Human-Computer Interaction

5—Golden Rules of Interface Design

TECHNISCHE UNIVERSITÄT DARMSTADT

SS 2013

Prof. Dr. Max Mühlhäuser
Dr. Jochen Huber
Mohammadreza Khalilbeigi
Roman Lissermann

Technische Universität Darmstadt
Department of Computer Science
Telecooperation Lab

Telecooperation Lab

Golden Rules of Interface Design

1. Keep the interface simple
2. Speak the user‘s language
3. Be consistent and predictable
4. Make things visible and provide feedback
5. Minimize the user‘s memory load
6. Design for error: Avoid errors, help to recover from errors, offer undo
7. Design clear exits and closed dialogs
8. Include help and documentation
9. Offer shortcuts for experts
10. Make the system responsive

Inspired by Jan Borchers, Ben Shneiderman, and Don Norman. The following slides are partly based on Jan Borcher‘s slide set Designing Interactive Systems 1.

1. Keep the interface simple

TECHNISCHE UNIVERSITÄT DARMSTADT

- Most important rule
- The first design is often not clear and too complex
  → Re-think and re-design the interface

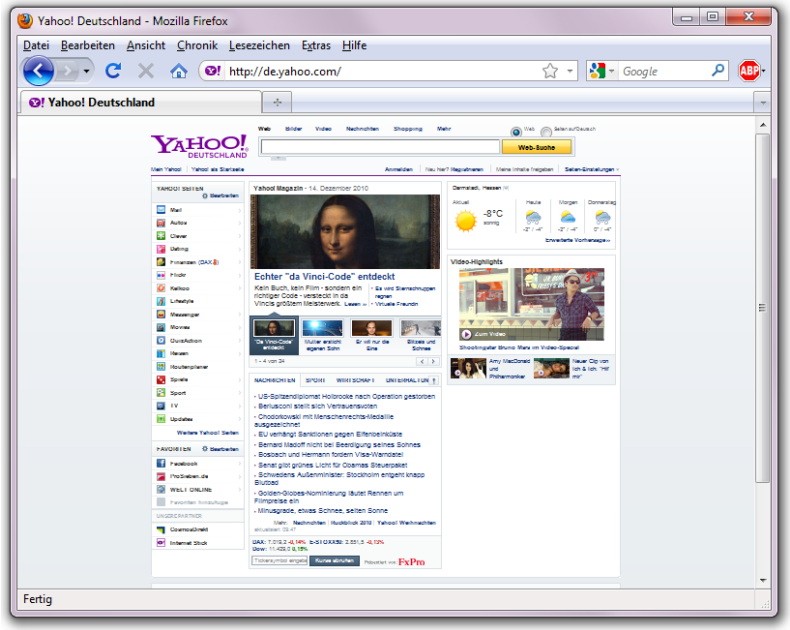
- Avoid to uncritically add more and more features with each version
  - Users will ask for them, but make sure that usability does not suffer
  - Experience is that 80% of users use only 20% of features

1. Keep the interface simple

TECHNISCHE UNIVERSITÄT DARMSTADT

→ Remove or hide irrelevant or rarely needed information

- All information on the screen is competing for the user‘s attention
- Irrelevant or rarely needed information decreases the relative visibility of relevant information

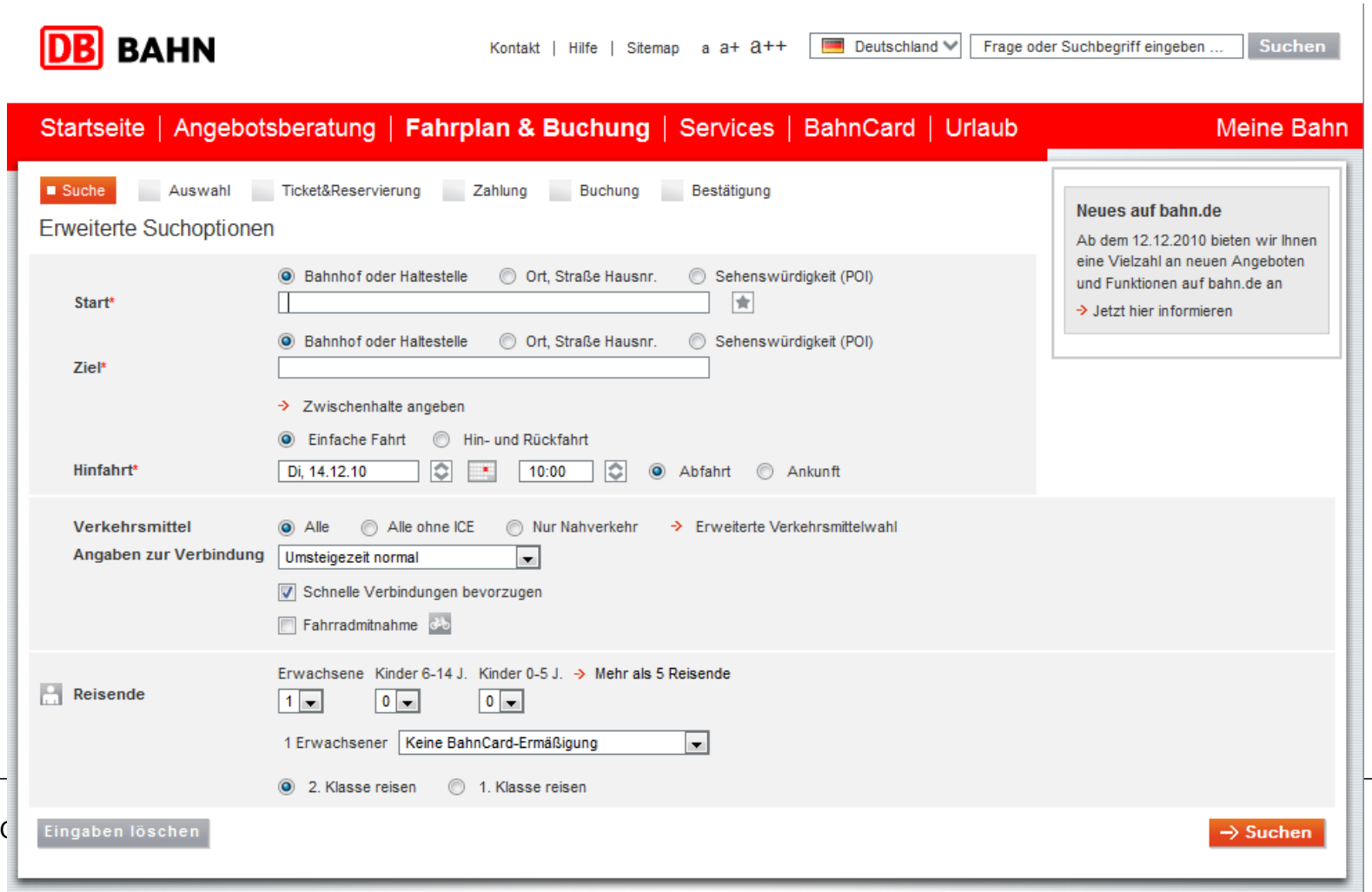


Telecooperation Lab

1. Keep the interface simple

TECHNISCHE UNIVERSITÄT DARMSTADT

→ Present information in natural order



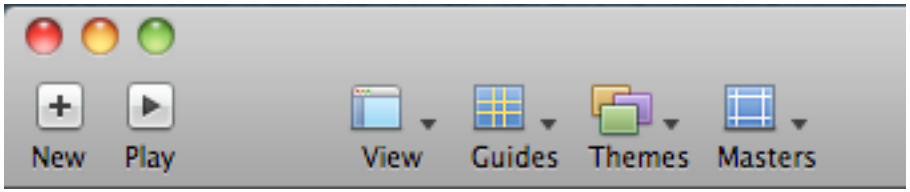
Telecooperation Lab

1. Keep the interface simple

TECHNISCHE UNIVERSITÄT DARMSTADT

→ Display information such that it is easy to parse

- E.g. Gestalt laws: closed shapes and spatial proximity convey groupings



Telecooperation Lab

2. Speak the user‘s language

TECHNISCHE UNIVERSITÄT DARMSTADT

- Take words and concepts from the application domain, not computer science
- Determine terminology during initial interviews and task analysis
- Example: *File* means less to an architect who is new to computers than *drawing*
- Example: iTunes uses terms *music*, *songs*, *playlists* – not *files*
- Applies to words for objects, but also to work processes and tasks

3. Be consistent and predictable

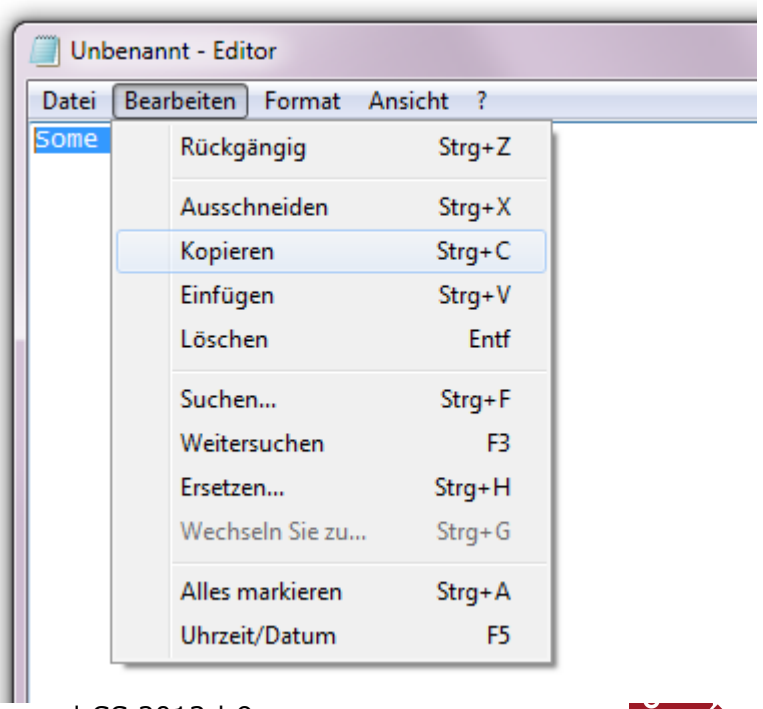
TECHNISCHE UNIVERSITÄT DARMSTADT

- Consistent command design
  - Similar commands for similar situations
  - Consistent terminology in menus, dialogs, help pages, …
- Consistent graphic design
  - Fonts, layout, color coding, upper/lower cases
- Consistent flow design
  - Similar tasks proceed in similar ways
- Only few obvious exceptions
  - E.g. no clear-text echo when entering passwords

Example: Cut / Copy / Paste

TECHNISCHE UNIVERSITÄT DARMSTADT

- The commands Cut & Paste and Copy & Paste have in all applications these commonly used names
- They are performed in this order (not e.g., first selecting target position, then selecting contents to copy or move)
- They are performed in all applications using the same keyboard shortcuts
- In most application menus, the commands are at a consistent place (Edit…)



Telecooperation Lab

Predictability

TECHNISCHE UNIVERSITÄT DARMSTADT

- Follow the *Principle of Least Surprise*
  - System should always react in a way that minimizes the user‘s surprise (and therefore confusion and irritation)
- Don‘t do unexpected things
- Users like to be „in control“
  - The user initiates actions, the systems respond
  - Usually not the other way round

4. Make things visible and provide feedback

TECHNISCHE UNIVERSITÄT DARMSTADT

- Make functionality visible – let the user discover them
  - Many applications don‘t follow this principle, unfortunately („How do I print this in duplex?“)
- Users should always be aware of what is going on
- Remember the 7 Stages of Action
  - Complete and continuous feedback bridges the Gulf of Evaluation

4. Make things visible and provide feedback

TECHNISCHE UNIVERSITÄT DARMSTADT

- Each user action requires some feedback
  - Subtle feedback for small / short / frequent actions
    E.g. key press, menu selection
  - More noticeable for main / long / infrequent actions
    E.g. saving or copying files
  - Icons in GUIs simplify visualizing object state and actions
    E.g. recycler, folder show if they contain files

- Feedback should be meaningful
  - Bad example: Card terminal in university cafeteria – What means *Error 214* ?

5. Minimize the user‘s memory load

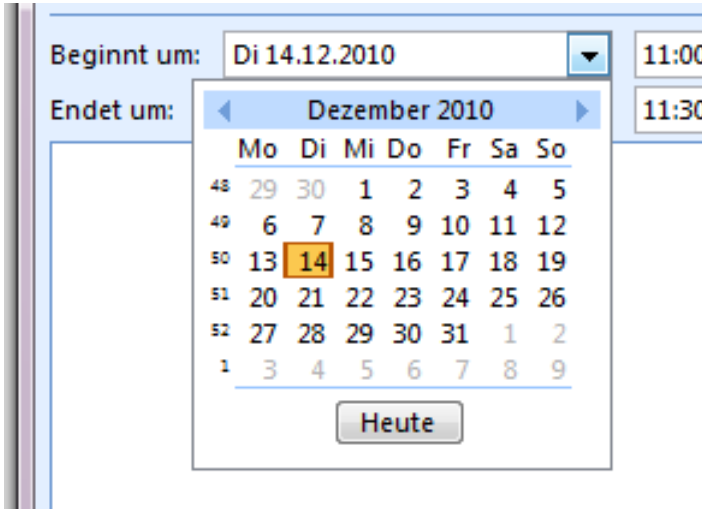
TECHNISCHE UNIVERSITÄT DARMSTADT

- Short-term memory has limited capacity
  - Ca. 7 +/- 2 chunks
- Avoid situations in which prior dialog information has to be reproduced from memory
  - User should not have to type anything twice
- Recognition is easier than remembering
  - It is easier to minimize memory load with GUIs than command line interfaces

6. Design for error: Avoid errors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Best solution: Design the system in a way that mistakes cannot be made. Examples:
  - Selection instead of (mis)typing
  - Prevent illegal input by appropriate constraints
    - E.g. select data; cannot type letters in numerical data fields
  - Automatic correction of illegal characters in file names



Telecooperation Lab

6. Design for error: Help to recover from errors

TECHNISCHE UNIVERSITÄT DARMSTADT

If errors cannot be avoided:

- Offer simple, concrete instructions to recover

- Provide meaningful error messages
  - Clearly explain the problem
  - Don ‘t make the user feel stupid
  - Offer a way to correct the problem

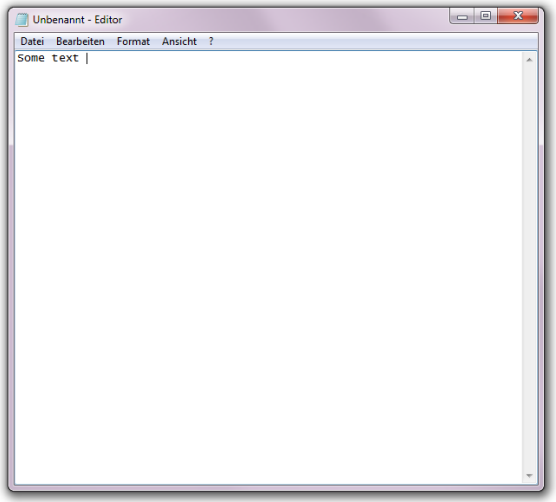
Bad example: Card terminal in university cafeteria – What means *Error 214* ?

6. Design for error: Offer undo

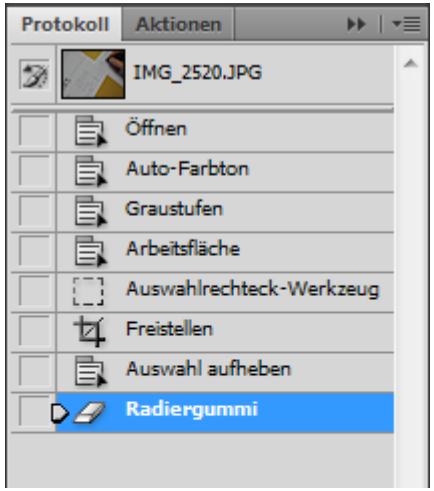
TECHNISCHE UNIVERSITÄT DARMSTADT

- As many actions as possible should be reversible
  - Lowers anxiety because users know errors are recoverable
  - Encourages users to try out new functions

Poor: Windows notepad offers only 1 single undo step



Good: Photoshop offers visible history

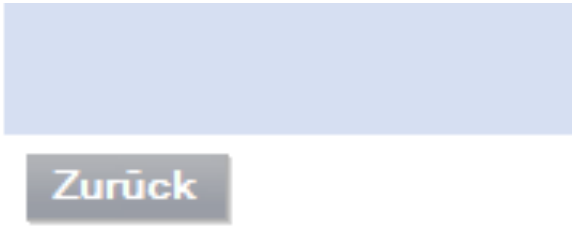


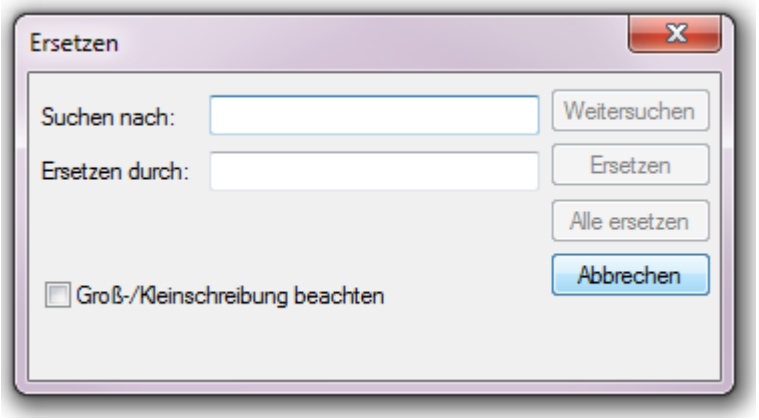
Telecooperation Lab

7. Design clear exits and closed dialogs

TECHNISCHE UNIVERSITÄT DARMSTADT

- Three most common questions of users during a dialog:
  - Where am I?
  - What can I do here?
  - How do I get back to where I was?
- Clear exits (*Back*, *Quit, Cancel*) help with Question 3



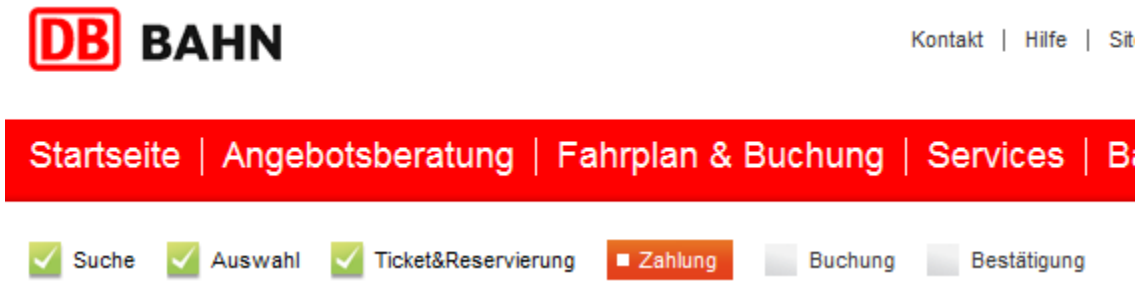


Telecooperation Lab

7. Design clear exits and closed dialogs

TECHNISCHE UNIVERSITÄT DARMSTADT

- Closed dialogs
  - Sequences of actions should be organized into groups (closed dialogs)
  - Provide feeling of having completed a step
  - Allows the user to relax, take a breath, free the mind for the next step

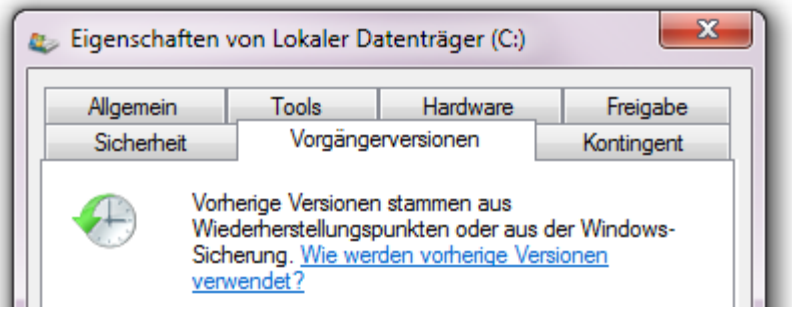


Telecooperation Lab

8. Include help and documentation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Providing help is not an excuse for poor design!
- Users don‘t like to read manuals
  - They prefer to learn while performing their tasks
  - Exception: expert users
- Most users will stay at an intermediary level
  - They need reminders and a clear learning path
  - They need a quick way to access critical information
    - E.g. tutorial and/or getting started manuals
    - E.g. tooltips
    - E.g. provide direct access to help that is relevant for the current context

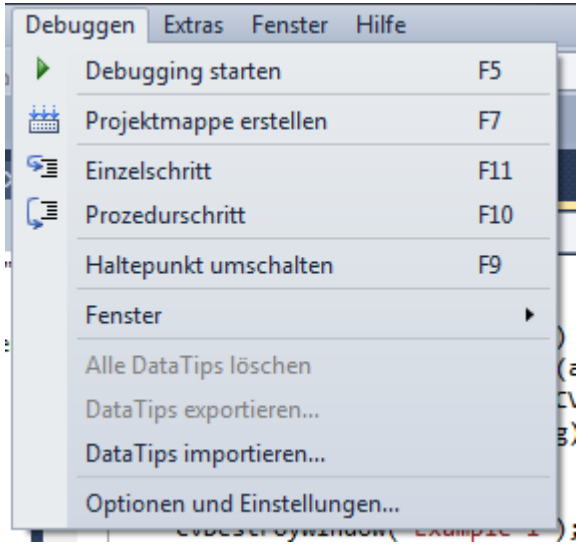


Telecooperation Lab

9. Offer shortcuts for experts

TECHNISCHE UNIVERSITÄT DARMSTADT

- Frequent users want faster interactions
- Enable transition to expert userse
  - Menus are used by beginners
  - Experts use toolbar buttons and keyboard shortcuts
  - To enable transition: Show toolbar and keyboard shortcuts in menu



Telecooperation Lab

10. Make the system responsive

TECHNISCHE UNIVERSITÄT DARMSTADT

- Responsiveness is key usability problem of interactive systems
  - Bad responsiveness opens Gulf of Evaluation („Where am I?“ „What is the system doing?“, „Is there a problem?“)
- Examples of bad responsiveness
  - Delayed response to button clicks
  - Sliders and scrollbars lag
  - Applications go „dead“ during disk operations

Example: Scrollbar

TECHNISCHE UNIVERSITÄT DARMSTADT

- Does text move as you scroll (good) or after you let go (bad)?
- If the designer does not specify, the developer will make a decision
- That will usually be the technically simplest
  - Developers are not trained in user interface theory and concepts

Responsiveness

TECHNISCHE UNIVERSITÄT DARMSTADT

- Responsiveness ≠ performance!
- Processing resources will always be limited
  - We still look at the hourglass as much as 15 years ago
- UIs are real-time systems with deadlines based on human cognition

Three Human Deadlines

TECHNISCHE UNIVERSITÄT DARMSTADT

- 0.1 seconds
  - Perception of cause and effect
  - E.g., delay between moving mouse and pointer following
- 1 second
  - Turn taking in conversation, min. reaction time for unexpected events
  - E.g., you have 1s max to show progress indicator, open window or finish system-initiated operations (like auto-save)
- 10 seconds
  - Typical human attention span
  - Max time for one step of a task
  - E.g., completing one step of a wizard
  - Max time to finish input to an operation
  - E.g., from selecting Print menu entry to sending off the print job

Golden Rules of Interface Design

1. Keep the interface simple
2. Speak the user‘s language
3. Be consistent and predictable
4. Make things visible and provide feedback
5. Minimize the user‘s memory load
6. Design for error: Avoid errors, help to recover from errors, offer undo
7. Design clear exits and closed dialogs
8. Include help and documentation
9. Offer shortcuts for experts
10. Make the system responsive