



# Requirements Engineering and User Experience

## CS421

### Usability

Dr. Salma Hamza

MedTech, SMU, Spring 2024

# Outline

- 1** What is Usability?
- 2** Learnability
- 3** Visibility
- 4** Error Prevention
- 5** User Control
- 6** Efficiency
- 7** Design
- 8** Heuristic Evaluation
- 9** Summary

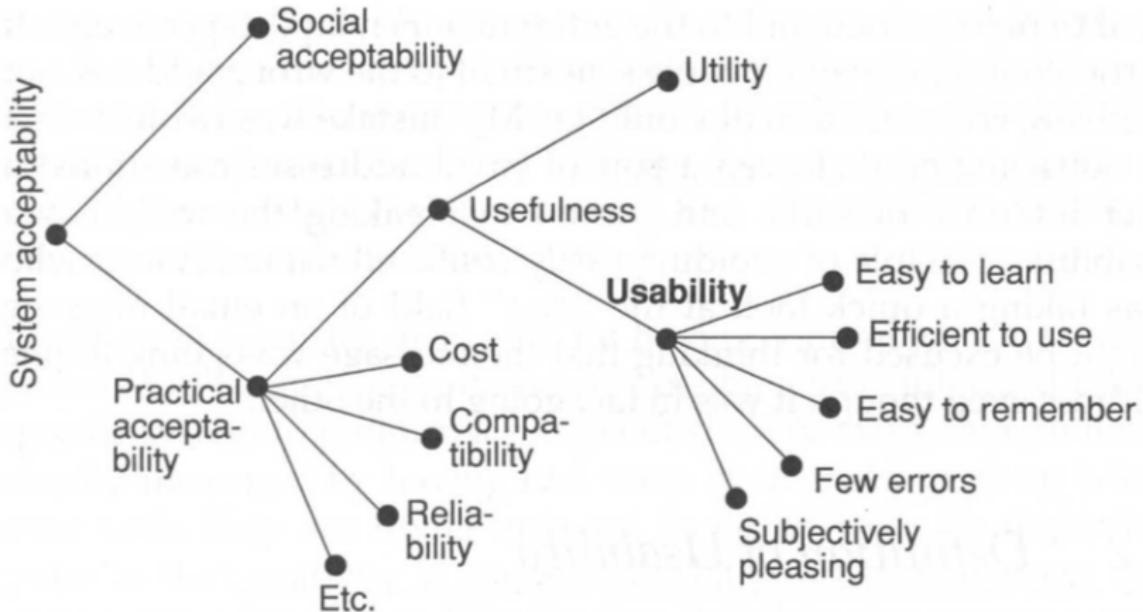
## What is Usability?

---

# The UI is important

- User Interface strongly affects perception of software
  - Usable software sells better
  - Unusable web sites are abandoned
- User Interfaces are Hard to design
  - You are not the user
    - Most software engineering is about communicating with other programmers
    - UI is about communicating with users
  - The user is always right
    - Consistent problems are the system's fault
    - ...but the user is not always right, either
      - Users aren't designers

# A Model of the Attributes of System Acceptability



# Usability vs. Utility

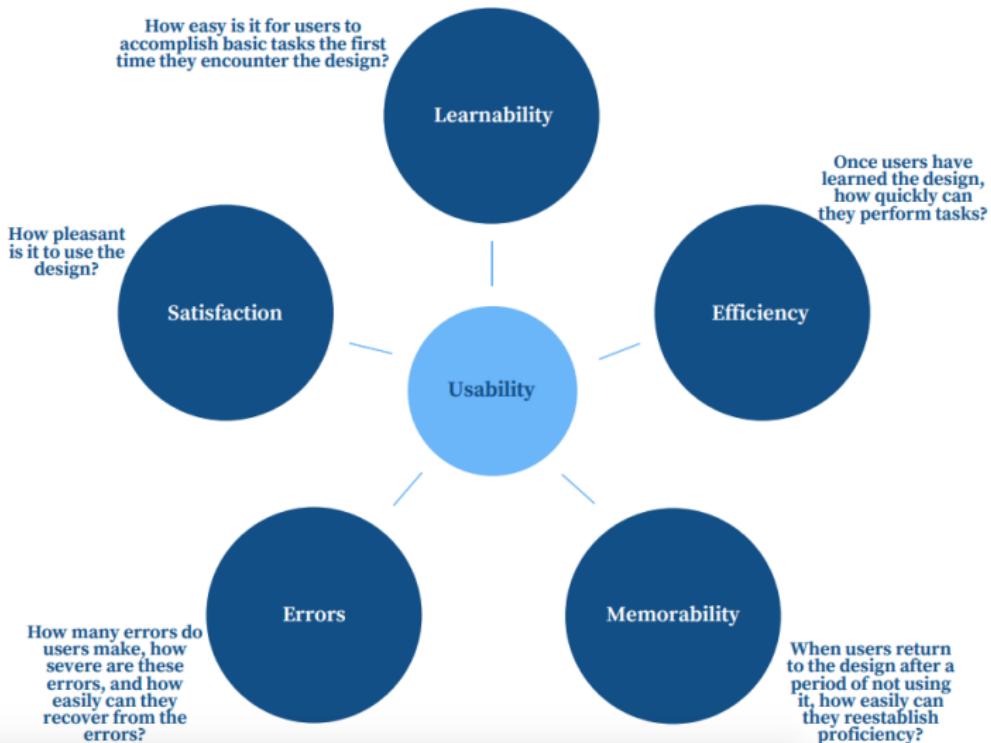


- Does the system provide the capabilities to allow the user to achieve their goal?
  - This is **utility** or **functionality**.
- Does the system allow the user to learn and to use the capabilities?
  - This is **usability**.
- **Both utility and usability are essential**
  - They must be measured in the context of particular types of users.

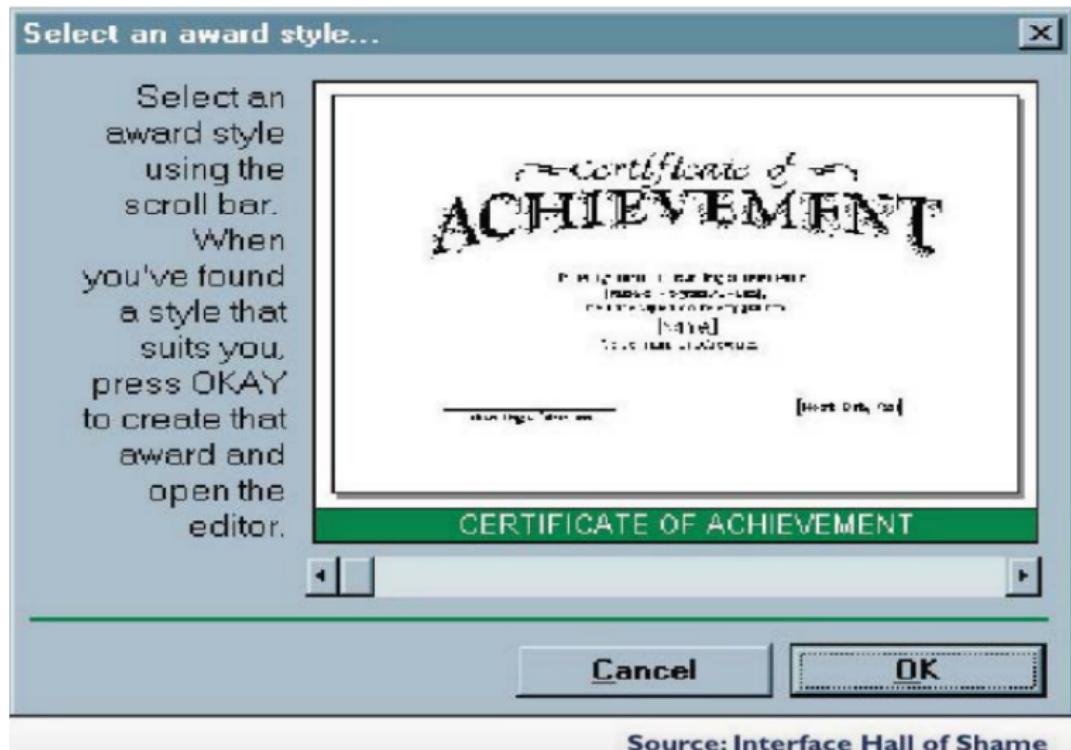
## Definition: Usability - ISO (9241-11, 1988)

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.

# 5 Quality Components of Usability

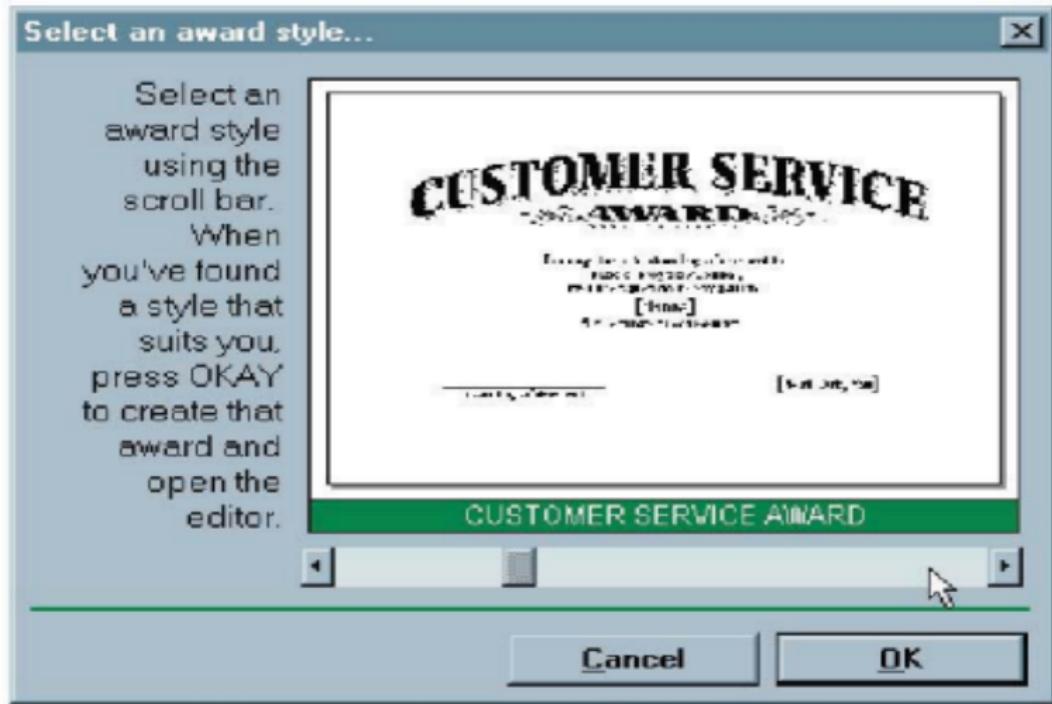


# User Interface Hall of Shame



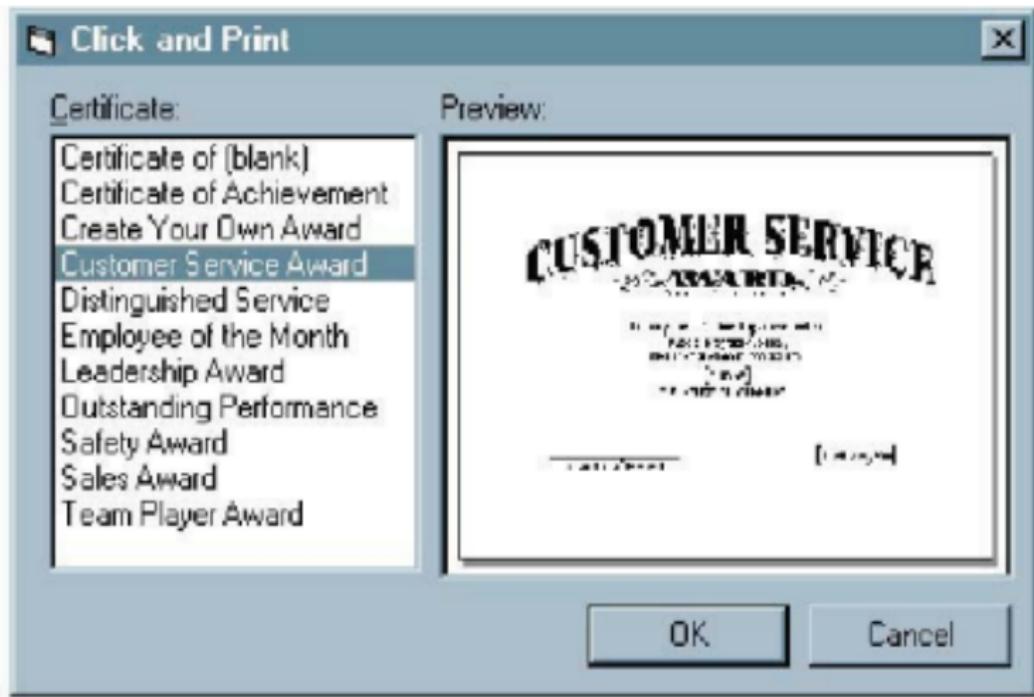
Source: Interface Hall of Shame

## User Interface Hall of Shame



Source: Interface Hall of Shame

## The Example Redesigned

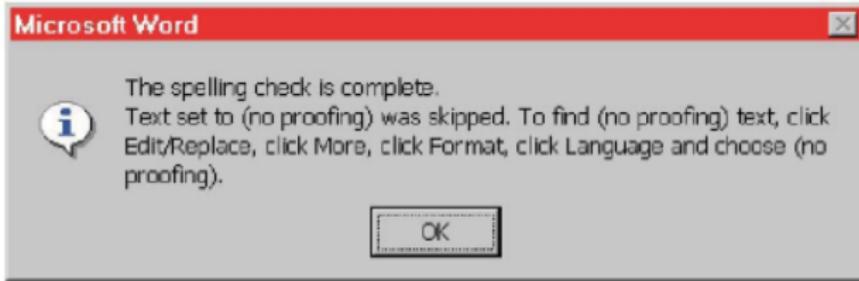


Source: Interface Hall of Shame

## Learnability

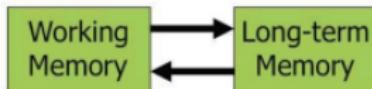
---

# People Don't Learn Instantly



Source: Interface Hall of Shame

# Memory



## ■ Working memory

- Small:  $7 \pm 2$  "chunks"
- Short-lived  $\sim 10$  sec

## ■ Long-Term memory

- Practically infinite in size and duration
- Elaborative rehearsal transfers chunks to long-term memory

## Recognition vs. recall

- Recognition: remembering with the help of a visible cue
  - aka "Knowledge in the world"
- Recall: remembering with no help
  - aka "Knowledge in the head"

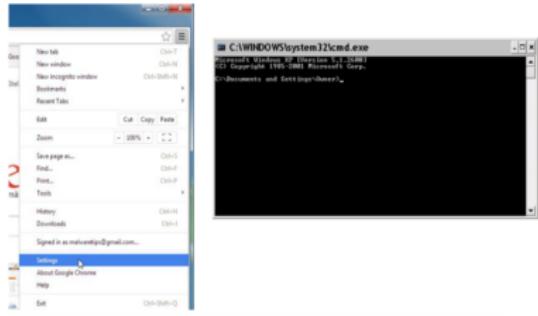
Recognition is much easier

### Rule

Promote recognition over recall in your UIs.

### ■ Example:

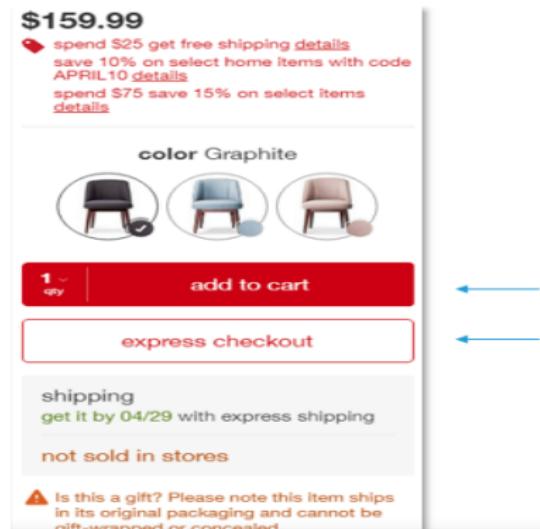
- 1 Command-line editor : You have to recall the name of a feature  
→ Recall is difficult and error-prone.
- 2 GUI: You easily recognize the feature  
→ Interfaces that promote recognition give users extra help in remembering information.



# Affordances

**Affordance:** a perceived signal or clue that an object may be used to perform a particular action

- Signaled by language or an object's physical appearance
- Any human being could guess at how to interact with the element, even if they have never been exposed to the interface before

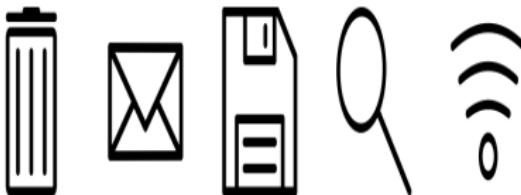


The screenshot shows a product page for a chair. At the top right, there is a promotional banner with offers like '\$159.99', 'spend \$25 get free shipping details', 'save 10% on select home items with code APRIL10 details', and 'spend \$75 save 15% on select items details'. Below the banner, there are three circular thumbnails showing the chair in 'color Graphite' (black), 'Graphite' (blue), and 'Graphite' (brown). To the right of these thumbnails are two blue arrows pointing left. Below the thumbnails is a red 'add to cart' button and a white 'express checkout' button with a red border. To the left of the 'add to cart' button is a dropdown menu showing '1' and 'qty'. Below these buttons is a green 'shipping' section with the text 'get it by 04/29 with express shipping'. Underneath that is a grey 'not sold in stores' section. At the bottom, there is a warning message in a grey box with a yellow warning icon: '⚠ Is this a gift? Please note this item ships in its original packaging and cannot be gift-wrapped or concealed.'

# Metaphors

Metaphors create a sense of familiarity, can draw the attention of users

- Strong metaphorical affordances:



- Weak metaphorical affordances:



Taxi?  
Travel?  
Uber?  
Directions?

Dashboard?  
Timer?  
Fuel?  
Clock?

Target?  
Goal?  
Aim?  
Dartboard?

Color?  
Overlap?  
Hazard?  
CMYK?

# Consistency & Standards



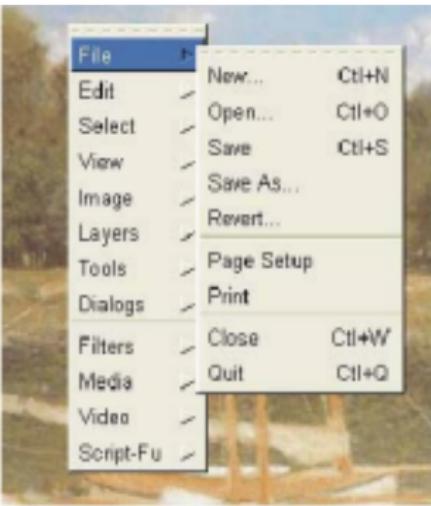
- Is everything working the same
- All terms should be consistent
- The interface should act the same across the product
- **example:** Shopping cart feature
- People are familiar with this feature!

Jakobâ€™s Law: "people spend most of their time on sites other than yours"

The screenshot shows the Jumia website homepage in Arabic. At the top, there is a navigation bar with a search bar containing " Cherchez un produit, une marque ou une catégorie", a "RECHERCHER" button, a "Se connecter" link, and an "Aide" link. A red circle highlights the "Panier" (Cart) icon. The main banner features the text "FATEZ VOS COURSES, DE CHEZ VOUS" and "JUMIA 40%". To the left is a sidebar with categories like "Sécurité", "Superroute", "Maison d'business", "Sport & Beauté", "Boutiques Officielles", "Téléphone & Tablette", "Informatique", "Techniques", "Mode", "Jeux vidéos & Consol", "Articles de sport", and "Autres catégories". To the right, there are promotional boxes for "Besoin d'aide?", "Contactez-nous", "Retour facile", "Remboursement rapide", "Produits authentiques et de qualité", and "COMMENT SE PROTÉGÉR ?". At the bottom, there are links for "Mes offres", "Hygiène et santé", "Abonnement Jumia", and "Jumia APP".

# Consistency of Layout

Consistency: Menubars appear at the top of the screen



© The GIMP Team. All rights reserved

# Match between the system and the real world

## ■ The system should:

- Speak the user's language: with words, phrases, and concepts familiar to the user
- Follow real-world conventions
- Example: Digital book

9:40 AM

THINKING, FAST AND SLOW

events. The word is, of course, *knew*. Some people thought well in advance that there would be a crisis, but they did not know it. They now say they knew it because the crisis did in fact happen. This is a misuse of an important concept. In everyday language, we apply the word *know* only when what was known is true and can be shown to be true. We can know something only if it is both true and knowable. But the people who thought there would be a crisis (and there are fewer of them than now remember thinking it) could not conclusively show it at the time. Many intelligent and well-informed people were keenly interested in the future of the economy and did not believe a catastrophe was imminent; I infer from this fact that the crisis was not knowable. What is perverse about the use of *know* in this context is not that some individuals get credit for prescience that they do not deserve. It is that the language implies that the world is more knowable than it is. It helps perpetuate a pernicious illusion.

The core of the illusion is that we believe we understand the past, which implies that the future also should be knowable, but in fact we understand the past less than we believe we do. *Know* is not the only word that fosters this illusion. In common usage, the words *intuition* and *premonition* also are reserved for past thoughts that turned out to be true. The statement "I had a premonition that the marriage would not last, but I was wrong" sounds odd, as does any sentence about an intuition that turned out to be false. To think clearly about the future, we need to clean up the language that we use in labeling the beliefs we had in the past.

8 more left in chapter

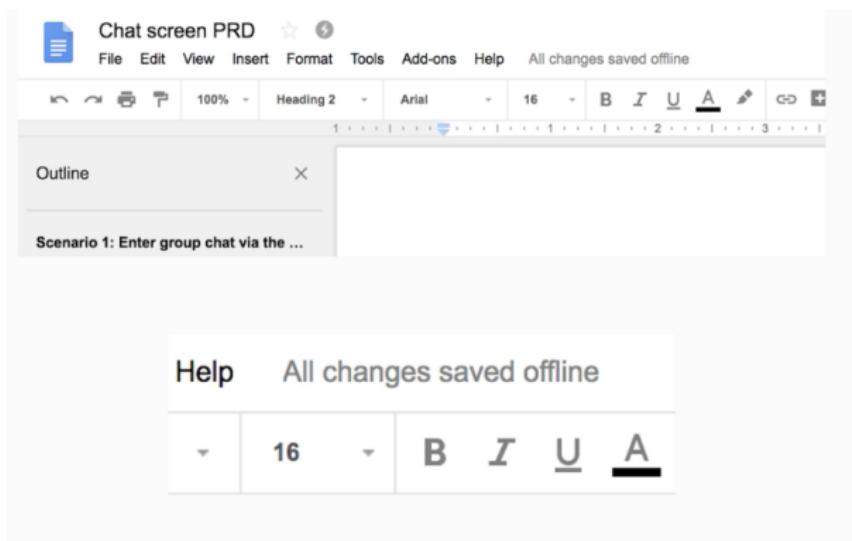
37%

## Visibility

---

## Visibility of System Status

- The System should keep users informed about what is going on



- Understanding a system's current state is all about allowing users to feel in control.

# Visible Navigation State

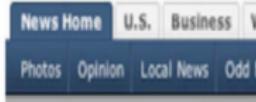
## ■ Breadcrumbs

[Travel](#) > [Guides](#) > North America

## ■ Pagination

Results Page:  
[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)  [Next](#)

## ■ Tabs



# Give Good Information Scent

- A link should smell like the content it leads to



© Microsoft. All rights reserved.

# Feedback

- Actions should have immediately visible effects
- Example: push button



- **Other Ex:** Elevator button, battery life, Wi-Fi connection indicator

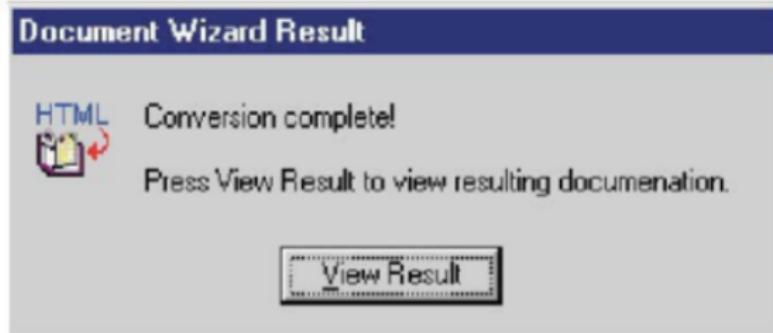
## Response Time

- < 0.1 s: seems instantaneous
- 0.1-1 s: user notices the delay
- 1-5 s: display busy indicator
- > 1-5 s: display progress bar



## Unnecessary Feedback

- Feedback is important, but don't overdo it!



**Source: Interface Hall of Shame**  
© Interface Hall of Shame. All rights reserved.

## Error Prevention

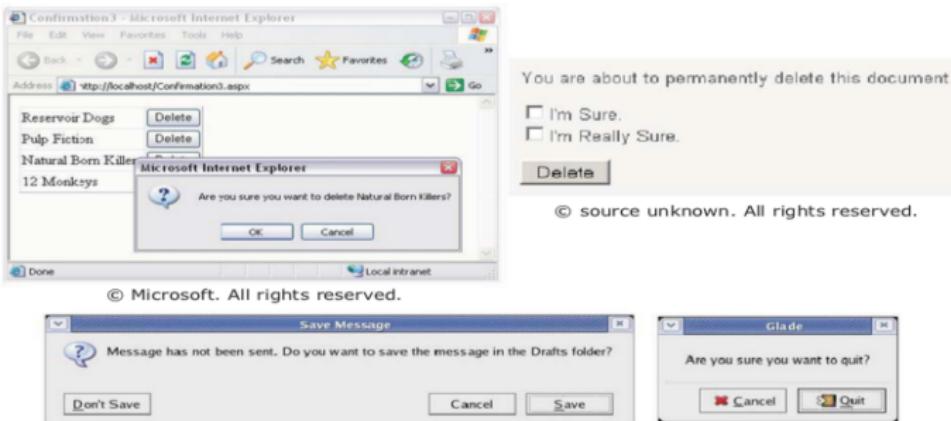
---

# Error Prevention

**Rule :** In your design, find out ways that you can stop errors!

## Example:

- 1 "reply all" button → That is not user error, it is the **design error** !
- 2 Are you sure you want to transfer all of your money to this other account?  
→ Prevent an accidental transfer
- 3 Confirmation Dialog



# Error reporting, diagnosis and recovery

- Inform users when an error has occurred

Ex: Error message: like red text or warning sign

## Shipping address

\*Required

### First name\*

Jess

### ▲ Last name\*

Enter the last name

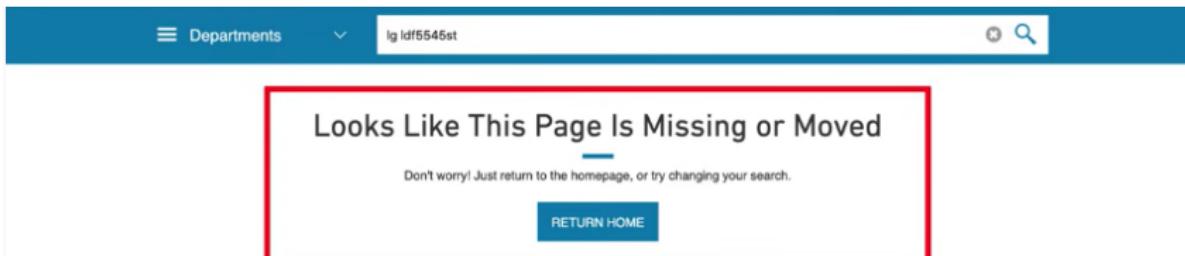
Ship to a Nordstrom Store

### ▲ Address\*

Enter the shipping address

# Error reporting, diagnosis and recovery

- Tell users what went wrong



The screenshot shows a search results page from a university website. At the top, there is a navigation bar with links for 'Departments' and a search bar containing the query 'lg ldf5545st'. Below the search bar, a large red rectangular box highlights an error message: 'Looks Like This Page Is Missing or Moved'. Underneath this message, a smaller text says 'Don't worry! Just return to the homepage, or try changing your search.' A blue 'RETURN HOME' button is located below the error message. The rest of the page content is visible but not highlighted.

## Connect With Us



## Sign Up for Email

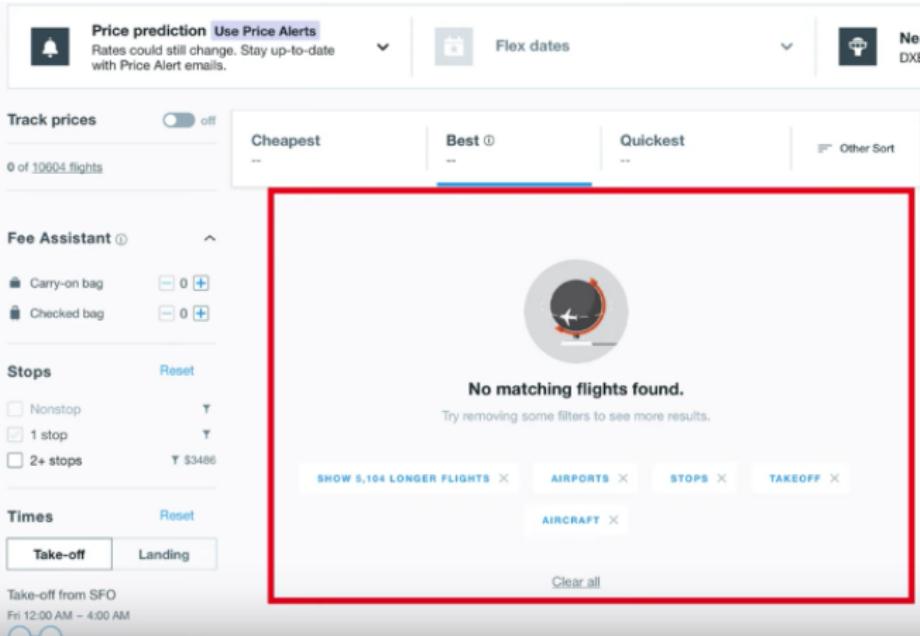
 Be the first to know: Sign up for exclusive offers, tips and more. [Sign Up >](#)

## Download Our Apps

 Learn more and download our FREE apps. [Discover >](#)

# Error reporting, diagnosis and recovery

## Offer users a solution



Price prediction [Use Price Alerts](#)  
Rates could still change. Stay up-to-date with Price Alert emails.

Flex dates

Nearby airports  
DXB has the cheapest flights in the area

Track prices  off  
0 of 10004 flights

Cheapest Best Quickest Other Sort

Fee Assistant

- Carry-on bag 0
- Checked bag 0

Stops Reset

- Nonstop
- 1 stop
- 2+ stops \$3486

Times Reset

Take-off Landing

Take-off from SFO Fri 12:00 AM – 4:00 AM

No matching flights found.  
Try removing some filters to see more results.

SHOW 5,104 LONGER FLIGHTS AIRPORTS STOPS TAKEOFF AIRCRAFT Clear all

REWARDED THE UNITED EXPLORER CARD EARN 40,000 BONUS MILES LEARN MORE Compare vs. KAYAK Emirates flyus.com priceline justfly.com cheapOair Expedia

# Be Precise and Comprehensible

## ■ Be precise

- "File missing or wrong format"
- "Line too long"
- "Name contains bad characters"

## ■ Restart user's input

- Not "Cannot open file", but "Cannot open file named paper.doc"

## ■ Speak the user's language

- Not "FileNotFoundException"
- Hide technical details (like a stack trace) until requested

## Suggested Reasons and Solutions

- Give **constructive** help
  - Why error occurred and how to fix it



## Be Polite

- Be polite and nonblaming



© Apple, Inc. All rights reserved.

- Avoid loaded words

- Fatal, illegal, aborted, terminated



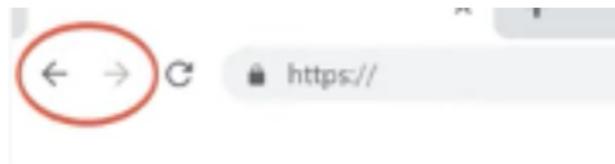
© Autodesk, Inc. All rights reserved.

## User Control

---

# User Control & Freedom

- Can the user recover from a mistake
- Can they 'undo' easily action?



Users often choose system functions by mistake and need a clearly marked "Emergency exit" to leave that unwanted state.

## Create New

Imagine a screen where new somethings are created. This is just an example to explain the need to include methods to cancel, so this is fake intro text. But imagine the possibilities.

Name

Something

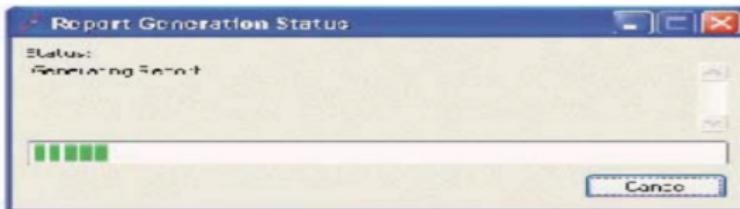
Something else

Cancel

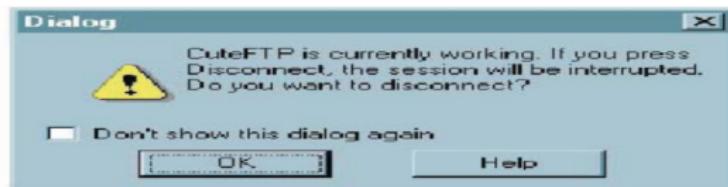
Create It!

## Clearly Marked Exists

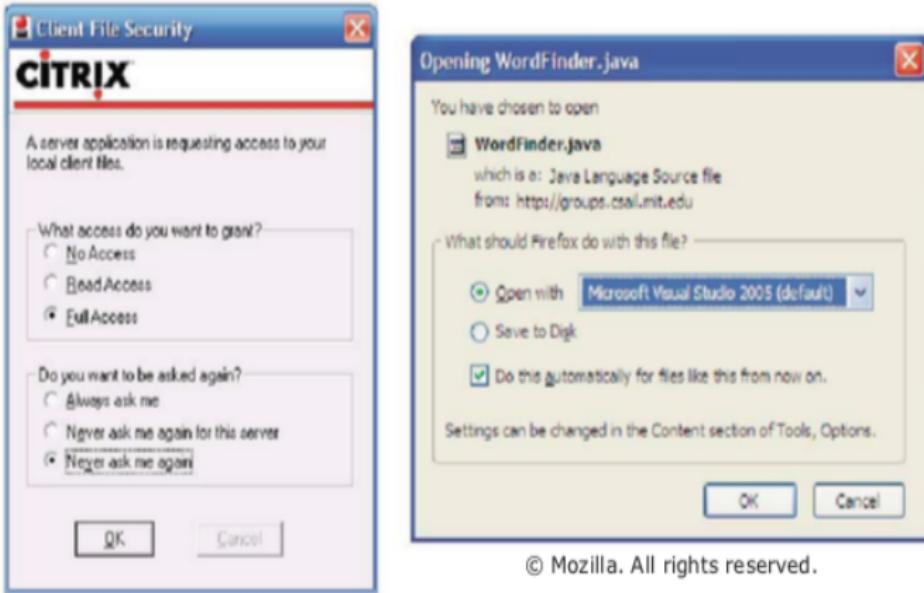
- Long operations should be cancelable



- All dialogs should have a cancel button



# Never Ask Me Again



© Mozilla. All rights reserved.

© Citrix Systems, Inc. All rights reserved.

## User Control Over Data

- Data entered by the user should be editable by the user
- UI should give the power to:
  - Create a data item
  - Read it
  - Update it
  - Delete it

## Efficiency

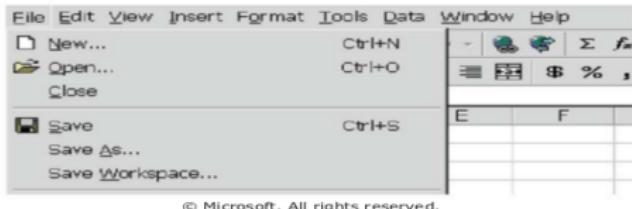
---

## Flexibility & Efficiency

- Flexibility helps people be more efficient.
  - Keyboard commands
  - Menu Accelerator
  - Aggregating Questions
  - Use Defaults and History
  - Anticipation
- Highly usable systems are flexible enough to be efficient for experts and friendly to newbies.

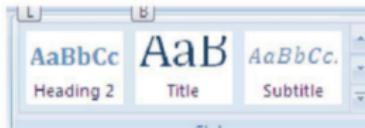
# Keyboard Shortcuts & Command Aggregates

- Keyboard commands
- Example: ctrl+C
- Menu Accelerator: An option or action that speeds up an interaction or process.



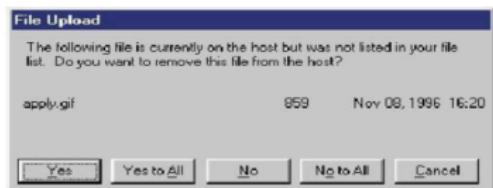
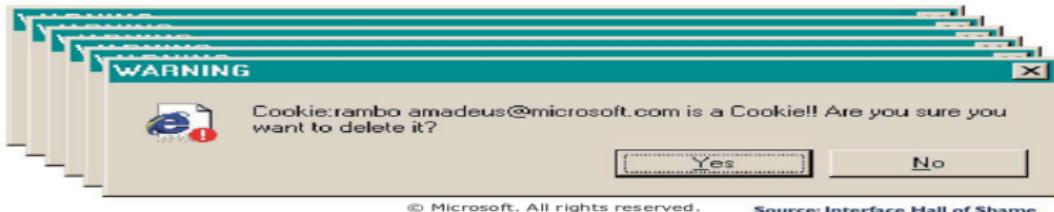
© Microsoft. All rights reserved.

- Command Aggregates
  - Styles
  - Scripts
  - Bookmarks

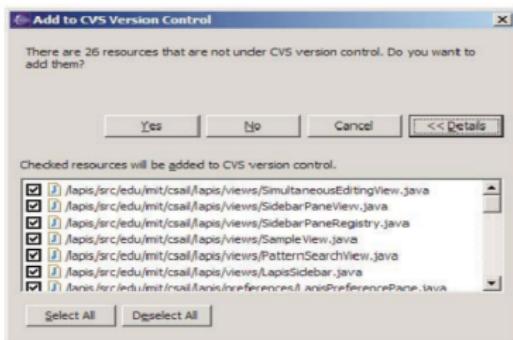


© Microsoft. All rights reserved.

# Aggregating Questions



© Microsoft. All rights reserved.



© The Eclipse Foundation. All rights reserved.

# Use Defaults and History

## ■ Use defaults

- Initially, most likely entry
- After use, previous entry

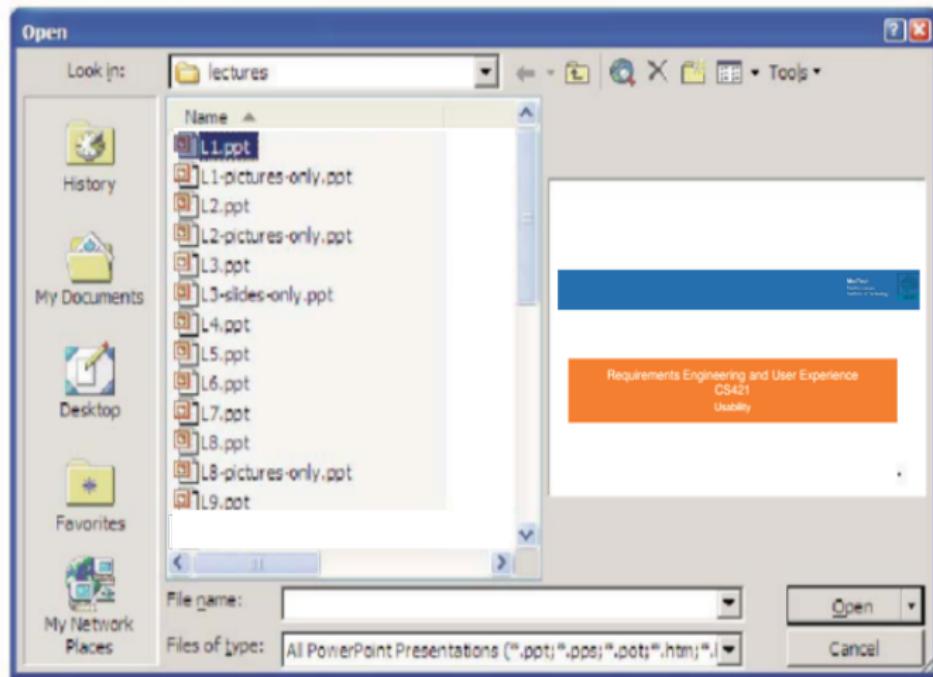
## ■ Keep History

```
1 VOLVO.DOC  
2 C\DOCUMENT\CLERICAL\RESUME.DOC  
3 C\DOCUMENT\CLERICAL\BUSCARD.DOC  
4 C\DOCUMENT\CONTACTS.DOC
```

Exit

## ■ Offer autocompletion

# Anticipation

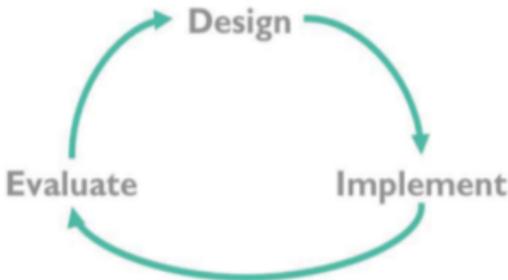


Design

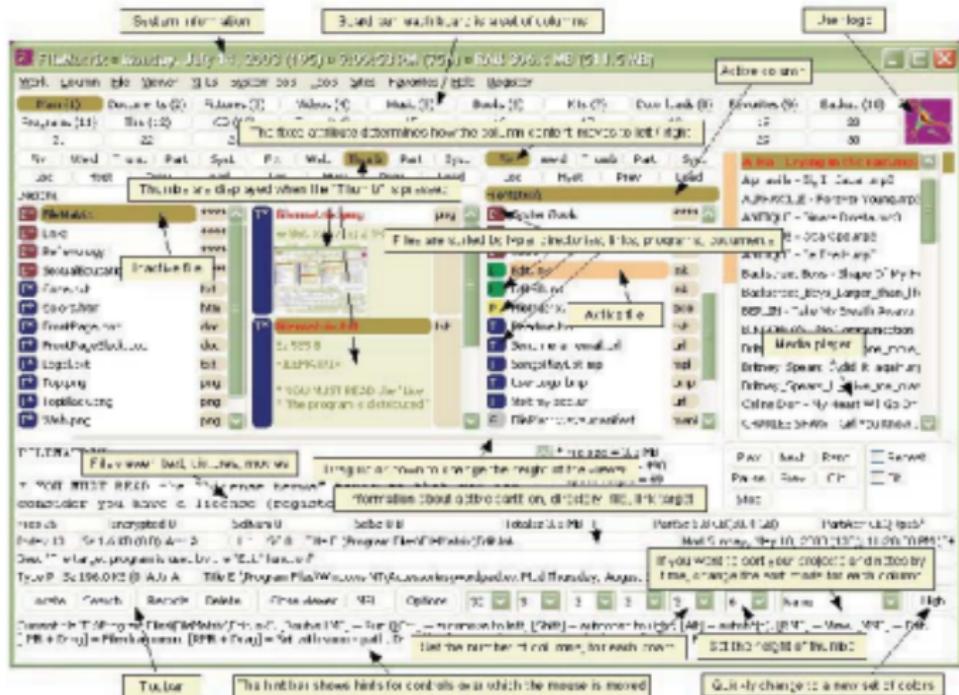
---

# User Centered Design Process

- Iterative, prototype-driven
  - Early iterations use cheap prototypes
  - More iterations generally means better UI
  - Early prototypes can detect usability problems
- Early focus on users and tasks
- Constant evaluation
  - Users are involved in every iteration
  - Only mature iterations are seen by the world



# Aesthetic & Minimalistic Design



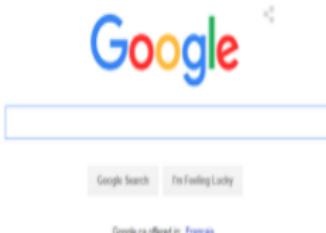
© FileMatrix. All rights reserved.

## Aesthetic & Minimalistic Design

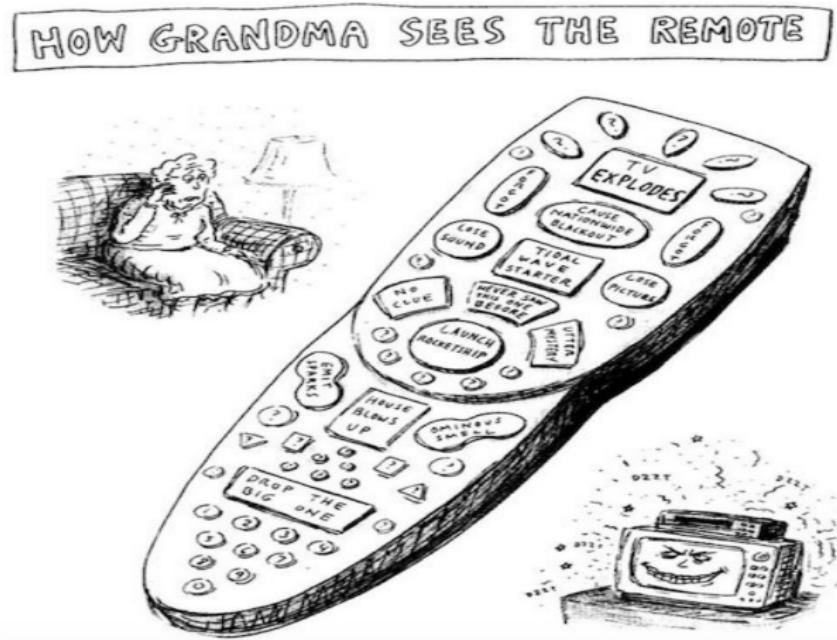
- Focus on the essential
- You should only have elements that support the primary goals
- You shouldn't have elements that are there to look pretty or to take up space.

---

Get Images  0 



## Aesthetic & Minimalistic Design



## Aesthetic & Minimalistic Design

Technique for Simplicity: Reduction

- Remove inessential elements



## Aesthetic & Minimalistic Design



## Heuristic Evaluation

---

# Usability Guidelines ("Heuristics")

- Plenty to choose from
  - Nielson's 10 principles
  - Norman's rules from Design of Everyday Things
  - Tognazzini's 16 principles
  - Shneiderman's 8 golden rules
- Help designers choose design alternatives
- Help evaluators find problems in interfaces ("Heuristic evaluation")

# Nielson Heuristics



- 1 Match the real world (L)
- 2 Consistency & standards (L)
- 3 Help & documentation (L)
- 4 User control & freedom (UC)
- 5 Visibility of system status (V)
- 6 Flexibility & efficiency (EF)
- 7 Error prevention (ER)
- 8 Recognition, not recall (ER)
- 9 Error reporting, diagnosis, and recovery (ER)
- 10 Aesthetic & minimalistic design (GD)

## Norman Principles



- 1 Affordances (L)
- 2 Natural mapping (L)
- 3 Visibility (V)
- 4 Feedback (V)

# Tog's First Principles

- 1 Anticipation (EF)
- 2 Autonomy (UC)
- 3 Color blindness (GD)
- 4 Consistency (L)
- 5 Defaults (EF)
- 6 Efficiency (EF)
- 7 Explorable interfaces (UC)
- 8 Fitts's Law (EF)
- 9 Human interface objects (L)
- 10 Latency reduction (V)
- 11 Learnability (L)
- 12 Metaphors (L)
- 13 Protect users' work (ER)
- 14 Readability (GD)
- 15 Track state (EF)
- 16 Visible navigation (V)

## Shneiderman's 8 Golden Rules

- 1 Consistency (L)
- 2 Shortcuts (EF)
- 3 Feedback (V)
- 4 Dialog closure (V)
- 5 Simple error handling (ER)
- 6 Reversible actions (UC)
- 7 Put user in control (UC)
- 8 Reduce short-term memory load (ER)

# Heuristic Evaluation



- Performed by an expert

- **Steps:**

- Inspect UI thoroughly
- Compare UI against heuristics
- List usability problems
- Explain & justify each problem with heuristics

# How To DO Heuristic Evaluation

- Justify every problem with a heuristic
  - "Too many choices on the home page (Aesthetic & Minimalistic Design")
  - Can't just say "I don't like the colors"
- List every problem
  - Even if an interface element has multiple problems
- Go through the interface at least twice
  - Once to get the feel of the system
  - Again to focus on particular interface elements
- Don't have to limit to the 10 Nielson heuristics

# How To DO Heuristic Evaluation

**Shopping Cart Contents**

Welcome, Ben Bitdiddle.

You have 2 item(s) in your shopping cart.  
 To remove an item, check "Remove" box & click "Recalculate".  
 Shipping Calculator below.

\*There is a problem with your order.\*

Product	Description	Quantity	Unit Price	Ext Price
	323022 Pinball Clean Plus Version 4.0 Retail ***(Free 2nd Day)*** <input type="checkbox"/> Remove	<input type="text" value="1"/>	\$61.00	\$61.00
	80098-21 Corsair VS1GBKIT400 1GB Kit DDR400 PC3200 Value Select Memory Retail (out of stock) <input type="checkbox"/> Remove Hardware	<input type="text" value="1"/>	\$179.00	\$179.00

Subtotal: \$240.00

For more information about tax, please [click here](#).

[Recalculate](#) [Clear Cart](#)

**Shipping Promotion details.** Please read.

\*Note: Discount will be applied during check out\*

Coupon Code:  [Apply](#)

Ship to Zip Code:  [Calculate Shipping Charge](#)

Have not made up your mind? Save all the items in your shopping cart!

Cart Title:  [Save shopping cart](#)

Return to old shopping cart.

Cart Name:  [Load shopping cart](#)

**Check Out**

## Heuristic Evaluation is Not User Testing

- Evaluator is not the user either
  - Maybe closer to being a typical user than you are, though
- Analogy: code inspection vs. testing
- HE finds problems that UT often misses
  - Inconsistent fonts
  - Fitts's Law problems
- But UT is the gold standard for usability

# Hints for Better Heuristic Evaluation

- Use multiple evaluators
  - Different evaluators find different problems
  - The more the better, but diminishing returns
  - Nielson recommends 3-5 evaluators
- Alternate heuristic evaluation with user testing
  - Each method find different problems
  - Heuristic evaluation is cheaper
- It's OK for observer to help evaluator
  - As long as the problem has already been noted
  - This wouldn't be OK in a user test

# Formal Evaluation Process

## 1 Training

- Meeting for design team & evaluators
- Introduce application
- Explain user population, domain, scenarios

## 2 Evaluation

- Evaluators work separately
- Generate written report, or oral comments recorded by an observer
- Focus on generating problems, not on ranking their severity yet
- 1-2 hours per evaluator

## 3 Severity Rating

- Evaluators prioritize all problems found (not just their own)
- Take the mean of the evaluators' ratings

## 4 Debriefing

- Evaluators & design team discuss results, brainstorm solutions

# Severity Ratings

- Contributing factors
  - Frequency: how common?
  - Impact: how hard to overcome?
  - Persistence: how often to overcome?
- Severity scale
  - 1 Cosmetic: need not be fixed
  - 2 Minor: needs fixing but low priority
  - 3 Major: needs fixing and high priority
  - 4 Catastrophic: imperative to fix

# Evaluating Prototypes

- Heuristic evaluation works on:
  - Sketches
  - Paper prototypes
  - Buggy implementations
- "Missing element" problems are harder to find on sketches
  - Because you're not actually using the interfaces, you aren't blocked by feature's absence
  - Look harder for them

# Writing Good Heuristic Evaluations

- Heuristic evaluations must communicate well to developers and managers
- Include positive comments as well as criticisms
  - "Good Toolbar icons are simple, with good contrast and few colors (minimalist design)"
- Be tactful
  - Not: "the menu organization is a complete mess"
  - Better: "menus are not organized by function"
- Be specific
  - Not: "text is unreadable"
  - Better: "text is too small, and has poor contrast (black text on dark green background)"

## Suggested Report Format

### ■ What to include:

- Problem
- Heuristic
- Description
- Severity
- Recommendation (if any)
- Screenshot (if helpful)

#### 12. Severe: **User may close window without saving data (error prevention)**

If the user has made changes without saving, and then closes the window using the Close button, rather than File >> Exit, no confirmation dialog appears.

Recommendation: show a confirmation dialog or save automatically



© Microsoft. All rights reserved.

## Summary

---

# Usability Dimensions Vary In Importance

- Depends on the user
  - Novice users need learnability Infrequent users need memorability
  - Experts need efficiency
- But no user is uniformly novice or expert
  - Domain experience
  - Application experience
  - Feature experience

# Usability Is Only One Attribute of a System

- Software designers have a lot to worry about:
  - Functionality
  - Performance
  - Cost
  - Security
  - Usability
  - Size
  - Reliability
  - Standards
- Many design decisions involve tradeoffs among different attributes

# Summary



- Prefer knowledge in the world over knowledge in the head
- Visibility of actions, state, and feedback
- Prevent errors as much as possible
- Give user control
- Keep it simple
- Heuristic evaluation finds usability problems by inspection