



Universität  
Zürich<sup>UZH</sup>

Institut für Informatik

---

# Software Quality

## Lecture 7 – UI Design, Usability & Testing

---

Thomas Fritz

Martin Glinz

*Many thanks to Meghan Allen and Daniel Greenblatt.*

---

# Overview

- Introduction to UI design
- User-centered design and human capabilities
- Usability and methods to achieve it
- Usability Heuristics
- UI testing

---

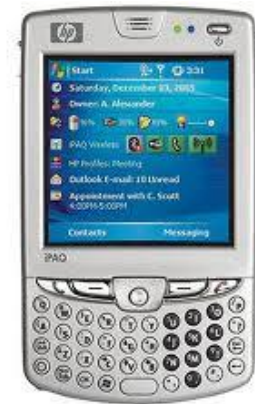
# Learning Goals

- Explain why designing interfaces is hard
- Understand why designers need to consider human abilities (memory, visual perception, ...)
- Explain why usability is important and how it can be achieved
- Analyze the usability of a UI by applying a usability method
- Analyze a GUI for problems using Nielsen's 10 usability heuristics and suggest aspects of the GUI that could be improved
- Explain how UI tests can be automated

# Designing interfaces is hard

How many of you can program or use all aspects of your

- digital watch?
- cell phone?
- DVD player?
- microwave?
- sewing machine?
- washer and dryer?
- stereo system (home or car)?
- unfamiliar water faucets?



# What causes these problems?

Now, to actually set the time, one does:

- Press and hold SET. (1 key press)
- Press MODE to select the hours place. (1 key press)
- Press SPLIT/RESET to advance hours. (6 key presses on average)
- Press MODE to select the minutes place. (1 key press)
- Press SPLIT/RESET to advance minutes. (30 key presses on average).
- Press SET when done. (1 key press)



# Symbolic issues...

- What does this mean?



---

What does this mean?



# How much is the gas?





---

Examples are obvious,

**but**

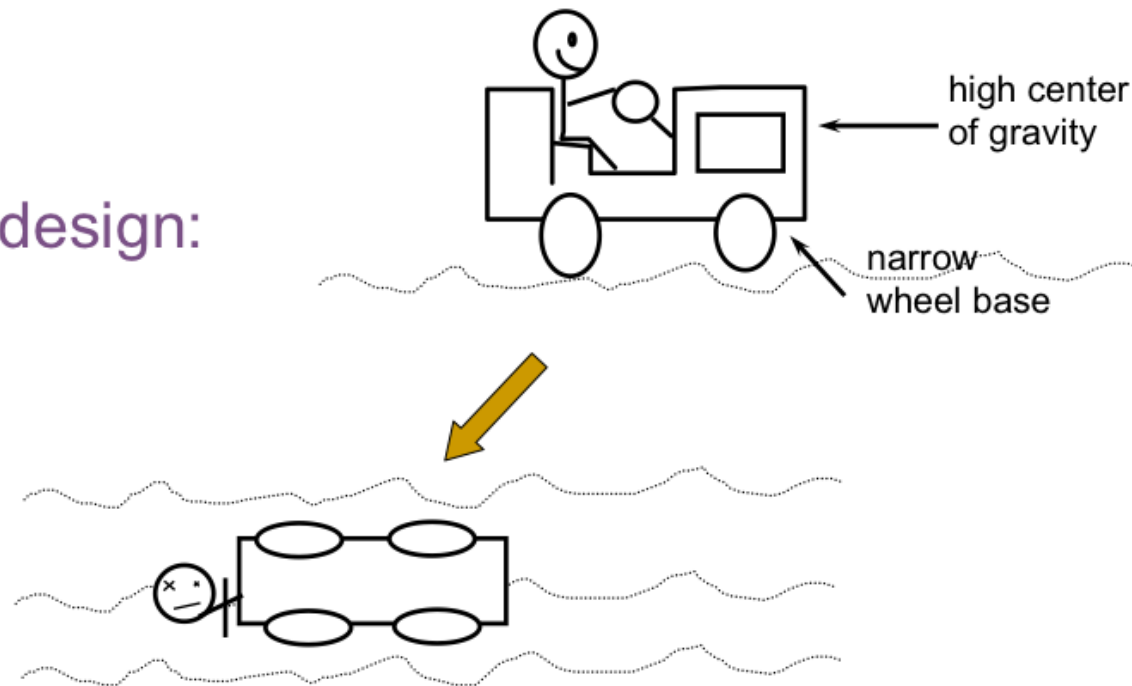
there are many subtle issues at work...

# In the past...

typical terrain:  
un-surfaced  
rough  
hilly

original design:

early tractors



**used to be called “Driver’s Error” ...**

*but*, accidents became infrequent when designs changed to low center of gravity & wider wheel bases

---

# Our approach now...



Make the tractors hard to tip...  
(make the interface easy to use and understand)

---

# Design is important

- many so-called human errors and “machine misuse” are actually errors in design
- designers help things work by providing a good conceptual model
- designers decide on a range of users as the design audience
- but design is difficult for a variety of reasons that go beyond design

*Good design avoids wasting the time of the users.*

---

---

# Group activity

Describe an interface or object that is poorly designed. How could it be improved?

eg. an elevator that will only go to the ground floor if you walk up to the first floor to press a button. This is not usable by people in wheelchairs. Could be improved by having buttons to call the elevator on the ground floor.

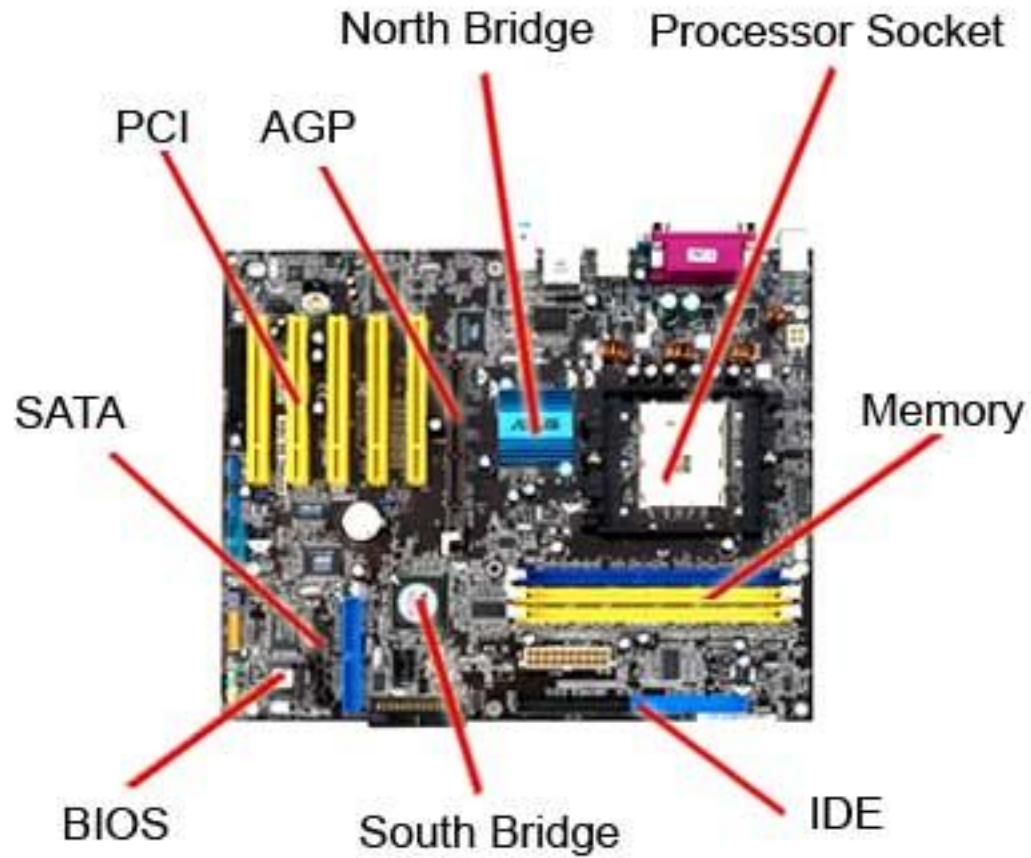
# Interface - visual



# Interface – physical



# Design





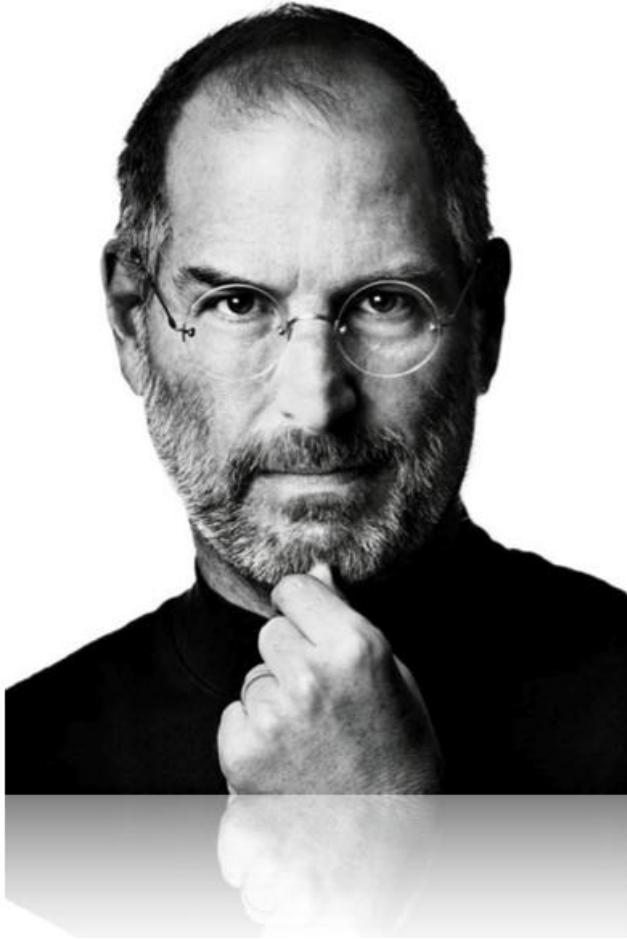
---

# Design



---

# What is design?



*Design is not just  
what it looks like  
and feels like.*

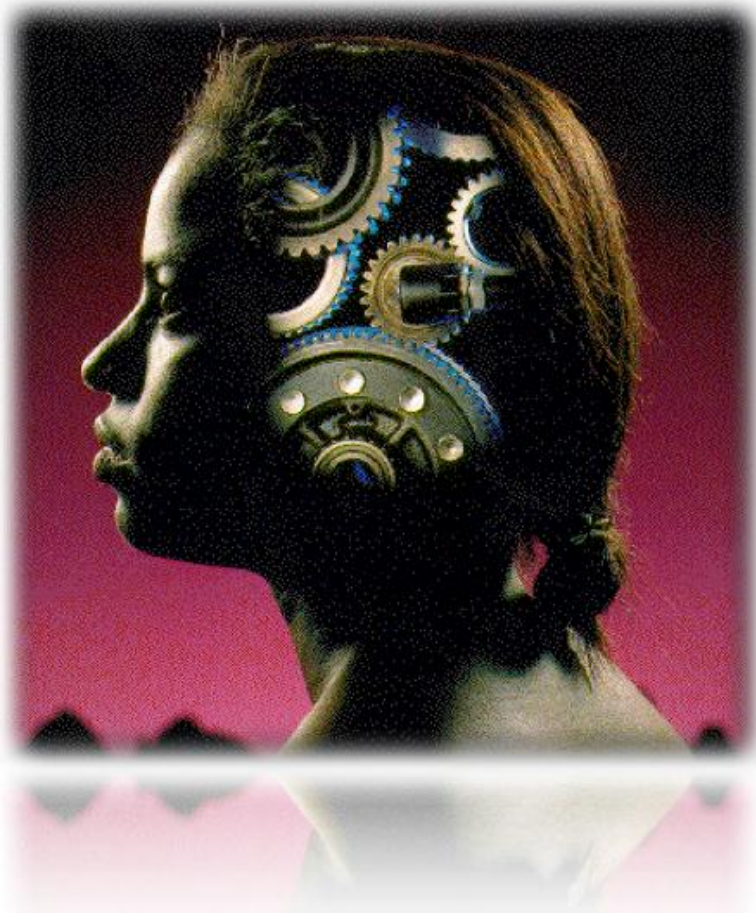
*Design is how it  
works.*

---

# User-Centered Design

- Why?
  - Cost saving
  - Competitive market
  - User expectations
  
- What?
  - Memory
  - Abilities
  - Color
  - Ergonomics
  - ...

# Human Capabilities



## Some facts on memory

- Associations are built by repetition.
- Scaffold model (more likely to remember items that have many associations).
- Recognition is easier than recall.
- Working memory has small capacity.
- Long-term memory has large capacity.

# Human Capabilities

## Visual Perception



- We excel at pattern recognition.
- We automatically try to organize visual displays and look for cues.
- Motion, grouping, contrast, color can make different parts of a display more or less salient.

---

# Human Capabilities



## Learning

- Incrementally presented information accelerates learning.
- Some users like to explore systems to learn; others will not.
- Workers focus on accomplishing tasks, not learning software.

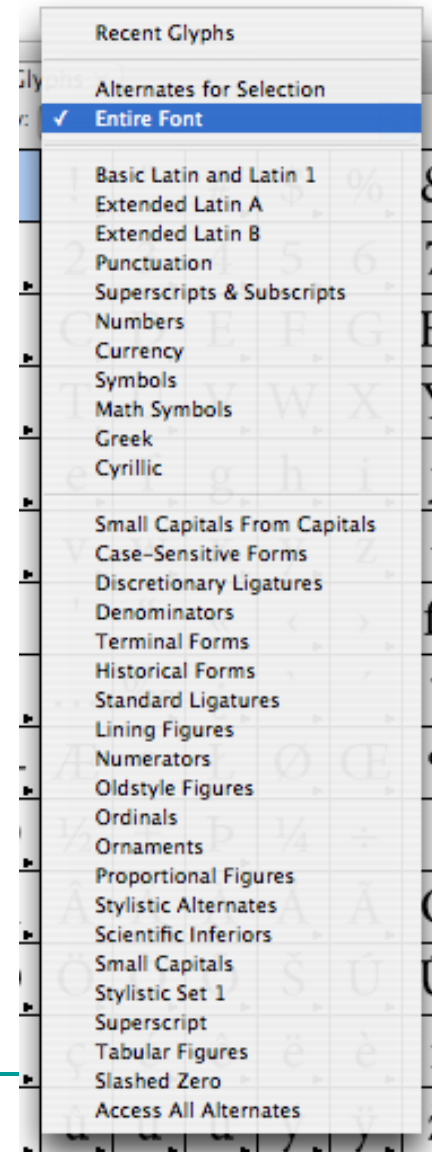
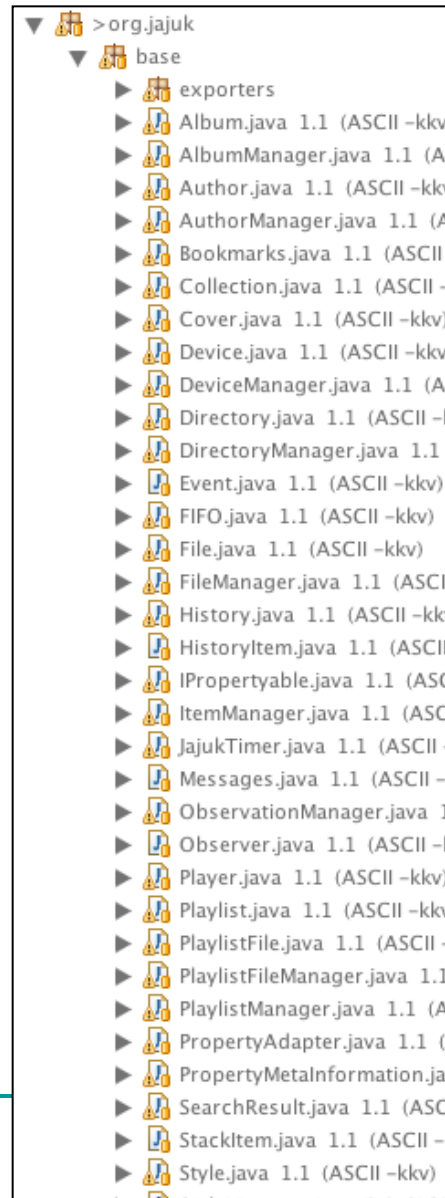
# Human Limitations



- Red-green color blindness (protonopia & deuteranopia)
  - 8% of males
  - 0.4% of females
- Blue-yellow color blindness (tritanopia)
  - Far more rare
- Guideline: don't depend solely on color distinctions
  - use redundant signals: brightness, location, shape

# Human Limitation

- Information overload





# Class activity

- Why is this street hard to shop on?
  - E.g., If you're trying to find a certain shop
  - How could you make it better?



---

# Usability

*The extent to which a product can be used by specified **users** to achieve specified **goals** with **effectiveness, efficiency and satisfaction** in a specified **context of use**.*

# Usability

- Studying and improving usability is part of Human-Computer Interaction (HCI)

- usability and good UI design are closely related
- a bad UI can have unfortunate results...



---

# Usability is NOT utility/functionality

- Utility is how useful something is
- Functionality is concerned with functions/features of the product
  
- Usability is how easy it is to use
  
- You can have something that is extremely usable, but has no utility
- A good product / software should have both

---

# Usability Methods

- Prototyping
- Surveys
- Card sorting
- Contextual inquiry
- Cognitive walkthrough
- Usability testing
- Heuristic evaluation
- ...

---

# Prototyping

- Creating a scaled-down or incomplete version of a system to demonstrate or test aspects of it
- Reasons to do prototyping:
  - aids UI design
  - provides basis for testing
  - team-building
  - allows interaction with user to ensure satisfaction

---

# Benefits of Paper Prototyping

- much faster to create than code
  - easier to change than code
  - encourages feedback, since it feels less permanent or final
  - focuses on big things vs. small (like the font)
  - implementation neutral
  - can be done by non-technical people
- 
- PP shows us “what” is in the UI, but also “how” the user can achieve their goals in the UI; helps uncover requirements

---

# Surveys

- used to collect quantitative data
- Helps to find information on who the users are, how they use your product, users' opinions, ...
- structured interviews
- consider mixing open-ended with closed questions
- plan questions well (usually no follow-up questions possible)
- keep it short



# Card Sorting

- Helps to determine mental model of users regarding a site/application's content and how it should be structured/categorized
- each index card contains a word or phrase, e.g. representing an information item on a web site
- Open card sort:
  - users are asked to group a stack of cards as it makes sense to them
  - then name each group of cards
- Closed card sort:
  - Users are asked to place the cards into predefined groups

# Contextual Inquiry

- Interviews with people in their places of work or homes – their natural work environment
- Helps to understand the task procedures that users follow to reach their goals
- Helps to get more qualitative data around context of use (physical characteristics, social/cultural environment,...)
- silently observe (watch and listen) the user and take notes; you may ask clarifying questions
- rather informal

# Cognitive Walkthrough

- group of usability experts *walk* through a set of typical user tasks, one-step-at-a-time
- define tasks and sub-tasks to be analyzed
- establish persona: who is the user of the system, specify experience or technical knowledge for user;
  - e.g., very computer-savvy teenage kid / senior without tech experience

# Cognitive Walkthrough

- at each step, evaluator answers 4 questions about her expectations of users' behavior:
  - Will the user try to achieve the right effect?
  - Will the user notice that the correct action is available? (visibility, understandability in the UI)
  - Will the user associate the correct action with the effect to be achieved? (action to UI element connection, label-following)
  - If the correct action is performed, will the user see that progress is being made toward solution of the task? (system feedback)
- try to create success (failure) story for each step

# Cognitive Walkthrough

## Common features of success

- ❑ “what effect to achieve”
  - part of original task, experience using a system, system tells them to do it
- ❑ “an action is available”
  - experience, seeing some device (eg. button), seeing a representation of an action
- ❑ “an action is appropriate”
  - experience, prompt or label that connects action to what they are trying to do, all other things look wrong
- ❑ “things are going OK”
  - experience, connection between system response and what they are trying to do

---

# Usability Testing

- Give people series of tasks representing important workflows in product
- identify any usability problems, collect quantitative data on users' performance (e.g., time on task, error rates), and determine user's satisfaction with product
- Identify target audience and typical tasks, establish success criteria
- User only gets task, but no other guidance
- Observers watch, listen and take notes

---

# Usability Testing

## Why do it

- Because you are not your users! Things obvious to you might not be obvious to your users
- Good way to find usability problems early on

# Class Activity 1/4 – Usability Test

- In groups of 2 (A,B), select one person (A) to perform the task one person to observe
- B: Stop the time, count all clicks, count the clicks that were made erroneously, ask how easy it was for user (1 : very easy, 2 : easy, 3 : moderate, 4 : difficult, 5 : very difficult)
- **Task:** Create a playlist called “onerepublic” with the song Good Life from OneRepublic on youtube
  - Success: playlist is created correctly
  - Target audience: youtube users
  - Start from [www.youtube.com](http://www.youtube.com), make sure to be logged out first



# Class Activity 2/4 – Cognitive WT

- Second person (B) does cognitive walkthrough, given the sequence of six steps
- For each step, observer (here we have an observer but not necessary, A) asks the four questions and writes down success/failure stories
- **Task:** Create a playlist called “onerepublic” with the song Good Life from OneRepublic on youtube
  - Target audience: youtube users with computer science and strong tech-background

# Class Activity 3/4 – Usability Test

- B now performs usability test
- A: Stop the time, count all clicks, count the clicks that were made erroneously, ask how easy it was for user (1 : very easy, 2 : easy, 3 : moderate, 4 : difficult, 5 : very difficult)
- **Task:** Change your facebook setting so that only friends of friends can send you messages (not everyone)
  - Success: settings are changed
  - Target audience: facebook users
  - Start from [www.facebook.com](http://www.facebook.com), make sure to be logged out first

# Class Activity 4/4 – Cognitive WT

- A does cognitive walkthrough, given the sequence of six steps
- For each step, observer (here we have an observer but not necessary) asks the four questions and writes down success/failure stories
- **Task:** : Change your facebook setting so that only friends of friends can send you messages (not everyone)
  - Target audience: facebook users with computer science and strong tech-background

---

# Heuristic Evaluation



- Nielsen's 10 Usability Heuristics (Jakob Nielsen)
- Shneiderman's 8 Golden Rules
- Tog's 16 Principles

---

# Nielsen's Principles - #1



































































## Match the Real World

- System should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms
- Follow real-world conventions
- Make information appear in natural and logical order
- Examples:
  - Files and folders on a desktop

# Nielsen's Principles - #2

## Consistency and Standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

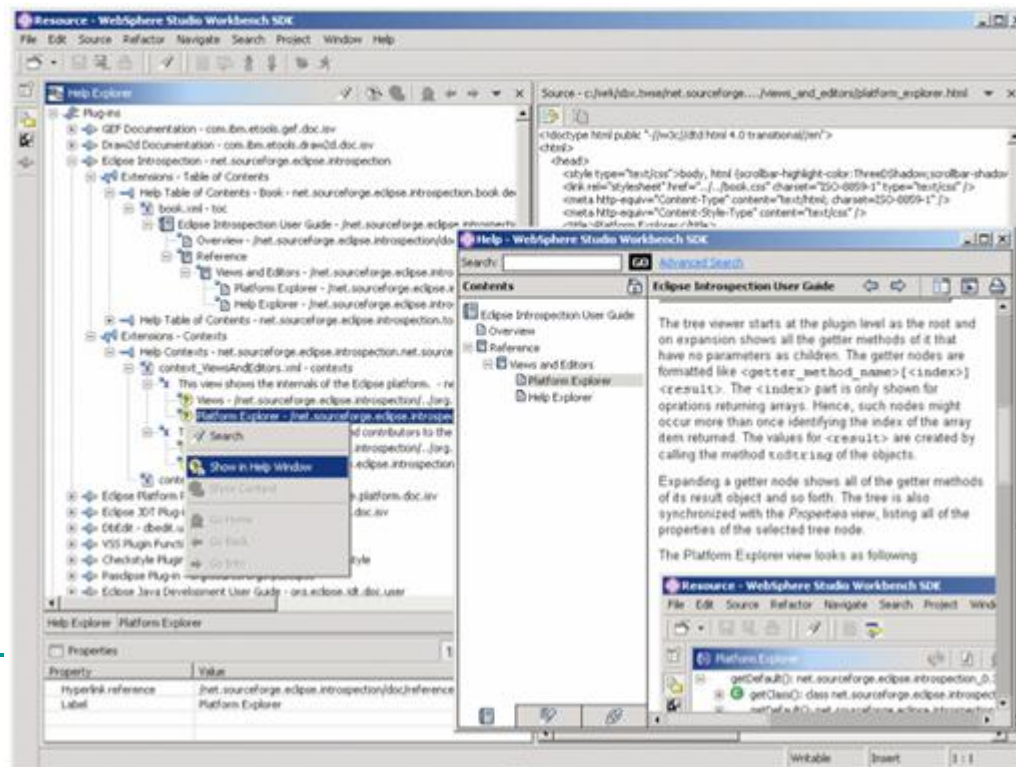
create, new		compare		forward		jar		plugin	
save		debug		backward		WAR		extension	
cut		run, execute		previous		EAR		extens'n point	
copy		import		next		window		thread	
paste		export		project		perspective		process	
add		play, resume		open project		property sheet		mapping	
remove		suspend		folder		table		error	
delete		terminate		open folder		database		warning	
erase, clear		stop		file		repository		alert	
search		undo		library		class		conflict	
find		redo		package		interface		public	
help		refresh		session bean		attribute		protected	
edit		filter		server		element		private	
								default	

- Eclipse's icons guidelines
- Mac, Windows, Gnome and KDE guidelines

# Nielsen's Principles - #3

## Help and Documentation

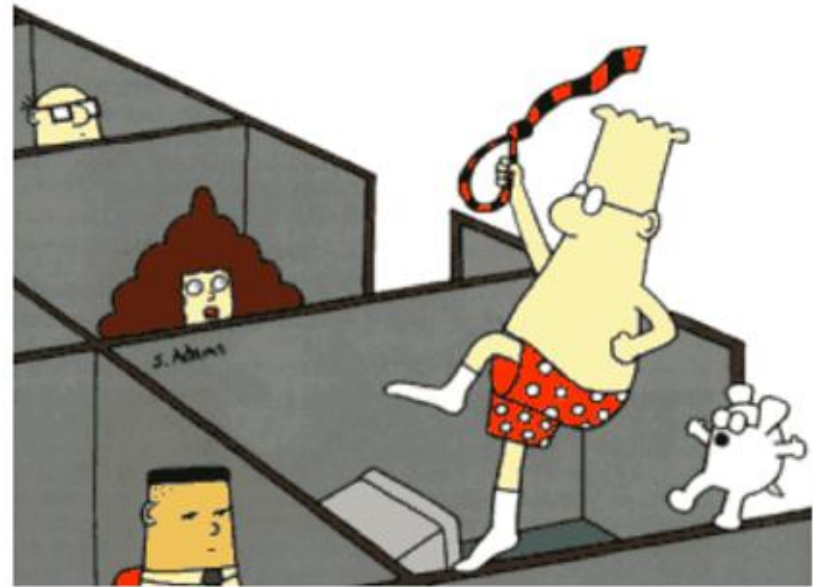
- Help should be searchable, focused on user's task, concrete and short.



# Nielsen's Principles - #4

## User Control and Freedom

- Provide “emergency exit” without having to go through extended dialogue
- Support undo and redo

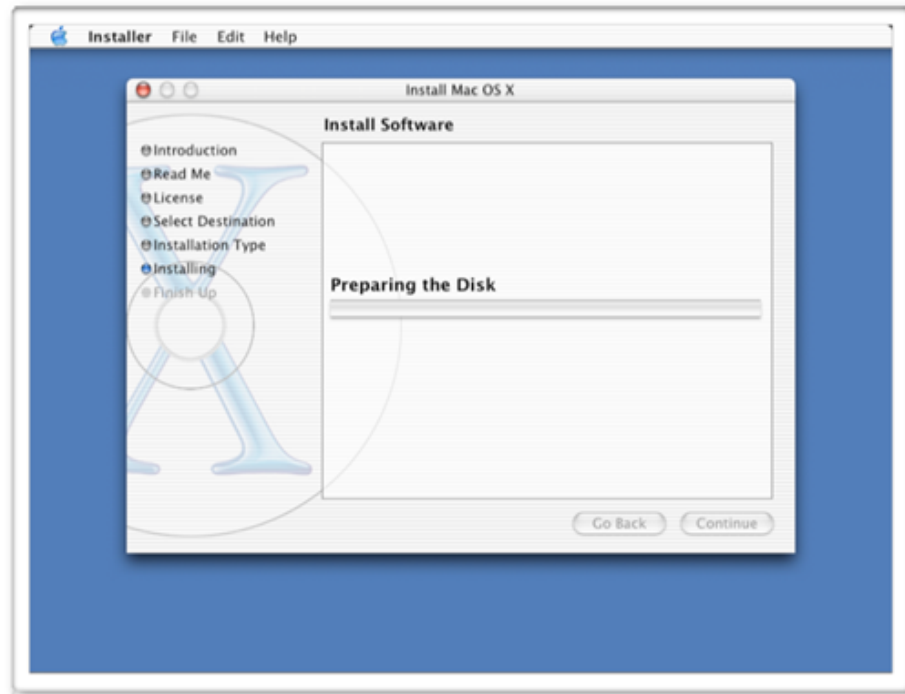




# Nielsen's Principles - #5

## Visibility of System Status

- keep users informed about what is going on, through appropriate feedback within reasonable time



# Nielsen's Principles - #6

## Flexibility and efficiency of use

- Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

### Keyboard shortcuts



Keyboard shortcuts help you save time by allowing you to never take your hands off the keyboard to use the mouse. *You'll need a Standard 101/102-Key or Natural PS/2 Keyboard to use the shortcuts.*

To turn these case-sensitive shortcuts on or off, click [Settings](#), and then pick an option next to **Keyboard shortcuts**.

Shortcut Key	Definition	Action
c	Compose	Allows you to compose a new message. <Shift> + c allows you to compose a message in a new window.
/	Search	Puts your cursor in the search box.
k	Move to newer conversation	Opens or moves your cursor to a more recent conversation. You can hit <Enter> to expand a conversation.
j	Move to older conversation	Opens or moves your cursor to the next oldest conversation. You can hit <Enter> to expand a conversation.

- e.g. gmail shortcuts

# Nielsen's Principles - #7

## Error Prevention

- preventing errors is better than good error message
- eliminate error-prone conditions or check and present users with confirmation option

The screenshot shows a web form titled "Movie Information for: Predator". The form is divided into several sections:

- Movie Collection:** Contains fields for MovieID (32), Title (Predator), Genre (Science Fiction), Region (2), Sound (5.1), and a Comments text area.
- Actors:** A list of actors with a dropdown menu. The dropdown is open, showing a list of names: Arnold Schwarzenegger, Carl Weathers, Bill Duke, Bill Duke, Charles S Dutton, Robert Duvall, Christopher Eccleston, Dakota Fanning, and Will Ferrell. The second "Bill Duke" entry is highlighted, and a mouse cursor is pointing at it. Below the list is a "Record:" indicator showing "3".
- Directors:** A dropdown menu with "John McTiernan" selected. Below it is another empty dropdown menu. Below that is a "Record:" indicator showing "1".
- Navigation:** At the bottom left, there are navigation buttons: "Movie 27 of 40" and "Close".

The form is designed to be error-prone by allowing multiple selections of the same actor name and by not providing a confirmation option for the selection.

# Nielsen's Principles - #8

## Recognition rather than recall

- Minimize user's memory load by making objects, actions, and options visible.



# Nielsen's Principles - #9

## Help users recognize, diagnose, and recover from errors

- Error messages in plain language (no codes), precise, and constructive

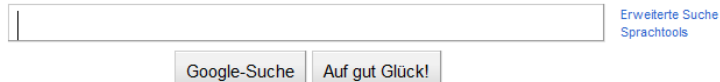


This example violates this principle!

# Nielsen's Principles - #10

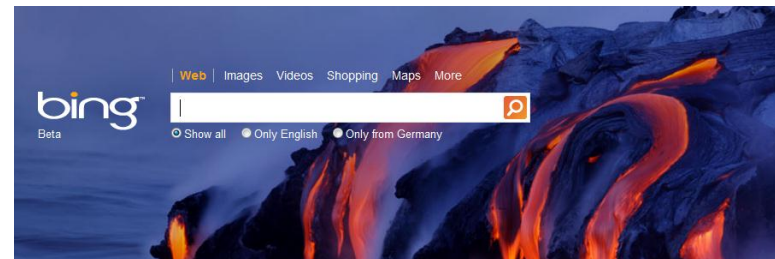
## Aesthetic and Minimalist Design

- Dialogues should not contain information which is irrelevant or rarely needed



[Werben mit Google](#) [Unternehmensangebote](#) [Über Google](#) [Google.com in English](#)

© 2011 - Datenschutz



---

# Group activity

- Find (or recall) a GUI that violates principle(s)
  - List principle(s) it violates
  - Name the program
  
- Find a great screenshot?

---

# When to apply Usability Methods

- Early and often!
- You can test with working code, a prototype, high-fidelity user interface mock-ups or even sketches on paper (paper prototyping)
- The earlier you can validate a certain approach, the less time spent writing wasted code
- **Prototype and test often!** before and during development; best to have an iterative testing cycle, not a one-off phase



---

# Usability Cost/Benefit

- Different methods take different amounts of time / money
- The expert evaluation methods are generally cheaper and you can still find the biggest usability problems

article from Nielsen on "Discount Usability Methods"  
[http://www.useit.com/papers/guerrilla\\_hci.html](http://www.useit.com/papers/guerrilla_hci.html)

---

# User Interface Testing

- Testing the functionality of the user interface, should be done after changes to the code
- Simulating user interaction is often difficult to achieve programmatically
- test plans: sequence of steps people have to perform
- Test frameworks for web application support the automation of tests

---

# UI Test – Selenium

- Testing framework for web applications
- Selenium IDE for Firefox to “record” test cases
- Test cases can be saved in multiple programming languages
- Automated in build process

<DEMO>

---

# Summary

- User Interface Design
  - Creative, but requires engineering
  - Can effect product success
  - Can cause happy/unhappy customers
  - Is as important as functionality
- Usability methods can help ensure success and should be applied early and often
- Selenium can help to automate UI testing (not usability testing!)