

User Interaction Design

*Andreas Monitzer
Cocoaheads Wien, 2010-03-11*

Gutes User Interface Design

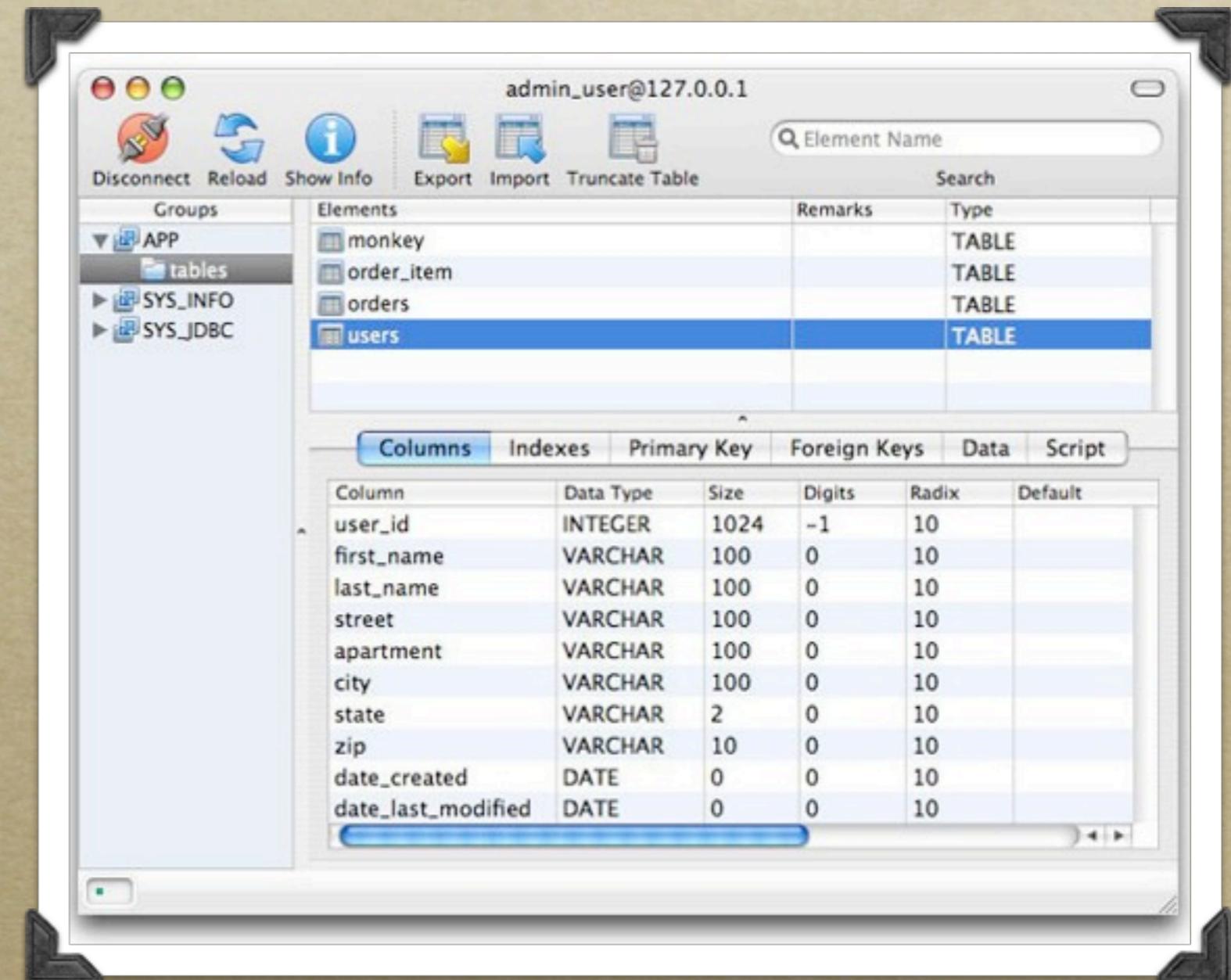


Schlechtes User Interaction Design



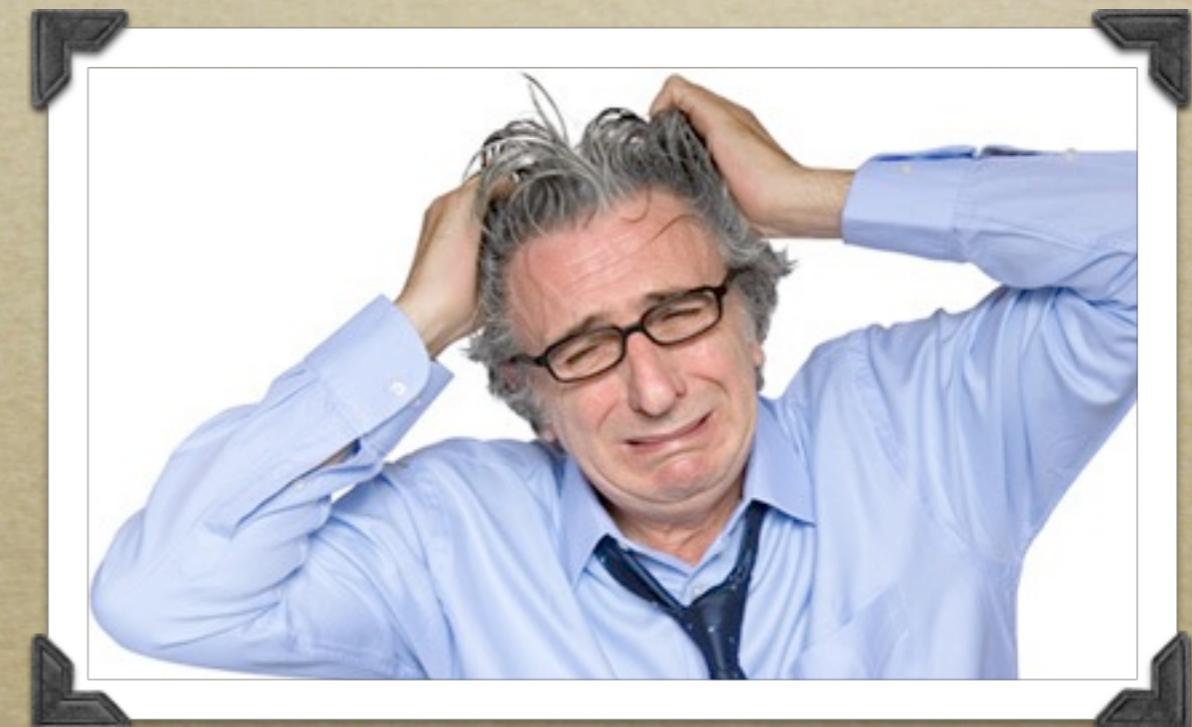
Traditionelle Vorgangsweise

- “*Lasst uns ein paar Controls nehmen und sie ins Fenster hauen*”
- *direktes Mapping der internen Datenstrukturen aufs UI*



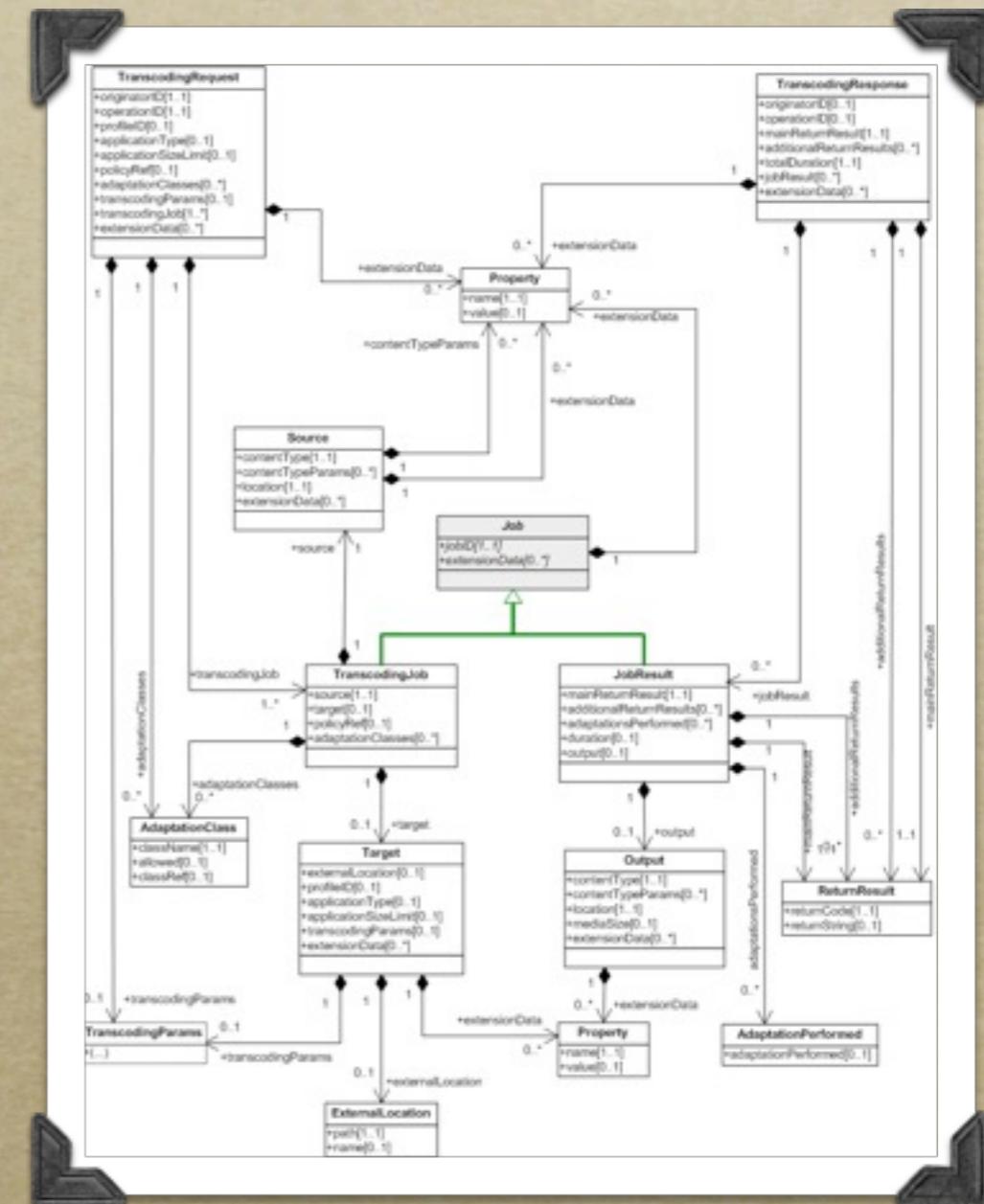
Für wen entwickeln wir?

- *der gradlinigste Ansatz: für einen selber*
- *bedeutet meistens, für den Programmierer*



Welche Art von Mensch sind Programmierer?

- *die meisten guten Programmierer haben mit 8-10 Jahren angefangen*
- *stark strukturierte Denkweise*



Welche Art von Mensch sind *Menschen*?

- *Großteil denkt nicht strukturiert*
- *will einfach eine Aufgabe erfüllen*
 - *wie: egal*
 - *warum es geht: egal*



Folgen der Diskrepanz

- *Technikfrustration*
 - “Der Computer versteht mich nicht”
 - “Daheim habe ich keinen Computer,
da ich dort entspannen will”
- *diese Leute sind das Hauptzielpublikum
von Macs/iPhones/iPads!*

“Dancing Bearware”

- *kann nicht gut tanzen*
- *Leute zahlen dafür*
- *toll, dass der Bär tanzt
(..irgendwie...)*

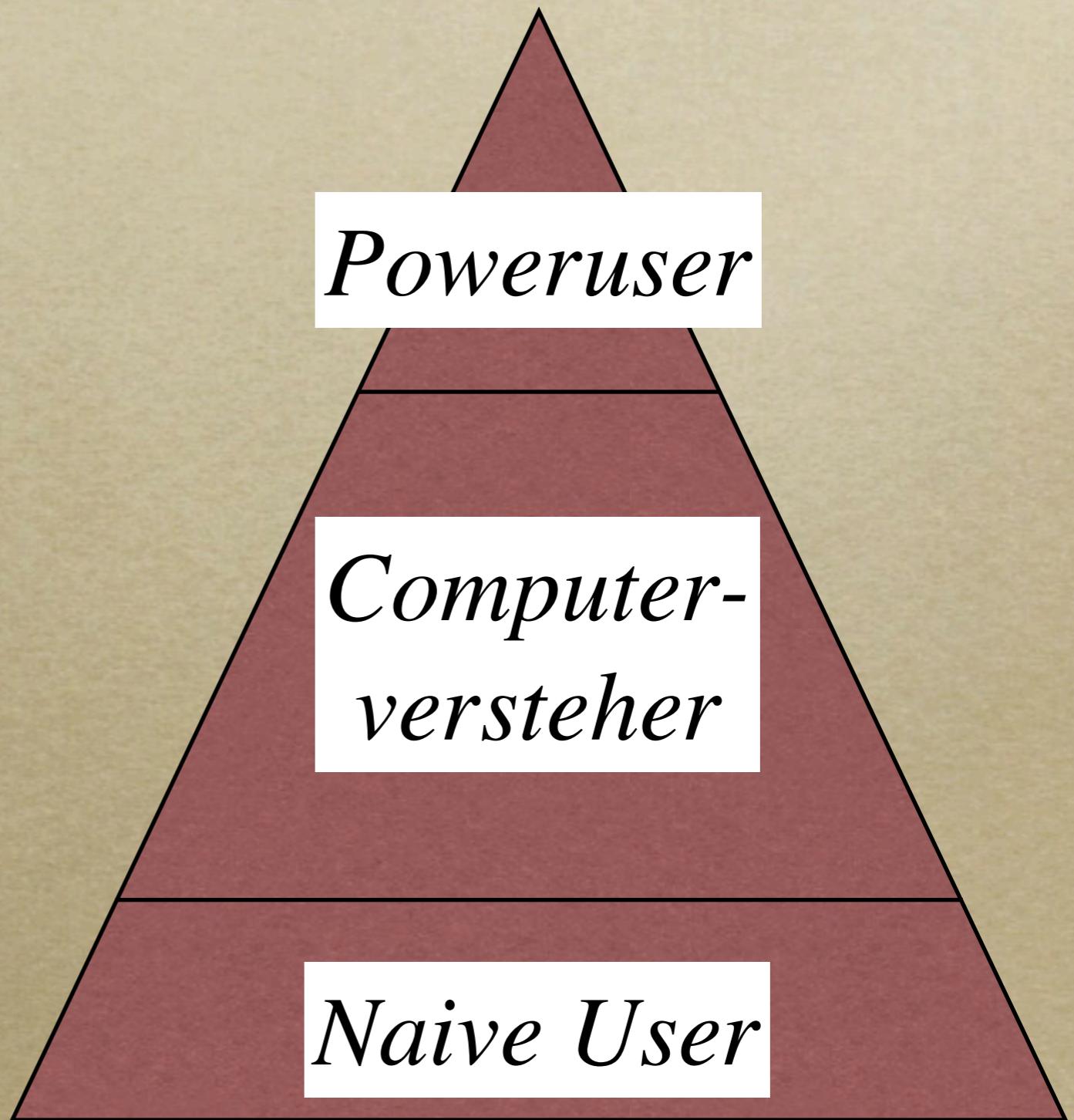


Reaktion der Menschen

- *Menschen sind es gewohnt, für Funktionalität Komfortabstriche zu machen.*
- *Viele Menschen sind schon so weit, dass sie es sogar erwarten!*

Benutzereinteilung

- *Poweruser:*
Programmierer-Mindset, “Apologist”
- *Computerversteher:*
Erfahren genug, um an die Schmerzen gewohnt zu sein
- *Naiver User:* *Großteil der Zielgruppe, “Normalsterbliche”*

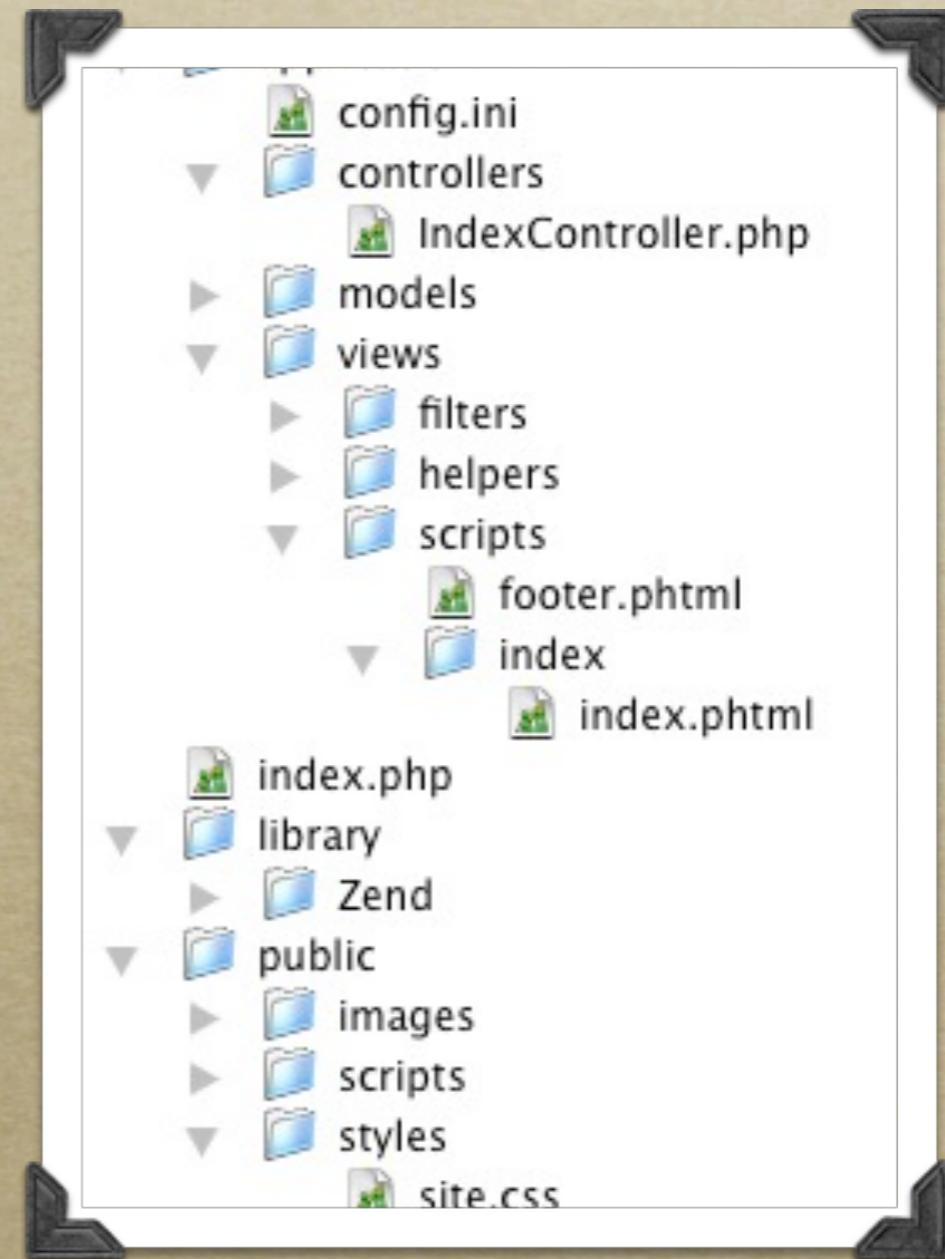


Umstellung der Planung

- *alt:*
 - *Interne Datenstrukturen → UI*
- *neu:*
 - *Aufgabe → Arbeitsablauf (Workflow)*
→ *UI → Interne Datenstrukturen*

Beispiel

- *eine Verzeichnisstruktur ist ein Baum*
- *daher die logische Darstellungsweise:*



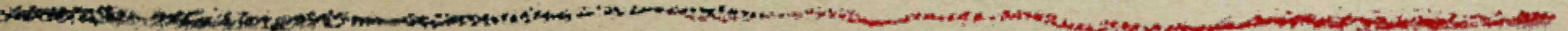


Was interessiert das den User?

Wann passiert Interaction Design?

- *in Wirklichkeit laufend in der Entwicklung*
 - *von jedem Programmierer*
 - *Daher müssen die Programmierer hier auch geschult sein!*

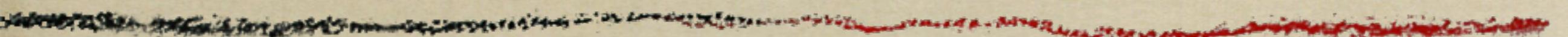
Schritte zum Erfolg



Definieren von Zielgruppen

- *Wer soll mein Programm verwenden?*
- *An wen muss ich mein Programm verkaufen?*

Problematik der Erfahrungsstufen

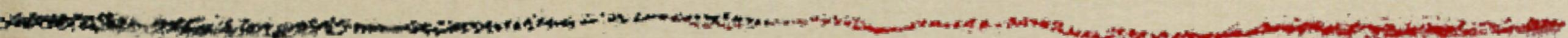


Anfänger

Mittlere

Experten

Problematik der Erfahrungsstufen



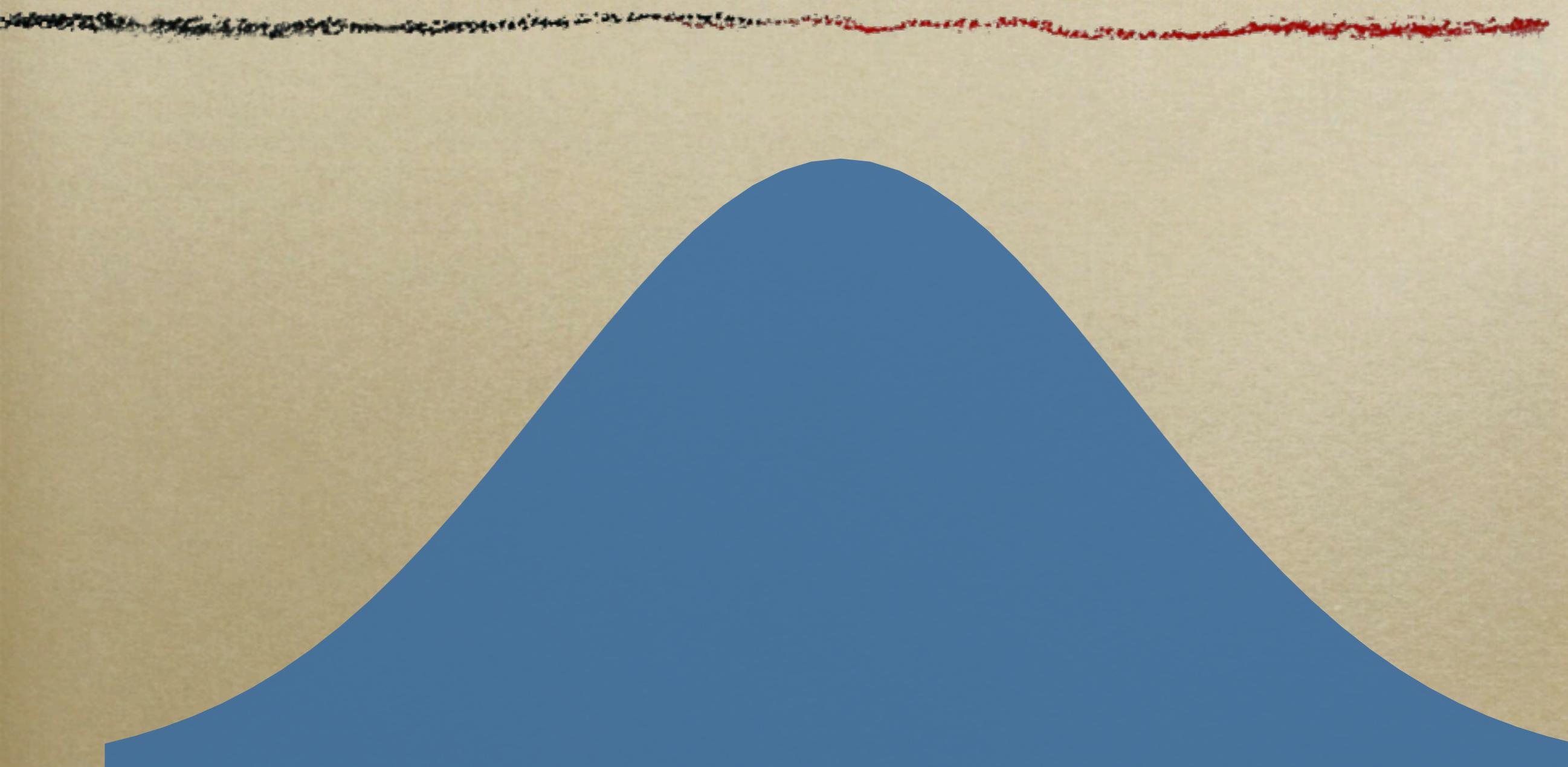
Anfänger



Mittlere

Experten

Problematik der Erfahrungsstufen

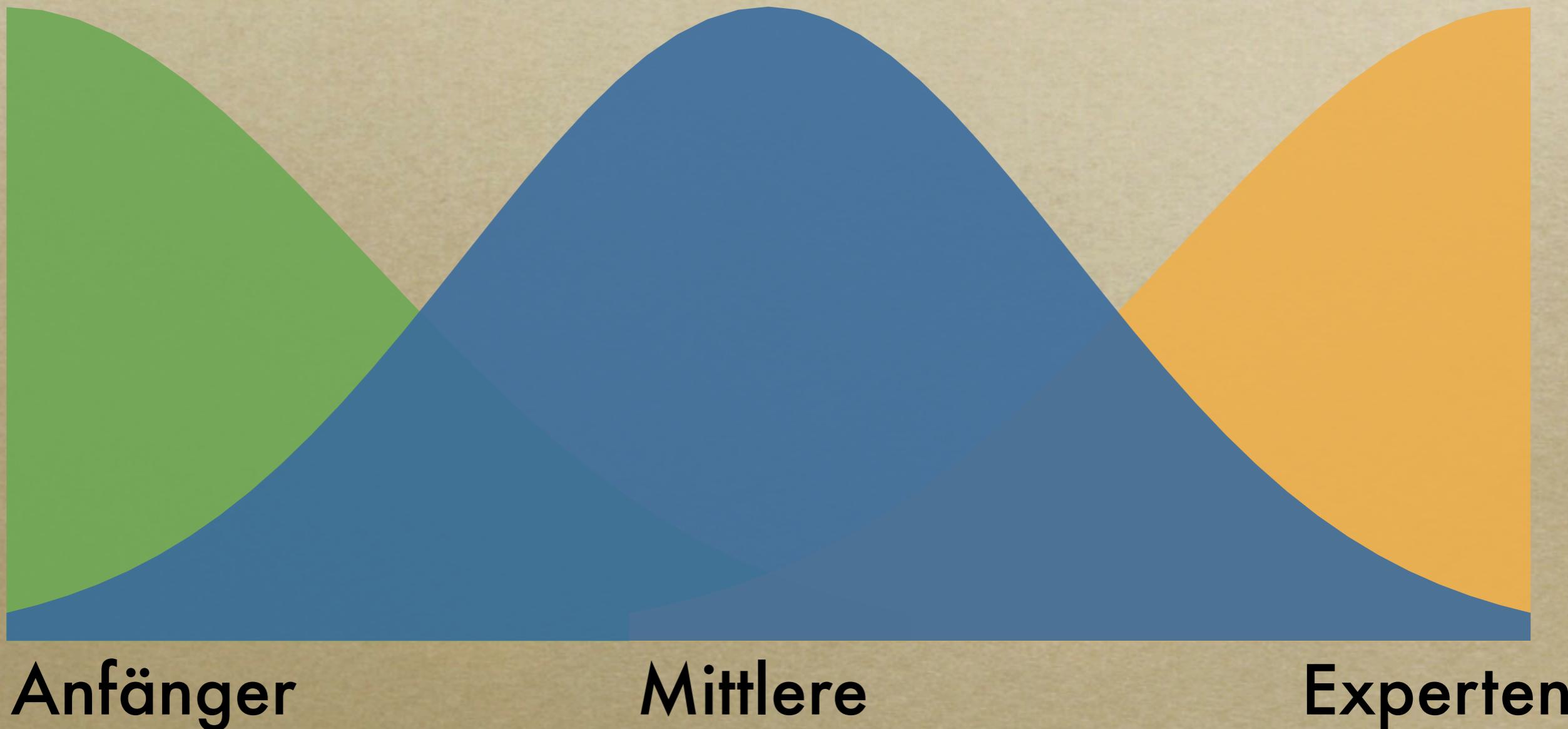


Anfänger

Mittlere

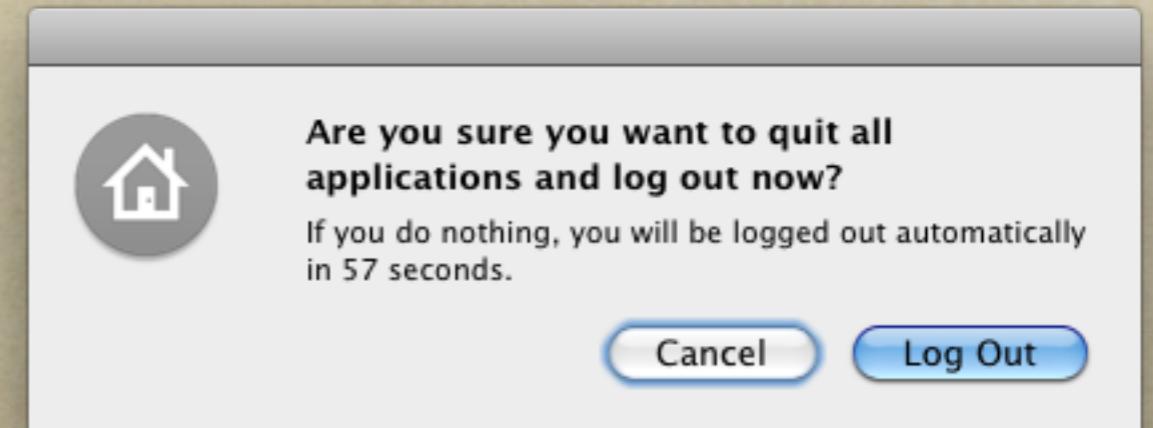
Experten

Problematik der Erfahrungsstufen

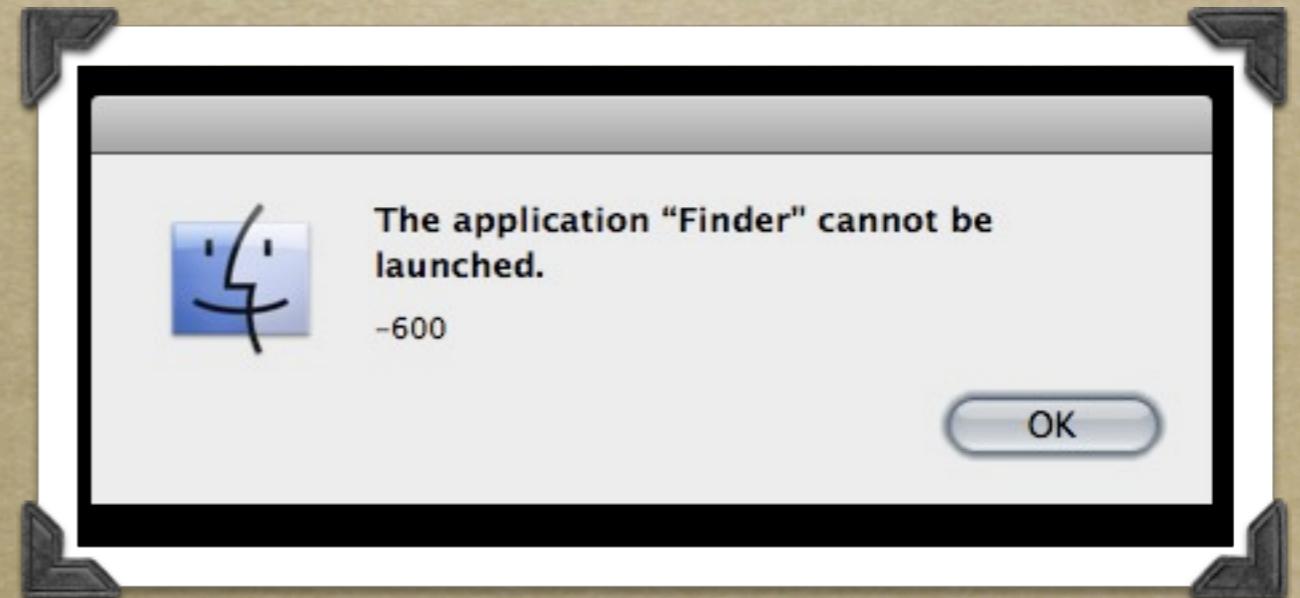


Der flexible Benutzer

- *Der User ist ein Idiot:*

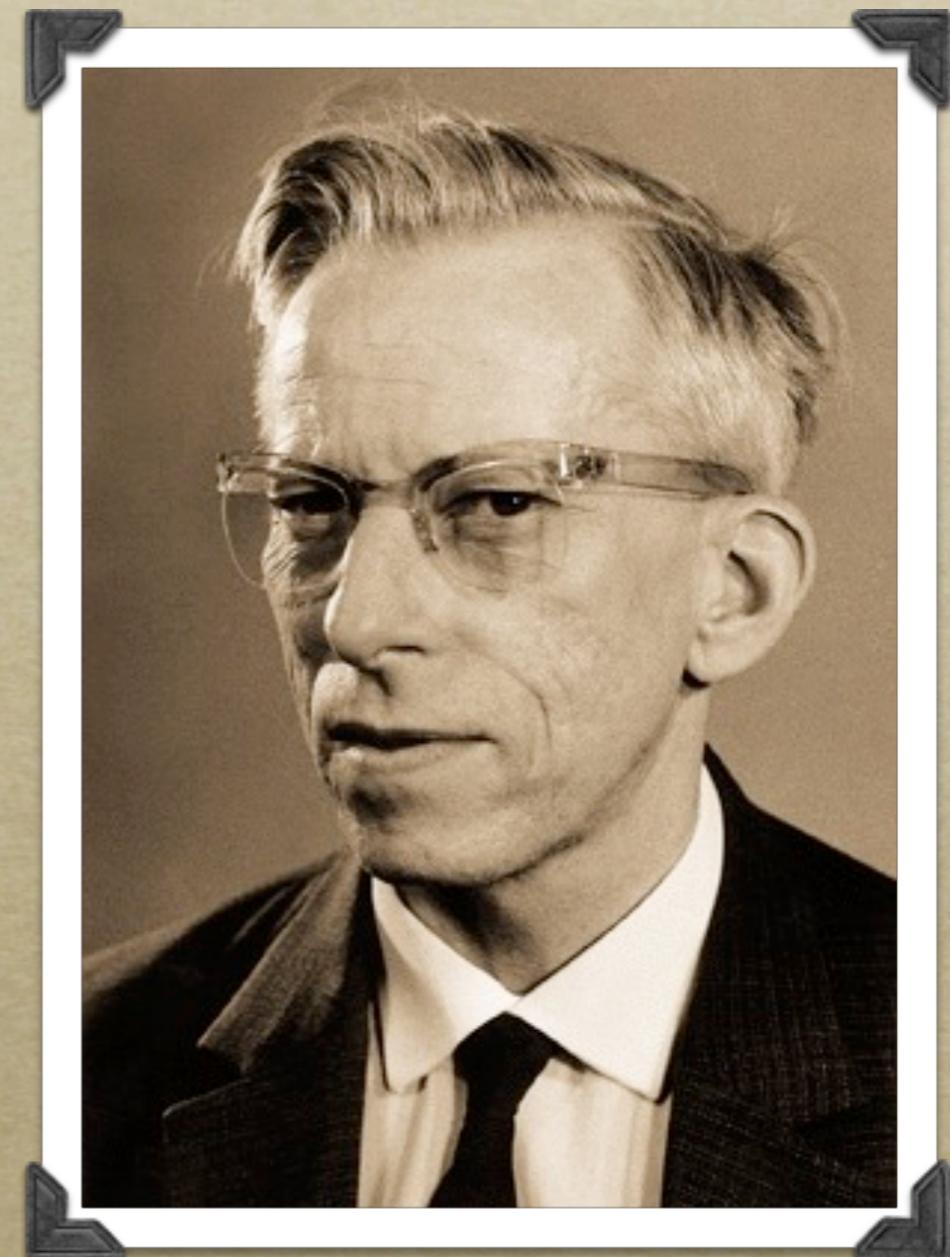


- *Der User ist ein Informatiker:*



Ausweg: Personas

- *Otto W. ist 45 Jahre alt*
- *Kaufhausangestellter*
- *macht gerne Digitalfotos im Urlaub*
- *kennt Computer nur aus dem Mediamarktkatalog*

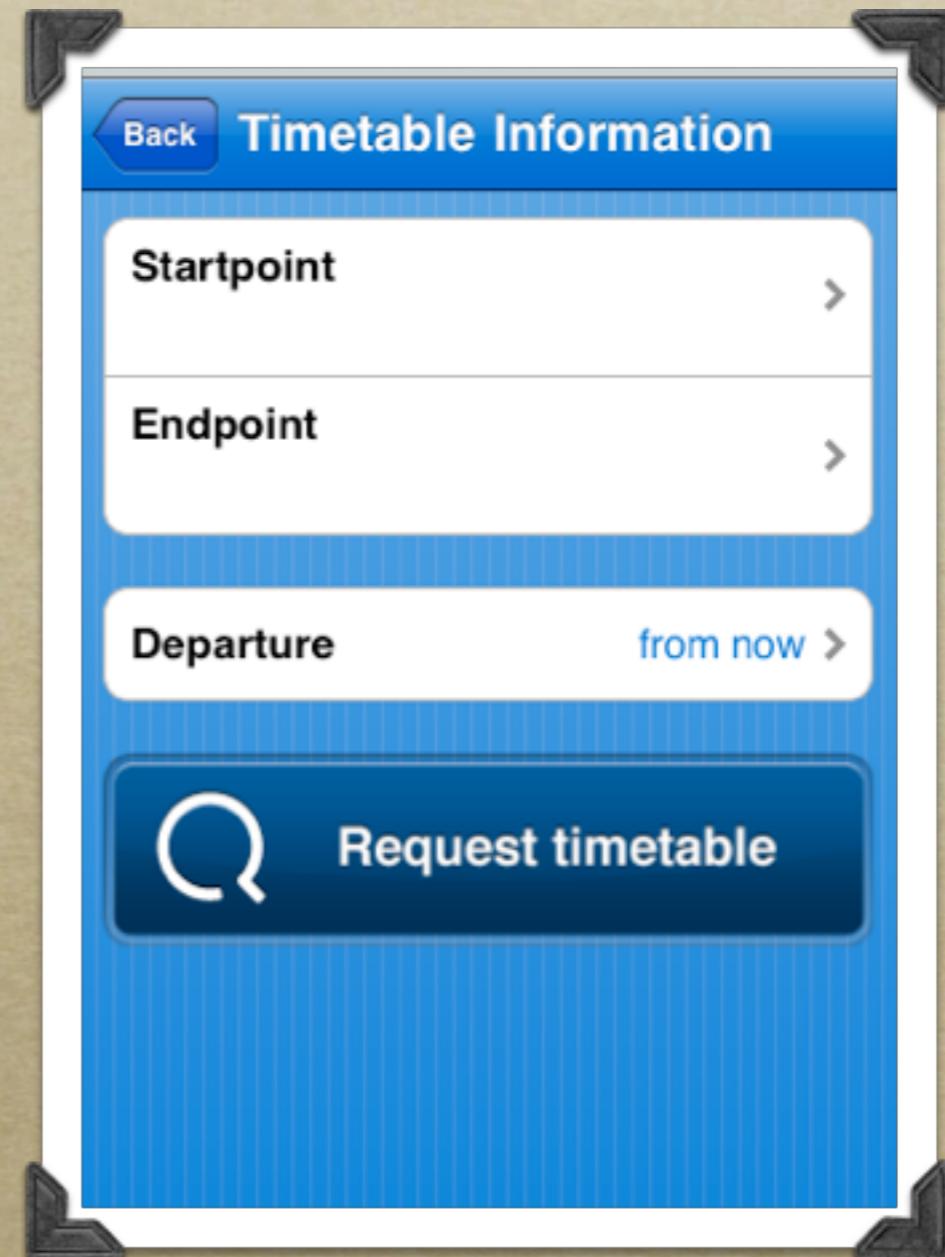


Aufgabenstellung

- *Welche Aufgaben soll der Benutzer mit dem Programm erfüllen können?*
- *Welche Information braucht das Programm als absolutes Minimum, um diese Aufgabe zu erfüllen?*

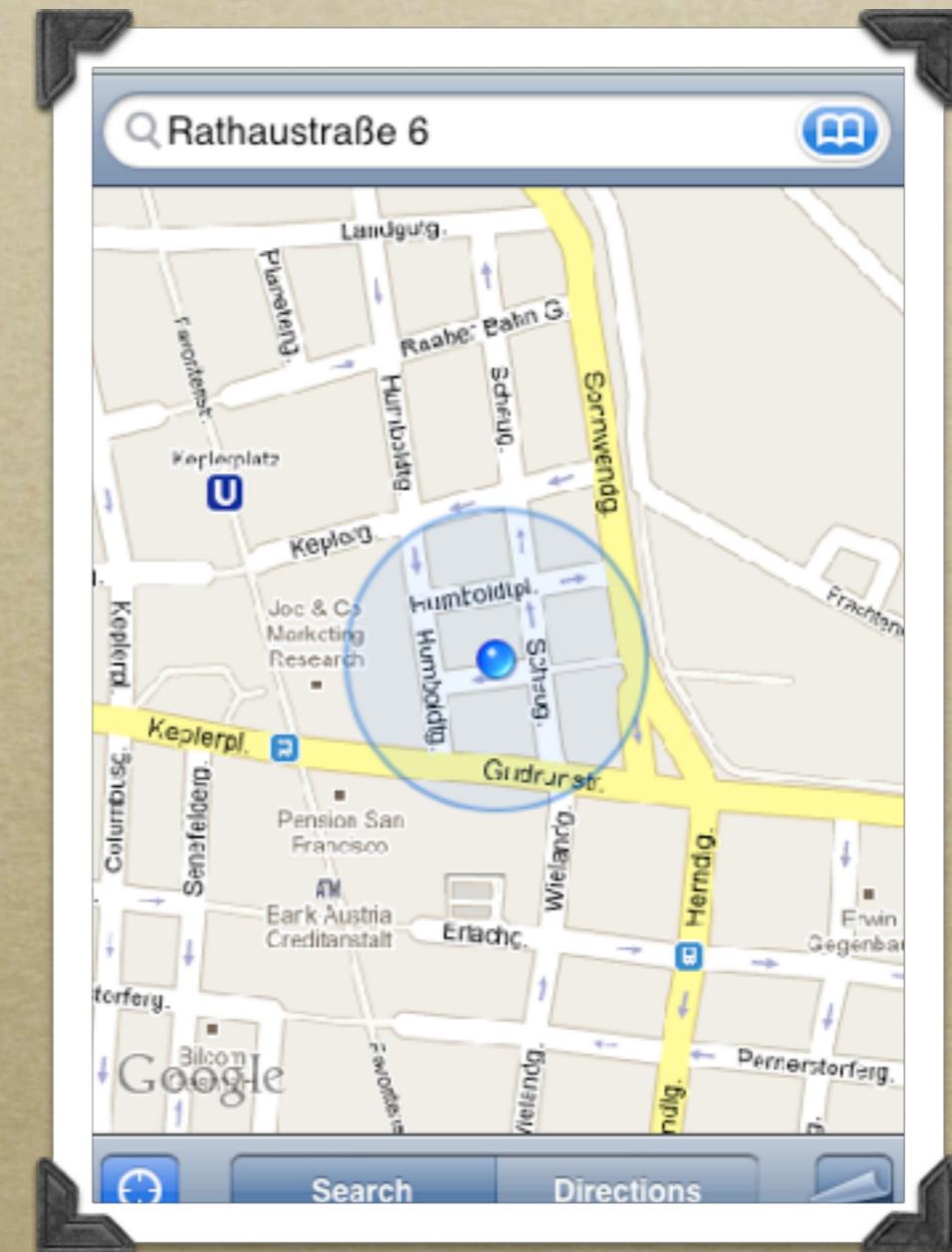
Keine dummen Fragen stellen!

- *hmm... von wo will ich wohl jetzt wegfahren?*



Keine Informationen verheimlichen!

- *ich weiß wo du bist!*



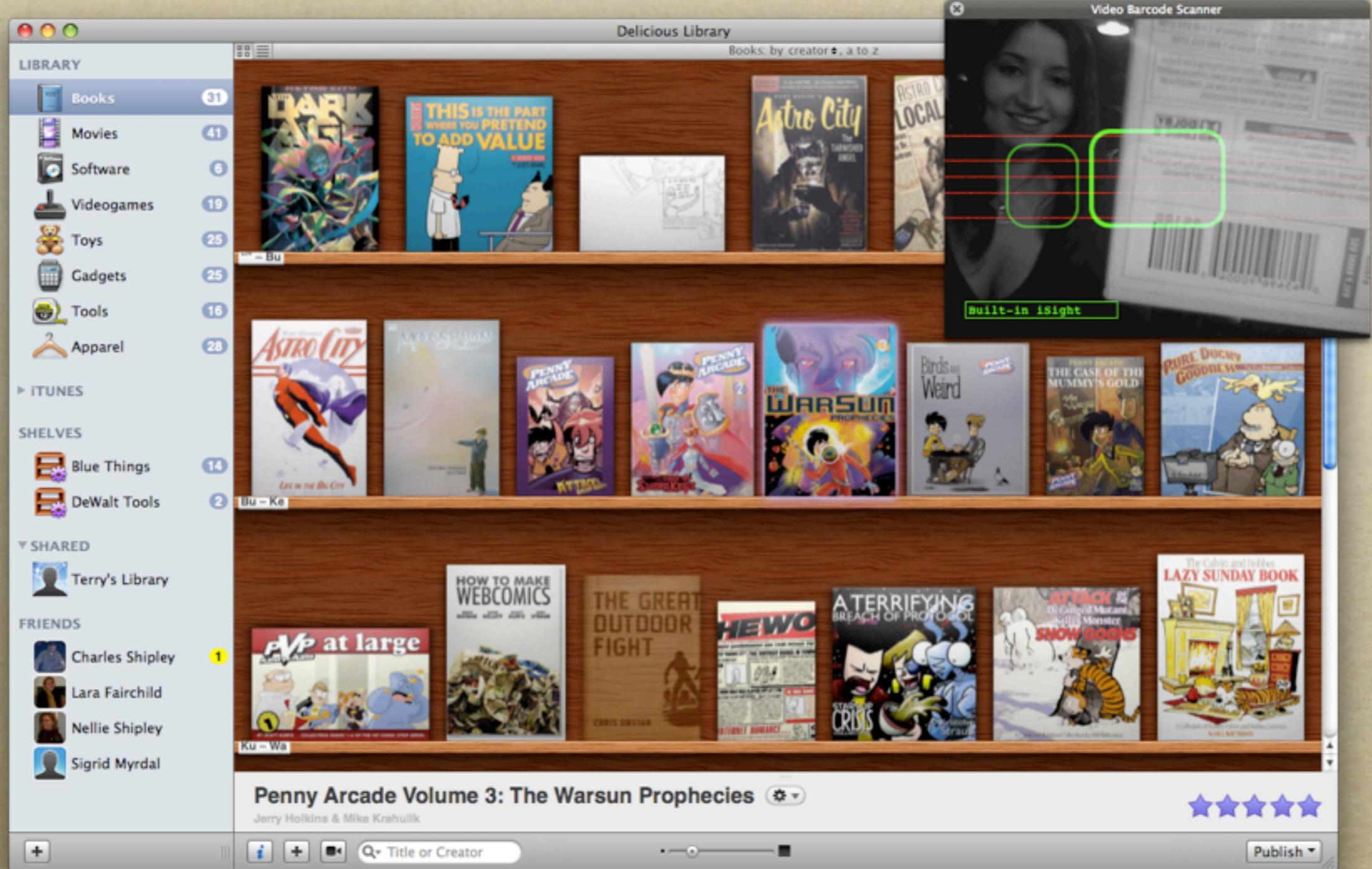
Bedienfehler tolerieren!

- *unlimitiertes Undo bei allen Operationen*
 - *(soweit technisch möglich)*
 - *statt “sind sie sicher”-Abfragen*

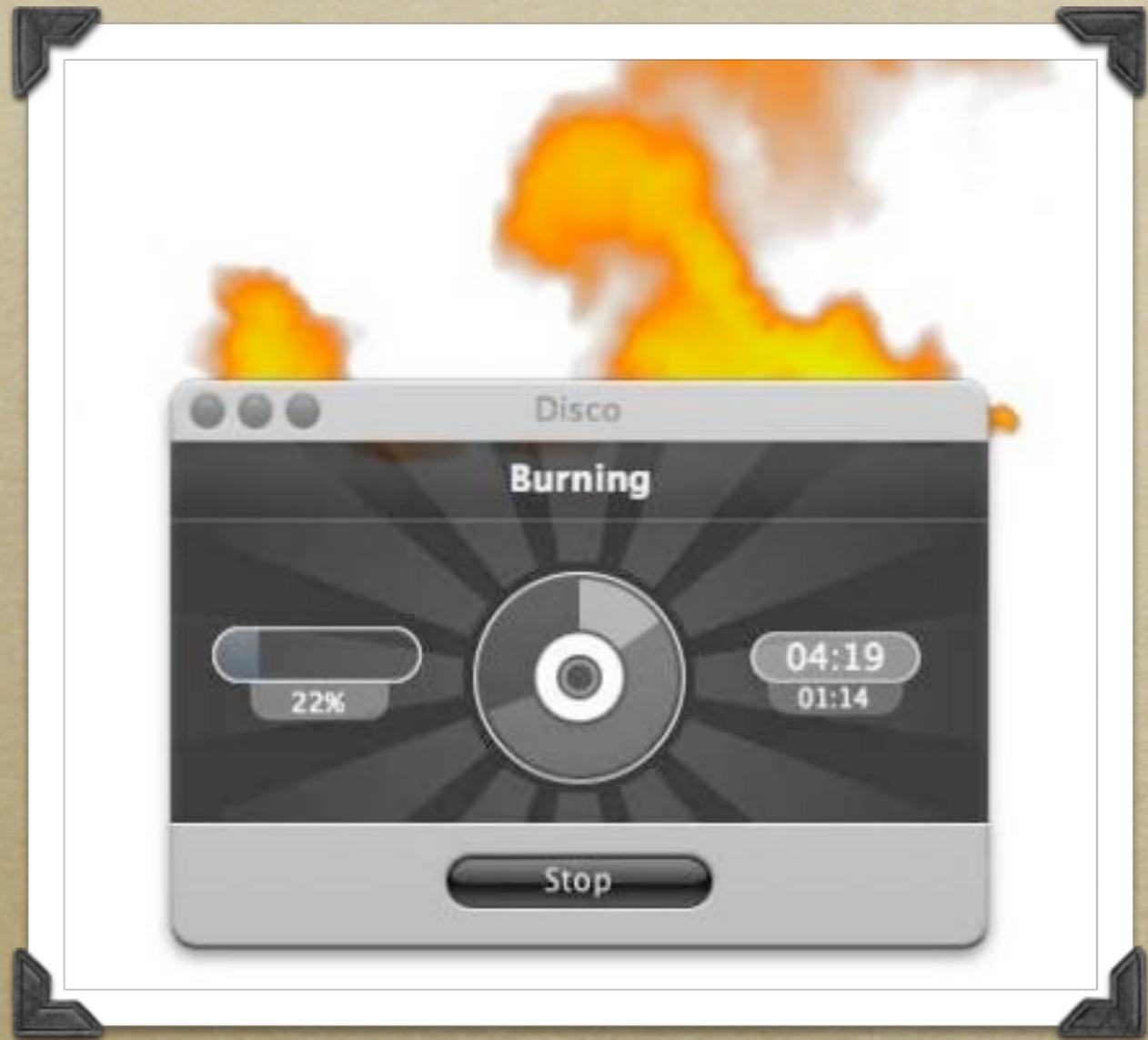


“Revolutionäre” Interfaces





Delicious Library



Disco



iPhone

Vorraussetzungen

- “*Outside the box*”-Denken
 - *Pretend it's Magic!*
- *sich trauen, diese Ideen auch wirklich umzusetzen*

Any sufficiently advanced
technology is indistinguishable
from magic.

Arthur C. Clarke, "Profiles of The Future", 1961

Pretend it's Magic



Entwicklungsablauf

- *klassisches Problem: wann ist das Produkt fertig?*
- *laufend Änderungswünsche von allen Seiten*
- *Lösung: Spezifikation?*
- *die einzige vollständige Spec ist der fertige Code selber*

Entwicklungsablauf (2)

- tatsächliche Lösung:
Beschreibung des Arbeitsablaufs
- Wenn die Arbeitsabläufe vollständig sind, dann ist das Produkt fertig.

Praktische Tipps

Ein Dateisystem ist komplex.

Wo hab ich das jetzt abgespeichert?

Backups?

Wie schicke ich das jetzt per Email?

Alles, was einer Erklärung bedarf,
ist problematisch.

*Was muss ich tun, um x zu tun?
Was bedeutet dieses grüne Ding da links unten?*

Den Erwartungen entsprechen.

*Typische Frage bei einem User-Test:
“Wenn Sie hier draufdrücken, was erwarten Sie dass
passiert?”*

Keine dummen Fragen stellen!

Kann man nicht oft genug betonen.

*Der Computer hat eine bessere Erinnerungsgabe als
Menschen, das sollte auch ausgenutzt werden.*

Negativbeispiel: Finanzonline

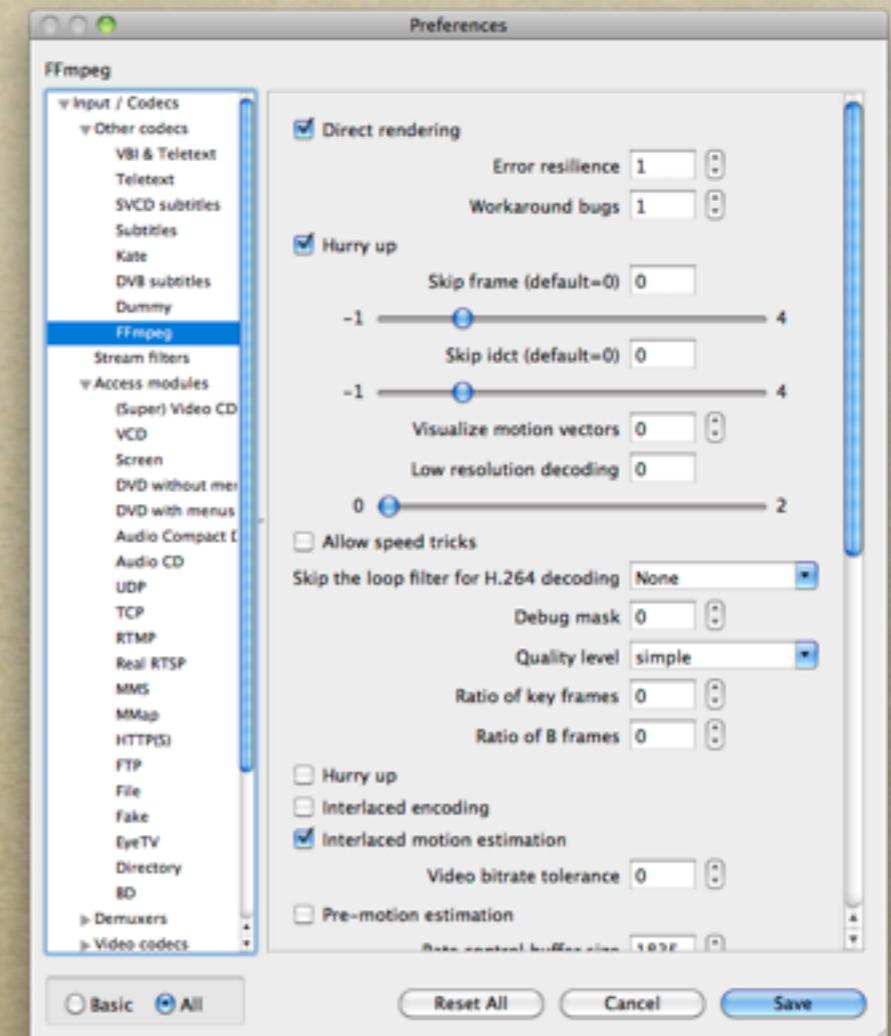
Keine Information,
die dem User nicht
hilft.

Beispiel: siehe rechts



Mehr Features bedeutet
oft auch komplexeres
User Interface.

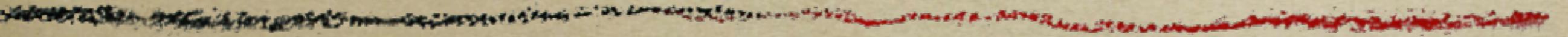
*Daher nur die Funktionen
einbauen, die auch wirklich
gebraucht werden!*



Zuerst denken, dann programmieren!

- *Bereits existierenden Code umschreiben ist ein extremer Aufwand!*
- *wird daher ungern bis gar nicht gemacht*
- *Daher zuerst planen, und dann erst ausprogrammieren.*

Was hat man davon?



Kundentreue

- *Apple hat eine der treuesten Fangemeinschaften überhaupt*
- *Woran liegt das?*



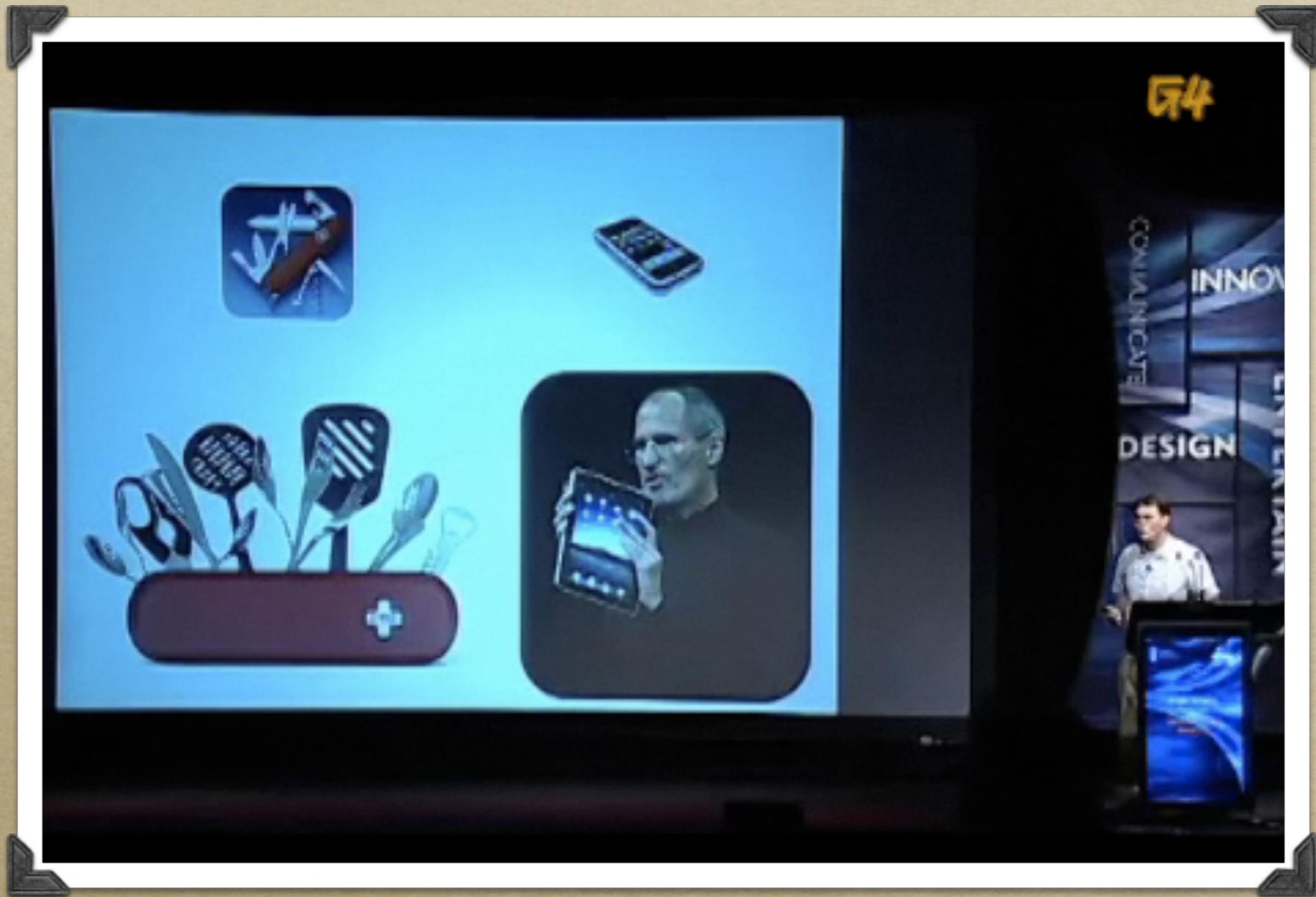
(NYC Apple Store Wedding)

Brauchen und wollen

- *ein Grafiker braucht ein Bildbearbeitungsprogramm*
- *viele Grafiker mögen aber Photoshop nicht, verwenden es trotzdem
→ es erfüllt den Bedarf, aber nicht das Verlangen*

Brauchen und wollen (2)

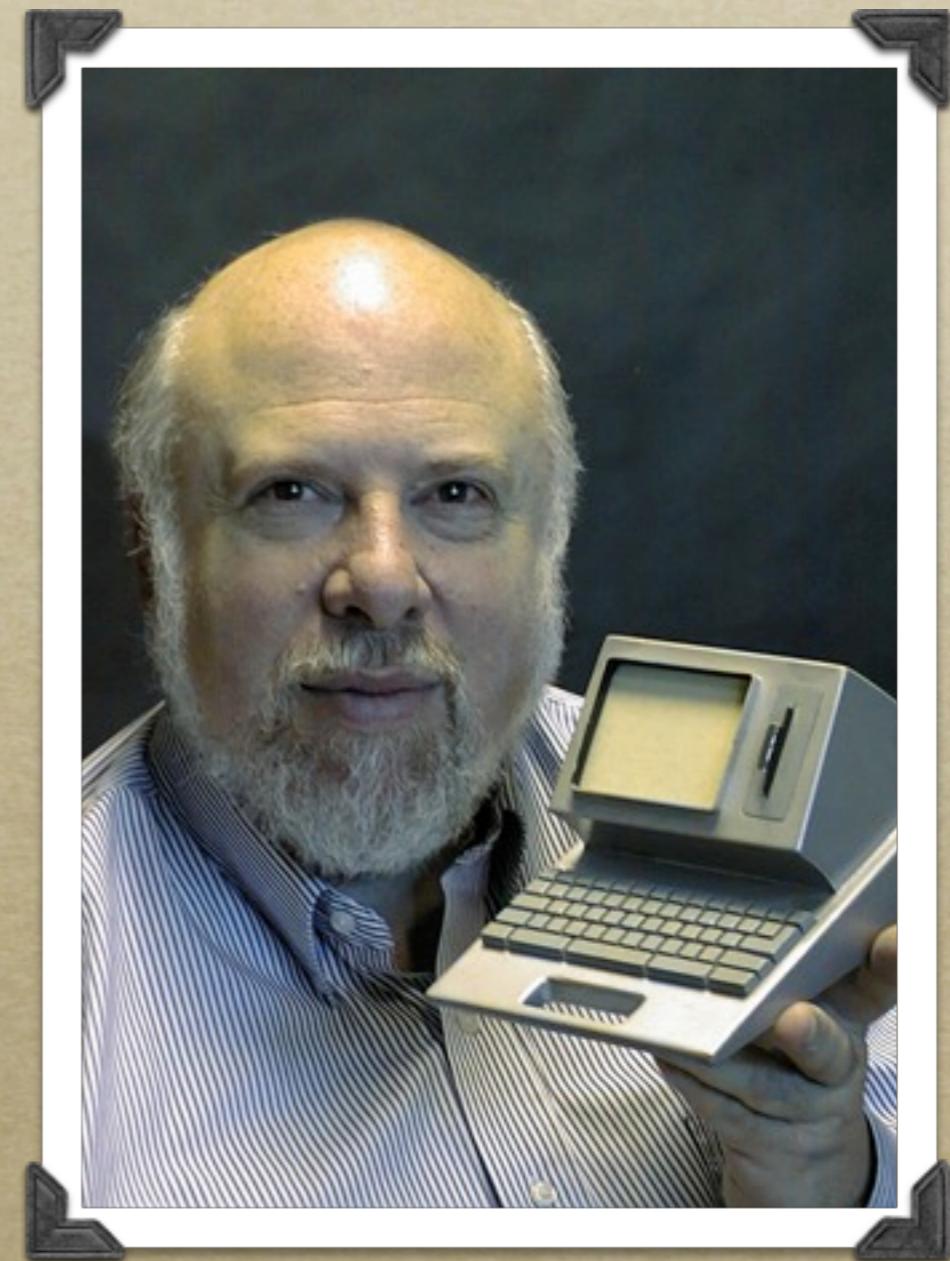
- *das iPhone ist total unnötig*
→ *niemand braucht es (ehrlicherweise)*
- *es ist angenehm zu bedienen*
→ *es erfüllt ein Verlangen*
- *Hier ist der Markt für das iPad!*



Einschub: Jesse Schell

Der morphende Computer

- *Jef Raskin's
“Information Appliance”*



Brauchen und wollen (3)

- *Das ideale Produkt erfüllt sowohl den Bedarf als auch das Verlangen.*
- *es ist angenehm zu verwenden*
- *man verwendet es gerne*
- *es geht einem nichts ab*

Brauchen und wollen (4)

- *Im Zweifelsfall verwenden normale Menschen eher das Produkt, das sie gerne bedienen als das, das wirklich alles kann was sie brauchen.*
- “*Das Produkt ist gut, es kann alles was der Benutzer machen will*” ist also *kein gültige Aussage!*

Wie ist das jetzt mit Apple?

- *Microsoft erschafft Produkte, die das tun, was die Leute brauchen.*
- *Apple erschafft Produkte, die Leute gerne verwenden.*

Zusammenfassung

- *User Interaction Design ist eigentlich der wichtigste Bestandteil der Softwareentwicklung.*
- *Er wird immer gemacht, auch wenn man sich dessen nicht bewusst ist.*
- *Wenn man konkret darauf achtet, kann man mit weniger Programmieraufwand aus einem so-la-la-Programm einen Hit machen.*

Wichtigster Punkt

*Auch ein Programmierer kann
Interaction Designer werden!*

Referenz

- *Alan Cooper:*
*The Inmates Are Running the Asylum:
Why High Tech Products Drive Us Crazy
and How to Restore the Sanity*
ISBN 978-0672326141