problems kann man den Entwickler zum Einlenken bewegen. Auch die Konfrontation mit Videoaufzeichnungen der Usability-Tests kann einen heilsamen Schock auslösen. Sehr sinnvoll ist die Argumentation auch mit quantitativen Daten aus Usability-Studien zu belegen. Die Anzahl notwendiger Mausklicks oder die Quote erfolgloser Versuche von echten Benutzern, eine Aufgabe mit dem Programm zu erledigen, sind Zahlenwerte, die der Denkweise der Entwickler entgegenkommen.

Es ist unumgänglich, dass die beiden Parteien im Laufe der Zeit Respekt und Verständnis füreinander entwickeln. Anders gibt es kaum eine Chance für eine erfolgreiche Zusammenarbeit.

### 4.1.2 Marketing

Auf den ersten Blick sind sich die Ziele von Marketing und Usability am ähnlichsten. Beide sind an einem gut gestalteten und benutzbaren Produkt interessiert. Für das Marketing sind das hervorragende Verkaufsargumente – für den Usability-Experten sind es die wichtigsten Qualitätsmerkmale der Software überhaupt.

Auch in ihrer Methodik überdecken sich die beiden Bereiche erheblich. Mit Fokusgruppen, Fragebögen, Interviews, Kundenbesuchen (Site Visits), Konkurrenzanalysen und sonstigen Marktforschungsmethoden versucht das Marketing, die Kundenanforderungen zu erheben. Jedoch zeigen sich hier trotz einer vergleichbaren Vorgehensweise die ersten Unterschiede. Überspitzt ausgedrückt, sieht das Marketing dort Kunden, wo es für den Interface-Designer um die Anwender geht. Das Marketing will Funktionen in Erfahrung bringen, die die zukünftige Version des Produkts vom Mitbewerber abhebt und sich deshalb gut verkaufen lässt. Welche Art von Programm ist in Zukunft gefragt? Welche Funktionen muss es enthalten? In erster Linie sammeln sie Anforderungen der Kunden und betrachten dabei die Bedienbarkeit der Software nur als sekundäres Ziel. Usability ist nur eine Produkteigenschaft unter vielen. Dieser Umstand schlägt sich auch in den Feature-Listen nieder, in denen die Funktionen der neuen Version aufgeführt werden. Man findet dort unter ferner liefen den Eintrag *Improve Usability – Verbessert die Bedienbarkeit!* Das stellt als solches zwar einen berechtigten Wunsch des Kunden dar, jedoch ist er in dieser Form für den Usability-Experten wertlos.

Der Usability-Experte muss sich viel intensiver mit den Beziehungen zwischen dem Benutzer, dem Computer, den zu erledigenden Aufgaben und der Organisation beschäftigen, als das für ein erfolgreiches Marketing notwendig wäre. Unter Organisation ist in der Software-Ergonomie der gesamte Arbeitskontext des Benutzers zu verstehen. Die zu gestaltende Software ist ein Teil davon, der sich in die existierende Arbeitswelt einfügen muss. Die Aufgaben und ihre Strukturen können mit einer Task-Analyse erforscht werden.

Und ob schließlich das mentale Modell, dass sich der Benutzer von dem System macht, für die intendierten Funktionen angemessen ist, kann durch Usability-Tests ermittelt werden. Dabei zu Tage tretende Schwierigkeiten und Fehlbedienungen sind eindeutige Indizien dafür, dass die Benutzungsoberfläche klarer strukturiert werden muss.

Don Norman fordert, dass User-Experience neben Technology und Marketing die dritte tragende Säule der Produktentwicklung sein sollte (Norman, 1998). Auf diese Weise löst er den Zielkonflikt zwischen dem Marketing, nämlich möglichst viele Kopien der Software zu verkaufen, und dem Software-Ergonomen, möglichst gute Produkte für den Benutzer zu gestalten. Solange diese Position aber für die User-Experience noch nicht erreicht ist, müssen sich die Usability-Experten in den vorhandenen Produktentwicklungsprozess von Marketing und Entwicklung einfügen.

Die Diskussion mit dem Management um einen vollwertigen und eigenständigen Usability-Bereich innerhalb des Product-Life-Cycle (PLC) wird häufig dadurch erschwert, dass die Methodik der Usability-Experten vom Ansatz her ähnlich der des Marketings ist. Die Sichtweise und die Fragestellungen, mit der Usability-Experten auf die Anwender zugehen, ergibt aber einen erheblichen Unterschied zwischen Usability und Marketing.

### **4.1.3 Produktmanagement**

Die Hauptaufgabe des Produktmanagements ist die Koordination aller Beteiligten im Software-Entwicklungsprozess. Da man als Usability-Experte den gesamten Prozess begleitet, ist die Beziehung zum Produktmanagement besonders eng. Der Usability-Experte hat die Aufgabe, auf die Belange des Anwenders hinzuweisen und für die notwendigen Maßnahmen im Projektplan zu werben. Denn je später Probleme in der Benutzbarkeit aufgedeckt werden, um so aufwändiger ist ihre Behebung.

In der ersten Phase der Produktentwicklung wurde vom Marketing eine Anforderungsdefinition (Product Requirements Document) erstellt. Die dort aufgeführten Einzelpunkte ergeben aber noch kein rundes Produkt, so wenig, wie ein Einkaufszettel einen fertigen Kuchen ergibt. Die Kundenwünsche müssen sinnvoll gruppiert und priorisiert werden. Hier ist das Produktmanagement gefordert, mit allen Beteiligten ein stimmiges Konzept zu erarbeiten und es verbindlich festzuhalten (Product Concept Document). Der Usability-Experte spielt dabei eine große Rolle, weil hier das künftige Produkt Kontur annimmt. Hier muss der Abgleich stattfinden, ob das Produkt die Marktanforderungen erfüllt. Es stellt sich insbesondere die Frage, ob es sinnvoll in den Arbeitskontext des Benutzers eingefügt werden kann. Dann folgt eine Phase, in der das nun fertige Produktkonzept im Detail ausgearbeitet wird. Es werden Functional-Specifications und UI-Specifications erstellt, die schriftlich festlegen, wie das Programm zu funktionieren hat, wie es sich verhalten und wie es aussehen soll. Der Sinn dieser Phase ist die Tatsache, dass man Änderungen auf Papier wesentlich leichter und schneller ausführen kann, als das mit Programm-Code möglich wäre. Die Konzepte sollten durchdacht und überprüft sein, bevor sie von der Entwicklungsabteilung implementiert werden.

#### 4.1.4 Customer-Support, Dokumentation und Qualitätsmanagement

Neben den Mitarbeitern der Qualität können und sollten auch die Kollegen aus den nicht primär an der Entwicklung beteiligten Abteilungen zur besseren Bedienbarkeit der Software beitragen. So verfügt die Support-Abteilung über schier unerschöpfliches Wissen zu Problemen hinsichtlich der aktuell am Markt befindlichen Software-Version. Sie führen in der Regel mittels einer Datenbank eine Statistik zu Anfragen der Kunden und erstellen daraus die so genannten FAQs (Frequently Answered Questions), Listen der am häufigsten gestellten Fragen. Dieses Reservoir gilt es, für den Usability-Experten anzupapfen, um die neue Version der Software besser bedienbar zu machen.

Die Autoren der Handbücher stehen vor der Herausforderung, ein Produkt zu dokumentieren, während es sich noch in der Entwicklung befindet. Als Quelle ziehen sie deshalb die UI- und Feature-Spezifikationen zu Rate und sind darum geradezu prädestiniert, die neuen Konzepte zu begutachten. Konsistente und plausible Interface-Elemente lassen sich nämlich weitaus leichter erklären als unnötig komplizierte Prozesse. Ihre Kommentare und Anmerkungen liefern wichtige Indikatoren, um die Interaktion mit der Software vor Fertigstellung nochmals zu verbessern.

Das Qualitätsmanagement hat natürlich in erster Linie den Auftrag, das Programm auf funktionale Fehlerfreiheit hin zu überprüfen. Da sich aber die Ansicht durchsetzt, dass gute Bedienbarkeit auch ein Qualitätsmerkmal ist, muss die Qualitätsabteilung diesen Bereich mit abdecken. Frei nach dem Motto *If it's not usable – it is a bug* tragen die Mitarbeiter der Qualität die problematischen Fälle in ihr Bug-Tracking-System ein.

Der Usability-Engineer fungiert als Ansprechpartner für die hier kurz vorgestellten Abteilungen. Er muss sicherstellen, dass die Vorschläge in die neue Produktversion einfließen können. Dies ist eine sehr angenehme Aufgabe, da die meisten Anregungen auf einem sehr hohen Niveau liegen.

#### 4.1.5 Upper-Management

Das Upper-Management umfasst diejenigen Manager, die für die gesamte Entwicklung des Produktes verantwortlich zeichnen. In kleinen und mittleren Firmen ist das häufig der Geschäftsführer. In großen Firmen kann es eine ganze Kaskade vom Entwicklungsleitern, CTOs (Chief Technology Officer), Vice-Presidents, bis hin zum CEO (Chief Executive Officer) geben. Der Einfluss, den das Upper-Management auf das Produkt nimmt, ist nicht zu unterschätzen. Und solange dies im Interesse des Erfolges des Gesamtunternehmens geschieht, ist es auch zu begrüßen. Dabei besteht jedoch die Gefahr, dass die Ideen des Upper-Managements nicht mit den Ergebnissen eines benutzerzentrierten Ansatzes übereinstimmen. Zwei Beispiele sollen das verdeutlichen.

Zu nennen wäre hier der Typus des *Demo-Feature-Chefs*, der den Einbau von Funktionen fordert, die bei Produktvorführungen auf Messen einen großen Eindruck hinterlassen. Sie werden dort von speziell trainierten Promotoren im Kontext einer speziell dafür konzipier-



ten Aufgabe gezeigt. Solche Funktionen können wichtige Marketinginstrumente sein, da Journalisten diese Funktionen gerne für ihre Artikel aufgreifen. Für den Arbeitsalltag sind diese Features allerdings nur bedingt geeignet. Sie sind nicht ausgereift und nicht in Usability-Studien verifiziert. Gemäß Alan Kay, dem Visionär des Personal-Computers, ist die Benutzungsschnittstelle vom Apple Macintosh die einzige, die es wert ist, kritisiert zu werden. Darum hier ein Beispiel für solch ein Demo-Feature aus dem Apple-Betriebssystem Mac OS X (siehe Abbildung 4.1). Das Dock ist grafisch ansprechend gestaltet und mit viel Liebe zum Detail animiert.



**Abbildung 4.1** Das Dock von Apple Mac OS X wächst am unteren Bildschirmrand in beide Richtungen, wenn es zusätzliche Objekte aufnehmen muss.

Unter Usability-Gesichtspunkten ist es jedoch stark verbesserungsbedürftig. Je nachdem, wie viele Objekte im Dock dargestellt werden, befindet sich nämlich der Papierkorb immer an einer anderen Stelle auf dem Bildschirm. Der Benutzer kann sich nicht an eine feste Position gewöhnen und muss daher immer erst suchen, bevor er eine Datei in den Papierkorb ziehen kann. Diese Geste mit der Maus kann folglich nicht motorisch gelernt werden, was sie zu einem ständigen Bremsfaktor in der Interaktion werden lässt. Zur Verdeutlichung stelle man sich nur vor, dass bei der Gangschaltung eines Autos der vierte Gang jedesmal woanders zu finden wäre.



**Praxistipp: Websites mit aktuellen Kolumnen**

Einige Websites behandeln regelmäßig aktuelle Fragestellungen des Interface-Designs. So kommentiert Jakob Nielsen unter [www.useit.com](http://www.useit.com) alle zwei Monate die Benutzbarkeit von Websites. Demgegenüber hat Bruce Tognazzinis Webzine AskTog einen allgemeineren Fokus. Unter dem Titel „Top 10 Reasons the Apple Dock Sucks“ diskutiert er die Probleme rund um das Dock von Mac OS X.

Die Webadresse lautet [www.asktog.com/columns/044top10docksucks.html](http://www.asktog.com/columns/044top10docksucks.html).

Nachdem der *Es-muss-aussehen-wie-Chef* zum Beispiel in dem Magazin „Wired“ etwas gesehen hat, ist er der Meinung, dass eine Funktion oder eine bestimmte Art der Darstellung unbedingt in das eigene Produkt eingebaut werden muss, da es sonst chancenlos am Markt ist. Was der Chef sagt, ist durch seine Rolle im Unternehmen für die Entwicklungsabteilung verbindlich. Keine Instanz wagt es mehr, die Forderung auf ihren Sinn hin zu hinterfragen. Es wird letztlich um den technischen Aufwand argumentiert, aber allenfalls der Usability-Experte fragt nach dem Gesamtkonzept. Werden damit nicht die für das Produkt ausgearbeiteten Interaktionsrichtlinien verletzt? Was würden Benutzertests ergeben, wenn man sie durchführen könnte?

Trotz dieser beiden überzeichneten Beispiele sollte nicht vergessen werden, dass das Upper-Management einen durchaus positiven Einfluss im Sinne der Usability haben kann. Maryam Mohit, V.P. of Site Development bei Amazon und zuständig für die Online-Customer-Experience, sagte in einem Interview (Hurst, 2002): „Von Anfang an war bei Amazon die Kundenorientierung ein oberstes Ziel. Nicht zuletzt Jeff Bezos selbst, der Gründer und CEO des Unternehmens, war immer ein Vorreiter, wenn es darum ging, den Fokus auf die Kunden zu richten. Dadurch hat er Mitarbeiter für die Firma gewinnen können, denen gleichfalls die Bedeutung von Usability bewusst ist.“ Wenn die Verantwortlichen die Bedeutung des Usability-Experten im Entwicklungsprozess verstanden haben, sind das die besten Voraussetzungen für ein professionelles und effektives Arbeiten.

## 4.2 Organisation der Usability-Experten

Nachdem im vorigen Abschnitt die Beziehungen zwischen dem Usability-Experten und den anderen Abteilungen im Unternehmen geschildert wurden, soll es nun um die Organisation der Usability-Experten untereinander gehen.

### 4.2.1 Was ist ein Usability-Experte?

Da es kein allgemein anerkanntes Rollenverständnis eines Software-Ergonomien und Interface-Designers gibt, wird das Tätigkeitsfeld auch von großen amerikanischen Software-Herstellern mit den unterschiedlichsten Bezeichnungen versehen. Recht üblich ist der Begriff *User Interface Design*. Um zusätzlich auszudrücken, dass die Arbeit mehr ist als reines Pixel-Design, dass sie insbesondere die Gestaltung der Gesamtstruktur der Software umfasst, ist *User Interface & Software Design* ein angemessener Titel. Dieser Begriff entstammt Terry Winograds Buch „Bringing Design to Software“ (Winograd, 1996), der wunderbar auf die Gestaltung eines Gesamtsystems hinweist, das eine viel tiefgreifendere Herangehensweise verlangt, als später an der Oberfläche sichtbar ist. Häufig trifft man auch auf den Begriff des *Interaktionsdesigns*, der die dynamischen Aspekte der Verwendung der Software hervorhebt. Der Name *User-Centered Software Design* betont mehr als alle anderen Begriffe die zentrale Bedeutung des Anwenders im Software-Entstehungsprozess. Es ist der Anwender, dem der Einsatz der Software nützlich sein soll und dessen Bedürfnisse bei seiner täglichen Arbeit durch ergonomisch gestaltete Programme zu berücksichtigen sind. Apple Computer hat dafür schließlich den Begriff *User Experience* geprägt. Er bezeichnet ein ganzheitliches Verständnis im Interface-Design. Die Interaktion des Benutzers mit dem Produkt beginnt nicht morgens um neun im Büro mit einem Doppelklick, sondern schon viel früher. Verpackung, Installation und Kundenservice gehören genauso zum Produkt, wie die eigentliche Benutzungsschnittstelle.

Bei der Gestaltung gebrauchstauglicher Software kann man einige spezielle Aufgabengebiete identifizieren. Experten dieser Teilgebiete bilden dann zusammen das Usability-Team.

### 4.2.2 Das Usability-Team

Damit ein Usability-Team gut funktioniert, müssen Spezialisten zweier Kategorien vertreten sein. Zum einen benötigt man Fachleute, die sich auf das Erheben von Usability-Daten verstehen, wie zum Beispiel den Usability-Requirements-Engineer und den Usability-Testing-Engineer. Ihre Aufgabe ist das objektive Beobachten der Arbeitssituation der Nutzer bzw. die genaue Durchführung von Usability-Tests in Usability-Labs. Die zweite Kategorie sind die kreativen Gestalter, die Interaktionsdesigner und Visual-Designer, deren Entwürfe wiederum von den erstgenannten validiert werden. Des Weiteren findet man den Usability-Engineer, der das Zusammenspiel dieser Gruppen im Usability-Engineering-Prozess plant, koordiniert und überwacht.

Es gibt Fachleute, die mehr als eine Teildisziplin beherrschen. Eine grenzüberschreitende Kombination ist aber eher selten anzutreffen. Zu groß sind die Mentalitätsunterschiede, die einen Forscher von einem Designer unterscheiden. Das wird klarer, wenn nachfolgend die einzelnen Gebiete etwas genauer erläutert werden.

Ein *Requirements-Engineer* ist für die Ermittlung der Kundenbedürfnisse zuständig. Die konkreten Eindrücke von Arbeitssituationen beim Kunden, die er durch Site-Visits gewinnt, werden von ihm zu Benutzerprofilen abstrahiert. Dadurch wird dem Projektteam klar, für welche Art von Benutzern die Software eigentlich entwickelt wird. Aus den geschilderten Tätigkeiten entstehen Szenarien typischer Arbeitsabläufe. Das Ausformulieren hilft dabei, sich den Kontext zu verdeutlichen, in dem die Software eingesetzt werden soll. Es ist wichtig, den Unterschied zu beachten zwischen dem, was die Kunden wollen, und dem, was sie wirklich brauchen. Ein häufiger Fehler ist nämlich, dass die Aussagen der Kunden direkt in Produkthanforderungen übersetzt werden, anstatt den gesamten Nutzungskontext zu betrachten.

Der *Usability-Testing-Engineer* untersucht wiederholt die aktuellen Prototypen auf Probleme in der Bedienbarkeit. Dazu führt er im Usability-Lab Studien durch, in denen eine Auswahl von echten Benutzern die neuen Konzepte der Software erproben. Werden hier akute Probleme aufgedeckt, können sie mit vertretbarem Aufwand in den Spezifikationsdokumenten nachgebessert werden, ohne den gesamten Terminplan in Gefahr zu bringen.

Obwohl sich ein eigener Abschnitt in diesem Buch dem Usability-Testing widmet, sei an dieser Stelle angemerkt, dass der Usability-Testing-Engineer eine rein beobachtende Aufgabe hat. Unvoreingenommen führt er die Tests durch und berichtet objektiv die Ergebnisse an die Interaktionsdesigner. Es ist nicht seine primäre Aufgabe, Verbesserungsvorschläge zu machen – dafür sind die Interaktions- und Visual-Designer besser geeignet.

Anhand der Ergebnisse der Requirements-Analyse entwickeln die *User-Interface- und Software-Designer* ein Gesamtkonzept, das die Anforderungen der Kunden erfüllt und das in sich stimmig ist. Dabei müssen sie zusätzlich darauf achten, dass ihre Ideen mit den jeweiligen Styleguides und HCI-Prinzipien in Einklang stehen. Ihre Entwürfe verfeinern sie dann zu den so genannten User-Interface-Spezifikationen – kurz UI-Specs, nach denen die Entwickler die Software implementieren werden.

Parallel zu den UI- & Software-Designern arbeiten die *Visual-Designer* an den grafischen Elementen der Software. Das Layout der Kontrollelemente in Dialogfenstern, die Gestaltung von Programm-Icons und von Icons in Werkzeugleisten sind nur einige ihrer Aufgaben. Für die Gestaltung dieser Elemente bedarf es eines besonderen Könnens in Grafikdesign und visueller Kommunikation. Als Beispiel sei hier das Icon für das Absenden einer E-Mail angeführt. Ein Briefkasten scheint dafür eine gute Metapher zu sein. Wer jetzt an eine gelbe Metallbox mit Schlitzfenstern denkt, hat für deutsche Kunden eine gute Idee zur grafischen Umsetzung. Außerhalb Deutschlands trifft man damit aber nur auf Unverständnis. In England sind Briefkästen rot – in den Vereinigten Staaten hingegen sind sie blau und haben ein gewölbtes Kopfteil.

Asien hat wieder andere kulturelle Normen. General Magic hatte gegen Konventionen verstoßen, als sie Mitte der neunziger Jahre in dem PDA-System Magic Cap die Animation eines Hundes über den Desktop des Benutzers laufen ließen, der in den Dokumenten nach einem Suchbegriff schnüffeln sollte. In Südostasien gehört ein Hund nicht auf den Tisch. Ein professioneller Visual-Designer ist für solche interkulturellen Fragestellungen sensibilisiert und kann damit Probleme vermeiden, ehe das Produkt im internationalen Markt scheitert. Es ist ein Design anzustreben, das unbedenklich in allen Ländern und Kulturen eingesetzt werden kann. General Magic hatte übrigens rechtzeitig vor der Markteinführung der asiatischen Version den Hund durch eine Eule ersetzt.

Die eigenständige Position eines *Technical-Writer* wird dann sinnvoll, wenn die Zahl der Spezifikationsdokumente unübersichtlich zu werden droht. Er sorgt durch formale Layoutvorgaben für eine gleichbleibend hohe Qualität der Spezifikationen und kümmert sich um eine angemessene Bereitstellung der Dokumente im Intranet. Spezifikationen von allgemeinerem Charakter gehen in seine Verantwortung über. Das führt mit der Zeit zu einem Satz firmeninterner User-Interface-Guidelines, mit deren Hilfe die Konsistenz zwischen Programmversionen, aber auch zwischen verschiedenen Produkten einer Firma erhöht werden kann.

Es liegt schließlich in der Verantwortung eines *Usability-Engineers*, alle Aktivitäten zur Verbesserung der Benutzbarkeit der Software zu koordinieren. Er kann am besten die Fähigkeiten seiner Kollegen einschätzen und sie bei ihren Aufgaben unterstützen. Typischerweise leitet er das Usability-Team und repräsentiert es nach außen. Einem Usability-Engineer obliegt auch die Planung im Entwicklungsprozess und darüber hinaus die Etablierung der notwendigen Schritte zur Verbesserung der Usability im gesamten Product-Life-Cycle.

### 4.2.3 Organisation der Usability-Experten bei mehreren Projektteams

Alle eben geschilderten Aufgaben müssen vom Usability-Team abgedeckt werden. In kleineren Firmen mit wenigen Usability-Experten kann es folglich dazu kommen, dass ein Software-Ergonom mehrere Aufgabenbereiche übernehmen muss. Größere Firmen, die mehrere Produkte entwickeln, haben einen entsprechend höheren Bedarf an voll besetzten

Usability-Teams. Es sei an dieser Stelle angemerkt, dass man in beiden Situationen zwischen Usability-Experten und Software-Entwicklern ein numerisches Verhältnis von eins zu zehn anstreben sollte. Bei einer geringeren Quote kommt es fast zwangsläufig zu Zugeständnissen bei der Benutzbarkeit des Produktes, da die anstehenden Aufgaben nicht mehr in angemessener Qualität und Zeit bewältigt werden können.

Es gilt, zwei grundsätzliche Formen zu unterscheiden: eine zentralisierte und eine dezentralisierte Organisation der HCI-Fachleute. Man spricht von einer *dezentralisierten Organisation*, wenn die Usability-Experten zu den jeweiligen Produktteams gehören. Sie sind dann der Entwicklung oder dem Produktmanagement angegliedert. Um ihren Status gegenüber den anderen Teams anzuheben, können sie aber auch eine dazu gleichberechtigte Interface-Abteilung bilden. Eine dezentralisierte Organisation hat den Vorteil, dass die Usability-Experten von den Entwicklern unmittelbar als Mitglieder des Teams wahrgenommen werden. Außerdem wännen sich die zuständigen Produktmanager die Kontrolle über den Usability-Bereich zu haben. Als Nachteil ist anzuführen, dass die zuständigen Produktmanager typischerweise keine Erfahrungen auf dem Gebiet der HCI besitzen. Es ist auch ungünstig, dass der Kontakt zu den Usability-Experten der anderen Produkte aktiv gesucht werden muss. Interne HCI-Mailinglisten und Newsletter, regelmäßige Design-Meetings sowie gemeinsame Workshops außerhalb der regulären Arbeit sind einige Beispiele, wie Zusammenhalt und Meinungsaustausch der Usability-Experten gefördert werden sollten.

Die *zentralisierte Organisation* fasst alle Usability-Experten in einem gemeinsamen Team zusammen, das von einem Manager oder HCI-Direktor geleitet wird. Diese Organisationsform begünstigt die Kommunikation zwischen den Fachleuten. Auch können einige Mitarbeiter zwischen Projekten aufgeteilt werden. So kann ein Visual-Designer an den Icons mehrerer Projekte arbeiten, was ohne zusätzlichen Aufwand zu einem einheitlicheren Erscheinungsbild aller Produkte der Firma beiträgt. Die Einrichtung eines eigenen Usability-Labors rentiert sich, da bei einer wachsenden Produktpalette für die Auslastung des Labs gesorgt ist. Man muss allerdings damit rechnen, dass es zu terminlichen Engpässen kommen kann, wenn mehrere Produkte auf die Entwürfe des Visual-Designs oder auf einen freien Termin im Usability-Lab warten. Diesen Problemen begegnet man am besten durch eine langfristige Koordination der jeweiligen Projektpläne; ein Garant für einen reibungslosen Ablauf ist das allerdings nicht, da es immer wieder zu kurzfristigen Anforderungen kommen kann. Eine zentralisierte Organisation der Usability-Fachleute birgt außerdem die Gefahr, dass man die Empfehlungen zur besseren Bedienbarkeit von den Produktteams als Last von außen sieht und sie deshalb pauschal abqualifiziert.

Es ist darum notwendig, dass Mitglieder des Usability-Teams ständig im Produktteam präsent sind und auf die Bedeutung von Usability hinweisen. In der Praxis führt das zu einer Mischform aus Zentralisierung und Dezentralisierung. Große Software-Unternehmen betrauen mindestens einen Usability-Experten mit der Hauptverantwortung für das Design eines jeden Produktes. Für diese Stellen ergibt sich eine mehr oder minder ausgeprägte Matrixorganisation, da sie auf der einen Seite eng mit den Entwicklern und dem Produktmanagement zusammenarbeiten und auf der anderen Seite auf die Kompetenz des gesam-

ten Usability-Teams zurückgreifen können, um Einzelfragen zu erörtern oder Teilaufgaben abzugeben.

Robin Jeffries, Distinguished Engineer und Leiterin des User Experience Office von Sun Microsystems, konstatiert, dass Usability-Gruppen, die eher auf firmeninternes Consulting ausgerichtet sind, die zentralisierte Organisationsform bevorzugen. Auch bei Projekten mit kurzen Entwicklungszyklen, wie sie zum Beispiel im Web-Bereich vorkommen, ist es meist nicht ratsam, die HCI-Experten für jedes Projekt neu zu organisieren (Jeffries, 2000).

#### 4.2.4 Usability in international operierenden Firmen

Mensch-Maschine-Kommunikation, Software-Ergonomie, selbst Informatik sind Begriffe, die einem amerikanischen Kollegen nicht geläufig sind. Das entsprechende Studienfach heißt an den dortigen Universitäten Computer Science. Der Teilbereich der Mensch-Maschine-Kommunikation ist am besten mit dem Begriff Human-Computer Interaction (HCI) zu übersetzen. Die Eigenschaft der Benutzbarkeit oder Gebrauchstauglichkeit von Software wird im Englischen Usability genannt. Neben den rein sprachlichen Hürden, die sich aber recht leicht überwinden lassen, kann es noch weitere Probleme in der Zusammenarbeit geben. Die meisten großen Software-Häuser haben sich an der Westküste der Vereinigten Staaten angesiedelt. Sie entwickeln ihre Software aber schon lange nicht mehr nur für den amerikanischen Markt, und so darf es nicht verwundern, dass auch die Entwicklungsabteilungen nicht mehr ausschließlich in Amerika angesiedelt sind. In einem international operierenden Unternehmen verteilen sich folglich auch die Usability-Experten auf die weltweiten Standorte. Ein Zeitunterschied von neun Stunden zwischen Silicon Valley und Deutschland schafft nicht gerade günstige Randbedingungen, um aus den räumlich zerstreuten Usability-Fachleuten ein gemeinsames Team zu bilden. Telefonkonferenzen sind nur ein unzureichendes Hilfsmittel, zumal im Usability-Bereich häufig visuelle Alternativen diskutiert werden müssen. Mittels Videokonferenzen und Webseiten im Intranet kann man dieses Handicap etwas ausgleichen. Es wird aber deutlich, wie notwendig ausgeprägte Fähigkeiten zur Kommunikation unter den Usability-Experten sind.

### 4.3 Das Web als zentrales Präsentations- und Arbeitsmedium

Das World Wide Web wurde 1989 am CERN in Genf von Tim Berners-Lee und Robert Cailliau entwickelt. Ihr Leitgedanke war ein plattformübergreifendes Hypertext-System zum Schreiben, Verknüpfen, Verteilen und Lesen von Informationen. Nach der Kommerzialisierung des Web ist aber das Erstellen eigener Web-Seiten nicht mehr so einfach möglich, wie es sein sollte. Das World Wide Web bietet ein enormes Potenzial, wenn es darum geht, aktuelle Ergebnisse zu verbreiten. Für den Interface-Designer ist darum das hauseigene Intranet das Medium der Wahl. Protokolle von Kundenbesuchen und Fokusgruppen,

Studien, Design-Entwürfe, Prototypen bis hin zu den UI-Spezifikationen lassen sich elegant über das Web dem übrigen Projektteam zur Verfügung stellen.

Wenn man das Web als sein primäres Arbeitsmedium begreift, bekommt man ohne Extraaufwand ein Forum im Intranet. Hyperlinks zu anderen Seiten im Intranet, die für das Produkt von Belang sind, wie zum Beispiel Marketing oder technische Spezifikationen, speichert man nicht in lokalen Bookmark-Listen seines Browsers, sondern setzt sie unmittelbar auf die internen Web-Seiten in den Kontext des UI-Designs des Produkts. Meeting-Protokolle und Mockups, die rein grafische Entwürfe darstellen, werden gleich im Web abgelegt. Sie sind so ohne großen Aufwand für alle Interessierten einsehbar. Durch Hyperlinks sind auch die Bezüge zwischen der ersten Phase, dem Erheben der Kundenanforderungen, und der zweiten Phase, der Konzeption und Gestaltung der Software, gut darstellbar.

Eine von der Geschäftsleitung oktroyierte Uniformität des Intranet ist meist zugleich sein Untergang. Daher sollte die Hemmschwelle so niedrig wie möglich gehalten werden, damit möglichst viele Informationen von möglichst vielen Kollegen im Web abgelegt werden. Die grafische Gestaltung ist dabei zunächst ein untergeordnetes Ziel. Einige Basisregeln können jedoch helfen, dass kein heilloses Durcheinander entsteht. Eine grobe Aufteilung der Website in Unterordner für jedes Team oder eine eigene Web-Domain für jedes Projekt ergeben die erste Strukturierungsebene. Um die einzelnen Dokumente besser zuzuordnen zu können, sollte der Autor und das Datum der letzten Änderung Bestandteil jeder Seite sein. Es kann auch die Aufgabe eines Technical-Writer im Team sein, die Inhalte im Intranet quasi redaktionell zu betreuen.



**Praxistipp: Pfade konstant halten**

Halten Sie die Pfade zu den HTML-Dateien und Dokumenten (die URLs) nach Möglichkeit konstant. Sie vermeiden damit eine Gefährdung der Konsistenz der Hyperlinks zwischen den Dokumenten und steigern das Vertrauen der Leser in Ihre Seiten. Für andere Abteilungen, wie zum Beispiel das Marketing, wird es möglich, auf die Seiten zum User-Interface zu verweisen. Sie können außerdem in Ihren E-Mails uneingeschränkt URLs versenden, da sie für die Empfänger nicht wertlos werden, wenn Sie Ihre Site neu strukturieren.

Die Web-zentrierte Arbeitsweise ersetzt nicht die direkte Kommunikation mit den Kollegen der anderen Abteilungen, aber sie unterstützt die Zusammenarbeit nachhaltig. Der zusätzliche Aufwand, den man zweifelsohne betreiben muss, um die Seiten des Intranet zu erstellen und zu pflegen, rentiert sich dadurch, dass man nicht mit allen Kollegen alle Details neu durchdiskutieren muss. Mitunter genügt einfach ein Hinweis auf die entsprechenden Seiten im Web.

## 4.4 Inspire, Inform, Consult

Für Usability-Experten kann jede Unterstützung, die sie aus anderen Bereichen bekommen, nur herzlich willkommen sein. Entwickler, die um ihre eigenen Stärken wissen – hoffentlich das Entwickeln – und ihre Schwächen kennen – wahrscheinlich den benutzerzentrierten Ansatz – und die deshalb proaktiv auf die Usability-Experten zugehen, sind Gold wert. Marketing-Leute, die nicht nur Käufer, sondern auch die Anwender dahinter bedenken, stehen einer Zusammenarbeit mit den Usability-Experten aufgeschlossen gegenüber. Damit solche Kollegen nicht die Ausnahme darstellen, sollte es sich der Usability-Experte zur Gewohnheit machen, durch stetiges Aufklären seine in Software-ergonomischen Belangen unerfahrenen Arbeitskollegen für Usability zu sensibilisieren und interessieren. Den in amerikanischen Firmen gerne verwendeten Begriff *Evangelizing* darf man nicht zu dogmatisch übersetzen. In gewissem Sinne kann man diese Aufgabe aber mit einer Mission vergleichen.

In den frühen siebziger Jahren trafen sich die Forscher am Xerox PARC wöchentlich zu den so genannten Bean-Bag-Konferenzen. Auf Sitzsäcken verteilt, lauschten sie den Ausführungen von Fachkollegen und ersannen die technischen und interaktiven Konzepte des Personal-Computings für die folgenden Dekaden. Heute veranstalten die größeren Firmen interne Vortragsreihen. Diese Seminare bieten den Mitarbeitern in regelmäßigen Abständen Einblick in Fragestellungen des Interface- und Software-Designs. Die Vortragenden können dabei aus der Firma selbst kommen, was dem Zusammenhalt und dem besseren Verständnis für Herangehensweisen in anderen Abteilungen dient. Durch eingeladene Vorträge werden dem Team neue Impulse gegeben. Sie können zusätzlich den schönen Effekt haben, dass die Position des Usability-Experten gestärkt wird. Bekanntlich gilt der Prophet im eigenen Lande nicht sehr viel. Erfahrene Usability-Berater und Größen der HCI-Disziplin, wie sie sich zum Beispiel in der Nielsen Norman Group zusammengefunden haben, wecken durch ihre Vorträge das Interesse der Kollegen für die eigene Arbeit.

Aber auch ohne großen finanziellen Einsatz lassen sich in kleinem Rahmen solche Veranstaltungen abhalten. Es bieten sich dazu Vorträge an, die auf Video oder DVD vorliegen. So hielt Tom DeMarco im Juni 2001 in Bonn einen Vortrag mit dem Titel „Requirements Engineering Then and Now“. Reich an Anekdoten präsentierte er eine neue Auffassung vom Qualitätsbegriff bei Software. Fehlerfreiheit sei ein einfaches, weil klar definiertes Ziel. Was aber nützt ein fehlerfreies Programm, das niemand benutzt, weil es niemand braucht? Gute Software zeichnet sich hingegen dadurch aus, dass sie dem Benutzer Möglichkeiten erschließt, die vorher undenkbar waren (DeMarco, 2001, S. 113). DeMarcos Vortrag liegt auf DVD vor (Broy/Denert, 2002).

Als eine weitere Anregung mag eine Veranstaltung zum Thema Video-Prototyping und Interface-Design in Hollywood-Filmen dienen. Anhand von Video-Ausschnitten zeigt und diskutiert man das User-Interface der Zukunft – oder zumindest das Bild, das die Produzenten in Hollywood basierend auf den aktuellen Arbeiten kreativer Forschungsinstitute entwerfen. Geradezu frappierend sind die Ähnlichkeiten zwischen „Minority Report“ (USA, 2002) und dem Starfire Projekt von Sun Microsystems aus dem Jahre 1996.



Tognazzini beschreibt das Projekt in „Tog on Software Design“ (Tognazzini, 1996). Statt eines heute üblichen Monitors, der nur beschränkt Platz für Dokumentfenster bietet, interagiert der Benutzer mit einem System, das einen schreibtischgroßen Computerbildschirm besitzt. Aktionen mit der Maus werden durch Gesten ersetzt. Das heißt, durch Zeigen auf ein Icon kann das zugehörige Dokument geöffnet werden. Fenster werden durch eine seitliche Handbewegung verschoben. Das Starfire Projekt umfasst weit mehr als diese Beispiele andeuten können. Es eignet sich daher bestens für eine inspirierende Diskussion über Entwicklungsmöglichkeiten im Bereich der Mensch-Maschine-Kommunikation.

Die zweite Etappe der Missionsarbeit ist das Informieren über die Methodik der Software-Ergonomie und über deren Möglichkeiten. Es besteht gemeinhin eine Unwissenheit über den Ansatz des Usability-Engineerings. Kein Entwickler ist bereit, die Vorschläge eines Usability-Experten anzunehmen, wenn er nicht von dessen Professionalität und der damit einhergehenden Verbesserung des Produktes überzeugt ist. Für ihn bedeuten sie in erster Linie zusätzlichen Aufwand. Ohne das Wissen um die Arbeitsweise von Interface-Designern schickt ein Entwickler den Designer mit seinen Papier-Mockups kalt lächelnd wieder nach Hause. Im Rahmen der oben angesprochenen Veranstaltungen bietet es sich an, auch den Kernbereich der Usability-Arbeit zu diskutieren. Das Intranet und direkte Gespräche sind weiterhin wichtige Aspekte des Informierens. Jedoch stellen sie nicht in dem Maße eine Öffentlichkeit her, wie dies mit Vorträgen möglich ist.

Wenn durch die erste inspirierende Etappe die Kollegen neugierig geworden sind, was Usability bewirken kann und wenn sie in der zweiten Etappe konkret über die Herangehensweise informiert worden sind, kann die Rolle des Usability-Experten und seine Bedeutung für die Produktentwicklung als etabliert gelten. Die Usability-Fachleute werden dann zu einem anerkannten und kompetenten Team, das im Entwicklungsprozess gerne konsultiert wird.

Die Bedeutung, die Usability-Experten im Entstehungsprozess von Software und Websites haben, hat in den letzten Jahren stark zugenommen. Daher ist noch nicht allen Beteiligten klar, wie die Zusammenarbeit im Alltag aussieht. Die potenziellen Zielkonflikte und Überschneidungen mit der Entwicklungsabteilung und dem Marketing sind in diesem Kapitel vorgestellt worden. Wenn man sie jedoch als sinnvolle Ergänzung wahrnimmt, wenn man sieht, dass die Mitarbeit der Usability-Experten eine Bereicherung darstellt, die sowohl den Prozess besser planbar als auch das Produkt besser bedienbar macht, wird Usability zu einem zentralen Qualitätsmerkmal.

### Quellen und Weiterlesen

*Zu allen im Folgenden mit „online“ gekennzeichneten Quellen stehen auf der buchbegleitenden Website die entsprechenden Links zur Verfügung.*

*Beyer, H., & Holzblatt, K.: Contextual Design: A Customer-Centered Approach to Systems Designs. Morgan Kaufman, 1998.*

*Brooks, Jr., F. P.: The Mythical Man-Month, Anniversary Edition. Addison-Wesley, 1995.*

*Broy, M., Denert, E. (Hrsg.): Software Pioneers. Springer, 2002.*

*Cooper, A.: The Inmates Are Running the Asylum. Macmillan Computer Publishing, 1999.*

*DeMarco, T.: Spielräume – Projektmanagement jenseits von Burnout, Stress und Effizienzwahn. Hanser, 2001.*

*Eberleh, E., Oberquelle, H., & Oppermann, R.: Einführung in die Software-Ergonomie, 2. Auflage. De Gruyter, 1994.*

*Gorny, P., Oberquelle, H.: HCI in Germany. SIGCHI Bulletin 31(2), S. 14 – 16, 1999. (online)*

*Hurst, M.: Interview: Maryam Mohit, Amazon.com. Good Experience Newsletter, 21. Nov. 2002. (online)*

*Jeffries, R.: Interaction Design: It's Infinite Variety. Interactions 7(2) März, S. 28 – 34, 2000.*

*Johnson, J. A.: GUI Bloopers – Don'ts and Do's for Software Developers and Web Designers. Morgan Kaufman, 2000.*

*Norman, D. A.: The Invisible Computer. The MIT Press, 1998.*

*Tognazzini, B.: Tog on Software Design. Addison-Wesley, 1996.*

*Winograd, T. (Hrsg.): Bringing Design to Software. Addison-Wesley, 1996.*

Außerdem die Zeitschrift *Interactions* der ACM. Beginnend mit dem Jahr 2000 enthält jeweils die März-Ausgabe eingehende Praxisberichte von HCI-Experten.