Paul Chlebek

Praxis der User Interface-Entwicklung

Paul Chlebek

Praxis der User Interface-Entwicklung

Informationsstrukturen, Designpatterns, Vorgehensmuster

Mit 126 Abbildungen

PRAXIS

VIEWEG+
TEUBNER

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über
<http://dnb.d-nb.de> abrufbar.

Höchste inhaltliche und technische Qualität unserer Produkte ist unser Ziel. Bei der Produktion und Auslieferung unserer Bücher wollen wir die Umwelt schonen: Dieses Buch ist auf säurefreiem und chlorfrei gebleichtem Papier gedruckt. Die Einschweißfolie besteht aus Polyäthylen und damit aus organischen Grundstoffen, die weder bei der Herstellung noch bei der Verbrennung Schadstoffe freisetzen.

1. Auflage 2011

Lektorat: Christel Roß | Walburga Himmel

Vieweg+Teubner Verlag ist eine Marke von Springer Fachmedien.
Springer Fachmedien ist Teil der Fachverlagsgruppe Springer Science+Business Media.
www.viewegteubner.de

Umschlaggestaltung: KünkelLopka Medienentwicklung, Heidelberg
Druck und buchbinderische Verarbeitung: STRAUSS GMBH, Mörlenbach
Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier
Printed in Germany

ISBN 978-3-8348-0728-1

Inhaltsverzeichnis

Kapitel 1: Funktion braucht Form

1.1 User Interface hin

In der modernen westlichen Zivilisation sind wir von Gebrauchsdingen umgeben. Wir haben Geräte und Maschinen für fast jeden Anwendungszweck gebaut. Wir ersinnen laufend neue und noch schlauere Geräte.

Warum? Weil wir ***das*** können. *Das* ist der Entwicklungsstand unserer Zivilisation. Geräte zu bauen und zu verwenden, ist unser üblicher Weg, unsere Umwelt zu technisieren und damit zu kontrollieren. Wir sind nun mal keine Delphine und kommen ohne Werkzeug nicht klar. Von den meisten Geräten, die wir uns anschaffen und gegebenenfalls verwenden, sind wir *nicht* existenzabhängig, dennoch dreht sich unser Denken und Streben um sie, weil sie – vermeintlich oder tatsächlich – unsere Lebensqualität verbessern.

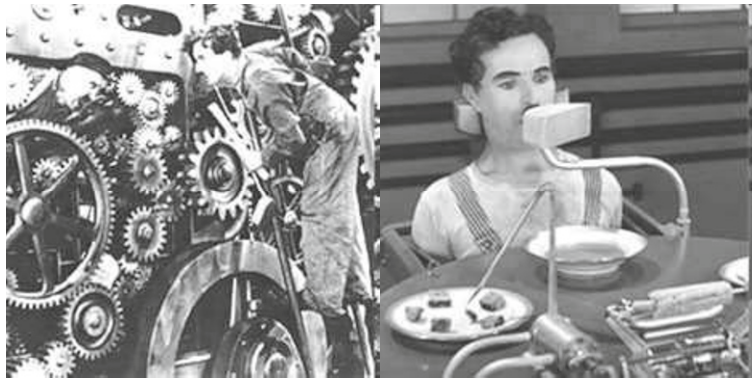


Bild 1: Maschinen sind unser Ding

Mit freundlicher Genehmigung von Roy Export S.A.S.

Das Buch, das Sie in den Händen halten, ist nicht zivilisations- oder technikkritisch. Es ist nicht einmal gegen ubiquitäre Verwendung von Geräten. Wie viele andere, trägt auch der Autor dieses Textes Smartphone und Laptop umher.

Dieses Buch ist aber über und für **Verwendungsqualität**, und damit tendenziell anwenderorientierter als ein technisches Handbuch. Es behandelt die technischen Möglichkeiten, Techniken und Methoden mit Fokus darauf, wie man Werkzeuge mit einem hohen Gebrauchswert baut.

P. Chlebek, *Praxis der User Interface-Entwicklung,* DOI 10.1007/978-3-8348-8219-6_1,

Neben den „klassischen“, mechanischen Werkzeugen zur Bearbeitung von Marmor, Stein und Eisen (machen wir von Faustkeil bis Akkubohrer schon seit über 20.000 Jahren) besitzen wir seit Mitte des zwanzigsten Jahrhunderts immer mehr und immer leistungsfähigere Informationsverarbeitungsgeräte: Mainframes, Minis, Workstations, Notebooks, Handhelds, Palmtops, Smartphones. Kurzum: Computer.

Computer unterscheiden sich in einem wesentlichen Punkt von herkömmlichen Geräten und Maschinen: Sie haben keinen unmittelbaren Einsatzzweck, sondern sind Ausführungsumgebung für „virtuelle“ (also nicht räumlich-materielle) Geräte.

Software ist ein virtuelles Gerät auf der Ausführungsplattform des Computers

Die virtuellen Geräte, die in Computern Informationen verarbeiten nennt man **Software**. Software macht aus der abstrakten Ablaufplattform des Computers eine konkret verwendbare, Informationen verarbeitende Maschine. Gegebenenfalls empfängt Software auch Daten von Sensoren und steuert mechanische Komponenten.

Die allgemeine Definition beschreibt Software als Gegenstück zur **Hardware**, wobei Software jede Art von digitalen Daten umfasst, die auf einer Hardware gespeichert sein können, von Firmware und Betriebssystem bis zu allen möglichen Dateien. Dieser Definition nach gehören zur Software sowohl Programme als auch z.B. Musikdaten auf einer CD, wobei diese selbst eine Hardware ist, auf dem die Software gespeichert wurde. Die CD mitsamt den Musikdateien ist selbst für eine weitere Hardware, welche ein softwaregesteuertes Gerät ist - den CD-Spieler - bestimmt. Software und Hardware ergänzen einander: Die Ablaufumgebung (Hardware + Betriebssystem + Gerätetreiber) gibt den Rahmen vor, die Applikation (Software) füllt den Rahmen aus.

Bild 2: Software ist ein virtuelles Gerät

In Kenntnis des Unterschiedes zwischen Computern nebst Software und mechanischen Geräten können wir beides - zumindest aus Sicht der Anwender - wieder in einen Sack stopfen.

Software auf einem Computer ist genauso ein **Gebrauchsding** wie eine Kuckucksuhr oder ein Dosenöffner, und ebenso ein Produkt der homosapiensischen Ingenieurkunst. Nur eben mit anderen Baustoffen und anderen Beschränkungen: Bei mechanischen Gebrauchsdingen sind es Werkstoffe und Gesetze der Mechanik; bei Software Algorithmen, Daten und Möglichkeiten der Ablaufplattform.

Im Übrigen wird Software nebst dem dazugehörigen Computer schon seit Jahrzehnten erfolgreich mit mechanischen Maschinen kombiniert. Eingebettete Systeme, d.h. Computer mir Software, die Maschinen steuert, findet man in Flug- und Fahrzeugen, in CNC Dreh- und Fräsmaschinen, in Fertigungsrobotern und sogar in Rasenmähern.

Das Verwenden eines Gerätes bzw. einer Software ist für sich genommen kein eigenes Ziel: Niemand (abgesehen von Testern) wird ein elektronisches Formular ausfüllen, wenn dies nicht dazu dient, eine Transaktion durchzuführen, z.B. einen Artikel zu bestellen, ein Ticket zu buchen, einen Vorgang anzustoßen, eine gewünschte Information abzurufen, oder wenigstens ein Spiel zu spielen.

Bild 3: Gerät ist Gerät, ob aus Software oder aus Hardware

Es zählt der Gebrauchswert

Man lässt sich auf das Gerät ein, weil es einen dem gewünschten Resultat näher bringen soll, nicht weil es da ist. Dementsprechend wird sich der normale Anwender nur insoweit mit der Konstruktion eines Werkzeugs auseinandersetzen, als dass es ihn in seiner Arbeit (oder bei seinem Hobby) weiterbringt. Das Gerät steht also nicht im Zentrum des Interesses, sondern das vom Benutzer angestrebte Ergebnis. „Kein Kunde kauft jemals ein Erzeugnis. Er kauft immer nur das, was das Erzeugnis für ihn leistet.“ (Peter F. Drucker)

Gebrauchsdinge (Mixer, Toaster, Mailclients, Browser, Armbanduhren, Zeichenprogramme, Staubsauger, Online-Shops, Handys, Spielkonsolen, Terminplaner und sonstiges Klimbim) haben (ungeachtet der Frage nach Sinn und Ausführung dieser Schöpfungen) Konstruktions- und Funktionsprinzipien, Verwendungszwecke und Verwendungsweisen.

Jedes Ding hat Funktions- und Bedienteile

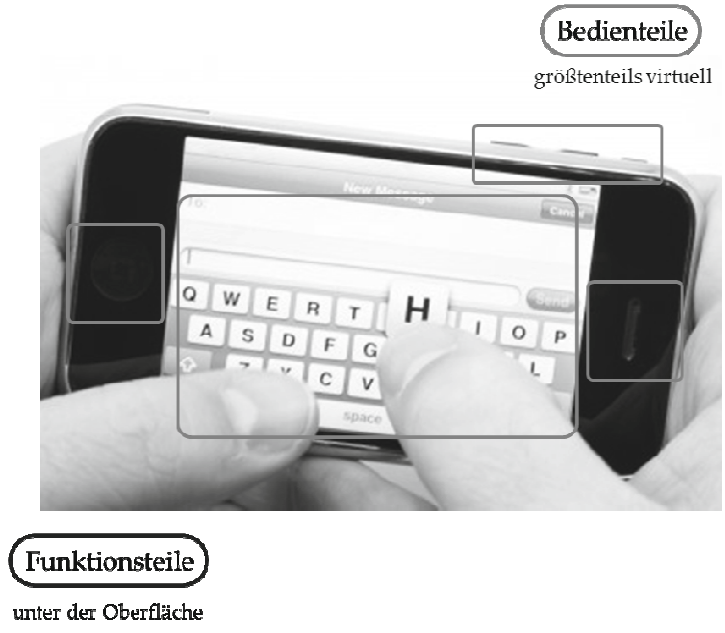
Jedes Gebrauchsding, ob ein mechanisches Gerät oder Software, besteht aus **Funktions- und Bedienteilen**. Die Bedienteile nennt man (bei Software) oft **User Interface** oder **Benutzeroberfläche** oder **Mensch-Maschine-Schnittstelle**. [WP05ui]. Die Funktionsteile nennt man Funktionsteile, weil sie funktionieren, d.h. sie bewerkstelligen das, wofür das Gerät an und für sich dient, z.B. Gemüse klein schneiden, Brot toasten oder Mails senden und empfangen.

Virtuelle und physische Bedienteile

Softwareprogramme haben virtuelle Bedienteile. Diese werden über die physischen Bedienelemente der Ablaufplattform angesteuert. Zum Beispiel sind an einem Personal Computer in der Regel eine Maus und eine Tastatur, einen oder mehrere Bildschirme, Lautsprecher, usw. angeschlossen. Das sind die **physischen Bedienelemente** dieser Ablaufplattform.

Ein Smartphone wie das iPhone hat einige Hardbuttons, ein Multi-Touchscreen sowie diverse Sensoren. Das sind die physischen Bedienteile dieses Geräts. In der Hauptsache sind seine Bedienteile (die User Interfaces der Apps) aber virtuell, und die Funktionsteile unter den Oberflächen der Apps verborgen. Die Funktionen der auf dem Smartphone ablaufenden Applikationen sind nur mittelbar über ihre Benutzeroberflächen erlebbar.

Bild 4: Ein Smartphone hat viele virtuelle Bedienteile



Bei mechanischen Geräten und auch bei Softwareanwendungen sind Funktions- und Bedienteile (oft untrennbar) zu einem Ganzen verflochten. Dennoch lassen sich Funktion und Bedienung als eng zusammenhängende - „symbiotische“ - jedoch unterschiedliche Anteile von Werkzeugen betrachten und bewerten.

Funktions- und Bedienteile sind oft verflochten.

Zum Beispiel besteht ein Spaten aus den Funktionsteilen „Blatt“ und „Stange“. Das Blatt ermöglicht den Aushub, die Stange sorgt für die Hebelwirkung. Die Bedienteile „Fußstütze“ und „Handgriff“ sind für das Graben nicht funktionell notwendig, doch ohne sie würde es so nicht gut gehen. Mit dem Spatenblatt allein kann man zwar vom Prinzip her graben, aber praktisch gesehen kommt man ohne Spatenstiel beim Umgraben eines Beets kaum voran.

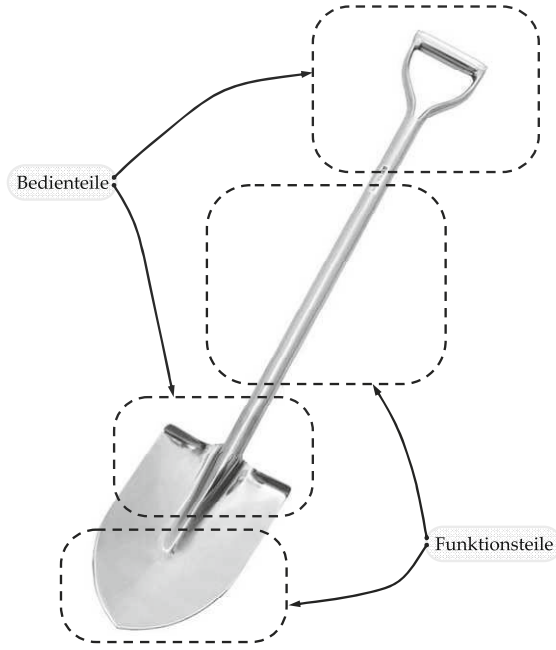


Bild 5: Funktions- und Bedienteile eines einfachen Werkzeugs

User Interfaces ermöglichen also Menschen das Verwenden von Werkzeugen, Geräten, Maschinen und Softwareanwendungen.

Umgekehrt ergibt sich daraus: Funktionen **brauchen** User Interfaces, um für Menschen nutzbar zu sein. Eine Fackel ermöglicht ein tragbares Feuer. Das Feuer braucht keinen Fackelstiel, es würde auch ohne brennen. Der Nutzer braucht den Stiel, sonst könnte er nicht lange fackeln, sondern würde sich die Hand verbrennen. Ein Besen braucht einen Besenstiel. Eine Lenkung muss eine Pine, ein Lenkrad oder etwas in dieser Art bieten.

Bei einem mechanischen Werkzeug finden die Interaktionen im Bedienteil nach den Gesetzen der Mechanik statt. Die Newtonschen Gesetze von Trägheit, Aktion und Reaktion sind

beim Verwenden von Maschinen intuitiv wahrnehmbar. Beim Betätigen der Gangschaltung in einem Fahrzeug spürt man das eigene Tun und dessen Wirkung in der Maschine unmittelbar mit mehreren Sinnen.

Auch ein Computerprogramm braucht irgendetwas, womit es vom Menschen nach seinem Willen gelenkt werden kann, und etwas, woran der Mensch sehen (oder sonst irgendwie wahrnehmen) kann, was aus seinem Wollen und Lenken in dem von ihm benutzten Programm wird.

Das Lenken nennt man „**Input**“, die Reaktionen des Programms heißen „**Output**“. Beides ist erhältlich in den Geschmacksrichtungen „interaktiv“ (ganz „in“) und „batch“ (sehr „out“, außer bei Entwicklern). Beim Verwenden einer Software macht der Anwender „Input“, die Applikation revanchiert sich mit „Output“. Die Abfolge dieser, optimalerweise im Zusammenhang zueinander stehenden, beiderseitigen Aktivitäten nennt man (sofern sie fein abgestuft sind und augenblicklich aufeinander folgen) „Interaktion“.

Eine der ersten Softwareanwendungen mit interaktivem Input und Output, die ich im Auftrag eines Dritten schrieb, war ein Programm zum Bedrucken von Überweisungsformularen. Der Auftraggeber (welcher zugleich der Anwender des Programms war) wollte die damals üblichen Endlos-Vordrucke nicht mehr mit der Schreibmaschine ausfüllen, weil er sich dabei oft vertippte, sondern vor dem Drucken auf dem Bildschirm sehen.

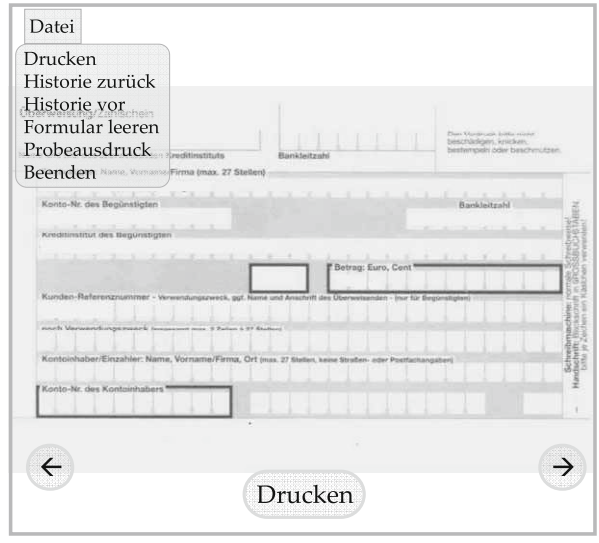


Bild 6: Ein Überweisungs-Druckprogramm

Später kam dazu der Wunsch, dass man die bereits erfassten Überweisungen abrufen, ändern und neu drucken können soll.

Das Programm sollte auf einem Atari ST (mit einem „hochauflösenden“ Schwarz-Weiß Bildschirm) unter GEM (Graphical Environment Manager) von Digital Research laufen.

Das User Interface dieser Anwendung sah in etwa wie oben abgebildet aus (mein Atari ST1024 weilt genauso wie der Commodore C16, der Schneider CPC, Joyce oder Amiga seit Langem in den ewigen Jagdgründen):

Das User Interface zeigte ein Überweisungsformular (das 1 zu 1 dem zu bedruckenden Vordruck entsprach), in das man an entsprechenden Stellen die zu druckenden Informationen eintragen konnte. Es verwendete also die Metapher des Papierformulars.

Unter dem Formular gab es einen Knopf mit der Aufschrift „Drucken“. Mit den Tasten ← und → konnte man in der Historie der gedruckten Überweisungen hin und her blättern. Dazu gab es zwei Bestätigungsabfragen: Beim Verlassen eines Formulars, auf dem Daten erfasst waren, das aber noch nicht gedruckt wurde, und vor dem Drucken. Die Anwendung hatte ein Menü, in dem sämtliche Anwenderaktionen (Drucken, Historie zurück, Historie vor, Formular leeren, Probeausdruck (zur Formularausrichtung) und Programm beenden) aufgelistet und zur Auswahl angeboten wurden.

Die Funktionsteile dieser Anwendung sind:

- Ausfüllen des Überweisungsformulars
- Ansicht des ausgefüllten Formulars
- Bedrucken des Formulars
- und Ansehen der Historie gedruckter Formulare.

Die Bedienteile dieser Anwendung sind:

- Die in das Formularbild eingebetteten Erfassungsfelder,
- die Schaltfläche „Drucken“,
- die Schaltflächen ← und →
- sowie das Drop-Down Menü mit der Liste aller vom Anwender auslösbaren Aktionen.

Ein solches virtuelles Überweisungsbedruckgerät ist überschaubar und lässt sich leicht nach Funktions- und Bedienteilen analysieren. Der Verwendungszweck und das Ergebnis der Anwendung sind ebenfalls leicht zu überblicken. Durch die direkte Anlehnung an das bekannte Formular wird dem Anwender sofort intuitiv klar, was man damit tun kann. So wie ein Korkenzieher unmissverständlich zum Flaschenöffnen taugt, ist dieses Programm zum Erstellen von Überweisungen gut. Inzwischen werden Überweisungen allerdings nicht mehr gedruckt, sondern gleich online überwiesen.

Die gelungene Überweisungsmetapher war Mitte der Neunziger Jahre einer der Erfolgsfaktoren von Intuits Quicken gegenüber Microsofts Money. Während Money das Erfassen der Überwei-

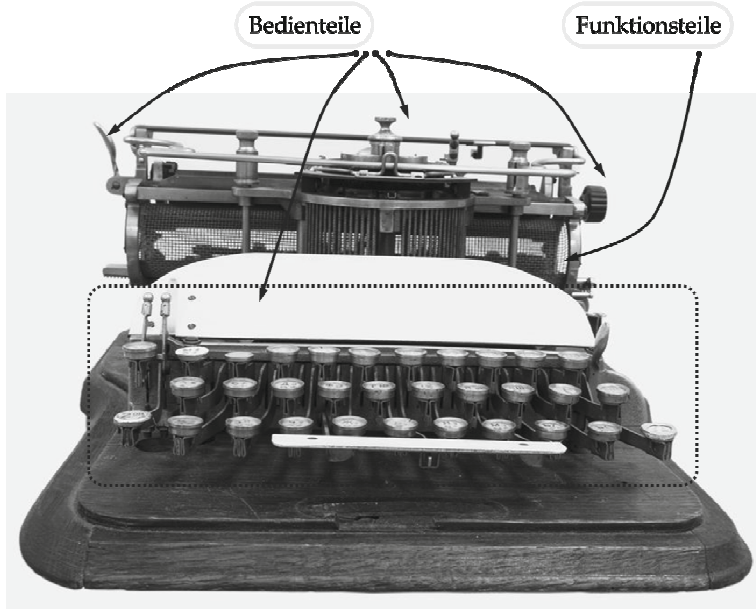
sungsdaten in einem Dialogfenster mit Standard-Eingabefeldern forderte, wartete Quicken mit dem originalgetreuen Formular auf – und gewann bei den Anwendern.

Vom einfachen zum komplexen Gerät.

Eine Schreibmaschine besteht aus Gehäuse, Walzenwerk, Typenwerk (Funktionsteil), Einzugs- und Vorschubbedienung, und Typenauswahl bzw. Tastatur (User Interface).

Neben guter Walzen- und Typenmechanik ist für die Verwendbarkeit der Schreibmaschine entscheidend, dass die Typenauswahl leicht ist, und dass der Einzug und der Vorschub leicht und zuverlässig zu bedienen sind. Die Verwendbarkeit ist also ein Produkt aus der Effektivität (Wirkung) und Effizienz (Wirkungsgrad) von Funktions- und Bedienteilen.

Bild 7: Eine frühe Schreibmaschine



Bei gleicher Anzahl und Qualität von Funktionen hängt die Verwendbarkeit zunehmend davon ab, wie gut die Bedienteile angeordnet und aufeinander abgestimmt sind. Bei gleich guten Bedienteilen ist der Funktionsumfang ausschlaggebend. Bedienen und Funktionieren ergänzen einander auf der sachlichen Ebene; auf der Marktebene stehen sie im Wettbewerb zueinander.

Vom einfachen zum komplexen User Interface.

Eine elektrische Schreibmaschine hat ähnliche Funktions- und Bedienungsteile wie ihr mechanisches Pendant. Oft hat sie zusätzliche Funktionen, wie z.B. das vorab Anzeigen der aktuellen Eingabezeile und Zeichen löschen oder einfügen mit den dazugehörigen Bedienschritten: **Form follows Function**.

Ein Schreibprogramm auf dem Computer hat vom Prinzip her die Funktion einer elektrischen Schreibmaschine. Doch bietet es auch Bearbeitungsmöglichkeiten, die weit über die Möglichkeiten der Schreibmaschine hinausgehen. Entsprechend umfangreich sind die Bedienungselemente: Menüs, Symbolleisten, Funktionstasten, Assistenten, Modussteuerungen, Formatierungshilfen usw.: **Form follows Use Case**.

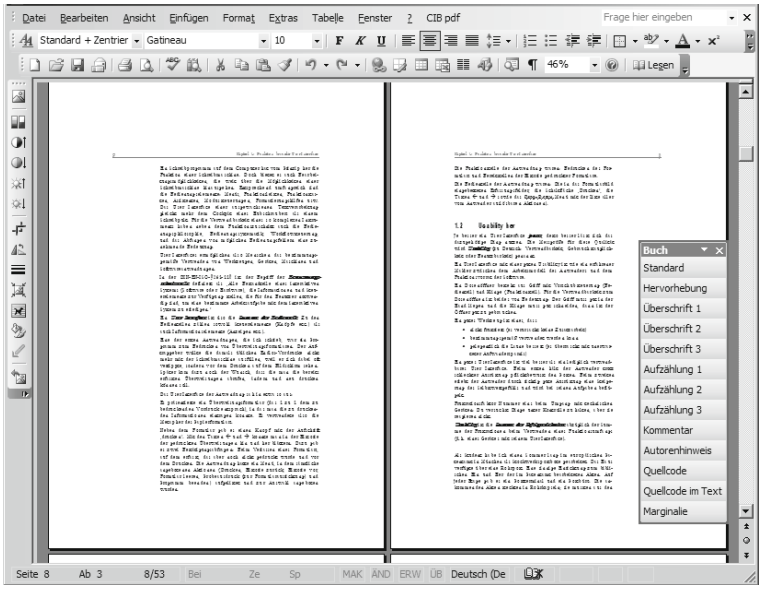


Bild 8: Eine Textverarbeitungssoftware

Das User Interface einer ausgewachsenen Textverarbeitung gleicht mehr dem Cockpit eines Düsenjets als einem Schreibgerät.

Bild 9: Cockpit eines Düsenjets

Schreib- wie Flugsituationen können komplex sein, dementsprechend vielfältig und spezialisiert sind die hierzu dienenden Werkzeuge.

Für die Verwendbarkeit von so komplexen Instrumenten wie Flugzeuge oder Office-Software haben – neben dem Funktionszuschnitt – auch die Bedienungsphilosophie, Bedienungssystematik, Workflowsteuerung, und das Abfangen von kritischen sowie „gebräuchlichen" (nahe liegenden) Bedienfehlern eine zunehmende Bedeutung: **Function follows Use Case**.

Bei einem softwaregestützten Werkzeug finden die Interaktionen in der Oberfläche nach den von dem dazugehörigen Anzeige- und Bedienkonzept aufgestellten Regeln statt. Falls ein solches Regelwerk im Projekt nicht zum Einsatz kommt, gelten die Regeln, die der Entwickler nach bestem Wissen, Gewissen und nach Tagesform implementiert.

Funktionsvielfalt steigert potentiell die Anzahl der beim Bedienen zu treffenden Entscheidungen.

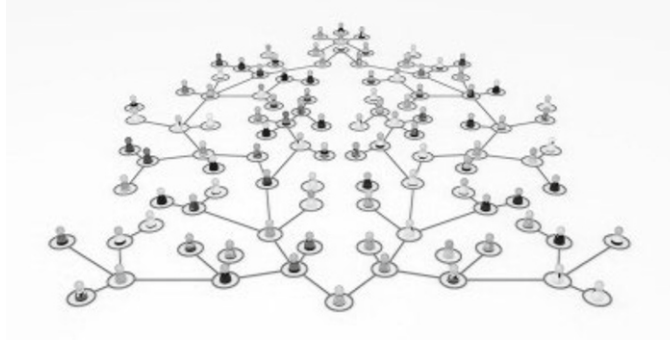


Bild 10: Viele Funktionen führen tendentiell zu vielen Bedienschritten

Generell lässt sich behaupten: Je mehr Funktionen und Verwendungsvarianten ein Softwarewerkzeug hat, desto mehr **Bedienteile** (z.B. Knöpfe) und/oder **Bedienschritte** (z.B. Menüauswahlstufen) hat es auch – *potentiell*. Die Anzahl der beim Bedienen vom Anwender (oder von der Applikation) zu treffenden Entscheidungen steigt – ebenfalls potentiell – mit der **Funktionsvielfalt** und mit der **Funktionsvarianz**.

Dabei führt Funktionsvielfalt praktisch nicht zwangsläufig zur erhöhten **Bedienkomplexität**, ebenso wie ein Werkzeug mit nur wenigen Funktionen nicht automatisch einfach zu bedienen ist. Die tatsächliche (bzw. vom Anwender wahrgenommene) Anzahl von Bedienteilen und Bedienschritten hängt nicht allein von der Funktionalität ab, sondern ebenso von der **Bedienmetapher**, die der Software zugrunde liegt.

Die Bedienmetapher umschreibt die vom Entwickler beabsichtigten Assoziationen des Anwenders beim Verwenden der Software. „Drag & Drop" und „Slider Control" sind Beispiele für solche Bedienmetaphern. Der Anwender assoziiert das Look und Feel

der Software mit Objekten aus der realen Welt. Er überträgt seine physischen Erfahrungen mit diesen Objekten auf seine Erwartungen bezüglich dessen, wie ihre Nachbildungen in der Software handzuhaben sind.

Eine enge Anlehnung der Handhabung und des Verhaltens von Oberflächenelementen an mechanische Gesetze und traditionelle Kulturtechniken kann sich positiv auf die intuitive Anwendbarkeit von Softwareprogrammen auswirken. Nachbildungen von realen Dingen und Vorgängen mittels virtueller Bedienteile sind insbesondere bei Spielen und Simulationen weit verbreitet.

Vor dem Aufkommen der GUIs (Graphical User Interface) waren beispielsweise Dame- und Schachspiele im Umlauf, bei denen die verschiedenen Figuren durch Buchstaben repräsentiert wurden.

Um Figuren zu bewegen, musste man die Start- und Zielposition über die Tastatur eingeben, z.B. „d2-d4“. Das Schachspielen im Textmodus war etwas, das man als Schachspieler in Kauf nahm, um eine Partie mit einem weit entfernten Gegner zu spielen.

```
  ---------------------------------
1 | R |   | B | K | Q | B | N | R |
  |---+---+---+---+---+---+---+---|
2 | P | P | P |   |   | P | P | P |
  |---+---+---+---+---+---+---+---|
3 |   |   |   |   |   |   |   |   |
  |---+---+---+---+---+---+---+---|
4 |   |   |   | P | N |   |   |   |
  |---+---+---+---+---+---+---+---|
5 |   |   |   |   |   |   |   |   |
  |---+---+---+---+---+---+---+---|
6 |   |   |   |   |   | *N|   |   |
  |---+---+---+---+---+---+---+---|
7 | *P| *P| *P| *P| *P|   | *P| *P|
  |---+---+---+---+---+---+---+---|
8 | *R| *N| *B| *K| *Q| *B|   | *R|
  ---------------------------------
    h   g   f   e   d   c   b   a
```

Bild 11: Schachspielen im Textmodus (telnet)

Um jedoch ein Gespür für die tatsächliche Spielsituation zu bekommen, bauten sich die Spieler nebenher ein richtiges Schachbrett mit echten Figuren auf und zogen die Partie darauf Zug um Zug nach.

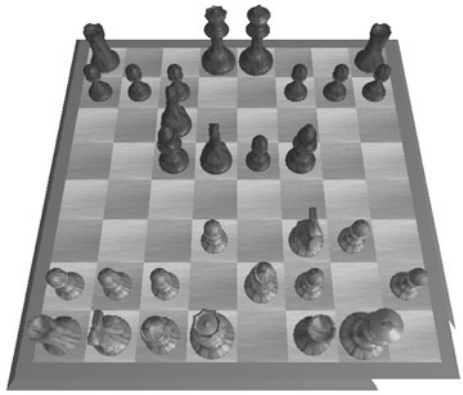
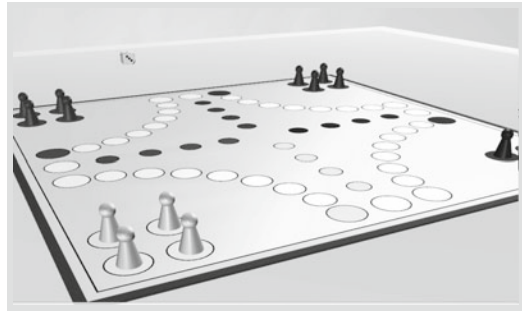


Bild 12: Schachspielen auf einem 3D-Brett (gnome)

Als graphische Schachspiele aufkamen, bei denen man auf einem 3D-Schachspiel die Figuren mit der Maus greifen und ziehen konnte, verschwand das Schachspielen im Textmodus fast vollständig wieder von der Bühne.

Virtuelle Nachbildungen realer Dinge lassen sich schneller verstehen und intuitiver bedienen, als abstrakte Modelle.

Bild 13: Virtuelle Nachbildungen realer Dinge (x-windows)



Im kommerziellen Umfeld überwiegen heute Applikationen, die

- einen Webbrowser als UI Framework verwenden,
- ein GUI (Graphical User Interface) haben,
- mit Maus und (ergänzend) mit Tastatur bedient werden,
- Töne, Sprache und Videos als Ausgabemedien verwenden,
- elektronische Formulare darstellen,
- und / oder grafische Modelle von Vorgängen bzw. durchzulaufenden Workflows bereitstellen,
- und /oder grafische Modelle von Bearbeitungsgegenständen bereitstellen,
- und /oder ergänzende Dienste (z.B. RSS Feeds) zur Verfügung stellen.

Bearbeitungsgegenstände und Vorgänge

Bearbeitungsgegenstände und **Vorgänge** sind bei User Interfaces von zentraler Bedeutung: Um das was bearbeitet wird, und darum, wie dies erfolgt, dreht sich alles Übrige.

Vorgänge kann man mit Produktionsprozessen oder Workflows vergleichen. Bearbeitungsgegenstände entsprechen den Waren, die in Produktionsprozessen aus Rohstoffen hergestellt werden.

Zum Beispiel sind Map24 oder Google Maps Webapplikation, mit denen man Karten ansehen, Routen berechnen und verschiedene Dinge auf der Landkarte suchen und finden kann. Sie ermöglichen die Navigation in einem grafischen Modell, haben aber auch Formularfelder zur Eingabe der Routenkriterien und geben die errechnete Route als Liste von Stationen aus.

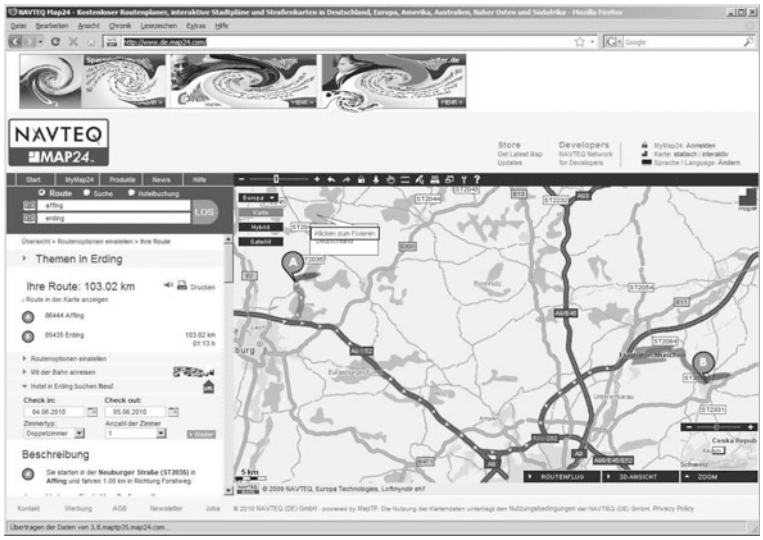


Bild 14: Map24 http://www.de.map24.com/ Mit freundlicher Genehmigung von NAVTEQ

Der in der Maps-Applikation (unter anderen) unterstützte Vorgang bzw. Anwendungsfall ist das Finden der passenden Route von A nach B. Dabei ist die Landkarte mit der eingezeichneten Route der Bearbeitungsgegenstand der Applikation.

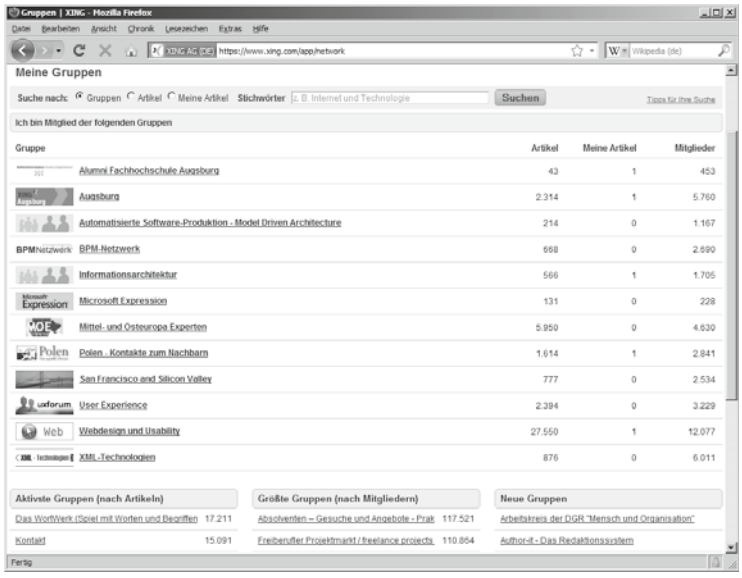


Bild 15: Xing Business Contacts Manager

https://xing.com Mit freundlicher Genehmigung von XING

Die Business Networking Anwendung Xing verwaltet eine Reihe von Bearbeitungsgegenständen wie Profile, Termine, Kontakte und Gruppen. Sie werden in Listen und in Einzelansichten angezeigt. Die von der Applikation unterstützten Aktivitäten drehen sich um Abrufen von Informationen über und das Kommunizieren mit anderen Netzwerkteilnehmern anhand vom Filtern, Sortieren, Ansehen und Pflegen dieser Objekte.

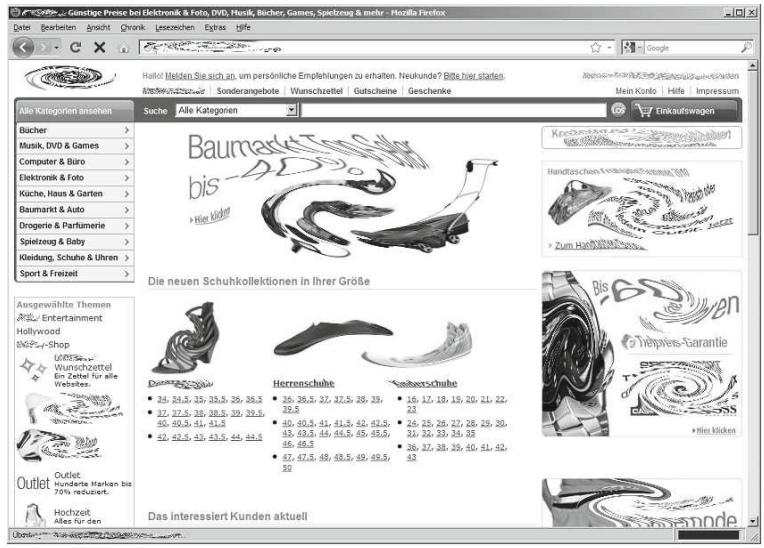


Bild 16: Leitseite eines Shopping Portals

Bei einem Shopping Portal geht es um den Einkaufsvorgang. Dieser (im User Interface im Hintergrund gehaltene) Vorgang ist in die Phasen des Aussuchens (Bearbeitungsobjekte = Trefferliste und einzelne Artikel) und des Kaufabschlusses (Bearbeitungsobjekte = Artikel im Einkaufswagen) aufgeteilt. Der Benutzer wird durch einen Workflow geführt: 1. Artikel ansehen 2. Auswahl in den Einkaufswagen hinzufügen 3. Ausgesuchte Artikel überprüfen 4. Zahlungsinformationen eingeben 5. Bestellung überprüfen und bestätigen.

Typische und weit verbreitete Vorgänge bzw. Anwendungsfälle in Softwareapplikationen sind:

- Exploratives Nachschlagen (typische Vertreter: Google, Google Maps, Wikipedia, Xing)
- Katalog-Shopping (z.B. Otto, Amazon, Ebay, Autoscout24)
- Kommunikation (Mail, Social Networking wie Xing, CRMs, Issue Tracker)
- Dokumenten- und Webseitenerstellung (Word, Powerpoint, Content Management und Publishing Systeme, Grafikprogramme)
- Konstruieren von Maschinen (CAD)
- Verwaltung und Unternehmenssteuerung (Sachbearbeitung, Finanzverwaltung, Kundenverwaltung, Excel, SAP, CRM)
- Spiele und Unterhaltung (Arcade, Strategie, Simulationen)
- Verwalten und Wiedergabe von Medien (Audio, Video, Webradio, Mediendateien und -streams)
- Softwareentwicklung (IDEs, Design Tools)

Obwohl alle diese Anwendungsfälle und Anwendungsformen in der Handhabung sehr unterschiedlich sind, haben sie gemeinsame (abstrakte) Aspekte der Verwendung, zum Beispiel:

- Suchen und Auswählen
- Auslösen von Aktionen
- Zuordnen und Verknüpfen
- Anordnen und Ändern von Daten

Definition des User Interface

In der DIN-EN-ISO-9241-110 ist der Begriff der **Benutzungsschnittstelle** definiert als „Alle Bestandteile eines interaktiven Systems (Software oder Hardware), die Informationen und Steuerelemente zur Verfügung stellen, die für den Benutzer notwendig sind, um eine bestimmte Arbeitsaufgabe mit dem interaktiven System zu erledigen."

Ein **User Interface** ist damit die **Summe der Bedienteile, deren Anordnung und Wechselwirkung**.

Zu den Bedienteilen zählen sowohl Steuerelemente (Knöpfe, Regler, Auswahllisten etc.) als auch Informationselemente (Anzeigen, Signale etc.). Bei Software sind damit alle (virtuellen) Eingabeelemente und alles gemeint, was die Software in Richtung Anwender ausgibt (also Schaltflächen, Eingabefelder, Anzeigen, Töne, haptische Signale etc.)

Aber auch Seitenlayout, Ablaufstruktur und Ablauflogik gehören zum User Interface einer Software. Das User Interface umfasst damit neben Eingabe, Ausgabe und Steuerung selbst auch die Anordnung, Ablaufreihenfolge, Verhalten und Situationsabhängigkeiten von Eingabe, Ausgabe und Steuerung.

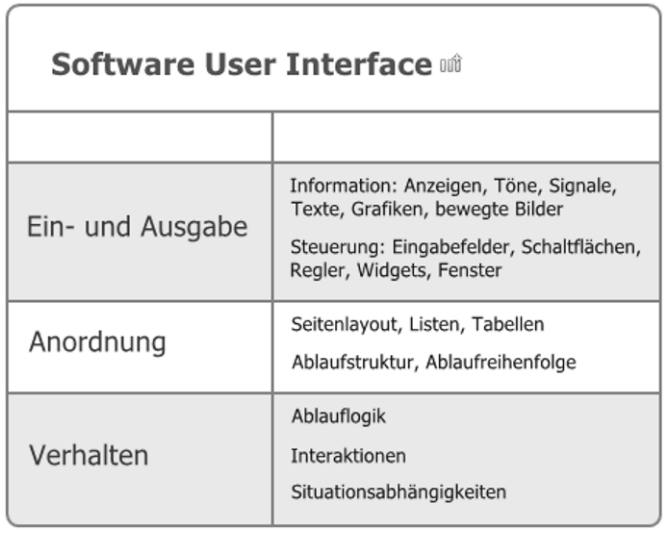


Bild 17: Umfang eines User Interface

Der Baustoff, aus dem Software User Interfaces gebaut sind, sind verschiedene Sorten von Informationsträgern. Primär sind das Träger für die Informationen, die zwischen Anwender und Applikation ausgetauscht werden.

Sekundär gehören dazu auch die Informationen, die dazu dienen, diesen Austausch in allen seinen Aspekten zu bewerkstelligen.

Strukturieren und Anordnen von Informationen

Hauptaspekt des Verwendens wie des Entwickelns von Software User Interfaces ist das geeignete Strukturieren und Anordnen von Informationen zur Darstellung auf dem Bildschirm und/oder für die Druckausgabe.

Nachgelagerter, jedoch notwendiger Aspekt ist das Strukturieren und Anordnen von Informationen zur Steuerung der Abfolge und der Wechselwirkung dieser Darstellungen in Abhängigkeit den Bedienaktionen des Anwenders.

Fazit: User Interfaces und Funktionen sind symbiotisch. Ohne Funktionalität ist ein User Interface sinnlos. Funktionalität ohne Bedienmöglichkeit auch.

Road Check „User Interface“:

RC01: Welche der unten genannten Dinge sind Bestandteile eines User Interface?

- ☐ *a). Eine API*
- ☐ *b). Anzeigen und Regler einer Maschine*
- ☐ *c). Darstellungs- und Steuerelemente einer Software*
- ☐ *d). Tastatur, Maus, Display, Drucker, Grafiktablett und andere am Computer angeschlossene Ein- und Ausgabegeräte*
- ☐ *e). Ein Treiber für ein am Computer angeschlossenes Gerät*
- ☐ *f). Das Ausgabeformat der Informationen (z.B. PDF)*

RC02: Welche der unten genannten Dinge sind Bestandteile eines User Interface?

- ☐ *a) Ablaufstruktur, Kontrollelement, Interaktion, Dialogmodalität*
- ☐ *b) Systemgerät, Softwaretreiber*
- ☐ *c) Ein- und Ausgabeelement, User Interface Logik*
- ☐ *d) Datenzugriffsschicht, Funktionsschnittstelle*
- ☐ *e) Grafiken, Texte, Töne, Animationen*
- ☐ *f) Eingabe, Verarbeitung, Ausgabe*
- ☐ *g) Ein- und Ausgabegeräte, Programmschnittstelle*
- ☐ *h) Eingabeempfänger, Interpretationsregeln, Ausgabemechanismen*

RC03: Identifizieren Sie (anhand einer Skizze) bei einem einfachen Gerät oder Werkzeug Ihrer Wahl dessen Bedien- und Funktionsteile. Notieren Sie, was die Funktionsteile tun und was die Bedienteile tun. Welche Bedienteile sind substantiell, welche fakultativ?

RC04: Was zählt bei einer Software zu den Bedienteilen eines User Interface? Was umfasst das User Interface einer Software neben den Bedienteilen?

1.2 Usability her

Je besser ein User Interface **zum User passt**, desto besser kann dieser das dazugehörige Gebrauchsding **nutzen**. Die Messgröße für diese Qualität wird **Usability** (zu Deutsch: Verwendbarkeit, Gebrauchstauglichkeit oder Benutzbarkeit) genannt.

Ein User Interface mit einer guten Usability ist wie ein erfahrener Vermittler zwischen dem Arbeitsmodell des Anwenders und dem Funktionsvorrat der Software. Ein Werkzeug mit einer guten Usability reagiert zuverlässig auf die Bedienschritte des Anwenders. Es tut zuverlässig, was es soll, frustriert nicht und spart Aufwand.

Usability ist die Summe der Erfolgserlebnisse abzüglich des Quadrats der Frustrationen beim Verwenden eines Funktionsumfangs (d.h. dem Bedienen eines Gerätes mittels seines User Interfaces).

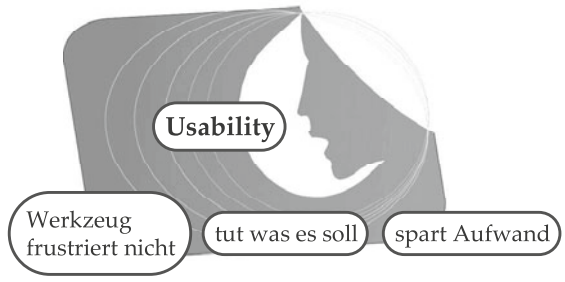


Bild 18: Faktoren der Usability

Usability des Werkzeugs entscheidet über den Erfolg des Anwenders

Erfolge und Misserfolge zählen nicht gleich. Ein erfolgreich abgeschlossener Arbeitsschritt motiviert zum Weitermachen. Bei einer Kette von Erfolgen ist man bereit, jeweils den nächsten Schritt zu machen.

Bei einem Misserfolg wiederholt man einmal oder zweimal die Arbeit. Dabei ist man, je nachdem, wie viel Aufwand man in die Wiederholung des bereits erledigt Geglaubten gesteckt hat, etwas frustriert oder aber richtig bedient. Bei mehreren Misserfolgen ist man einfach nur beleidigt, und will das Werkzeug, oder die Software nicht weiter verwenden.

Usability Kriterien

Ein gutes Werkzeug ist eines, das:

- bestimmungsgemäß verwendet werden kann
- keine Zusatzarbeit durch überflüssige Bedienschritte verursacht
- nicht zu Fehlern verleitet
- gelegentlich die Laune bessert (es überrascht mit unerwarteter Aufwandersparnis)