



Maastricht University

Department of Advanced Computing Sciences

Human Computer Interaction

Usability principles and testing

Konstantia Zarkogianni

Summary of this lecture

- Usability basics
- Principles, guidelines, standards..
- What do we measure?
 - Metrics
 - Targets
- How do we measure?
 - Usability testing. From welcoming our user to analyzing data

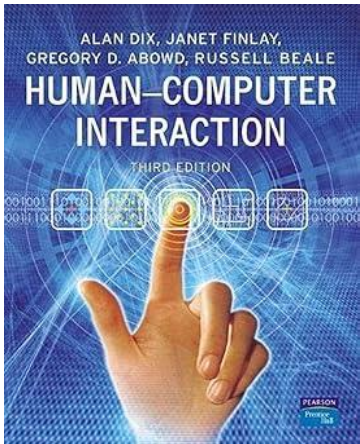
Check usability on the prototype

- Set aims according to user needs
- Define design principles and interaction style
 - Will there be browsing?
 - Screens? Introductory, central, basic screen for each functionality
- Use prototyping tools and start from the screens!
- Ask users' opinions on this prototype!

Usability directives

- Principles
 - Abstract design rules, High generality
- Standards
 - Concrete rules. Prestigious and, sometimes, of limited usability.
- Guidelines
 - Concrete rules. Not so prestigious
- Patterns (templates)
 - Easy to use when originality is not an issue
 - Highly re-usable

Principles



Principles to support usability

Learnability

- Predictability
- Synthesizability
- Familiarity
- Consistency
- Generalizability

Flexibility

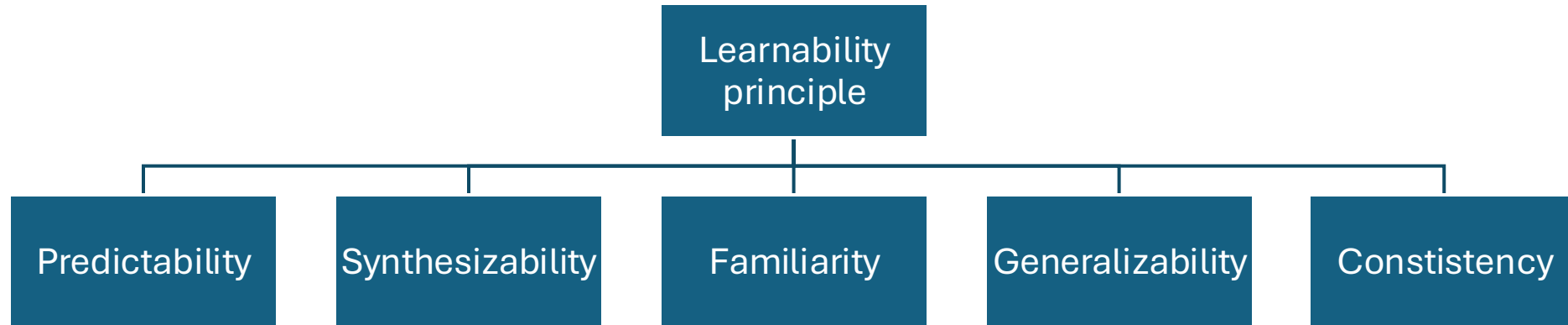
- Dialogue initiative
- Multithreading
- Task Migrability
- Substitutivity
- Customizability

Robustness

- Observability
- Recoverability
- Responsiveness
- Task conformance

Learnability

The ease with which new users can begin effective interaction and achieve maximal performance



Learnability

Predictability

- System behaviour is observably deterministic:
 - *Non-deterministic delays should be avoided*
 - *Operation effect determinable by interaction history*

Operation visibility

- *User actions should be matched by a response*
- *Affordance/logical constraints should be used to indicate available actions*

The diagram compares two UI design styles. On the left, under 'Navigation Menu Styles', it shows a traditional menu with links 'Home Services About Us Contact Us' and a 'Hamburger' menu represented by three horizontal lines. On the right, under 'Button Styles', it shows a 'Blocked button' with the text 'Join Us' inside a white box with a green border, and a 'Text & icon' button with a play icon and the text 'Join Us'. Both sections are separated by a 'VS' label.

Navigation Menu Styles

1. Traditional Menu
2. “Hamburger” Menu

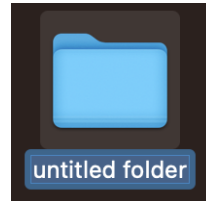
Button Styles

1. Blocked button
2. Text & icon

Learnability

Synthesizability

- assessing the effect of actions on the system state
 - **immediate honesty:** mainly in GUIs. For example, when creating a new folder the user can immediately see a folder icon named “New folder”



- **eventual honesty:** The user needs to perform some actions to confirm the change. For example, in command-driven interfaces when a new folder is created, the user has to remember the destination directory and verify the creation after running some command

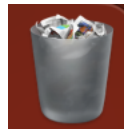
```
konstantia — -bash — 80x24
Last login: Sun Feb 11 16:49:34 on console

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) 192:~ konstantia$ mkdir "New Folder"
(base) 192:~ konstantia$ ls
Desktop                New Folder
Documents              OneDrive - Maastricht University
Downloads              Pictures
Library                Public
Movies                 Zotero
Music                  anaconda3
(base) 192:~ konstantia$
```

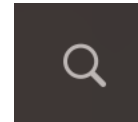

Learnability

Familiarity

- how prior knowledge applies to new system
- guessability; affordance; metaphors



Recycle Bin



Search

Learnability

Generalizability

- extending specific interaction knowledge to new situations
- example: drawing circles → drawing ellipses
- UI standards and guidelines assist/enforce generalizability
 - Applications should offer cut/copy/paste whenever possible

Learnability

Consistency

- likeness in input/output behaviour arising from similar situations or task objectives

Confirmation

Are you sure you want to delete this document?

Yes

Cancel

Confirmation

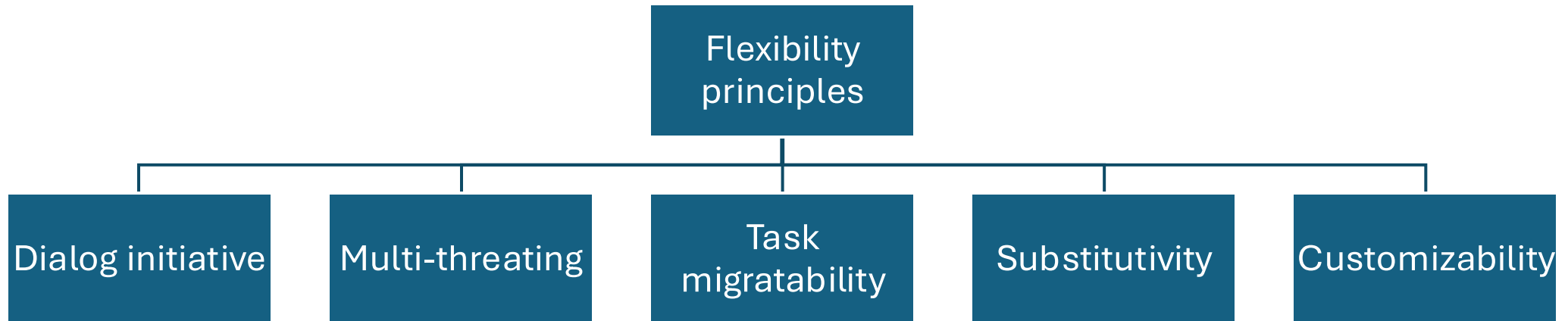
Are you sure you want to save this document?

Cancel

Yes

Flexibility

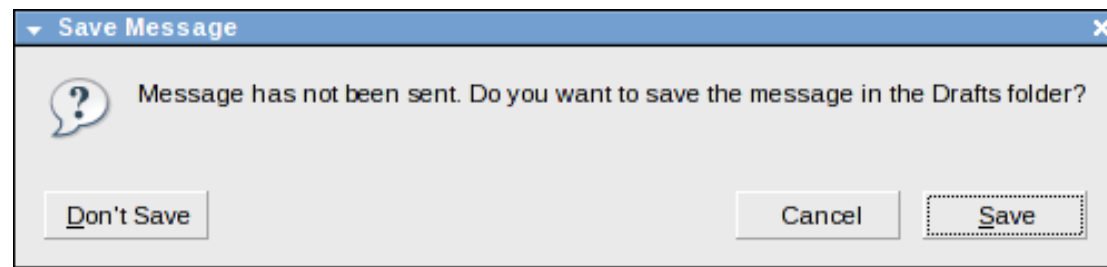
The multiplicity of ways the user and system exchange information



Flexibility

Dialogue initiative

- ▶ freedom from system imposed constraints on input dialogue
 - ▶ user should be able to abandon, suspend or resume tasks at any point
 - ▶ modal dialog boxes are system pre-emptive (initiated by the system)
 - ▶ direct manipulation is user pre-emptive (initiated by the user)
 - ▶ minimise system pre-emptive dialogue and
 - ▶ maximise user pre-emptive dialogue



Flexibility

Multithreading

- ▶ ability of system to support user interaction for more than one task at a time
 - ▶ Concurrent: Simultaneous execution of multiple tasks. For example, editing a document on a word processor while it also increments and displays the word count.

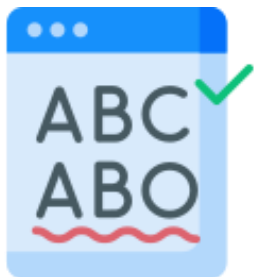
VS

- ▶ Interleaving: The dialog is restricted to a single task at a time but has a temporal gap between multiple tasks. For example, users can have multiple documents open on a word processor but can only work on one document at a time.
- ▶ Multimodality: For example, the user can copy a text using either the right-click option or the keyboard shortcut.

Flexibility

Task migratability

- ▶ passing responsibility for task execution between user and system
 - ▶ automate routine tasks, but don't fix function allocation



It is a waste of time for a user to manually check a very long document and correct. A spell checking facility in a word processing application can check words against its own computerised dictionary

Obvious
contribution from
AI

Flexibility

Substitutivity

- offers a user alternative ways of specifying input or viewing output.



Analog clock

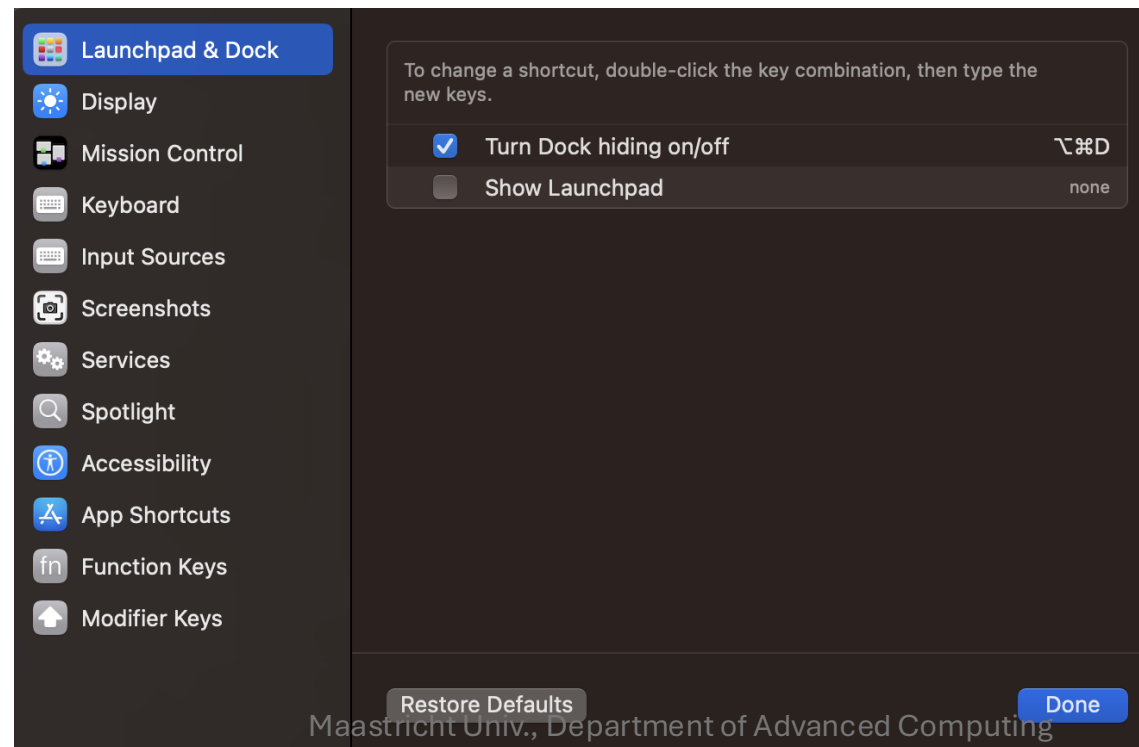


Digital clock

Flexibility

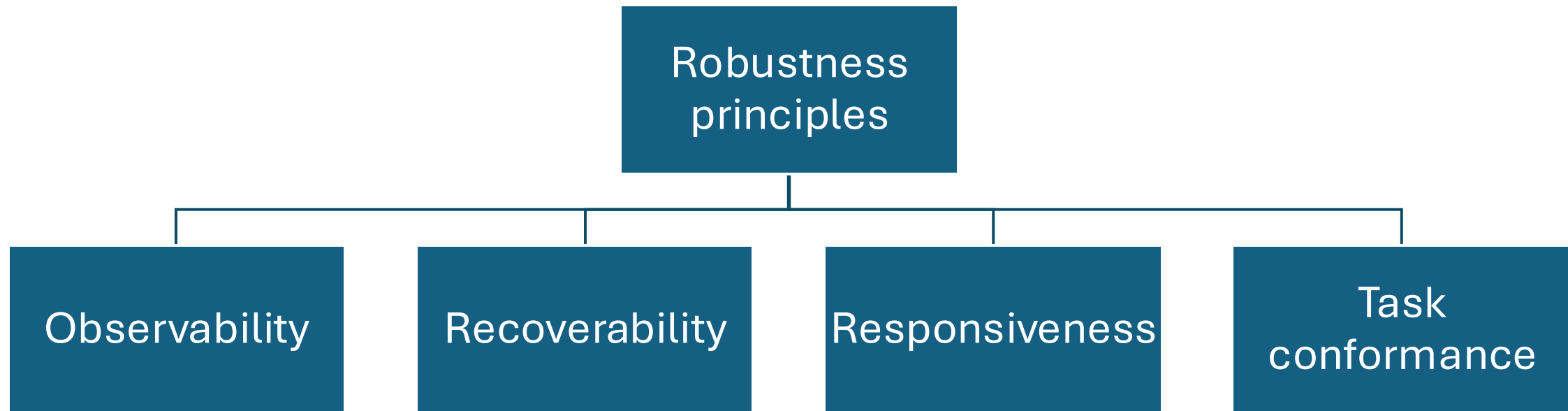
Customizability

- modifiability of the user interface by user or system
 - provide choice of methods; allow short-cuts; permit users to change features: deferred design



Robustness

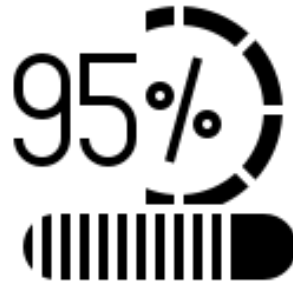
The level of support provided to the user in determining successful achievement and assessment of goal-directed behaviour



Robustness

Observability

- ▶ ability of user to evaluate the internal state of the system from its perceivable representation
- ▶ Browsability
 - **W**here am I? — immediate honesty wrt system state
 - **W**here am I going? — operation predictability
 - **W**here have I been? — synthesizability
 - **W**hat can I do now? — predictability



Robustness

Recoverability

- ▶ ability of user to take corrective action once an error has been recognized
- ▶ forward/backward recovery
- ▶ The system must be able to detect user errors and correct them or help the user in doing so
- ▶ Commensurate effort, which states that performing an action should be difficult if, once performed, reversing it would be difficult. For example, the delete option should be kept in a location that is difficult to reach by the user.

Robustness

Responsiveness

- how the user perceives the rate of communication with the system
- if delay is inevitable, provide reassurance: time affordances
- stability



Robustness

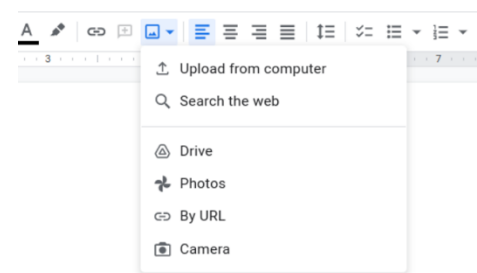
Task conformance

- degree to which system services support all of the user's tasks
 - task completeness: whether a system is capable of supporting the entire task that a user wishes to perform. Google Docs provides all of the operations that a user may potentially execute while creating a text document



- task adequacy: the required task should be supported in a way that the user wants and can understand.

There is no option for the user to select a photo from the drive



Images taken from
<https://www.educative.io/>

Standards

- International Institutes propose them, so that designers are conformed to them.
- They are usually result of intense and mature research
- They are based on prevailing technologies in the field
- ISO 9241-11: Guidance on Usability (1998)

ISO 9241-11 standard part

goal	Measurement of effectiveness	Measurement of efficiency	Measurement of satisfaction
Appropriateness for performing specific tasks	% of attained goals	Time for attaining a goal	Satisfaction on a (Likert) scale
Appropriate for non-experienced users	Number of functions used	Relation with other experienced users	Satisfaction to handle advanced functionalities
Easiness to learn	% of learnt functionalities	Time needed to learn	How easy learning looks
Error handling	% of errors corrected	Time needed for correction	How the procedure looked

Guidelines

- More descriptive and generic than standards
- Practically, they are usability principles applied during the last phase of designing
- A plethora of guidelines can be used

Guidelines collection

- Heuristic Nielsen rules
- Shneiderman golden rules
- Norman principles
- NASA
- Apple
- IBM
- GNOME
- Microsoft
- ...

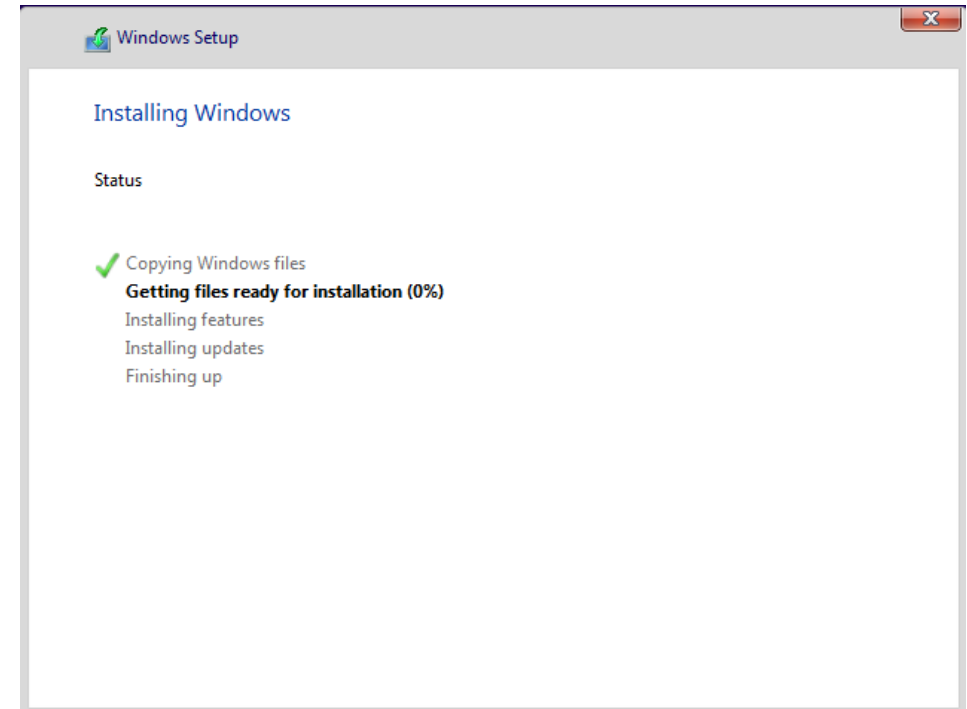
Nielsen's heuristics

- Visibility of system status.
- Match between system and real world.
- User control and freedom.
- Consistency and standards.
- Error prevention.
- Recognition rather than recall.
- Flexibility and efficiency of use.
- Aesthetic and minimalist design.
- Help users recognize, diagnose, recover from errors.
- Help and documentation.

Nielsen's heuristics 1

Visibility of system status

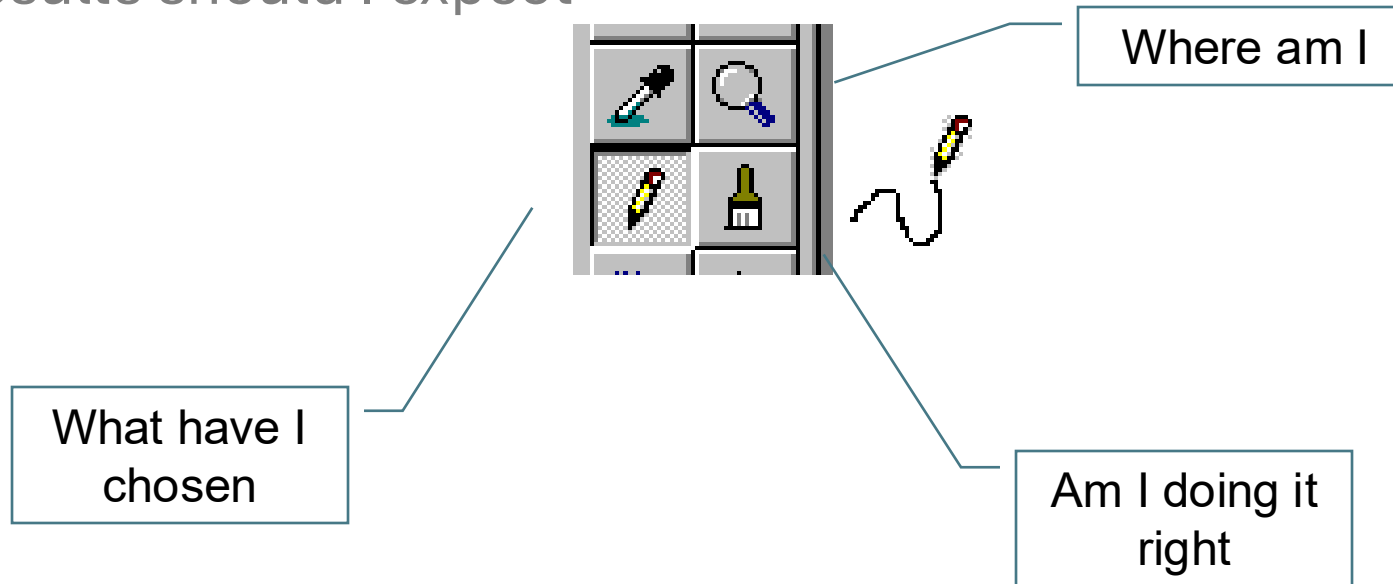
- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- Design dialogues that yield closure



Nielsen's heuristics 1

Visibility of system status

- What is the system doing
- How it interpreted my input
- What results should I expect



Nielsen's heuristics 1

Upload the file or [Cancel](#)



picnik



Fluffing clouds....



Your password has been emailed.

Theresa Neil sign in

Type new password:

Six-characters minimum; case sensitive

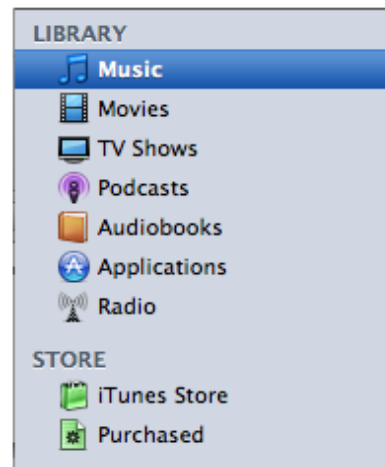
Password strength:

Strong

Nielsen's heuristics 2

Match between the system and the real world

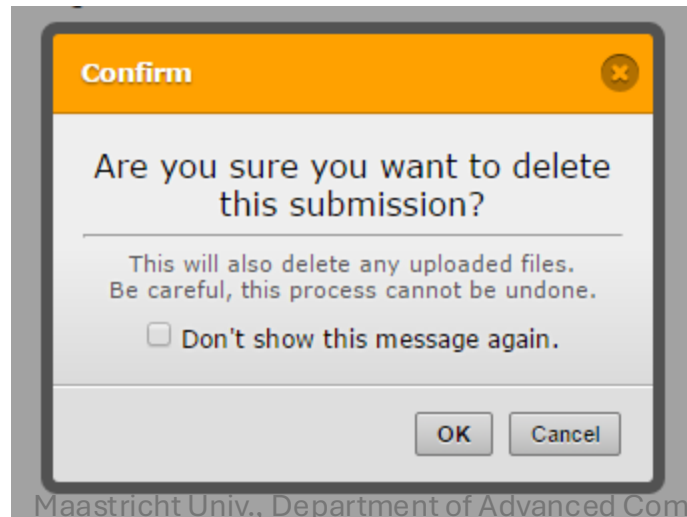
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.



Nielsen's heuristics 3

User control and freedom

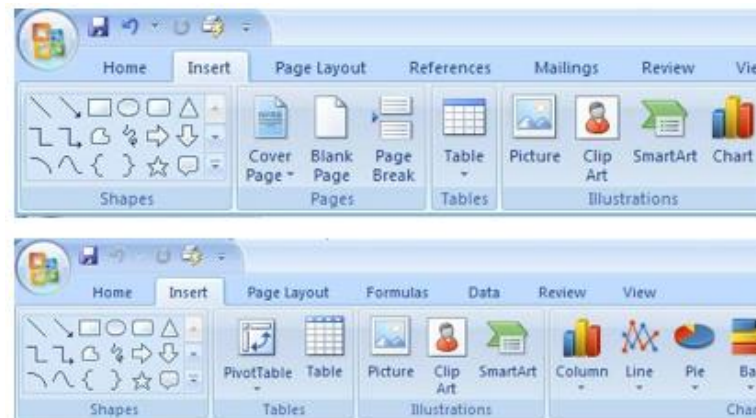
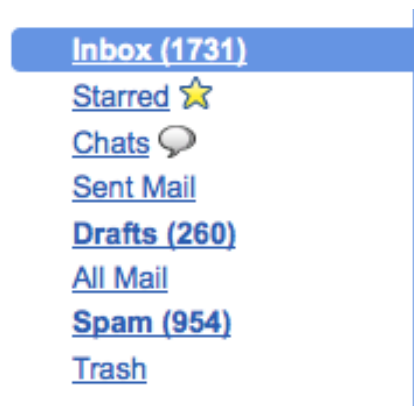
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. "Undo" is important here.



Nielsen's heuristics 4

Consistency and standards

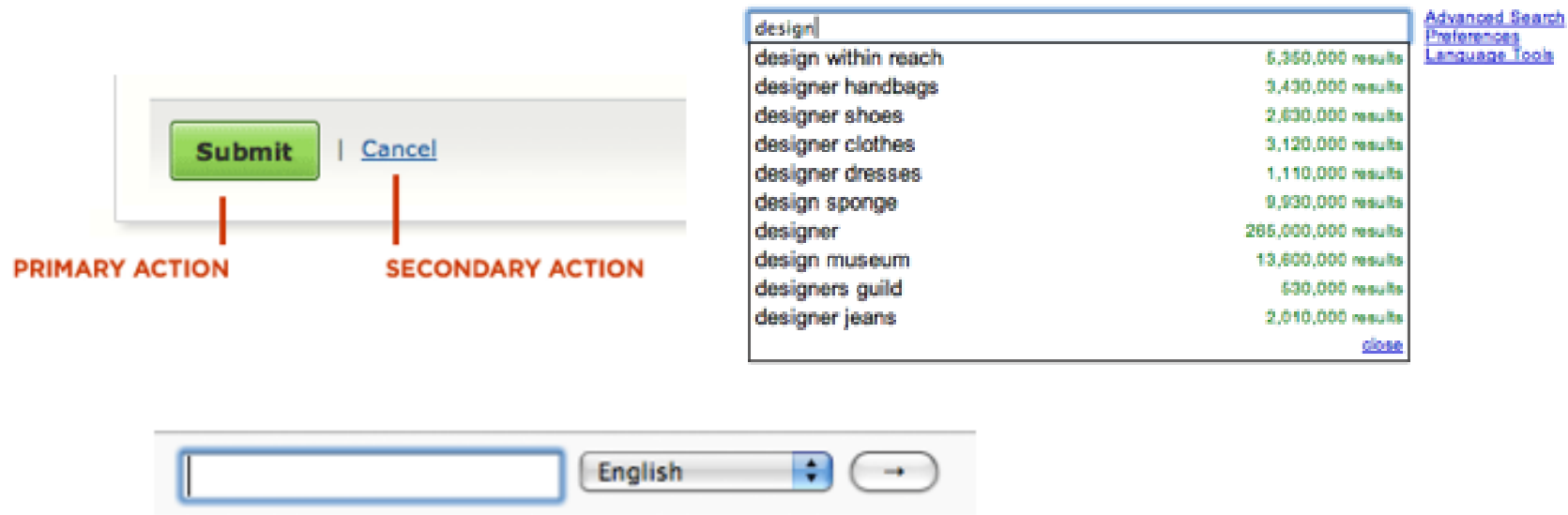
- Users should not have to wonder whether different words, situations, or actions mean the same thing.
- Follow platform conventions.
- Consistency across applications
- Re-use commands and layouts across different parts of the application



Nielsen's heuristics 5

Error prevention

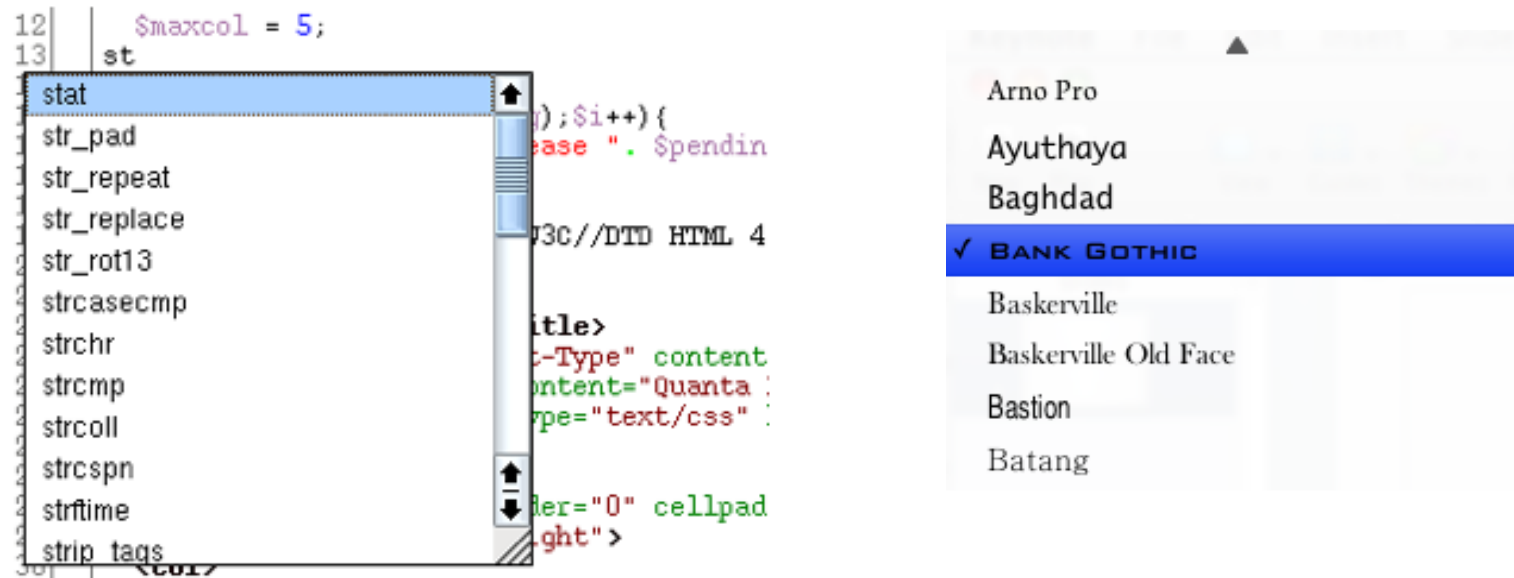
Even better than good error messages is a careful design which prevents a problem from occurring in the first place.



Nielsen's heuristics 6a

Recognition rather than Recall

- Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- People can typically hold 7+/-2 things in their short memory



Nielsen's heuristics 6b

Show the possibilities

Make objects, actions, and options visible. Don't hide them unless there's a reason.

I want to convert...using live currency rates

this amount

1

enter any amount

of this type of currency

CAD Canada Dollars

USD United States Dollars

EUR Euro

GBP United Kingdom Pounds

DEM Germany Deutsche Marks

scroll down to see more currencies

into this type of currency.

USD United States Dollars

EUR Euro

CAD Canada Dollars

GBP United Kingdom Pounds

DEM Germany Deutsche Marks

scroll down to see more currencies

Click Here to Perform Currency Conversion

Nielsen's heuristics 7

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.
- The user should drive the user interface and not the other way around (e.g a frequent action should not require starting always from the home screen)

Edit	View	Insert	Format
Can't Undo		⌘Z	
Repeat Save		⌘Y	
Cut		⌘H	
Copy		⌘C	
Paste		⌘V	
Paste Special...			
Paste as Hyperlink			
Clear			
Select All		⌘A	
Duplicate		⌘D	
Delete Slide			
Find...		⌘F	
Replace...		⌘H	
Go to Property...			
Object			

Nielsen's heuristics 8

Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

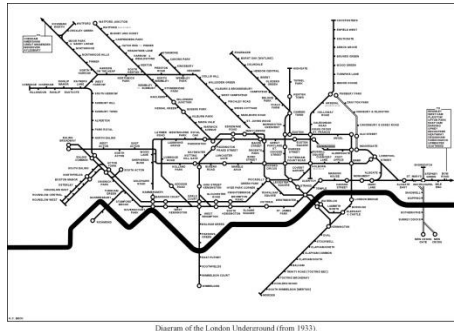


Diagram of the London Underground (from 1933).

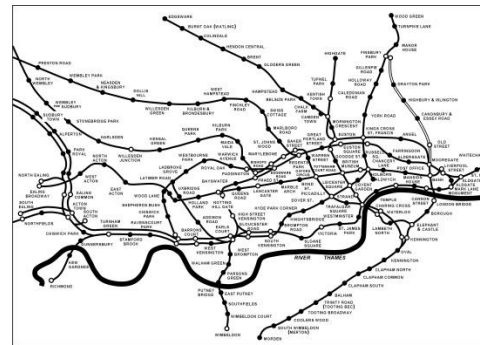
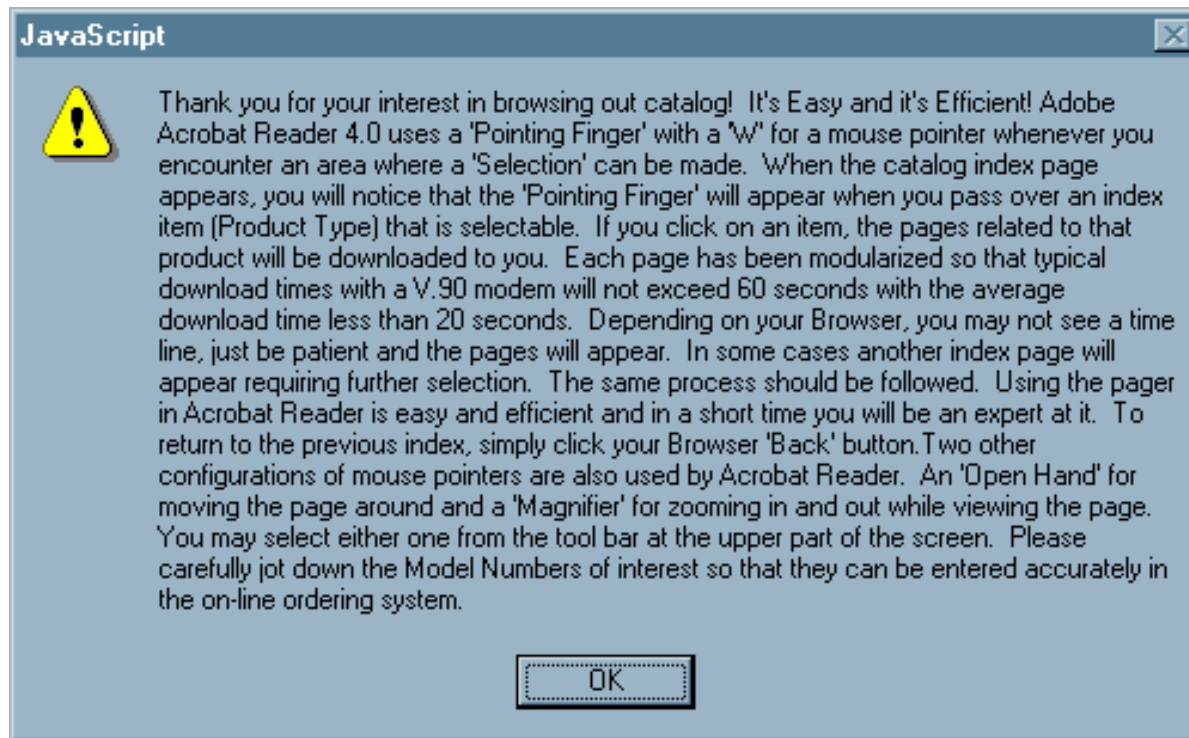


Figure 2-7: Geographical map of the London Underground (from 1932).

Nielsen's heuristics 8

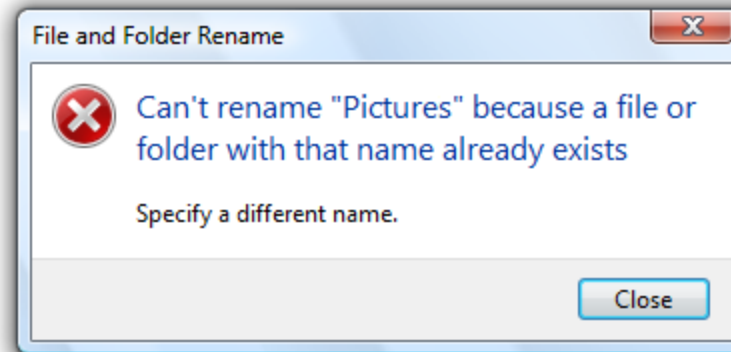
Aesthetic and minimalist design



Nielsen's heuristics 9

Help users recognise, diagnose and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem and constructively suggest a solution.



Nielsen's heuristics 9

Help users recognise, diagnose and recover from errors

- Errors are inevitable
 - prevention
 - Interface aspects and functions that can help avoid errors (e.g. numeric spinners vs free digit entry)
 - detection
 - Through feedback
 - correction
 - Through 'undo' command

Nielsen's heuristics 9

Help users recognise, diagnose and recover from errors

- Ignore
 - If non-allowed actions have no effect
 - Warning message is not obligatory
 - example
 - Drop a file in a not allowed space
- Auto-correction
 - Applies the most probable action
 - If applied correctly, it can lead to an increase in usability
 - if not, it can lead to mistakes that have negative impact on the trust in the interface
 - example
 - Text autocorrect

Or start a new account

Choose a username (no spaces)

bert

⚠ bert is already taken. Please choose a different username.

Choose a password

...

⚠ Passwords must be at least 6 characters and can only contain letters and numbers.

Retype password

Email address (must be real)

not an email

⚠ The email provided does not appear to be valid

☒ Send me occasional Digg updates.



Oh no!

It seems the page you were trying to find on my site isn't around anymore (or at least around here).

[Report it missing using my contact form](#) and I'll see what I can do about it.

Whilst your here why not check out my [articles listing](#) or [browse my blog](#)? You never know - you may just

Nielsen's heuristics 10

Help and documentation

- Not a substitute for a bad design!
- Discrete but ubiquitous
- Organized in layers
- Users do not often resort to help
 - **They prefer experimentation (exploratory learning)**
- Possibility to use a search tool
- In categories

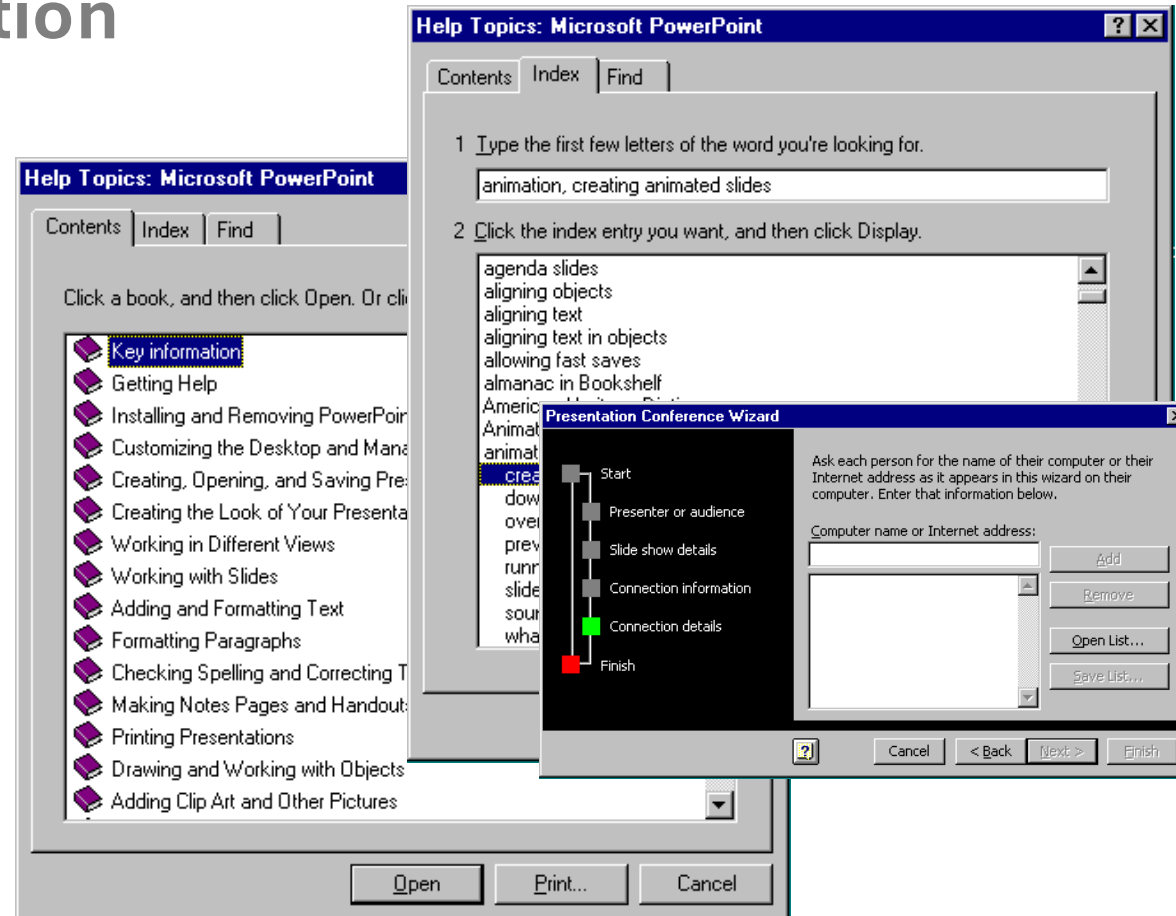
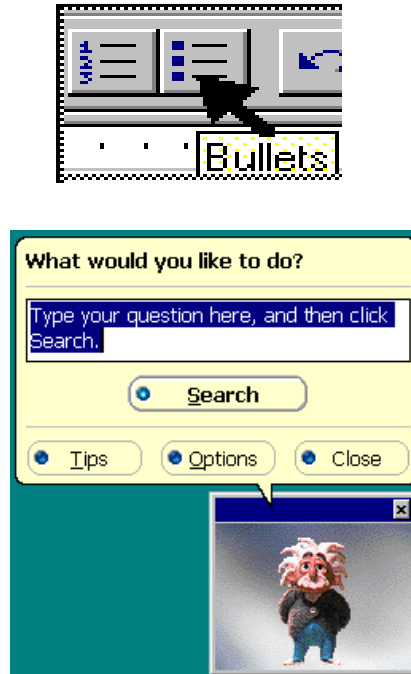
Nielsen's heuristics 10

Help and documentation

- Types of help
 - Local/Network-based (Official fora)
 - Suggestions
 - During interaction
 - Tutorials
 - Simple task description
 - Construction of an analytical tutorial. Use metaphors when possible.
 - Demo
 - Functioning examples/prototypes
 - Agents
 - Anthropomorphism

Nielsen's heuristics 10

Help and documentation



Patterns (templates)

- Reusability of knowledge
- Patterns focus on:
 - Practice, instead of theory
 - The use of successful exemplars
 - Familiarity
- They do little for
 - Innovation in the design
 - Openness to new, derived patterns





Metrics

Measuring usability

- Time needed for a task
- Tasks executed per time unit
- Successful vs failed attempts in a task
- User errors
- Time needed to manage errors
- How frequently manuals are used
- Positive vs negative user comments
- How many times did the user give up?

Targets

Qualitative targets

- Useful at first, design steps
- Jump out of user requirements elicitation
- Examples:
 - Allow tourists to look for information easily
 - Know the user
 - Allow team-working
 - Consistency in functionalities

Quantitative targets

- Comparative
 - Increase in speed
 - Error minimization
- Absolute
 - Time limits
 - Number of errors
 - ...can think of more?

Categorized targets

- Easiness to learn
- Easiness to use
- Efficiency
- Errors
- Knowledge sustainability
- User satisfaction

Easiness to learn

- Predictability of system response
- Familiarity
- Understanding of system response
- Consistency in system response

Easiness to use

- The best interfaces are almost invisible to the user. They avoid unnecessary elements and are clear in the language they use on labels and in messaging.

Efficiency

- Time
 - E.g. how easy is it to find your gate at a busy port?
- Steps needed
 - Any ATM user should be able to perform an action in three steps
 - Travelers must have access to info related to next departure, without needing to check the whole timetable



Effectiveness (Errors)

- Formal definition
 - A user action that does not allow to reach the desired target
- Frequency
 - Avg number per time unit
 - Avg percentage of users who make an error
- Error classification
 - How critical was the loss?
 - How easy to recover?
- Errors must be easy to recognize immediately!

Knowledge sustainability

- ▶ Most interfaces are used on a frequent basis
- ▶ Interfaces should help sustain knowledge
 - ▶ Previous interactions must be used to make easy the new ones
- ▶ Some parts of a system that may be done quite frequently
 - ▶ Installation
 - ▶ Back-up
 - ▶ Restore at previous point
- ▶ Hard to control if it can be supported or not
 - ▶ A system, easy to learn, is usually easy to remember

User Satisfaction

- Expectations
 - What are the targets? Who are the users? What are their skills? Will the interface challenge their self-confidence?

affect significantly the extent to which a system is desirable
- Key-factors:
 - Edutainment apps
 - Self-teaching
 - Personalized apps

Intuitive evaluation regarding expected satisfaction

- ▶ Aesthetics
 - ▶ Good appearance
 - ▶ Nice to interact with
- ▶ Supports
 - ▶ creativity
 - ▶ collaboration
- ▶ Motive
 - ▶ praise
 - ▶ challenge
- ▶ Donts:
 - ▶ Boring
 - ▶ Too complex
 - ▶ Too challenging

Usability testing

Usability Testing

- The conduct of empirical studies of a product's usability by:
 - observing users perform tasks,
 - analyzing log-files,
 - questionnaires,
 - Interviews,
 - combination(s) of the above



Image from: <https://www.zignuts.com>

Step 1: User Analysis & Profiles

- If needed, break your users into categories and create profiles:
 - **Demographics:** age, gender, education level, cultural background
 - **Experience level** with similar products, with required technology
 - **Frequency of expected use:** occasional, frequent user
 - **Type of user:** e.g. closer to one of your personas, primary, secondary, etc.
 - **Other things:**
 - subject matter knowledge
 - location of use
 - disabilities or impairments (from color blindness and learning disabilities to more severe ones)

Step 1: User Analysis & Profiles

- Choose user profiles to test:
 - Choose users from all major profiles to test with them
 - If limited testing is to be considered:
 - Choose most common profiles
 - Choose profiles that you expect will have the greatest usability issues

Step 2: Decide what to test

Choose an overall purpose

- Example: How useful and easy to use is our new app?

Define your objectives:

- Example: Does our search engine provide usable results in the first 5 links returned?
- Example: Are results clear to the users?
- Example: How often did they ask for help/manuals?
- Example: How many users simply gave up?

Measure them and group them (targets & metrics):

1. Effectiveness: Can they do it? How many errors occurred?
2. Efficiency: How fast? To what extent?
3. Satisfaction: Can they understand it? Can they find it?

Step 2: Decide what to test

Select specific tasks to have tested:

- Those with a high chance of user failure (complex tasks, one-of-a-kind tasks, highly abstract or technical tasks)
- Consider tasks with a high cost of user failure (e.g. tasks where data could be damaged or lost)
- Consider:
 - First impressions (look and feel)
 - First tasks (forgive the user)

Step 3: Preparing for the testing

Create written test materials:

- Task list for users
- Written welcome speech/ Intro to be read to user
- Consent forms
- Pre-task and post task questionnaires & interview questions
- Other materials

Define team member's roles:

- Facilitator/Briefer (necessary): Often only team member to interact with users
- Observation recorder/note taker (necessary)
- Camera operator (optional)
- Help desk operator (optional)
- Test administer (optional)

Step 4: Conducting the test

Greet & Brief participant:

- a. Read/say welcome
- b. Emphasize that you are not testing *them* but the product and that they should act as natural as possible
- c. Emphasize how user tells you s/he has completed a task
- d. Stress that the testing is “anonymous”

Be unbiased (especially the Facilitator/Briefer)

Intervene carefully (avoid as much as possible)

Observe and record data

Debrief user

Step 5: Analyzing the Data

Examine findings related to your metrics for each user, user profile, and task

- Plot your data and visualize them graphically
- Use analysis techniques such as statistics (even averages + standard deviations help, but deeper analyses are usually needed to test statistical significance)
- Usually through significance tests
- Determine cause of problems
- Determine scope/severity of problems
- Make recommendations/changes

Report Findings