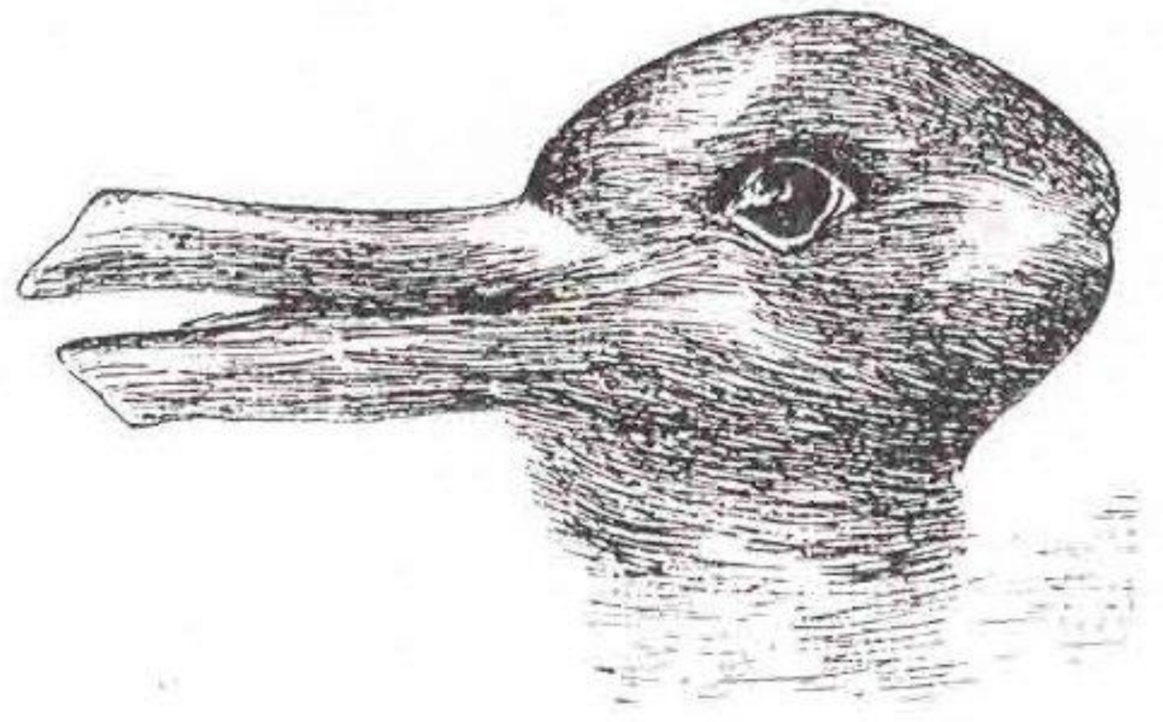
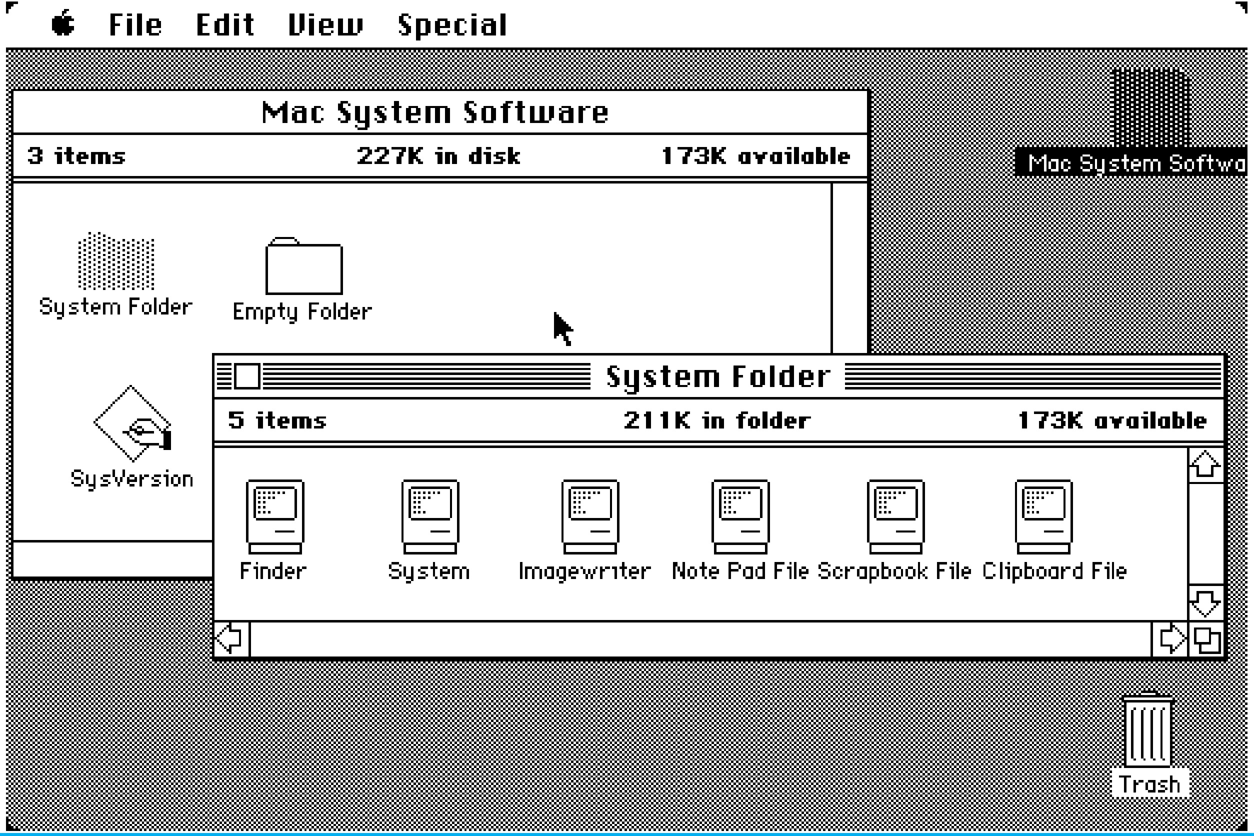




User Illusions

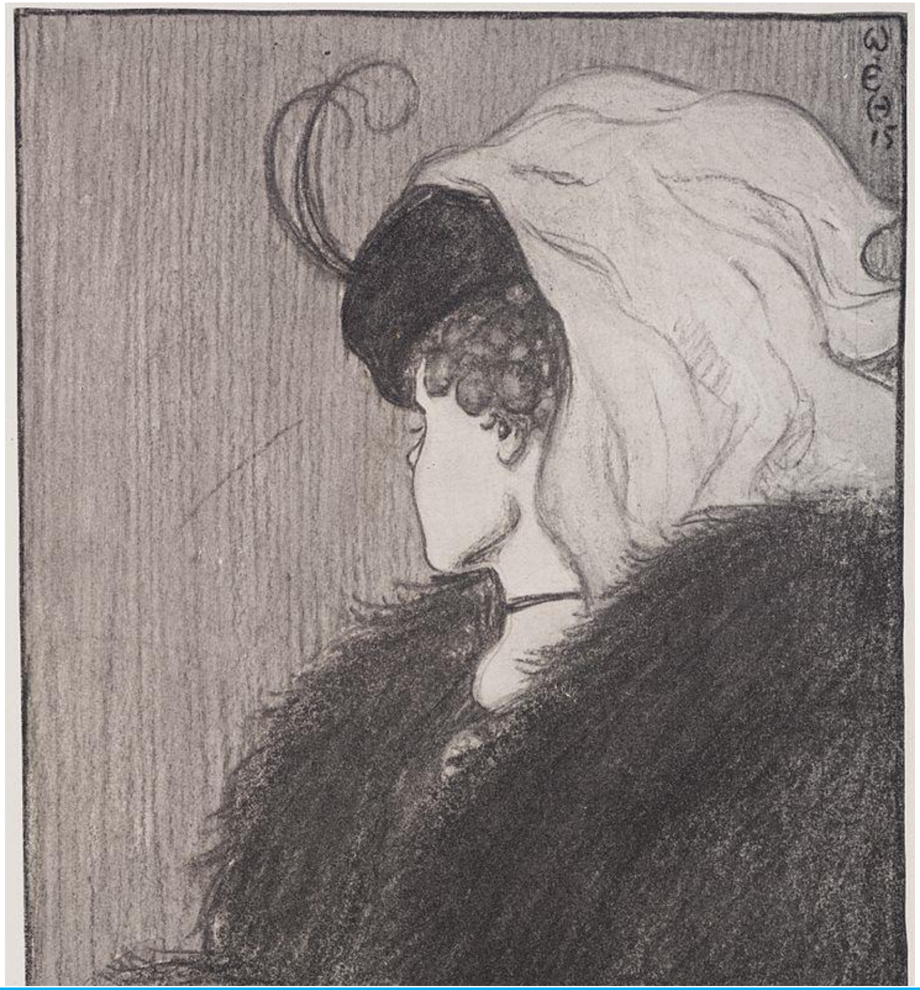
Albrecht Schmidt

Learning Goals

- Understand …
  - How users see a user interface,
  - The concept of user illusion,
  - Why it is hard for developers to spot usability issues.
- Be able to explain …
  - how to use the concept of user illusion to improve a user interface or interaction design.

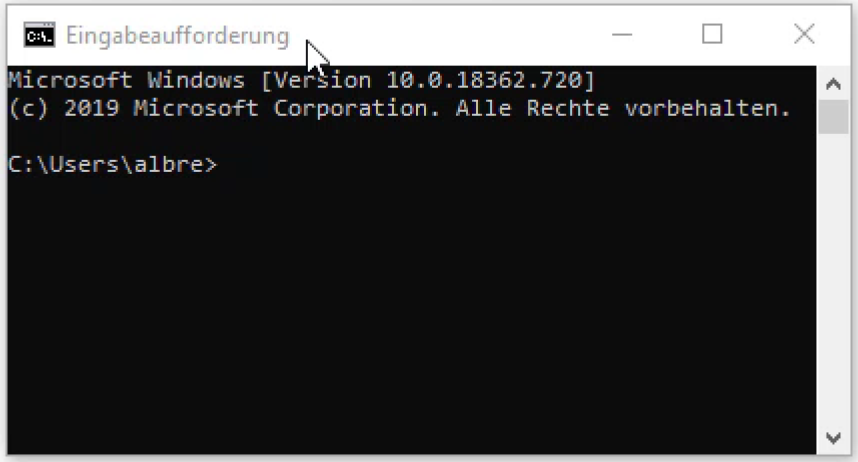
We see what we want to see

We see what make sense to us



We see what we want to see

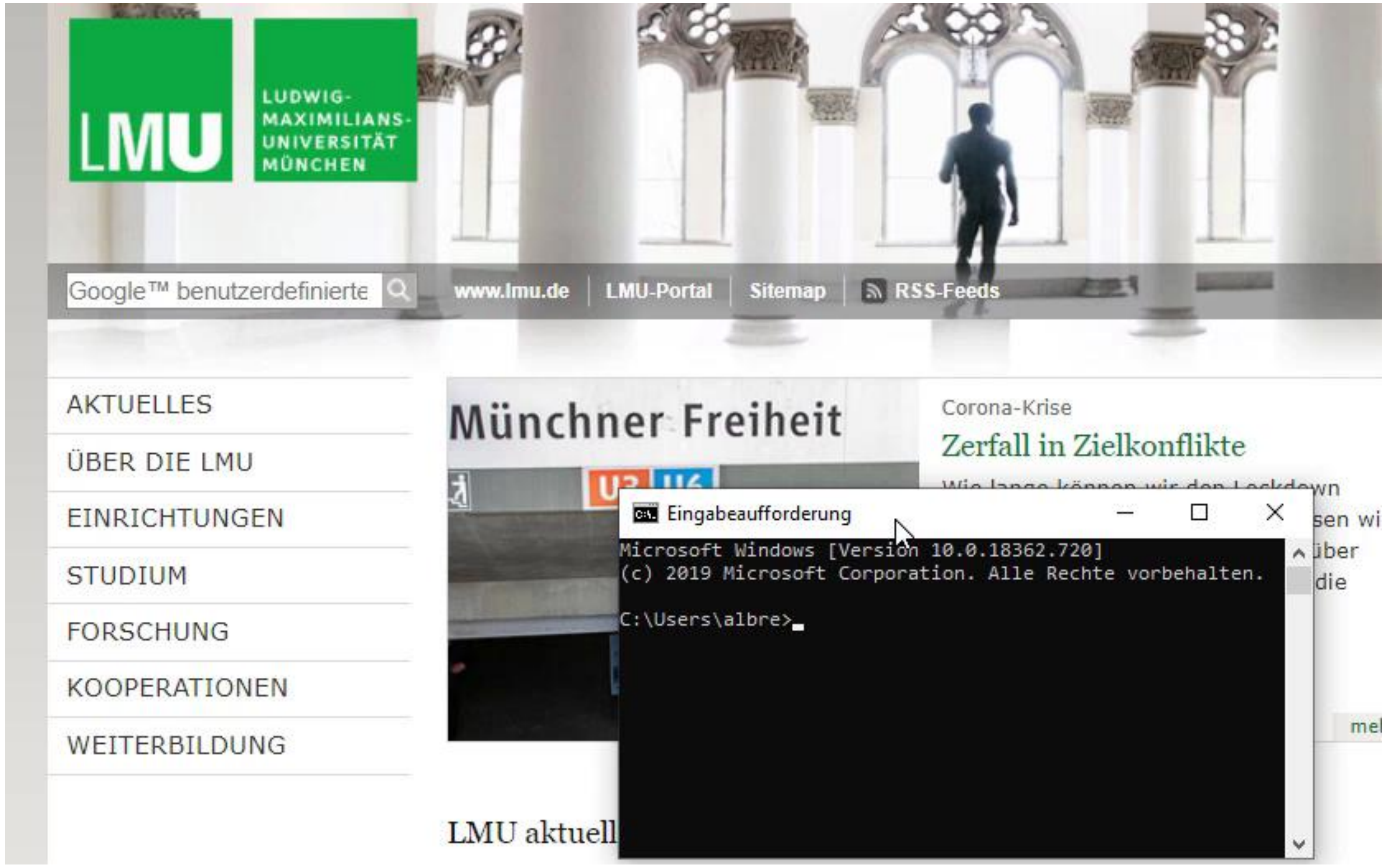
Example: Window



```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\albre>
```

We see what we want to see

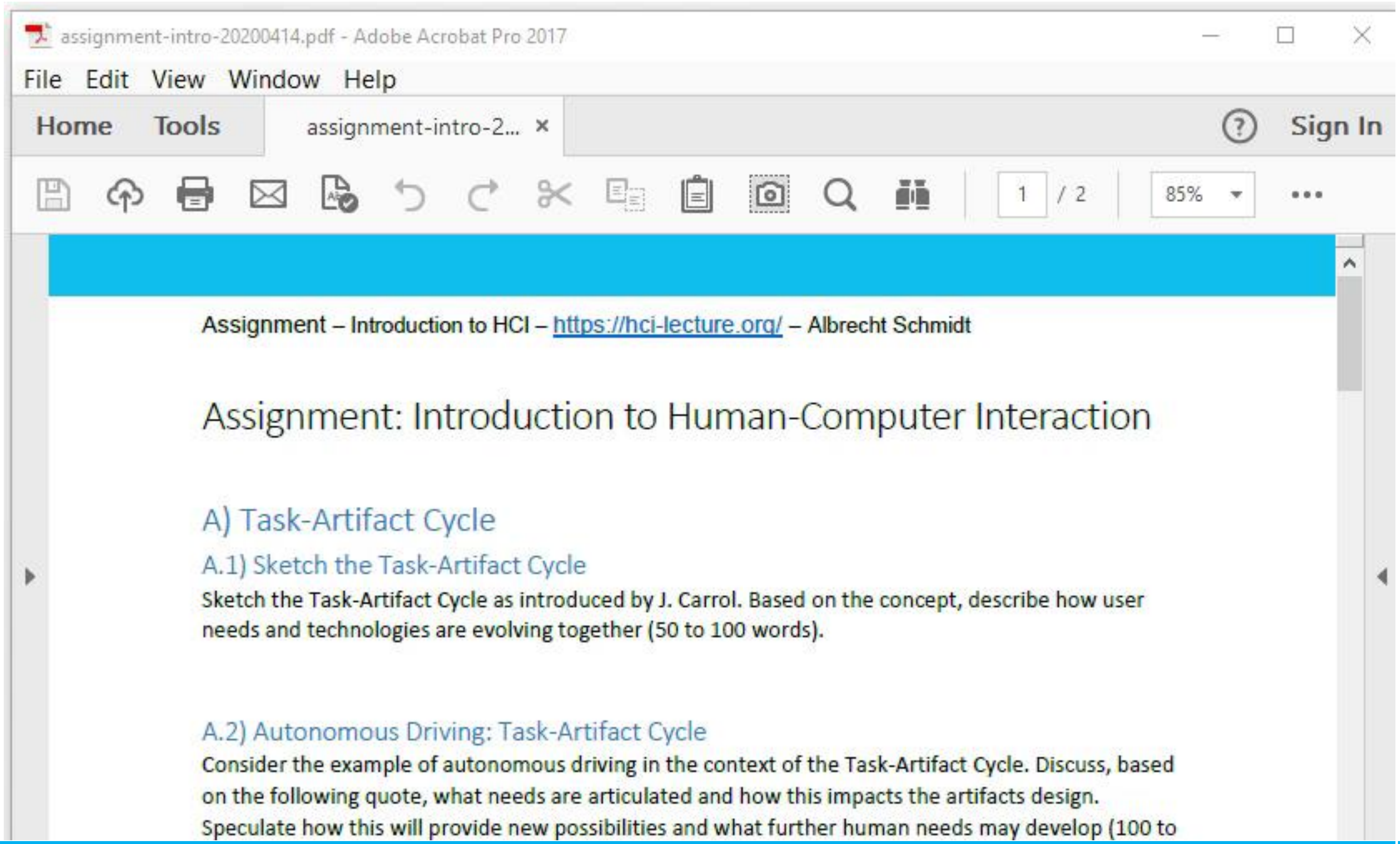
Example: Background vs. Foreground



```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\albre>_
```

We see what we want to see

Example: Scrolling


assignment-intro-20200414.pdf - Adobe Acrobat Pro 2017
File Edit View Window Help
Home Tools assignment-intro-2... ×
Sign In
1 / 2
85%

Assignment – Introduction to HCI – https://hci-lecture.org/ – Albrecht Schmidt

Assignment: Introduction to Human-Computer Interaction

A) Task-Artifact Cycle

A.1) Sketch the Task-Artifact Cycle

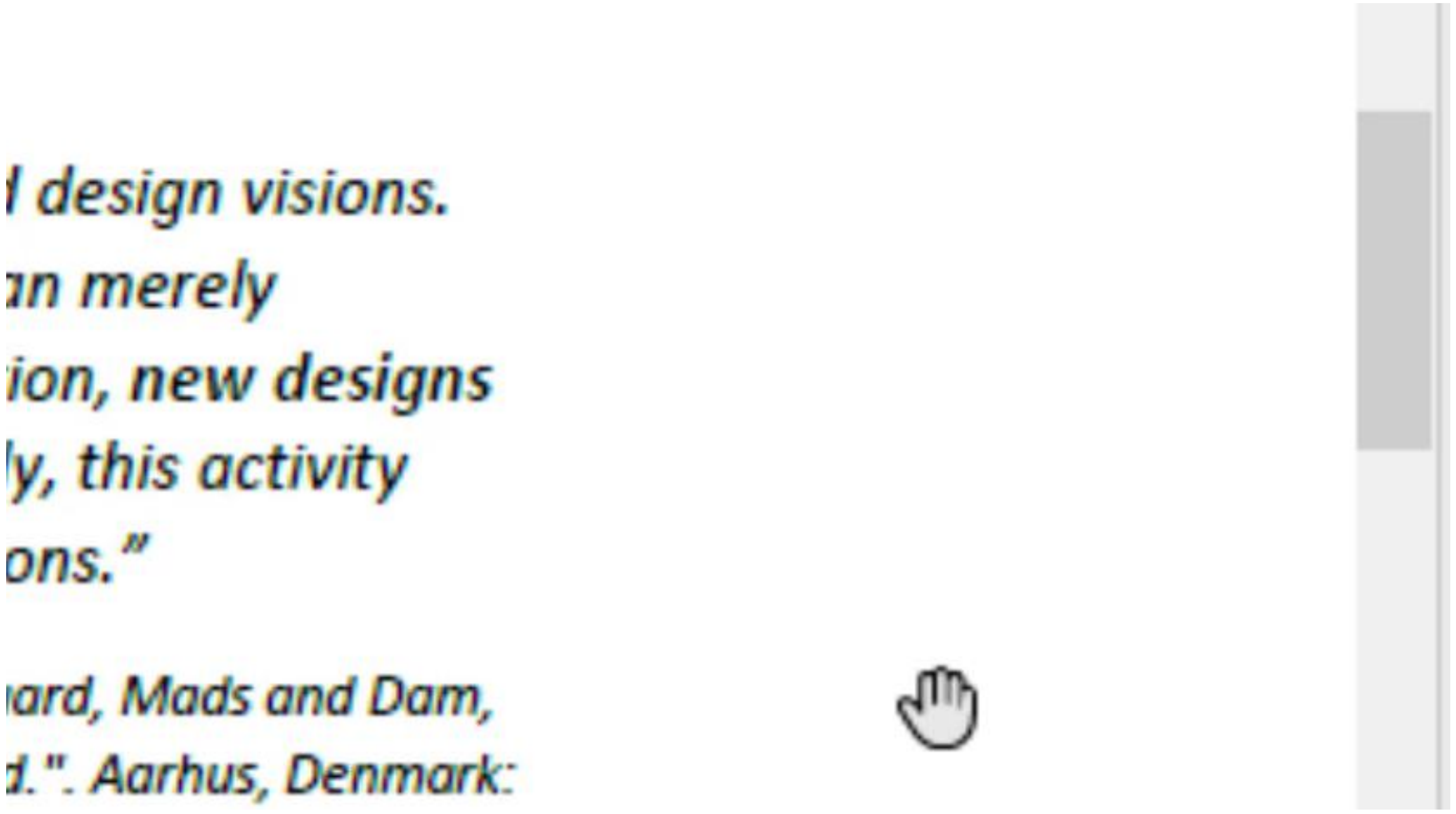
Sketch the Task-Artifact Cycle as introduced by J. Carrol. Based on the concept, describe how user needs and technologies are evolving together (50 to 100 words).

A.2) Autonomous Driving: Task-Artifact Cycle

Consider the example of autonomous driving in the context of the Task-Artifact Cycle. Discuss, based on the following quote, what needs are articulated and how this impacts the artifacts design. Speculate how this will provide new possibilities and what further human needs may develop (100 to

We see what we want to see

Example: Scrolling

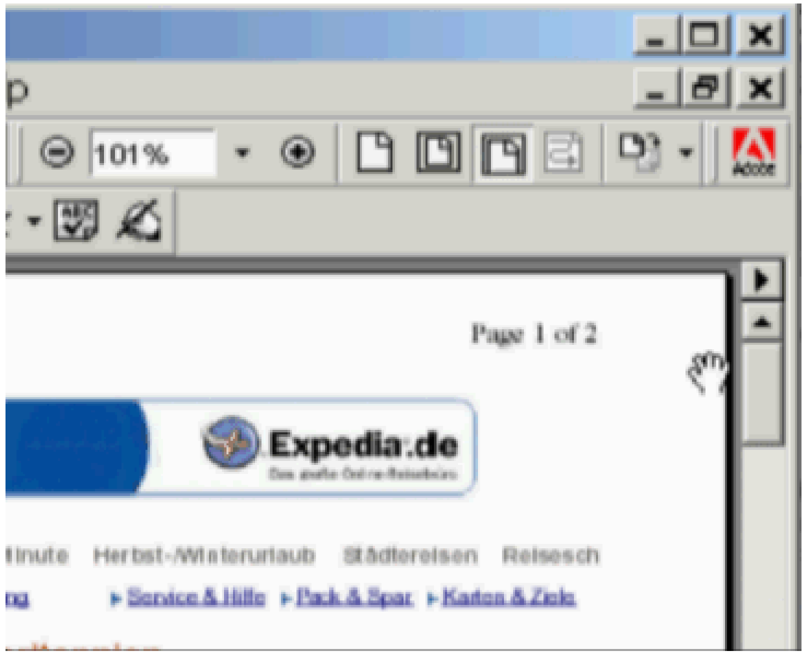


l design visions.
an merely
ion, new designs
ly, this activity
ons."

ard, Mads and Dam,
d.". Aarhus, Denmark:

We see what we want to see

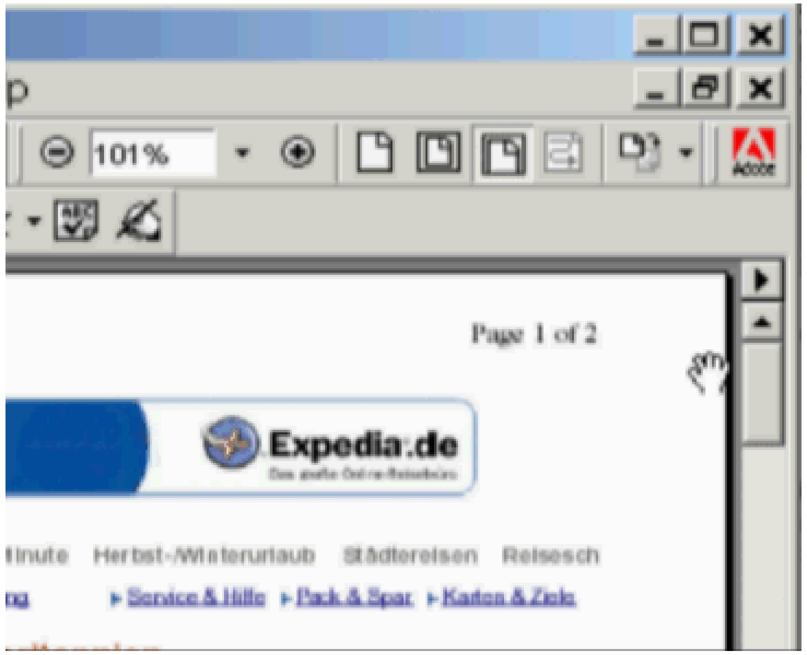
Example: Scrolling

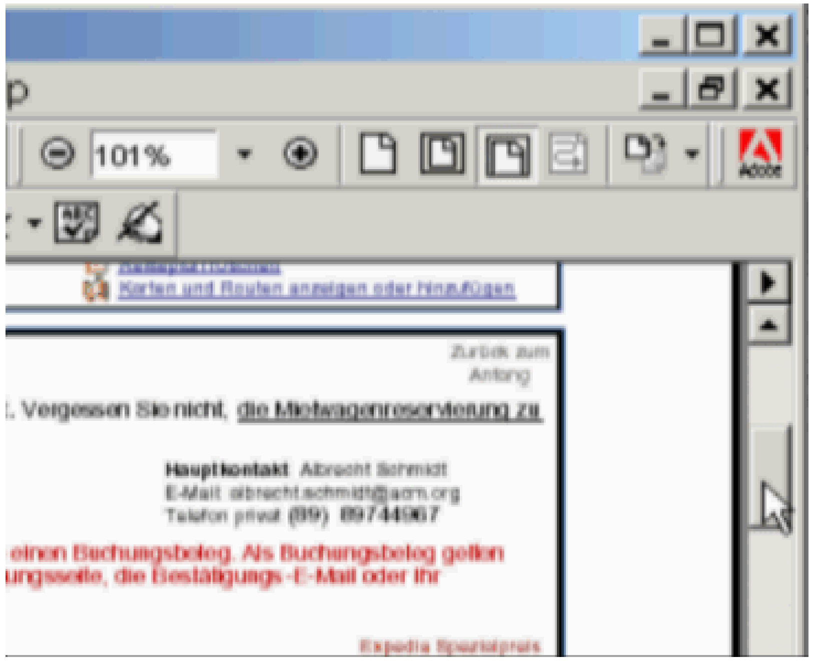


- Moving up the hand
  Moves up the document
- What really happens?
- What happens on a graphics level?
- What do we imagine?
- What is the metaphor?

We see what we want to see

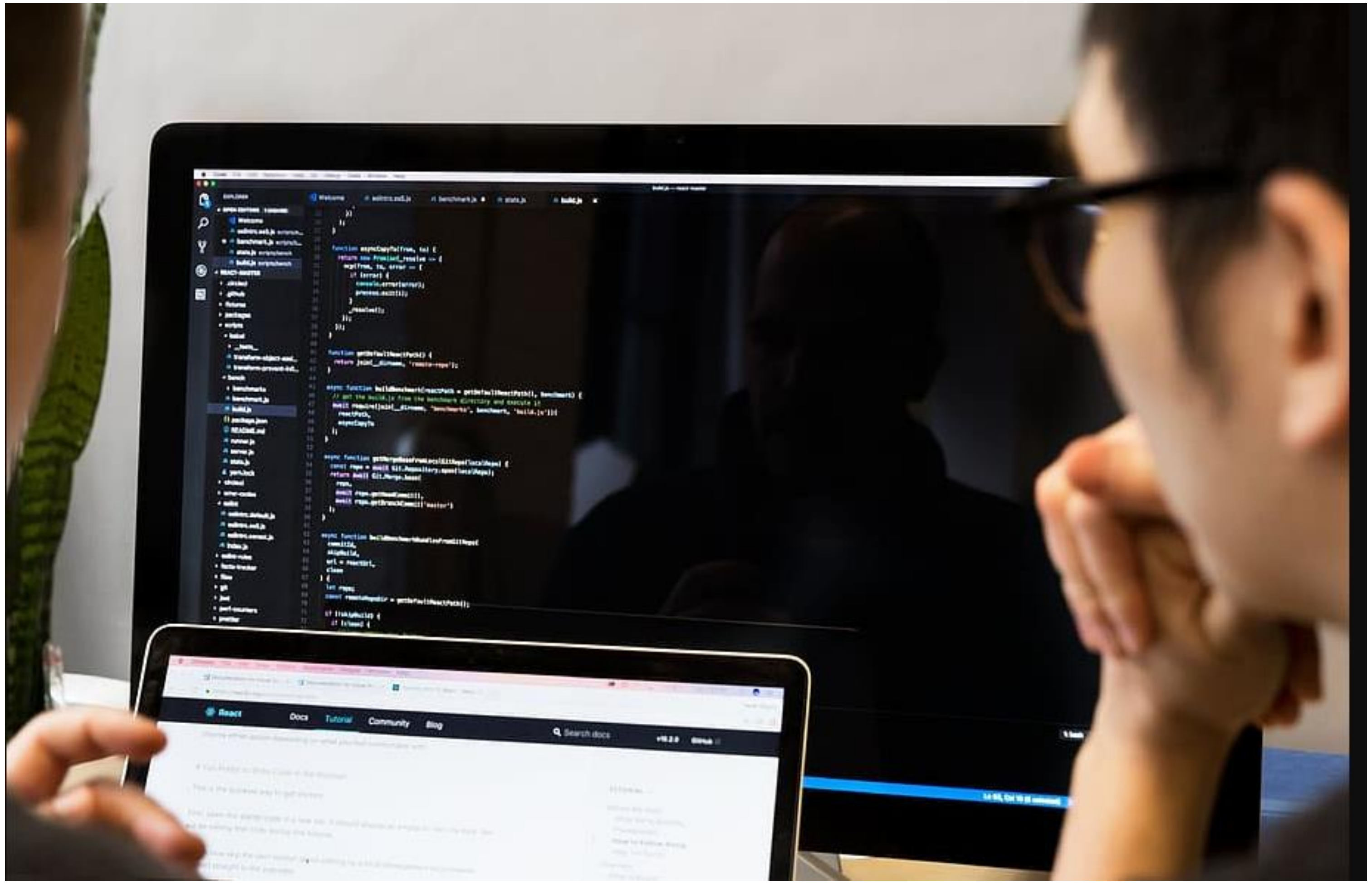
Example: Scrolling




101%
Karten und Routen anzeigen oder hinzufügen
Zurück zum Anfang
Vergessen Sie nicht, die Mietwagenreservierung zu
Hauptkontakt Albrecht Schmidt
E-Mail albrecht.schmidt@acm.org
einen Buchungsbeleg. Als Buchungsbeleg gelten
ungsseite, die Bestätigungs-E-Mail oder Ihr
Expedia Spezialpreis

Why does the developer not have usability issues (with their software)?

Why are many open source products are hard to use?



Why does the developer not have usability issues (with their software)?

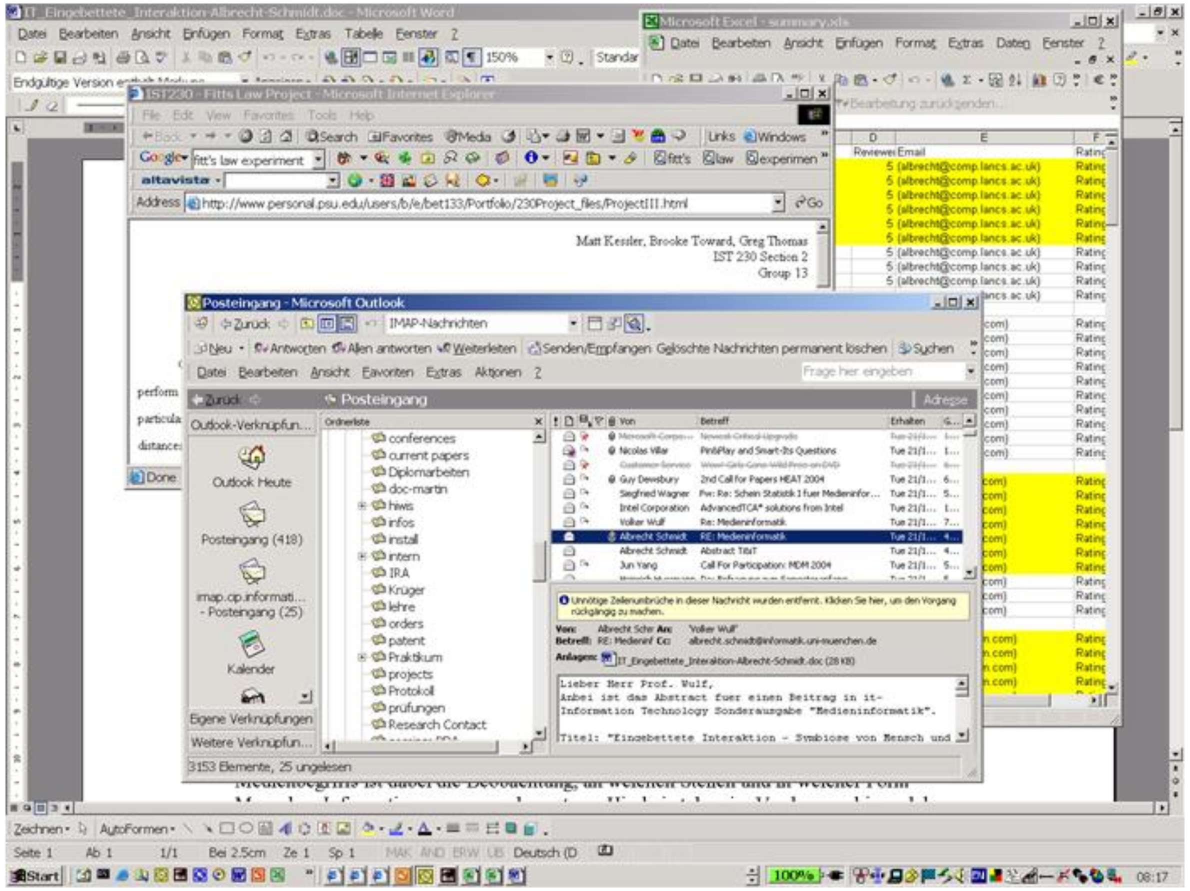
Why are many open source products are hard to use?

- End users have little idea about
  - system and software architecture
  - remote database connections
  - state transitions and dependencies
  - internal query syntax
  - application context
  - system restrictions
  - …
- … and the do not care about it –
  and they should not need to care about it!

What do you see?
What do you know?

Colored pixels? Overlapping windows? Email Program?

Meeting notes and an urgent request by a colleague?


IT_Eingebettete_Interaktion-Albrecht-Schmidt.doc - Microsoft Word
Microsoft Excel - summary.xls
IST230 - Fitts Law Project - Microsoft Internet Explorer
Posteingang - Microsoft Outlook
Posteingang
Outlook Heute
Posteingang (418)
Kalender
conferences
current papers
Diplomarbeiten
hiwis
infos
intern
lehre
orders
patent
Praktikum
projects
Protokoll
prüfungen
Research Contact
3153 Elemente, 25 ungelesen
Start

Matt Kessler, Brooke Toward, Greg Thomas
IST 230 Section 2
Group 13

Von	Betreff	Erhalten
Nicolas Villar	PinBPlay and Smart-Its Questions	Tue 21/1...
Guy Dewsbury	2nd Call for Papers HEAT 2004	Tue 21/1...
Siegfried Wagner	Fw: Re: Schein Statistik I fuer Medieninfor...	Tue 21/1...
Intel Corporation	AdvancedTCA* solutions from Intel	Tue 21/1...
Volker Wulf	Re: Medieninformatik	Tue 21/1...
Albrecht Schmidt	RE: Medieninformatik	Tue 21/1...
Albrecht Schmidt	Abstract TiltT	Tue 21/1...
Jun Yang	Call For Participation: MDM 2004	Tue 21/1...

Unnötige Zeilenumbrüche in dieser Nachricht wurden entfernt. Klicken Sie hier, um den Vorgang rückgängig zu machen.

Von: Albrecht Schr An: 'Volker Wulf'
Betreff: RE: Medieninf Cc: albrecht.schmidt@informatik.uni-muenchen.de
Anlagen: IT_Eingebettete_Interaktion-Albrecht-Schmidt.doc (28 KB)

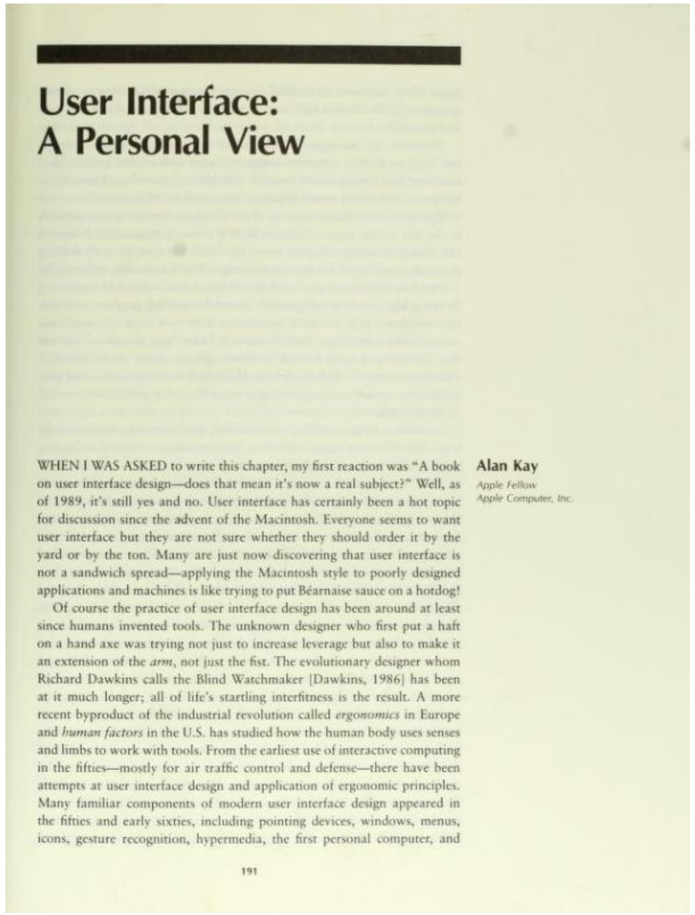
```
Lieber Herr Prof. Wulf,
Anbei ist das Abstract fuer einen Beitrag in it-
Information Technology Sonderausgabe "Medieninformatik".

Titel: "Eingebettete Interaktion - Symbiose von Mensch und
```

User-Illusion and Metaphors

Alan Kay on metaphors

“At PARC we coined the phrase **user illusion** to describe what we were about when designing the user interface. **There are clear connotations to the stage, theatrics, and magic - all of which give strong hints as to the direction to be followed**. For example, the screen as "Paper to be marked on" is a metaphor that suggests pencils, brushes, and typewriting. Fine as far as it goes. But it is the magic - understandable magic - that really counts. Should we transfer the paper metaphor so perfectly that the screen is as hard as paper to erase and change? Clearly not.” (p. 199)

Kay, A. (1990). User interface: A personal view. In B. Laurel, (Ed.), *The art of human computer interface design* (pp. 191–207). Reading , MA : Addison-Wesley.



User Interface:
A Personal View

Alan Kay
Apple Fellow
Apple Computer, Inc.

WHEN I WAS ASKED to write this chapter, my first reaction was “A book on user interface design—does that mean it's now a real subject?” Well, as of 1989, it's still yes and no. User interface has certainly been a hot topic for discussion since the advent of the Macintosh. Everyone seems to want user interface but they are not sure whether they should order it by the yard or by the ton. Many are just now discovering that user interface is not a sandwich spread—applying the Macintosh style to poorly designed applications and machines is like trying to put Béarnaise sauce on a hotdog!

Of course the practice of user interface design has been around at least since humans invented tools. The unknown designer who first put a haft on a hand axe was trying not just to increase leverage but also to make it an extension of the *arm*, not just the fist. The evolutionary designer whom Richard Dawkins calls the Blind Watchmaker [Dawkins, 1986] has been at it much longer; all of life's startling interfitness is the result. A more recent byproduct of the industrial revolution called *ergonomics* in Europe and *human factors* in the U.S. has studied how the human body uses senses and limbs to work with tools. From the earliest use of interactive computing in the fifties—mostly for air traffic control and defense—there have been attempts at user interface design and application of ergonomic principles. Many familiar components of modern user interface design appeared in the fifties and early sixties, including pointing devices, windows, menus, icons, gesture recognition, hypermedia, the first personal computer, and

191

User interfaces create (understandable) Illusions

Magic and User Illusions

A Constructive Approach to Support User Interface Design

1. Describe the perfect magical experience or the optimal illusion you want to create
2. List all parameters that have an influence on the user experience
   1. What aids the user experience / illusion?
   2. What breaks the user illusion?
3. Prioritize the parameters and use this as input for design decisions.

Did you understand this block?

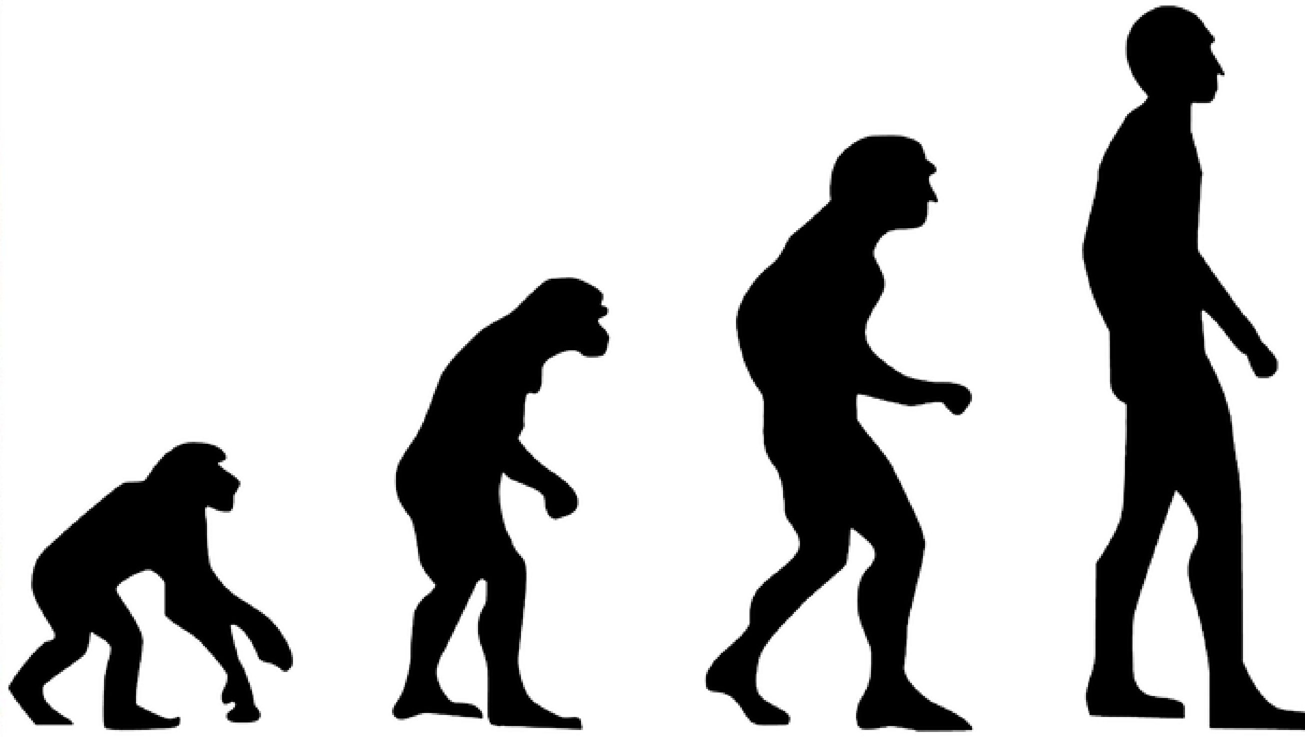
Can you answer these questions?

- How do we recognize a window in the user interface as window? How does this relate to gestalt laws?
- Why do developers often no realize usability problems?
- Explain the concept of user illusion.
- Explain how to use the concept of user illusion to improve a user interface or interaction design.

References

- Kay, A. (1990). User interface: A personal view. In B. Laurel, (Ed.), The art of human computer interface design (pp. 191–207). Reading , MA : Addison-Wesley.

License



Focus on the Human

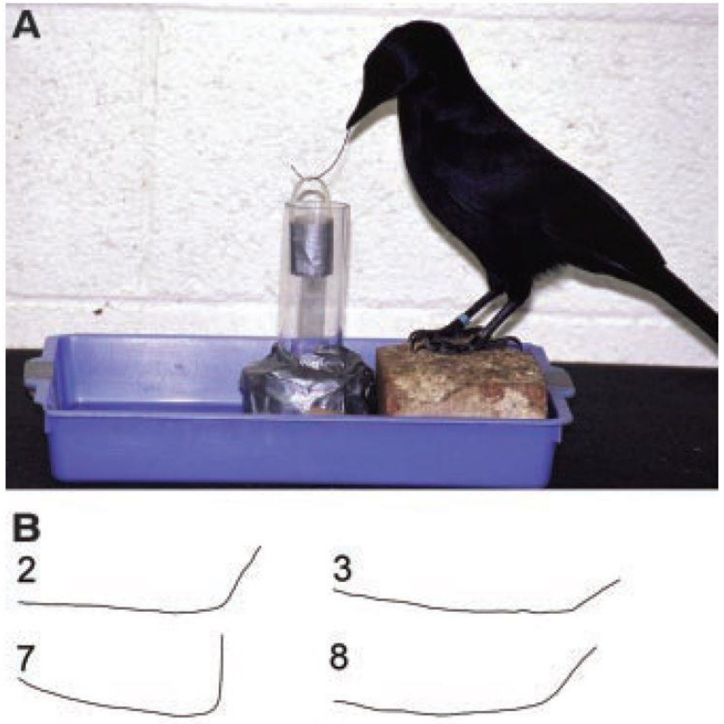
Learning Goals

- Understand …
  - the basic activities in a human centered design process
  - what are potential products in the design process
  - what typical design objectives are
- Be able to
  - explain design products and design objectives in the context of a real world scenario
  - explain the key principles for usable systems by Gould and Lewis
  - discuss the statement “Make Mistake Early and Recognize Them” in the context of usability

We build tools that extend our abilities

Tool use, intelligence, and evolution

The way humans make and use **tools** is perhaps what **sets our species apart** more than anything else. Now scientists are more and more uncovering the forces that drove our lineage to our heights of tool use — and how **tool use**, in turn, might have **influenced our evolution**."

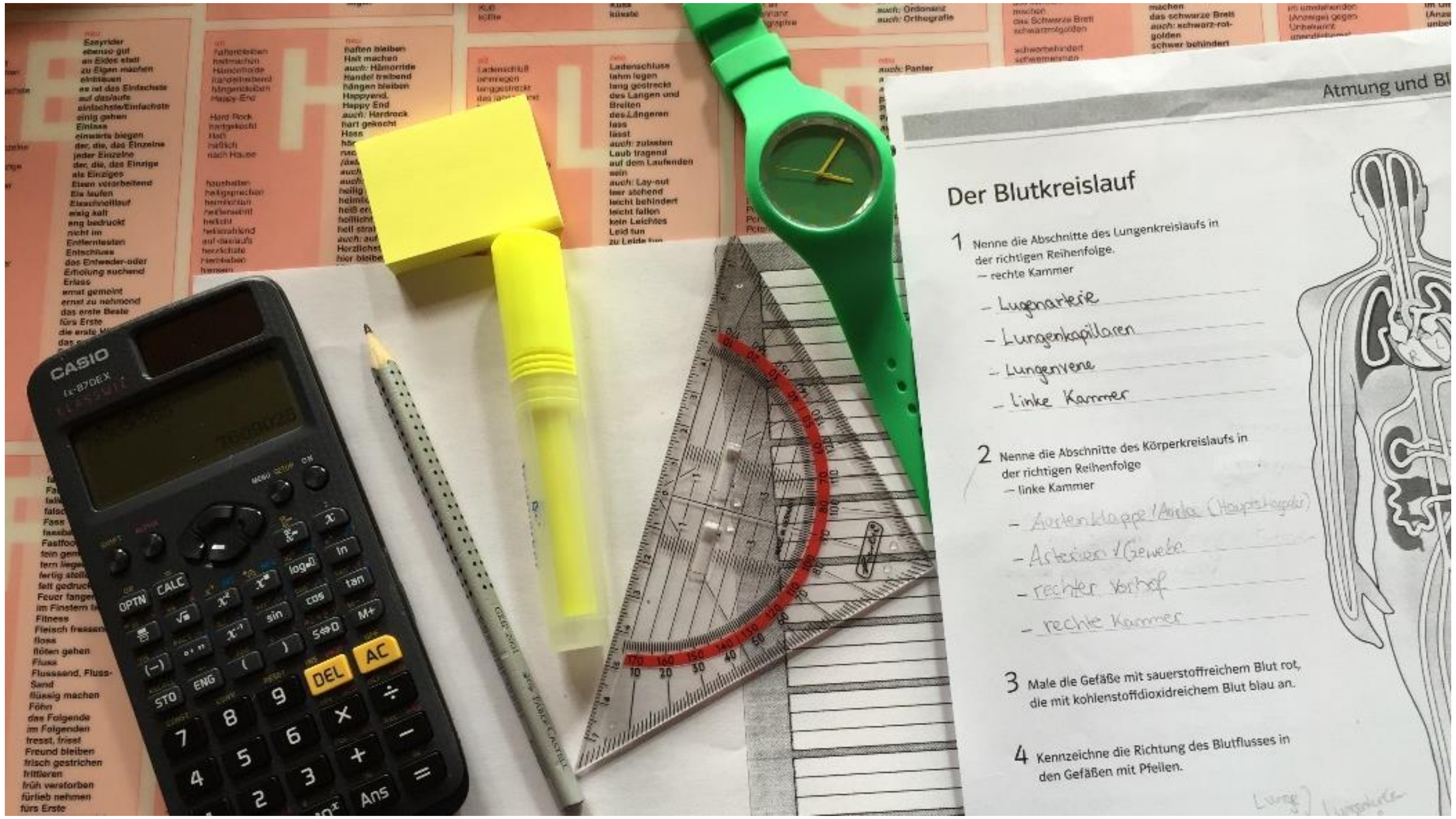


Charles Q. Choi. 2009. Human Evolution: The Origin of Tool Use Live Science, November 11, 2009.

Weir, A.A., Chappell, J. and Kacelnik, A., 2002. Shaping of hooks in New Caledonian crows. *Science*, *297*(5583), pp.981-981.

Tools to Augment Cognition

Tool use, intelligence, and evolution



CASIO
Atmung und Bl
Der Blutkreislauf
1 Nenne die Abschnitte des Lungenkreislaufs in der richtigen Reihenfolge.
– rechte Kammer
– Lugenarterie
– Lungenkapillaren
– Lungenvene
– linke Kammer
2 Nenne die Abschnitte des Körperkreislaufs in der richtigen Reihenfolge
– linke Kammer
– Aortenklappe / Aorta (Hauptschlagader)
– Arterien / Gewebe
– rechter Vorhof
– rechte Kammer
3 Male die Gefäße mit sauerstoffreichem Blut rot, die mit kohlenstoffdioxidreichem Blut blau an.
4 Kennzeichne die Richtung des Blutflusses in den Gefäßen mit Pfeilen.

It is not Simple to Make Good Interactive Products!

Examples are all around us…, some basic misconceptions

- If I (the developer) can use it, everyone can use it
- If our non-technical staff can use it, everyone can use it
- Good designs/user interfaces are applied common sense
- A system is usable if all norms and style guidelines are met

That is why a process is required.

How it does NOT work

…. Don’t fall for this

- Usability tests at the end when the product is ready and needs to be shipped
- Designing a new and pretty skin or look & feel to a product
- Introducing HCI issues after the system architecture and the foundations are completed
- Have the developers and testers test the usability of the system

Creating Interactive Systems

Structured process along the development cycle

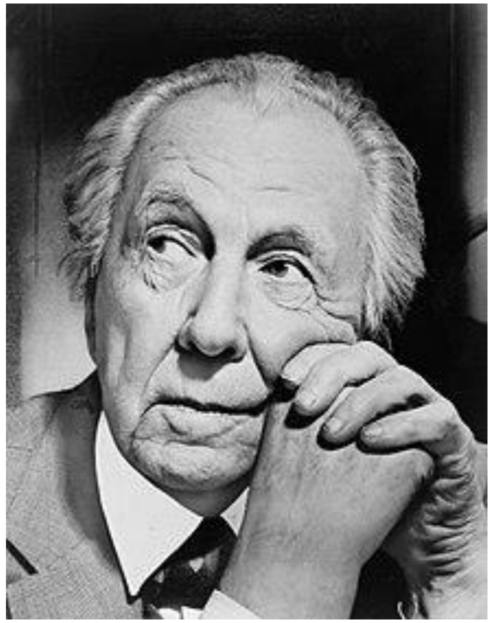
- An interior designer can not make a great house if the architect and engineers forgot windows, set the doors at the wrong locations, and created an unsuitable room layout.
- Creating the user interface at the end of the project will not work – it has been considered from the very beginning.

Make Mistake Early and Recognize Them!

Early involvement is key

Frank Lloyd Wright

“You can use an eraser on the drafting table or a sledgehammer on the construction site.”



Slide Adapted from Lewis Chuang

Focus on Technology or User?

Cylinder or Disk?

GRAPHOPHONE
GRAPHOPHONE
MICRO
MAISON
DIETRICH

- Thomas Alva Edison (1880)
- Emile Berliner (1887)

Slide Adapted from Lewis Chuang

Focus on Technology or User?

Cylinder or Disk?

Benefits of the disc

- mass-production
- less storage space
- easier shipping
- allow double-sided recordings
- better marketing (famous artistes)

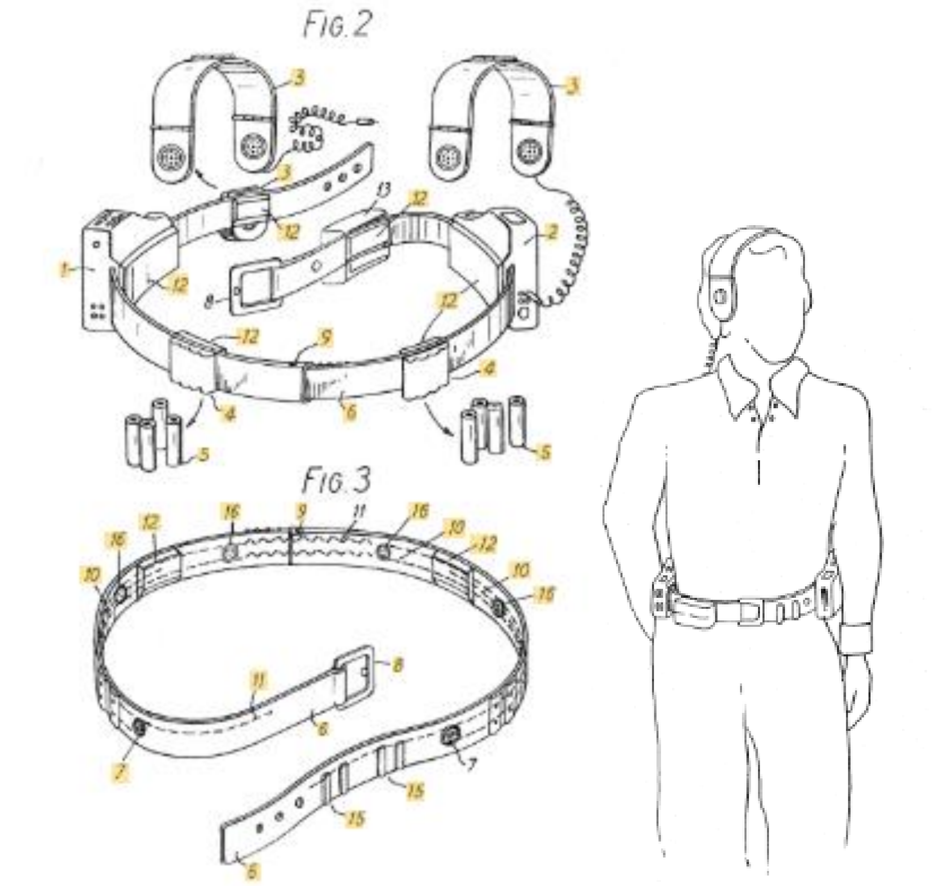
- Emile Berliner (1887)

Slide Adapted from Lewis Chuang

Personal and Mobile Stereo Music Players

Mini-Exercise: Stereo-Belt vs. Walkman

U.S. Patent Oct. 25, 1983 Sheet 2 of 3 4,412,106



Sony TPS-L2 (1979) by Binarysequence
CC BY-SA Wikimedia

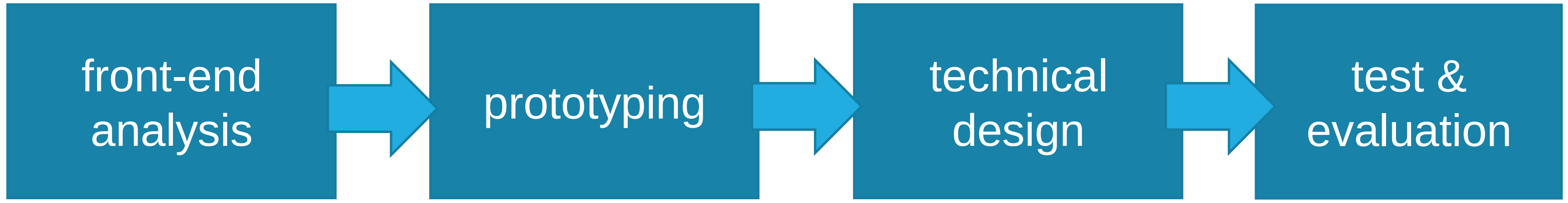
Design products

- Interfaces
- interactions
- environments
- tasks and processes
- user experience
- organizational structures
- society
- needs

Why to design?

Design objectives

- create new products, systems, & experiences
- improve existing products
- ensure safety: features and training
- develop performance support
- develop methods for training and assessment
- guide team and organization formation



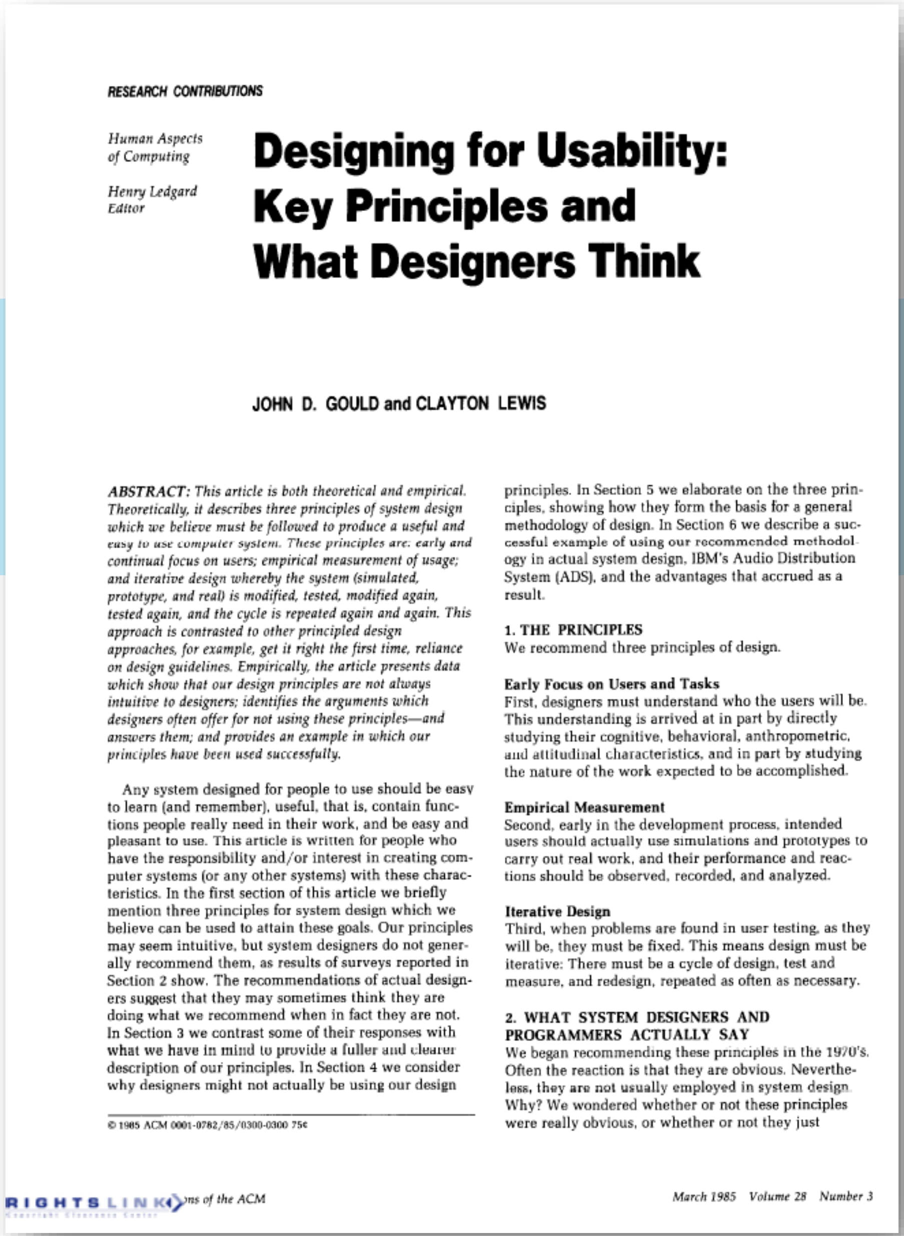
Slide Adapted from Lewis Chuang

Desiging for Usability

Key Principles by Gould and Lewis, 1985

- "Any system designed for people to use should be **easy to learn** (and remember], **useful**, that is, contain functions **people really need** in their work, and be **easy and pleasant to use**."
- "…three principles of system design which we believe must be followed to produce a useful and easy to use computer system […]
  - Early Focus on Users and Tasks
  - Empirical Measurement
  - Iterative Design"

Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM*, *28*(3), 300-311.



RESEARCH CONTRIBUTIONS

Human Aspects of Computing

Henry Ledgard Editor

Designing for Usability: Key Principles and What Designers Think

JOHN D. GOULD and CLAYTON LEWIS

ABSTRACT: This article is both theoretical and empirical. Theoretically, it describes three principles of system design which we believe must be followed to produce a useful and easy to use computer system. These principles are: early and continual focus on users; empirical measurement of usage; and iterative design whereby the system (simulated, prototype, and real) is modified, tested, modified again, tested again, and the cycle is repeated again and again. This approach is contrasted to other principled design approaches, for example, get it right the first time, reliance on design guidelines. Empirically, the article presents data which show that our design principles are not always intuitive to designers; identifies the arguments which designers often offer for not using these principles—and answers them; and provides an example in which our principles have been used successfully.

Any system designed for people to use should be easy to learn (and remember), useful, that is, contain functions people really need in their work, and be easy and pleasant to use. This article is written for people who have the responsibility and/or interest in creating computer systems (or any other systems) with these characteristics. In the first section of this article we briefly mention three principles for system design which we believe can be used to attain these goals. Our principles may seem intuitive, but system designers do not generally recommend them, as results of surveys reported in Section 2 show. The recommendations of actual designers suggest that they may sometimes think they are doing what we recommend when in fact they are not. In Section 3 we contrast some of their responses with what we have in mind to provide a fuller and clearer description of our principles. In Section 4 we consider why designers might not actually be using our design principles. In Section 5 we elaborate on the three principles, showing how they form the basis for a general methodology of design. In Section 6 we describe a successful example of using our recommended methodology in actual system design, IBM's Audio Distribution System (ADS), and the advantages that accrued as a result.

© 1985 ACM 0001-0782/85/0300-0300 75¢

1. THE PRINCIPLES

We recommend three principles of design.

Early Focus on Users and Tasks

First, designers must understand who the users will be. This understanding is arrived at in part by directly studying their cognitive, behavioral, anthropometric, and attitudinal characteristics, and in part by studying the nature of the work expected to be accomplished.

Empirical Measurement

Second, early in the development process, intended users should actually use simulations and prototypes to carry out real work, and their performance and reactions should be observed, recorded, and analyzed.

Iterative Design

Third, when problems are found in user testing, as they will be, they must be fixed. This means design must be iterative: There must be a cycle of design, test and measure, and redesign, repeated as often as necessary.

2. WHAT SYSTEM DESIGNERS AND PROGRAMMERS ACTUALLY SAY

We began recommending these principles in the 1970's. Often the reaction is that they are obvious. Nevertheless, they are not usually employed in system design. Why? We wondered whether or not these principles were really obvious, or whether or not they just

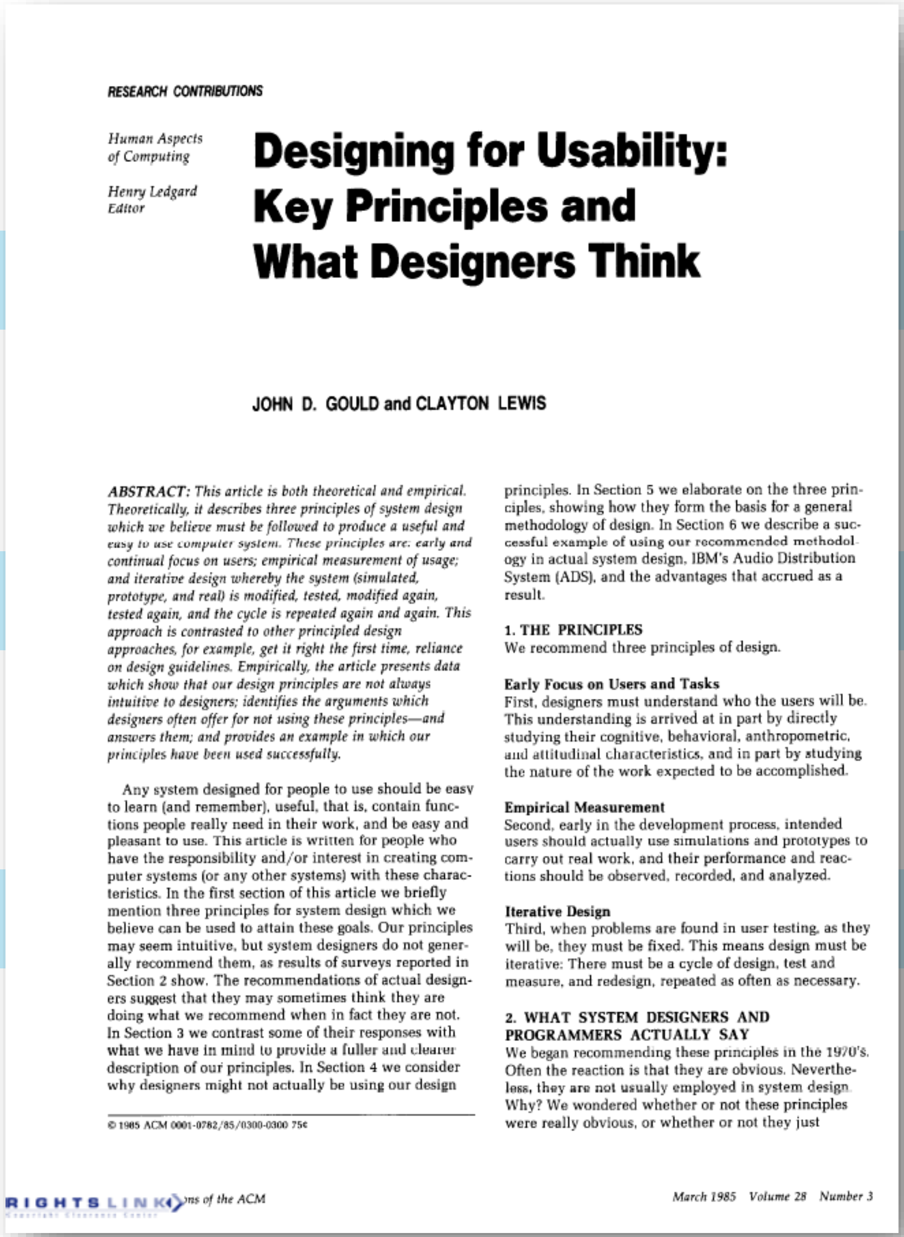
RIGHTSLINK ons of the ACM

March 1985 Volume 28 Number 3

Desiging for Usability

Key Principles by Gould and Lewis, 1985

- **“Early Focus on Users and Tasks**
  First, designers must understand **who the users** will be. […] studying their cognitive, behavioral, anthropometric, and attitudinal characteristics, and in part by studying the nature of the work expected to be accomplished.
- **Empirical Measurement**
  Second, early in the development process, intended users should actually use simulations and prototypes to carry out real work, and their performance and reactions should be observed, recorded, and analyzed.
- **Iterative Design**
  Third, when problems are found in user testing, as they will be, they must be fixed. This means design must be iterative: There must be a cycle of design, test and measure, and redesign, repeated as often as necessary.”


RESEARCH CONTRIBUTIONS

Human Aspects of Computing

Henry Ledgard Editor

Designing for Usability: Key Principles and What Designers Think

JOHN D. GOULD and CLAYTON LEWIS

ABSTRACT: This article is both theoretical and empirical. Theoretically, it describes three principles of system design which we believe must be followed to produce a useful and easy to use computer system. These principles are: early and continual focus on users; empirical measurement of usage; and iterative design whereby the system (simulated, prototype, and real) is modified, tested, modified again, tested again, and the cycle is repeated again and again. This approach is contrasted to other principled design approaches, for example, get it right the first time, reliance on design guidelines. Empirically, the article presents data which show that our design principles are not always intuitive to designers; identifies the arguments which designers often offer for not using these principles—and answers them; and provides an example in which our principles have been used successfully.

Any system designed for people to use should be easy to learn (and remember), useful, that is, contain functions people really need in their work, and be easy and pleasant to use. This article is written for people who have the responsibility and/or interest in creating computer systems (or any other systems) with these characteristics. In the first section of this article we briefly mention three principles for system design which we believe can be used to attain these goals. Our principles may seem intuitive, but system designers do not generally recommend them, as results of surveys reported in Section 2 show. The recommendations of actual designers suggest that they may sometimes think they are doing what we recommend when in fact they are not. In Section 3 we contrast some of their responses with what we have in mind to provide a fuller and clearer description of our principles. In Section 4 we consider why designers might not actually be using our design principles. In Section 5 we elaborate on the three principles, showing how they form the basis for a general methodology of design. In Section 6 we describe a successful example of using our recommended methodology in actual system design, IBM's Audio Distribution System (ADS), and the advantages that accrued as a result.

© 1985 ACM 0001-0782/85/0300-0300 75¢

1. THE PRINCIPLES
We recommend three principles of design.

Early Focus on Users and Tasks
First, designers must understand who the users will be. This understanding is arrived at in part by directly studying their cognitive, behavioral, anthropometric, and attitudinal characteristics, and in part by studying the nature of the work expected to be accomplished.

Empirical Measurement
Second, early in the development process, intended users should actually use simulations and prototypes to carry out real work, and their performance and reactions should be observed, recorded, and analyzed.

Iterative Design
Third, when problems are found in user testing, as they will be, they must be fixed. This means design must be iterative: There must be a cycle of design, test and measure, and redesign, repeated as often as necessary.

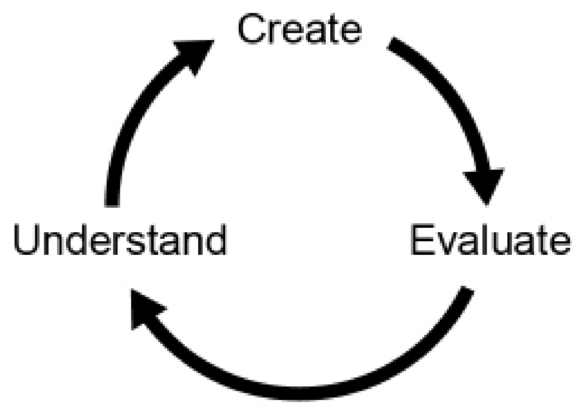
2. WHAT SYSTEM DESIGNERS AND PROGRAMMERS ACTUALLY SAY
We began recommending these principles in the 1970's. Often the reaction is that they are obvious. Nevertheless, they are not usually employed in system design. Why? We wondered whether or not these principles were really obvious, or whether or not they just

RIGHTSLINK ons of the ACM *March 1985 Volume 28 Number 3*

Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM*, *28*(3), 300-311.

Four basic activities of interaction design

- **Identifying needs** and establishing requirements for the user experience
- **Developing alternative designs** that meet those requirements
- **Building interactive versions** of the designs
- **Evaluating** what is being built throughout the process and the user experience it offers



Did you understand this block?

Can you answer these questions?

- Name the key principles for usable systems according to Gould and Lewis
- Give an example of an interactive systems design, that shows the importance of the statement “Make Mistake Early and Recognize Them”
- Describe typical design products and design objectives.
- What is the difference between design objectives and design products?
- What are the basic activities in a human centered design process?

References

- Pavel, A. (1983). U.S. Patent No. 4,412,106. Washington, DC: U.S. Patent and Trademark Office. http://www.google.com/patents/US4412106
- Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. Communications of the ACM, 28(3), 300-311.

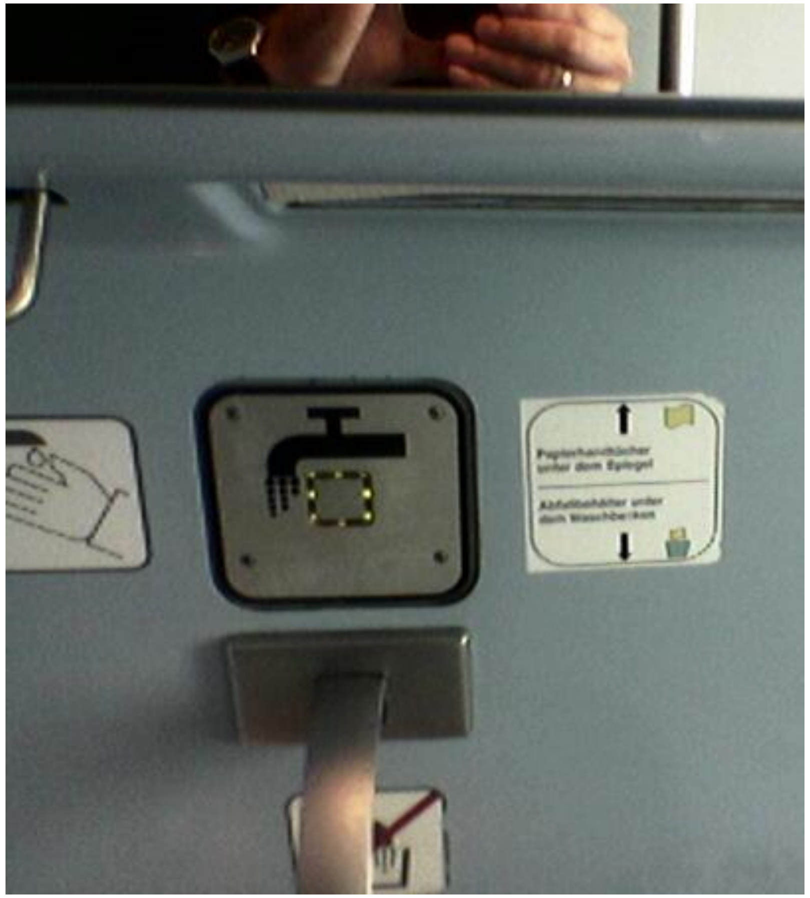
Mental Modal and Metaphors

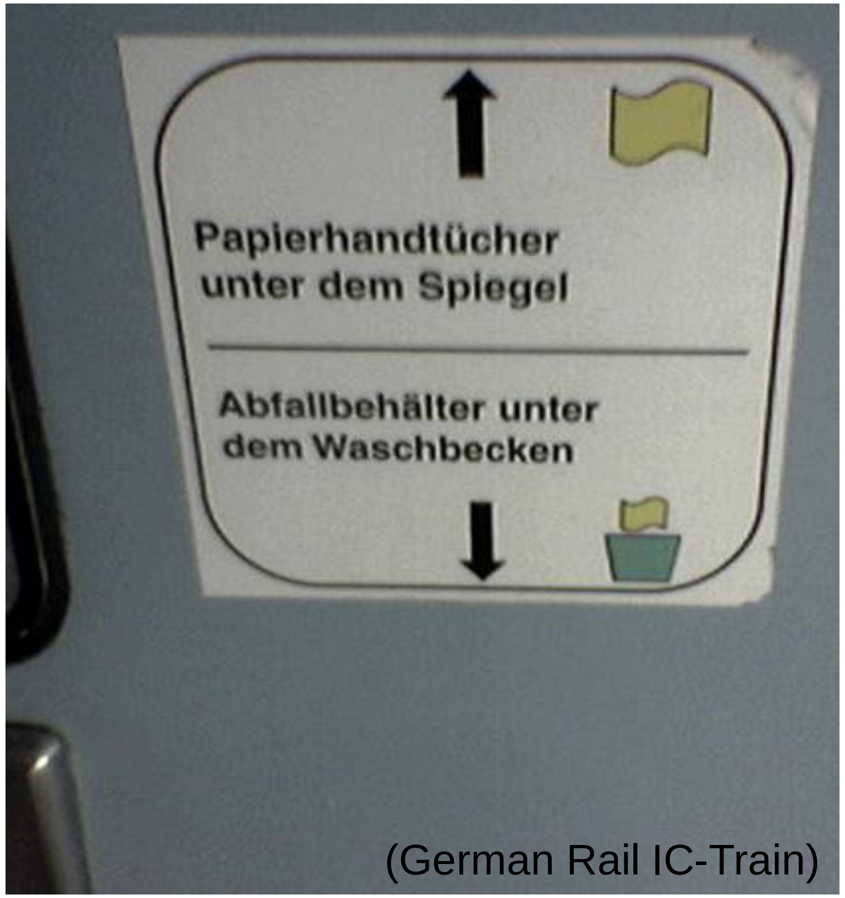
Learning Goals

- Understand …
  - Why it is hard for developers to spot usability issues.
  - How mental model, conceptual model, and metaphor relate
- Be able to explain
  - the terms mental and conceptual model,
  - The difference between mental and conceptual model,
  - what metaphors are, why they are useful, and where their limitations are.

Why is Something Easy to Use?

Signs and explanations for things that are usually obvious are an indicator for a potential design problem.




Papierhandtücher unter dem Spiegel
Abfallbehälter unter dem Waschbecken

(German Rail IC-Train)

What is a Mental Model

How the user thinks it works

‘In human-computer interaction research, the notion of "**mental models**" has come to be a very general catch-phrase for **anything having to do with end users' knowledge** of an application (van der Veer, 1990). There is a feeling that **if we could "capture" mental models, then we could build good interfaces** […]
But many […] are much less convinced of the alleged benefits of mental models and of our ability to use them for reasoning or other complex cognition. […] Researchers who have investigated mental images have been struck by how **incomplete and inflexible they** are…’

Nardi, B. A., & Zarmer, C. L. (1993). Beyond models and metaphors: Visual formalisms in user interface design. *Journal of Visual Languages & Computing*, *4*(1), 5-33.

HEWLETT PACKARD

Beyond Models and Metaphors: Visual Formalisms in User Interface Design

Bonnie A. Nardi, Craig L. Zarmer
Software and Systems Laboratory
HPL-90-149
September, 1990

models; metaphors; visual formalisms; user interface design

The user interface has both syntactic functions - supplying commands and arguments to programs - and semantic functions - visually presenting application semantics and supporting problem solving cognition. In this paper we argue that though both functions are important, it is time to devote more resources to the problems of the *semantic* interface. Complex problem solving activities, e.g. for design and analysis tasks, benefit from clear visualizations of application semantics in the user interface. Designing the semantic interface requires computational building blocks capable of representing and visually presenting application semantics in a clear, precise way. We argue that neither mental models nor metaphors provide a basis for designing and implementing such building blocks, but that *visual formalisms* do. We compare the benefits of mental models, metaphors and visual formalisms as the basis for designing the user interface, with particular attention to the practical solutions each provides to application developers. We describe our implementation of a visual formalism to show the potential for visual formalisms to serve as reusable computational structures that support the development of semantically rich applications.

(c) Copyright Hewlett-Packard Company 1990

Mental Model

How the user reasons and understands

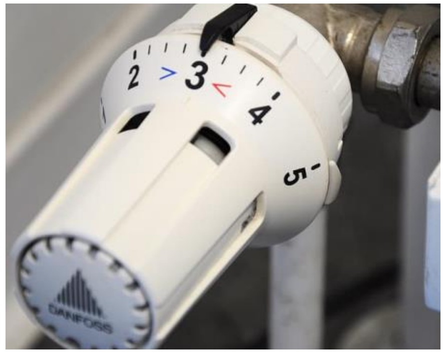
- Kenneth Craik (1943) ‘the mind constructs "small-scale models" of reality that it uses to anticipate events, to reason, and to underlie explanation’
- Users acquire mental models by
  - Interaction / observation
  - Explanation
- Two types
  - Functional – users know what to do, but not why
  - Structural – users know why to do something

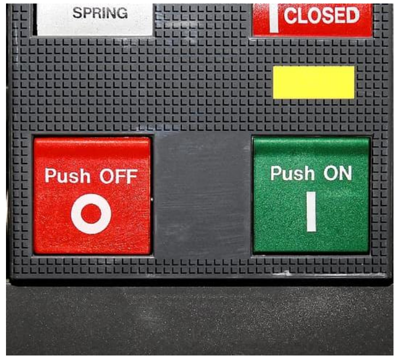
William Hudson. Mental Models, Metaphor and Design (2003).UK UPA and HCI2003
http://www.syntagm.co.uk/design/articles/mmmad.pdf

Mental Model

Example: Central Heating

- You came back and it is only 15°C in your room. You want a room temperature of 21°C. What do you do?
    1. You put the thermostat to 21°C and wait
    2. You put the thermostat to max (=35°) and switch it back to 21° once it is warm?
- What different mental models do people have who chose strategy 1 or 2?





William Hudson. Mental Models, Metaphor and Design (2003).UK UPA and HCI2003
http://www.syntagm.co.uk/design/articles/mmmad.pdf

Mental Model

Definition by Jakob Nielson

’A mental model is **what the user believes** about the system at hand.’

- ‘A mental model is based on **belief, not facts**: that is, it's a model of what users know (or think they know) about a system such as your website. Hopefully, users' thinking is closely related to reality because they **base their predictions** about the system on their mental models and thus plan their future **actions** based on how that model predicts the appropriate course.’
- ‘It's a **prime goal for designers** to make the user interface **communicate the system's basic nature** well enough that users form reasonably accurate (and thus useful) mental models.’

Jakob Nielsen. Mental Models on October 17, 2010
https://www.nngroup.com/articles/mental-models/

Mental Model

Definition by Jakob Nielson

- ‘Individual users each have their own mental model. […] different users might construct different mental models of the same user interface. Further, one of usability's big dilemmas is the common **gap between designers' and users' mental models**. Because designers know too much, they form wonderful mental models of their own creations…, leading them to believe that each feature is easy to understand. Users' mental models of the UI are likely to be somewhat more deficient, making it more likely for people to make mistakes and find the design much more difficult to use.

Jakob Nielsen. Mental Models on October 17, 2010
https://www.nngroup.com/articles/mental-models/

Conceptual Model

The model the designers wants the user to have

- “A conceptual model is a high-level description of how a system is organized and operates.”
  Johnson, J., & Henderson, A. (2002). Conceptual models: begin by designing what to design. *interactions*, *9*(1), 25-32.

- The conceptual model
  - is deliberately designed
  - it allows to user to understand and operate the UI
  - it draws on prior knowledge of the user
  - is communicated through the interface and interaction design

William Hudson. Mental Models, Metaphor and Design (2003).UK UPA and HCI2003
http://www.syntagm.co.uk/design/articles/mmmad.pdf

Designer Model, User Model, and System Model

Don Norman

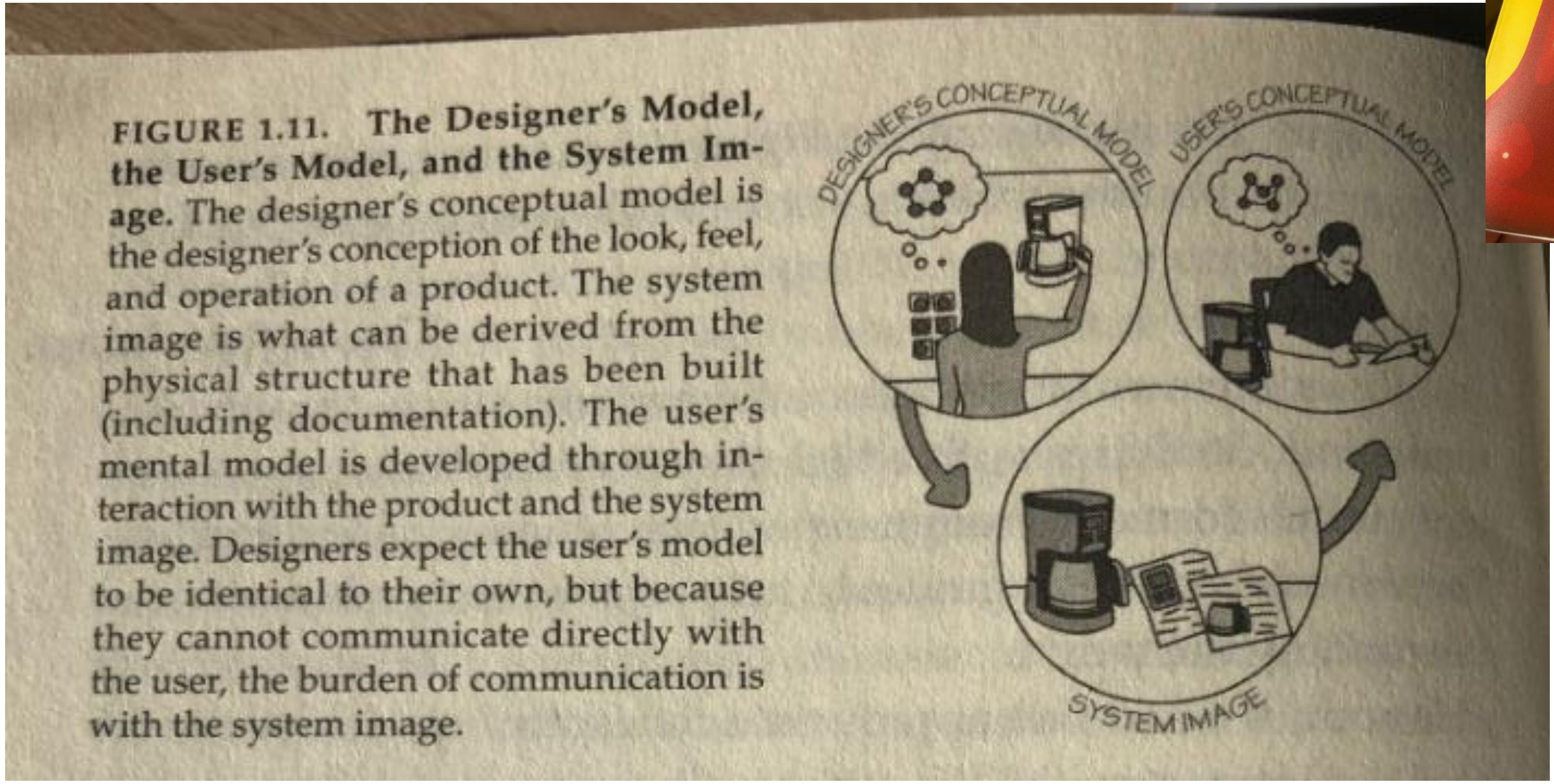
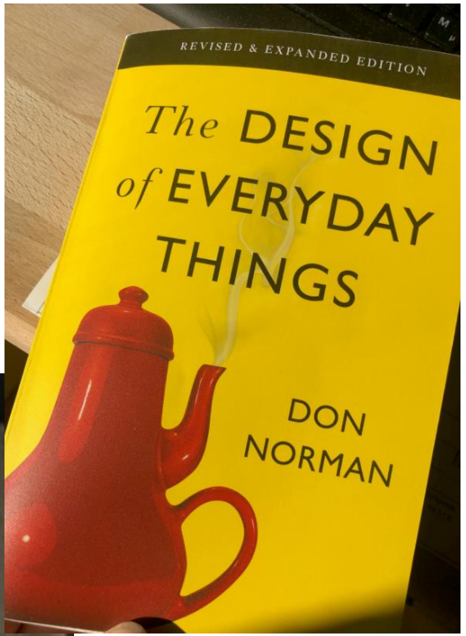


FIGURE 1.11. The Designer's Model, the User's Model, and the System Image. The designer's conceptual model is the designer's conception of the look, feel, and operation of a product. The system image is what can be derived from the physical structure that has been built (including documentation). The user's mental model is developed through interaction with the product and the system image. Designers expect the user's model to be identical to their own, but because they cannot communicate directly with the user, the burden of communication is with the system image.

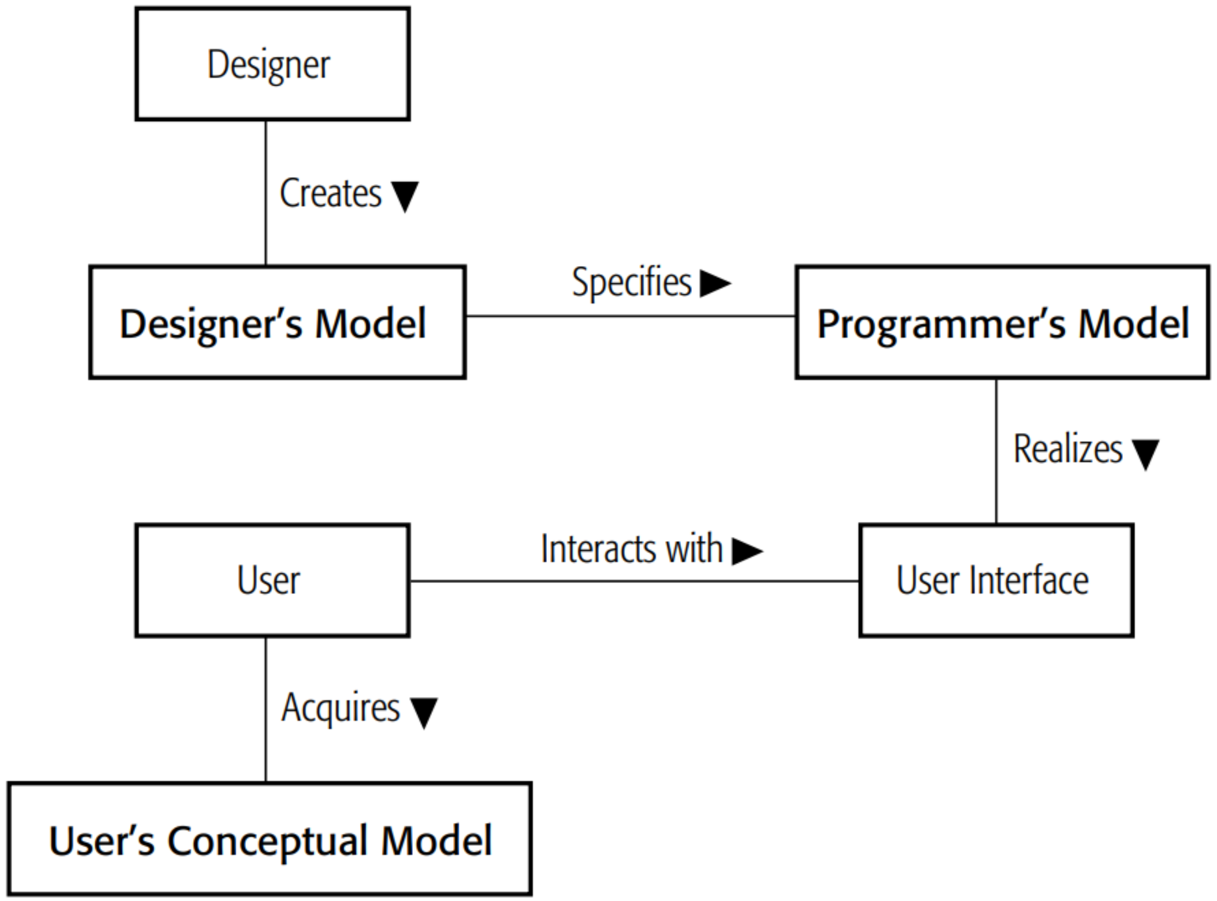
DESIGNER'S CONCEPTUAL MODEL
USER'S CONCEPTUAL MODEL
SYSTEM IMAGE



REVISED & EXPANDED EDITION
The DESIGN of EVERYDAY THINGS
DON NORMAN

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Models – Designer, Programmer, User



Hudson, W. (2001). Toward unified models in user-centered and object-oriented design. Object Modeling and User Interface Design: Designing Interactive Systems, 313-362.

Clarification of Terms

Different things with the same intention

- **Mental Model** = User Model = User’s Conceptual Model
- **Conceptual Model** = Designer Model
- **System Model** = Programmer’s Conceptual Model = Programmer’s Model = Implementation Model

Metaphors

Build on what the users know

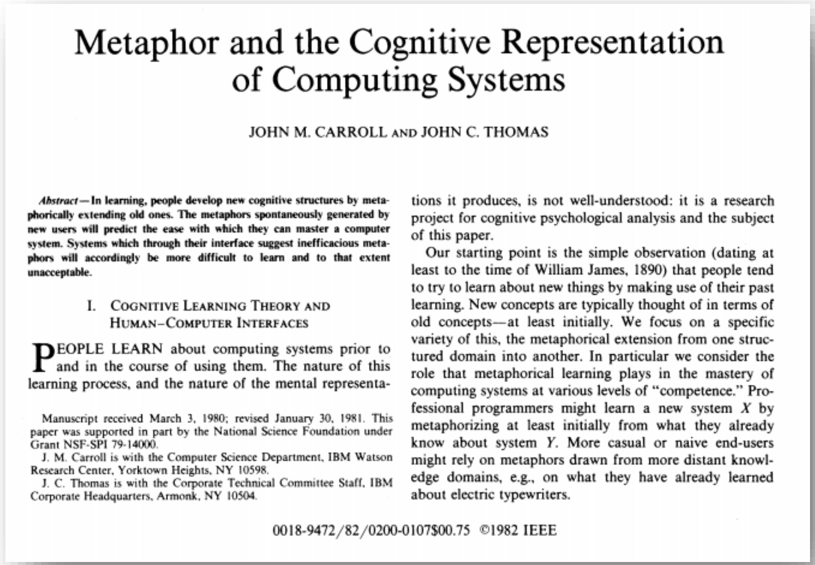
‘Given the analogy, the new **user can draw upon his knowledge about the familiar situation** in order to reason about the workings of the mysterious new computer system. For example, if the new user wants to **understand about how the computer file system** works, he need only **think about how an office filing cabinet works** and then carry over this same way of thinking to the computer file system’

Frank Halasz and Thomas P. Moran. 1982. Analogy considered harmful. In Proceedings of the 1982 Conference on Human Factors in Computing Systems (CHI ’82). Association for Computing Machinery, New York, NY, USA, 383–386. DOI:https://doi.org/10.1145/800049.801816

Metaphors

Build on what the users know

‘If people employ metaphors in learning about computing systems, the designers of those systems should anticipate and support likely **metaphorical constructions to increase the ease of learning and using** the system.’



Metaphor and the Cognitive Representation of Computing Systems

JOHN M. CARROLL AND JOHN C. THOMAS

Abstract—**In learning, people develop new cognitive structures by metaphorically extending old ones. The metaphors spontaneously generated by new users will predict the ease with which they can master a computer system. Systems which through their interface suggest inefficacious metaphors will accordingly be more difficult to learn and to that extent unacceptable.**

I. COGNITIVE LEARNING THEORY AND HUMAN–COMPUTER INTERFACES

PEOPLE LEARN about computing systems prior to and in the course of using them. The nature of this learning process, and the nature of the mental representations it produces, is not well-understood: it is a research project for cognitive psychological analysis and the subject of this paper.

Our starting point is the simple observation (dating at least to the time of William James, 1890) that people tend to try to learn about new things by making use of their past learning. New concepts are typically thought of in terms of old concepts—at least initially. We focus on a specific variety of this, the metaphorical extension from one structured domain into another. In particular we consider the role that metaphorical learning plays in the mastery of computing systems at various levels of “competence.” Professional programmers might learn a new system *X* by metaphorizing at least initially from what they already know about system *Y*. More casual or naive end-users might rely on metaphors drawn from more distant knowledge domains, e.g., on what they have already learned about electric typewriters.

Manuscript received March 3, 1980; revised January 30, 1981. This paper was supported in part by the National Science Foundation under Grant NSF-SPI 79-14000.

J. M. Carroll is with the Computer Science Department, IBM Watson Research Center, Yorktown Heights, NY 10598.

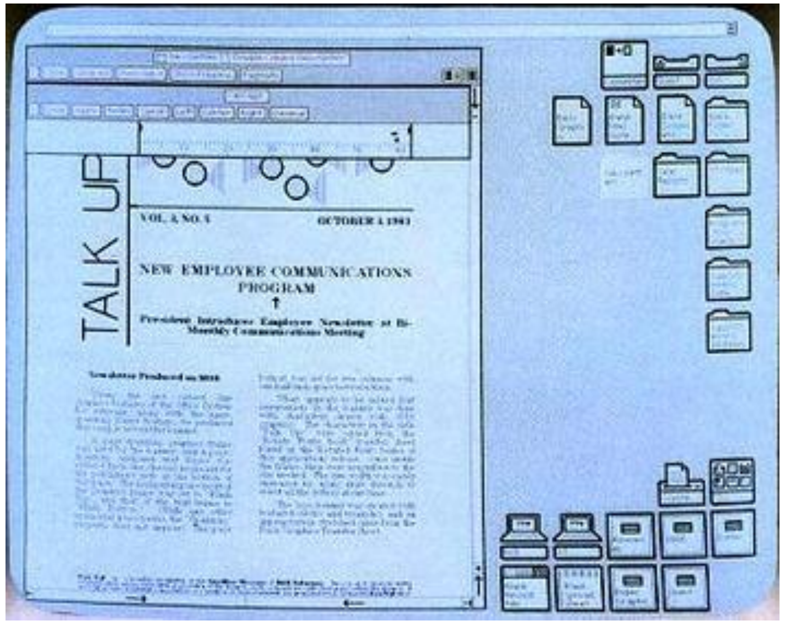
J. C. Thomas is with the Corporate Technical Committee Staff, IBM Corporate Headquarters, Armonk, NY 10504.

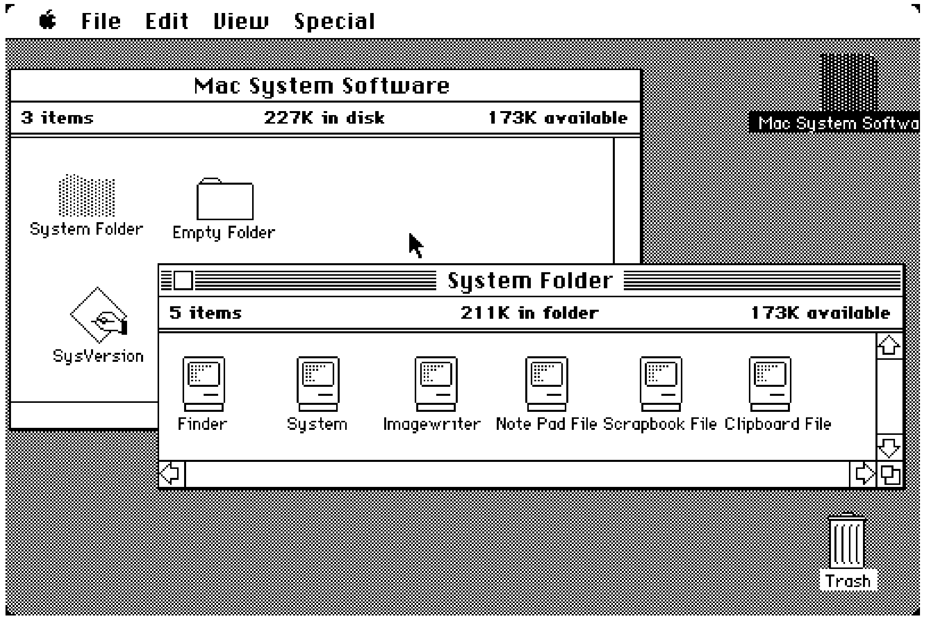
0018-9472/82/0200-0107$00.75 ©1982 IEEE

Carroll, J. M., & Thomas, J. C. (1982). Metaphor and the cognitive representation of computing systems. IEEE Transactions on systems, man, and cybernetics, 12(2), 107-116.

Metaphors - Examples

Desktop Metaphor – Alan Kay at Xerox Parc

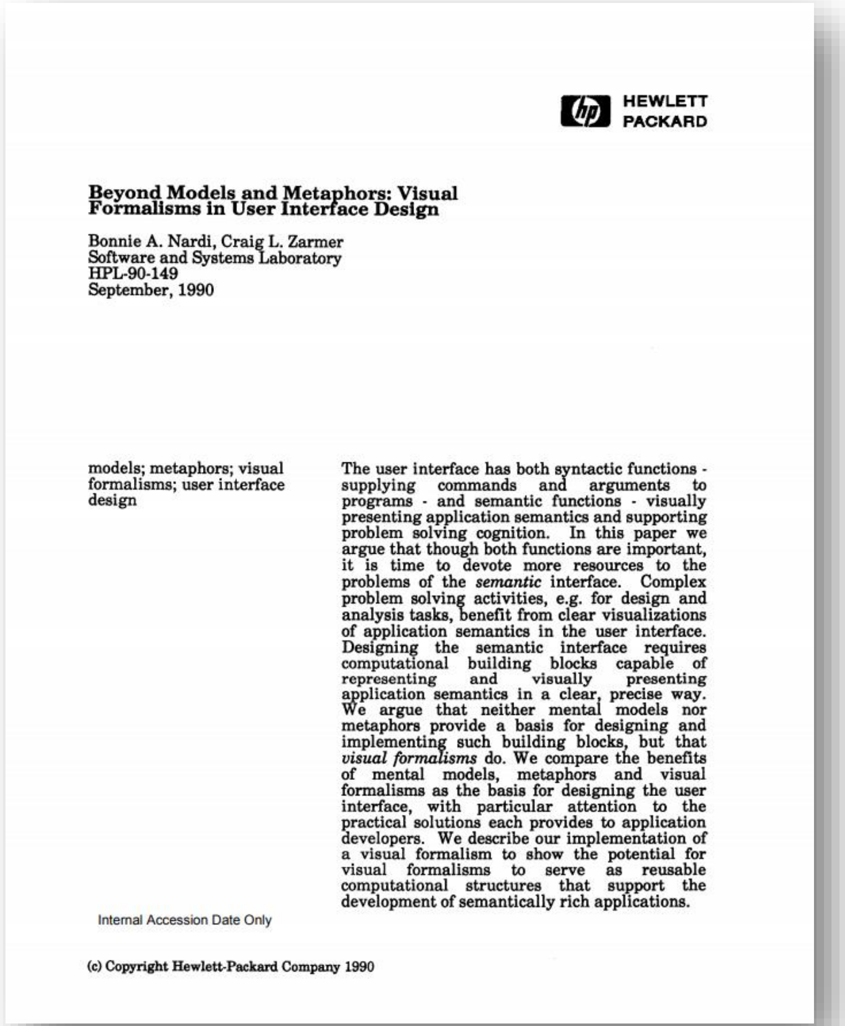




Metaphors

… are also causing problems

‘However, metaphors suffer from numerous problems that make them unsuitable for expressing rich application semantics, and inappropriate for the reusable computational structures we seek. […] Metaphors are slippery things, and not just because **they contain irrelevancies** (do we set the trash can out for garbage pick-up on Friday mornings?) and **incompletenesses** with respect to the domain they are meant to represent. […] Metaphors tempt us to **over-generalize** and to forget distinctions that we should be remembering.’


HEWLETT PACKARD

Beyond Models and Metaphors: Visual Formalisms in User Interface Design

Bonnie A. Nardi, Craig L. Zarmer
Software and Systems Laboratory
HPL-90-149
September, 1990

models; metaphors; visual formalisms; user interface design

The user interface has both syntactic functions - supplying commands and arguments to programs - and semantic functions - visually presenting application semantics and supporting problem solving cognition. In this paper we argue that though both functions are important, it is time to devote more resources to the problems of the *semantic* interface. Complex problem solving activities, e.g. for design and analysis tasks, benefit from clear visualizations of application semantics in the user interface. Designing the semantic interface requires computational building blocks capable of representing and visually presenting application semantics in a clear, precise way. We argue that neither mental models nor metaphors provide a basis for designing and implementing such building blocks, but that *visual formalisms* do. We compare the benefits of mental models, metaphors and visual formalisms as the basis for designing the user interface, with particular attention to the practical solutions each provides to application developers. We describe our implementation of a visual formalism to show the potential for visual formalisms to serve as reusable computational structures that support the development of semantically rich applications.

Internal Accession Date Only

(c) Copyright Hewlett-Packard Company 1990

Nardi, B. A., & Zarmer, C. L. (1993). Beyond models and metaphors: Visual formalisms in user interface design. Journal of Visual Languages & Computing, 4(1), 5-33.

Metaphors

Mini Exercise

- Search for examples of user interface designs that draw on metaphors?
- Where do these metaphors work well?
- Where are they limited?

- A starting point for your search could be the early Book shelf app on iPads / iOS

Did you understand this block?

Can you answer these questions?

- What is a mental model?
- What is the difference between functional and structural mental models?
- What is a conceptual model?
- What is the difference between a mental and a conceptual model?
- What are metaphors? What are their advantages and what are their limitations?

References

- Nardi, B. A., & Zarmer, C. L. (1993). Beyond models and metaphors: Visual formalisms in user interface design. Journal of Visual Languages & Computing, 4(1), 5-33.
- William Hudson. Mental Models, Metaphor and Design (2003).UK UPA and HCI2003. http://www.syntagm.co.uk/design/articles/mmmad.pdf
- Jakob Nielsen. Mental Models on October 17, 2010. https://www.nngroup.com/articles/mental-models/
- Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.
- Hudson, W. (2001). Toward unified models in user-centered and object-oriented design. Object Modeling and User Interface Design: Designing Interactive Systems, 313-362.
- Frank Halasz and Thomas P. Moran. 1982. Analogy considered harmful. In Proceedings of the 1982 Conference on Human Factors in Computing Systems (CHI '82). Association for Computing Machinery, New York, NY, USA, 383–386. DOI:https://doi.org/10.1145/800049.801816
- Carroll, J. M., & Thomas, J. C. (1982). Metaphor and the cognitive representation of computing systems. IEEE Transactions on systems, man, and cybernetics, 12(2), 107-116.

License

For more content see: https://hci-lecture.de

Human Centered Design

ISO 9241-210

Learning Goals

- Understand …
  - The terms user centered design (UCD) and human centered design (HCD)
  - The general structure of the ISO 9241 Ergonomics of human–system interaction
  - The ISO 9241-210 Human-centered design for interactive systems
  - The problems of user centered design
- Be able to explain
  - Rationale for adopting human-centered design according to ISO 9241-210
  - Principles and activities of human-centered design according to ISO 9241-210
  - The separation between interaction design and technical realization

User Centered Design vs. Human Centered Design

- The terms are used interchangeable
- Human centered is the more modern term
- With the term “human centered” the focus should be on the person as a whole not only the user

ISO Standards and DIN Norms

ISO 9241
Ergonomics of human–system interaction

Many parts with specific focus

- Part 1: General introduction
- Part 2: Guidance on task requirements
- Part 3: Visual display requirements
- Part 4: Keyboard requirements
- …
- Part 110: Dialogue principles
- Part 151: Guidance on World Wide Web user interfaces
- Part 171: Guidance on software accessibility
- Part 210: Human-centered design for interactive systems
- …
- Part 420: Selection procedures for physical input devices
- Part 910: Framework for tactile and haptic interaction
- Part 920: Guidance on tactile and haptic interactions

INTERNATIONAL STANDARD

ISO 9241-210

First edition
2010-03-15

Ergonomics of human–system interaction —

Part 210:
Human-centred design for interactive systems

Ergonomie de l'interaction homme–système —

Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs

Reference number
ISO 9241-210:2010(E)

© ISO 2010

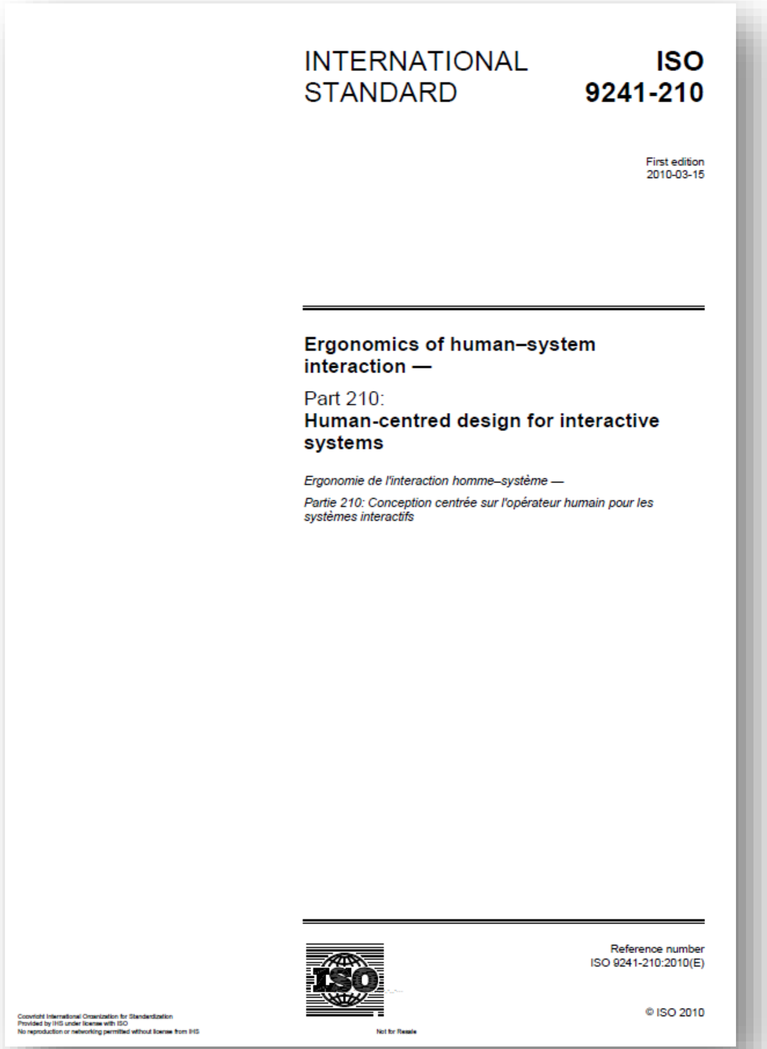
ISO 9241 Ergonomics of human–system interaction

ISO 9241
Ergonomics of human–system interaction

Part 210: Human-centered design for interactive systems

Table of Contents

1. Scope
2. Terms and definitions
3. Rationale for adopting human-centered design
4. Principles of human-centered design
5. Planning human-centered design
6. Human-centered design activities
7. Sustainability and human-centered design
8. Conformance



INTERNATIONAL STANDARD

ISO 9241-210

First edition 2010-03-15

Ergonomics of human–system interaction —

Part 210:
Human-centred design for interactive systems

Ergonomie de l'interaction homme–système —

Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs

Reference number
ISO 9241-210:2010(E)

© ISO 2010

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

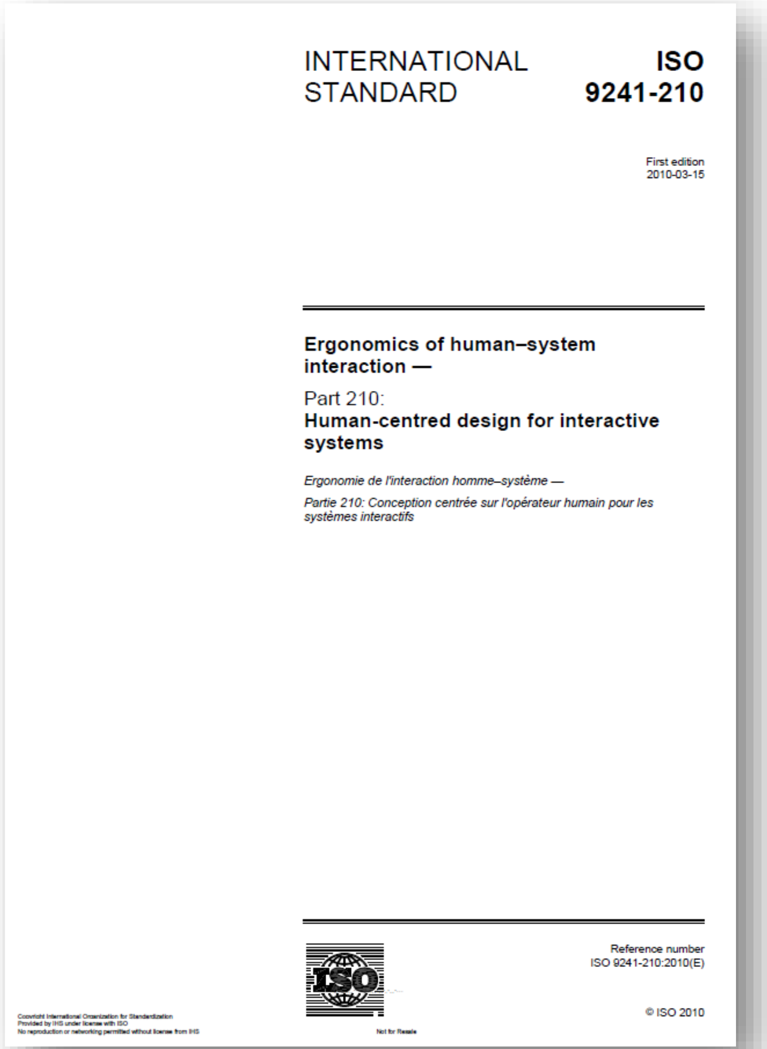
ISO 9241-210:2019(EN) Human-centered design for interactive systems

ISO 9241-210

Human-centered design for interactive systems

3. Rationale for adopting human-centered design

- “Using a human-centered approach to design and development has substantial **economic and social benefits for users, employers and suppliers**. Highly usable systems and products tend to be more **successful both technically and commercially**.“
- “Systems designed using human-centred methods improve quality, for example, by:
  - a) increasing the **productivity** of users and the operational efficiency of organizations;
  - b) being **easier to understand and use**, thus reducing training […] costs;
  - c) increasing usability for **people with a wider range of capabilities** […]
  - d) improving user experience;
  - e) **reducing** discomfort and **stress**;
  - f) providing a **competitive advantage** […]
  - g) contributing towards sustainability objectives.”


INTERNATIONAL STANDARD

ISO 9241-210

First edition
2010-03-15

Ergonomics of human–system interaction —

Part 210:
Human-centred design for interactive systems

Ergonomie de l'interaction homme–système —

Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs

Reference number
ISO 9241-210:2010(E)

© ISO 2010

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9241-210:2019(EN) Human-centered design for interactive systems

ISO 9241-210

Human-centered design for interactive systems

4. Principles of human-centered design

- a) the design is based upon an explicit **understanding of users, tasks and environments** […]
- b) **users are involved** throughout design and development […]
- c) the design is **driven and refined by** user-centered **evaluation** […]
- d) the process is **iterative** […]
- e) the design addresses the **whole user experience** […]
- f) the design team includes **multidisciplinary skills** and perspectives”

INTERNATIONAL STANDARD

ISO 9241-210

First edition
2010-03-15

Ergonomics of human–system interaction —

Part 210:
Human-centred design for interactive systems

Ergonomie de l'interaction homme–système —

Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs

Reference number
ISO 9241-210:2010(E)

© ISO 2010

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9241-210:2019(EN) Human-centered design for interactive systems

ISO 9241-210

Human-centered design for interactive systems

6. Human-centered design activities

"a) understanding and specifying the **context of use**"

- *What are the tasks or objectives associated with the design?*

"b) specifying the **user requirements"**

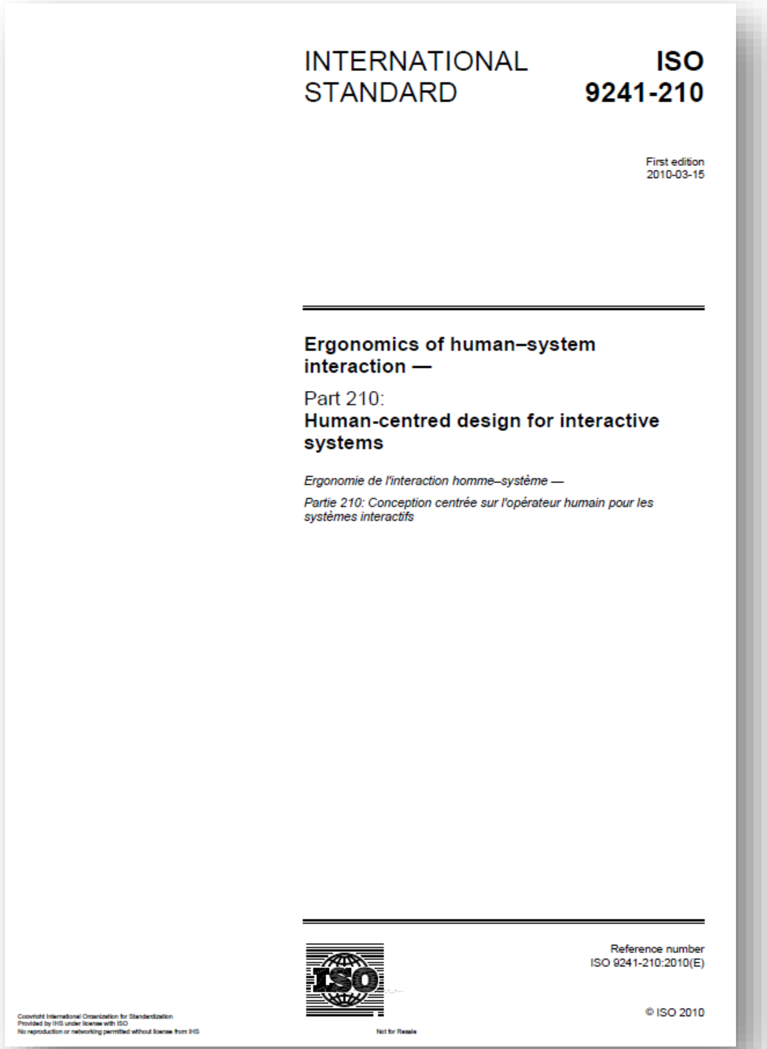
- *What expectations or requirements must the design accommodate?*

"c) producing **design solutions"**

- *prototyping, rendering, mockup building, implementation*

"d) **evaluating** the design"

- Conduct initial evaluations, usability testing, and ergonomic assessment



INTERNATIONAL STANDARD

ISO 9241-210

First edition
2010-03-15

Ergonomics of human–system interaction —

Part 210:
Human-centred design for interactive systems

Ergonomie de l'interaction homme–système —

Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs

Reference number
ISO 9241-210:2010(E)

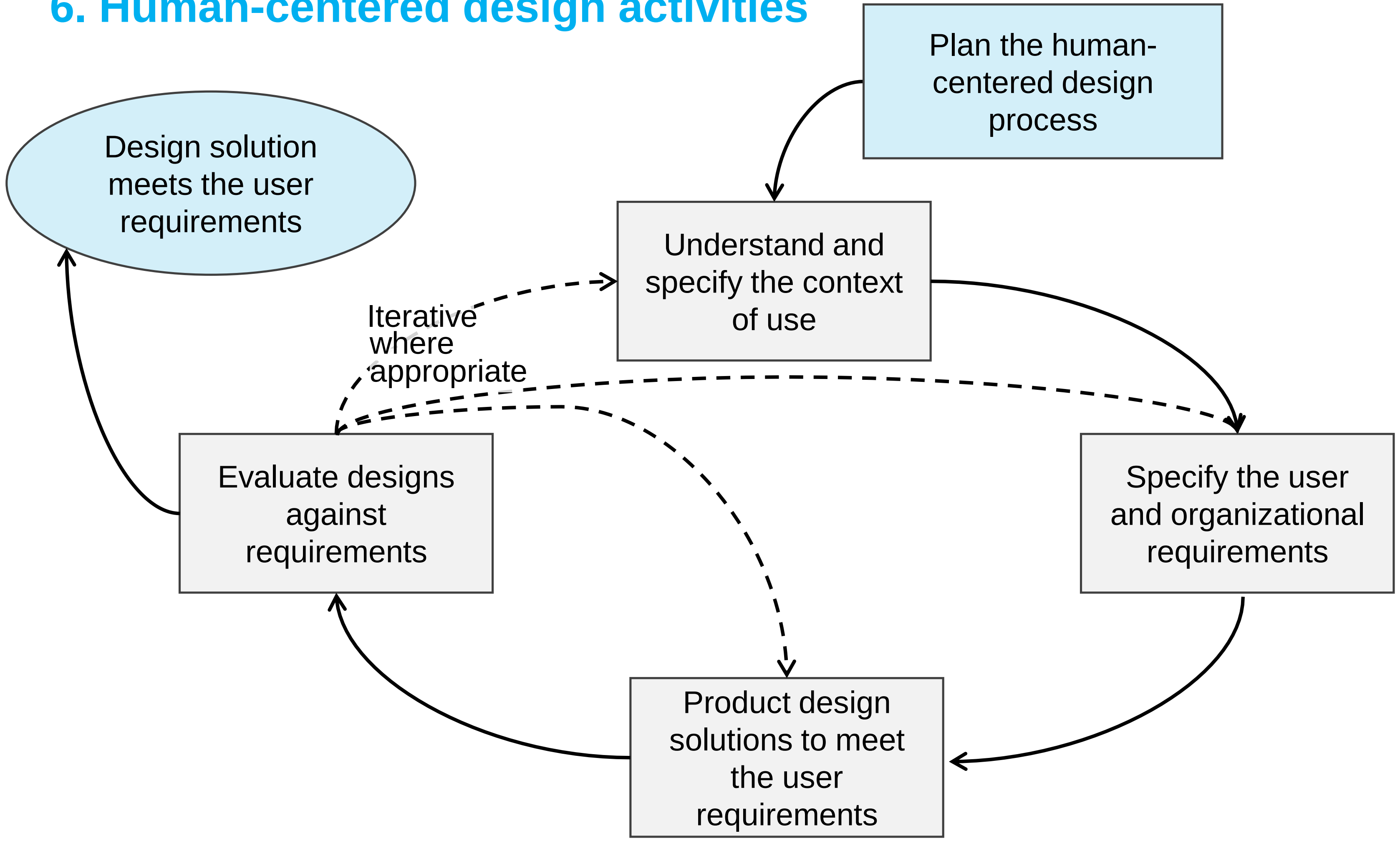
© ISO 2010

ISO 9241-210:2019(EN) Human-centered design for interactive systems

ISO 9241-210

Human-centered design for interactive systems

6. Human-centered design activities


Plan the human-centered design process
Understand and specify the context of use
Specify the user and organizational requirements
Product design solutions to meet the user requirements
Evaluate designs against requirements
Design solution meets the user requirements
Iterative where appropriate

INTERNATIONAL STANDARD

ISO 9241-210

First edition
2010-03-15

Ergonomics of human–system interaction —

Part 210:
Human-centred design for interactive systems

Ergonomie de l'interaction homme–système —

Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs

Reference number
ISO 9241-210:2010(E)

© ISO 2010

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

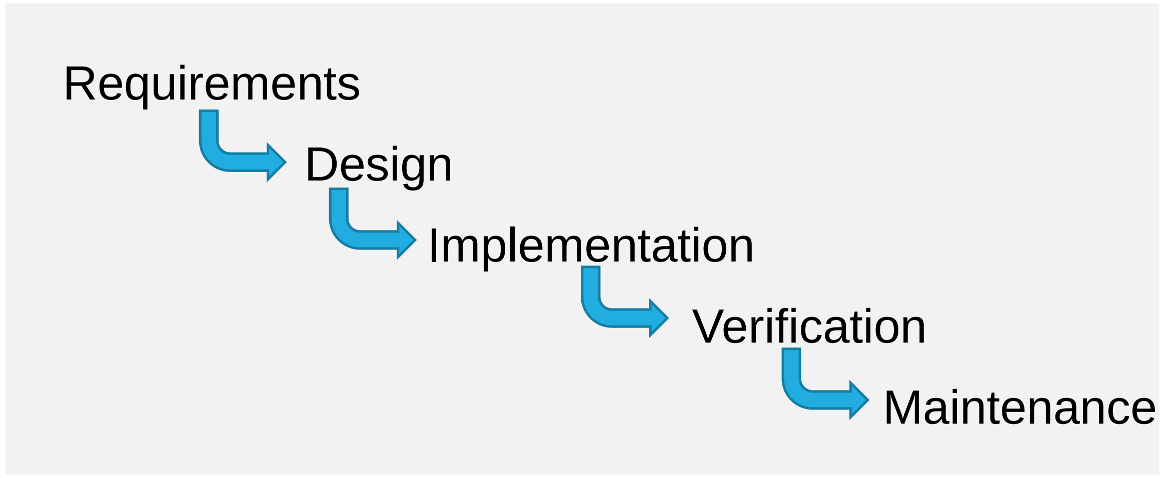
Not for Resale

ISO 9241-210:2019(EN) Human-centered design for interactive systems

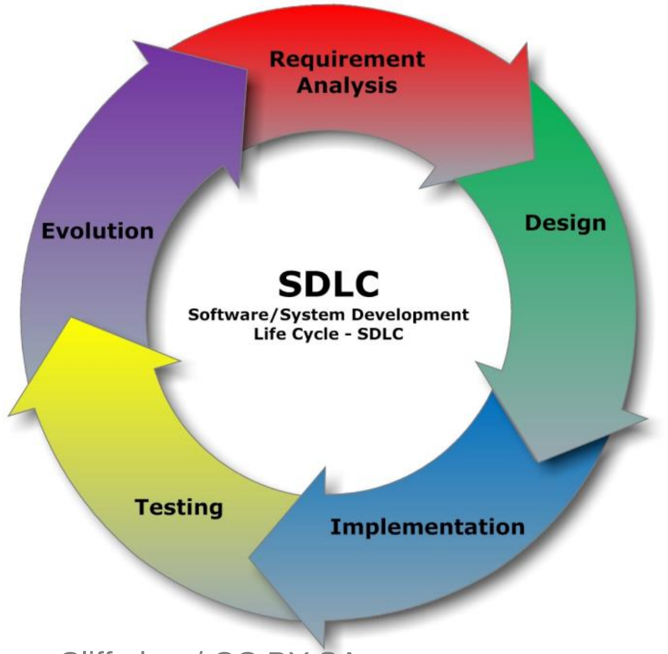
Software Process vs Human Centered Design Process

Mini-Exercise

- Will the human centered design process work well with the waterfall model?



- Does it fit agile software development methods?



Cliffydcw / CC BY-SA

Separation between interaction design and technical realization

Separation into a two stage process for interactive applications

- 1st – Concept development and Interaction design (quick iterations)
  - Application and interaction concept
  - Interaction design
  - Prototypes to evaluate the concept and interaction design
- 2nd – technical realization (slow iterations)
  - Technical analysis
  - Technical specification (e.g. architecture, platform)
  - Implementation
  - Evaluation and Quality management

Problems of User Centered Design

- Users may expect disadvantages (e.g. being replaced by software)
- Users may have conflicting views
- Users may be wrong
- Users may be resistant to change

- In a “business environment” you are expected to create a system with regards to the **goals specified** and this is unfortunately NOT necessarily the system users would like to have
- There is often a **trade-off between** the goals of **employers (customer) and employees (user)**

How easy is it to work in multidisciplinary teams?

- Many people are involved in the process of designing and implementing an interactive product
  - **Different background** (e.g. design, business, CS, marketing, administration)
  - Different and **conflicting** low level **objectives** objectives
- Communication can be very difficult!
- To be able to work in a team is essential!
  - Team work is a skill that can be learned

Did you understand this block?

Can you answer these questions?

- What is the ISO 9241-210 standard about?
- What are potential problems of user centered design
- What is the rationale for adopting human-centered design according to ISO 9241-210
- Name the Principles and activities of human-centered design according to ISO 9241-210
- How do the design activities according to ISO 9241-210 relate to each other? Make a sketch.
- Explain how interaction design and technical realization can be separated and why this may be useful.

References

- ISO 9241 Ergonomics of human–system interaction
- ISO 9241-210:2019(en) Human-centered design for interactive systems

INTERNATIONAL STANDARD

ISO 9241-210

First edition
2010-03-15

Ergonomics of human–system interaction —

Part 210:
Human-centred design for interactive systems

Ergonomie de l'interaction homme–système —

Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs

Reference number
ISO 9241-210:2010(E)

© ISO 2010

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Principles to Support Usability

according to Alan Dix et al.

Learning Goals

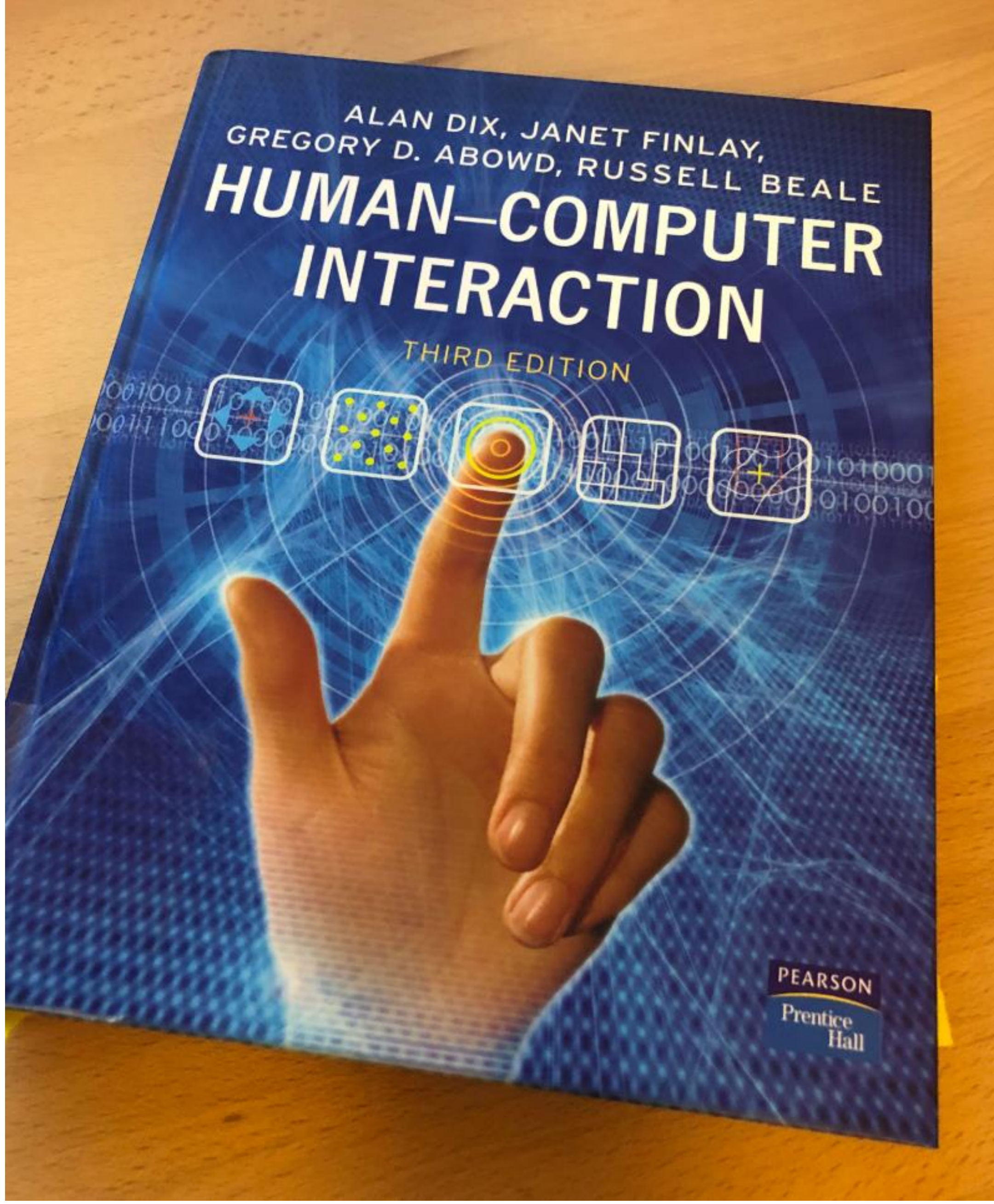
- Understand …
  - the principles that support usability according to Dix et al.
  - what points contribute to each of the principles

- Be able to …
  - explain these principles and give examples
  - discuss user interface designs with regard to these principles

Principles to Support Usability

By Dix et al.

- **Principle 1: Learnability**
- Principle 2: Flexibility
- Principle 3: Robustness


ALAN DIX, JANET FINLAY, GREGORY D. ABOWD, RUSSELL BEALE
HUMAN–COMPUTER INTERACTION
THIRD EDITION
PEARSON
Prentice Hall

Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education, p260ff, https://hcibook.com/.

Principle 1: Learnability

Principles to Support Usability Dix et al.

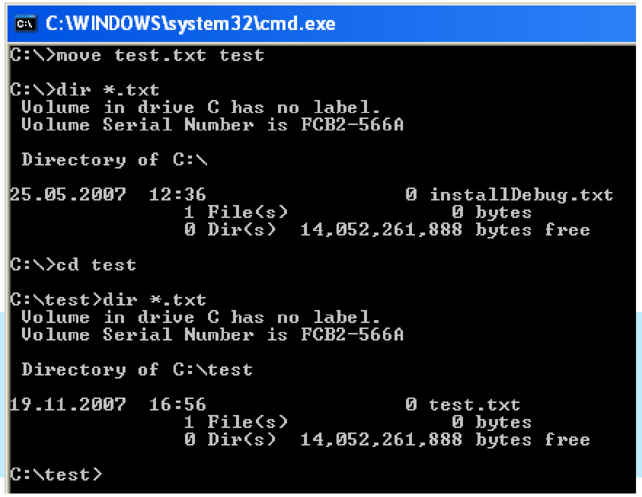
The ease with which new users can begin effective interaction and achieve maximal performance.

- Predictability
- Synthesizability
- Familiarity
- Generalizability
- Consistency

Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 1: Learnability

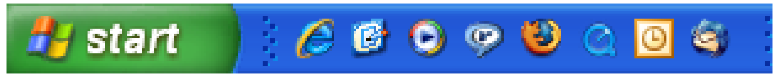
Principles to Support Usability Dix et al.

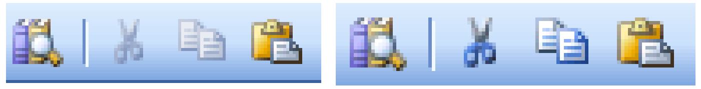


```
C:\>move test.txt test

C:\>dir *.txt
 Volume in drive C has no label.
 Volume Serial Number is FCB2-566A

 Directory of C:\

25.05.2007  12:36                 0 installDebug.txt
               1 File(s)              0 bytes
               0 Dir(s)  14,052,261,888 bytes free

C:\>cd test

C:\test>dir *.txt
 Volume in drive C has no label.
 Volume Serial Number is FCB2-566A

 Directory of C:\test

19.11.2007  16:56                 0 test.txt
               1 File(s)              0 bytes
               0 Dir(s)  14,052,261,888 bytes free

C:\test>
```

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

- **Predictability**
  - Determining effect of future actions based on past interaction history
  - Visibility of operations and effects
- Synthesizability
- **Familiarity**
  - how prior knowledge applies to a new system
  - affordance ('guessability')
- Generalizability
- Consistency





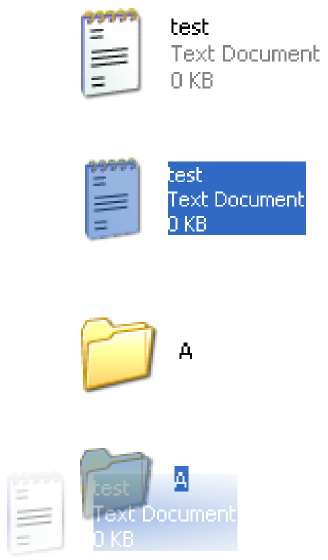
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 1: Learnability

Principles to Support Usability Dix et al.

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

- Predictability
- **Synthesizability**
  - ability of the user to assess the effect of past operations based on the current state
  - the user should see the changes of an operation
  - immediate vs. eventual feedback
- Familiarity
- Generalizability
- Consistency



Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 1: Learnability

Principles to Support Usability Dix et al.

Star Trek IV: The Voyage Home

25 seconds
Show from Movie: Star Trek IV: The Voyage Home
https://www.youtube.com/watch?v=hShY6xZWVGE

Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

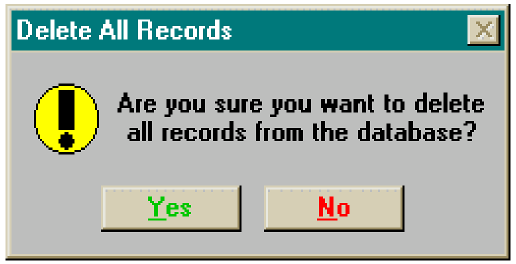
Principle 1: Learnability

Principles to Support Usability Dix et al.

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

- Predictability
- Synthesizability
- Familiarity
- **Generalizability**
  - extending specific interaction knowledge to new situations
- **Consistency**
  - likeness in input/output behavior arising from similar situations or task objectives

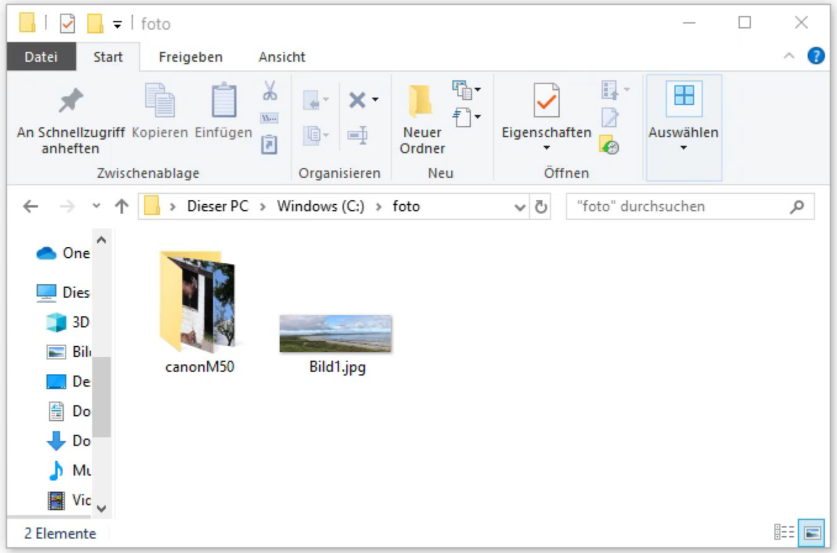
25 seconds
Show from Movie: Star Trek IV: The Voyage Home
https://www.youtube.com/watch?v=hShY6xZWVGE

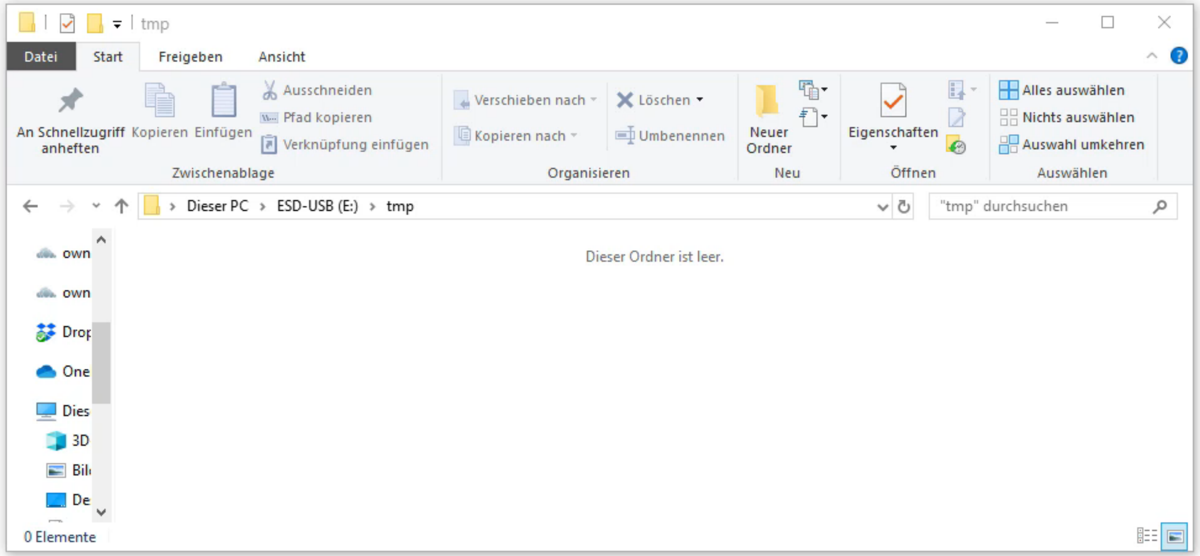

Delete All Records

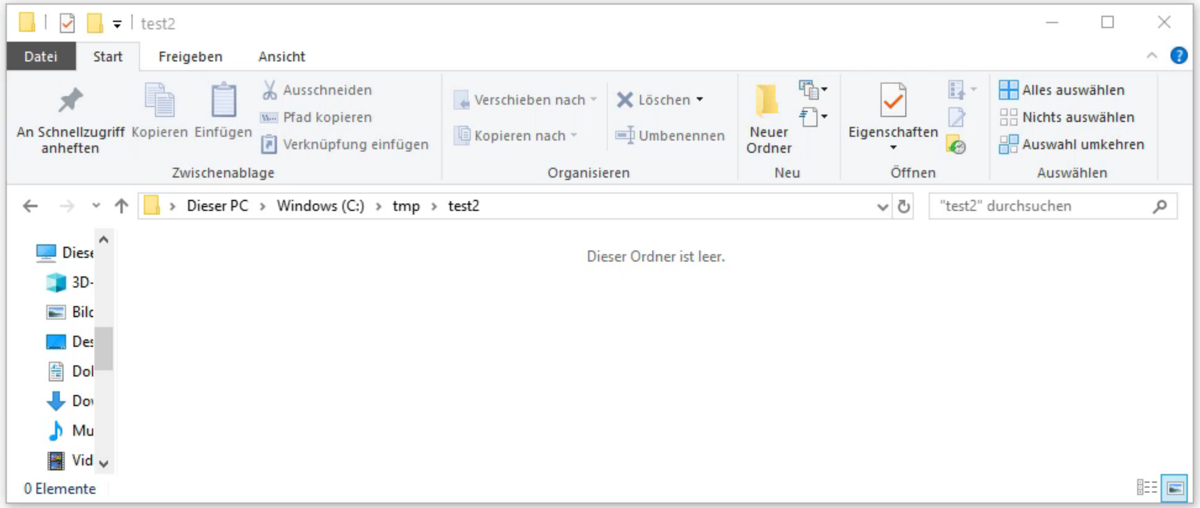
Are you sure you want to delete all records from the database?

Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Example: Predictability, Consistency



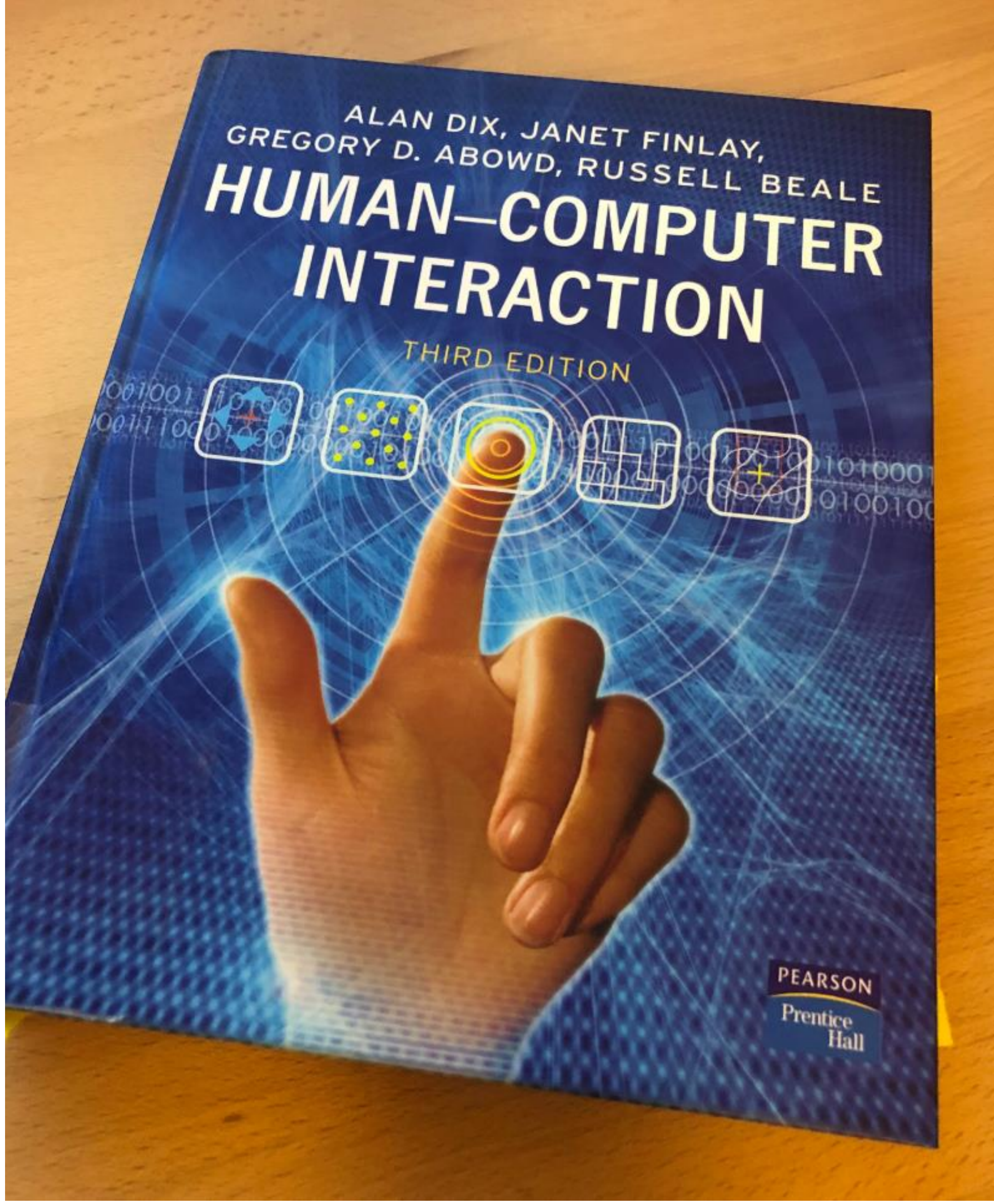




Principles to Support Usability

By Dix et al.

- Principle 1: Learnability
- **Principle 2: Flexibility**
- Principle 3: Robustness



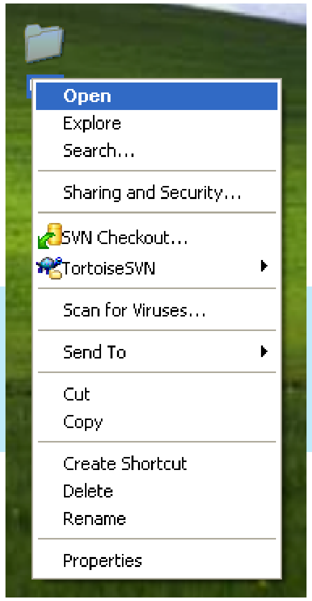
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education, p260ff, https://hcibook.com/.

Principle 2: Flexibility

Principles to Support Usability Dix et al.

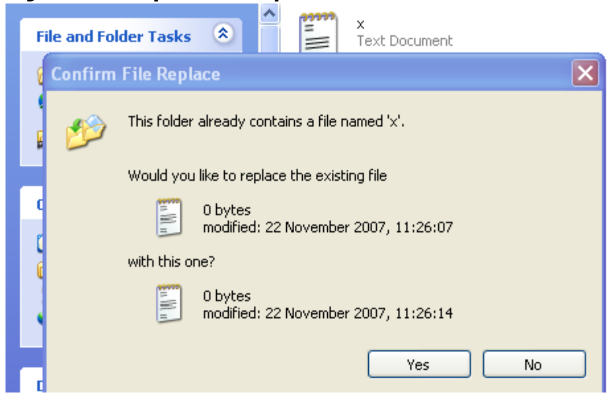
The multiplicity of ways the user and system exchange information.

- **Dialogue initiative**
  - freedom from system imposed constraints on input dialogue
  - user preemptiveness: user initiates dialog
  - system preemptiveness: system initiates dialog
- Multithreading
- Task migratability
- Substitutivity
- Customizability



user preemptiveness

system preemptiveness



This folder already contains a file named 'x'.

Would you like to replace the existing file

0 bytes
modified: 22 November 2007, 11:26:07

with this one?

0 bytes
modified: 22 November 2007, 11:26:14

Slide adapted from Dr. Paul Holleis

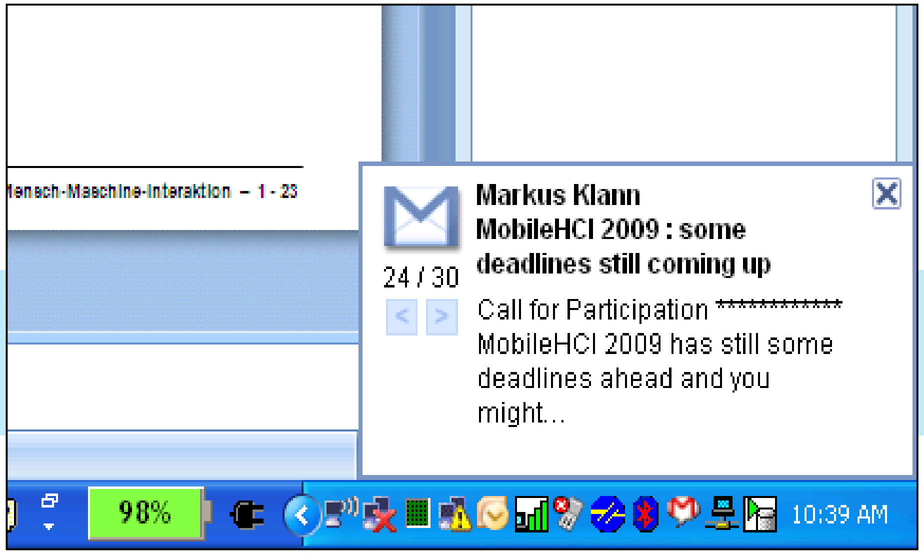
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 2: Flexibility

Principles to Support Usability Dix et al.

The multiplicity of ways the user and system exchange information.

- Dialogue initiative
- **Multithreading**
  - system supports user interaction for several tasks at a time
    - concurrent multimodality: simultaneous communication of information pertaining to separate tasks
    - interleaving multimodality: permits temporal overlap between separate tasks, dialog is restricted to a single task
- Task migratability
- Substitutivity
- Customizability


Markus Klann
MobileHCI 2009 : some deadlines still coming up
24 / 30
Call for Participation ************ MobileHCI 2009 has still some deadlines ahead and you might...

Slide adapted from Dr. Paul Holleis

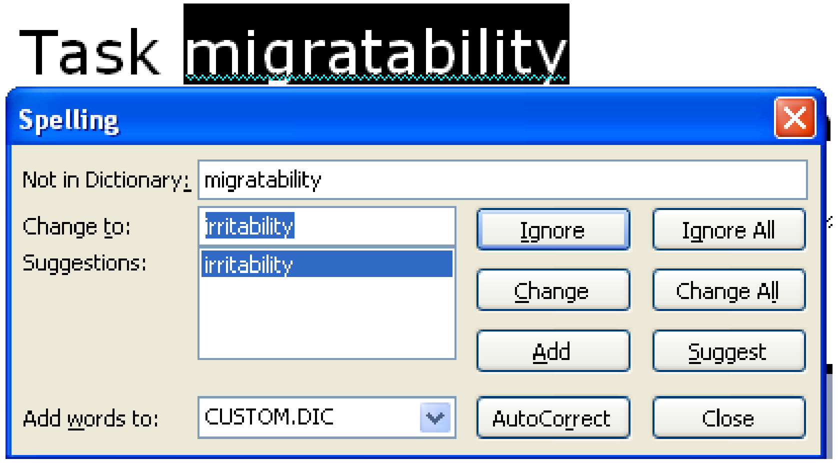
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 2: Flexibility

Principles to Support Usability Dix et al.

The multiplicity of ways the user and system exchange information.

- Dialogue initiative
- Multithreading
- **Task migratability**
  - passing responsibility for task execution between user and system, e.g. spell checking
- Substitutivity
- Customizability



Not in Dictionary: migratability
Change to: irritability
Suggestions: irritability
Add words to: CUSTOM.DIC

Slide adapted from Dr. Paul Holleis

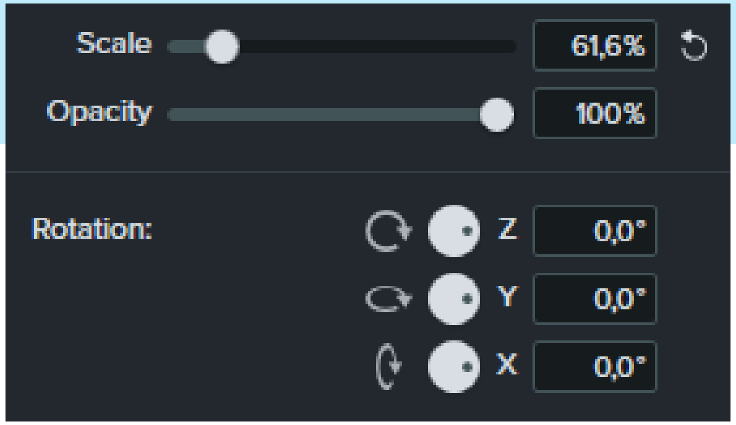
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 2: Flexibility

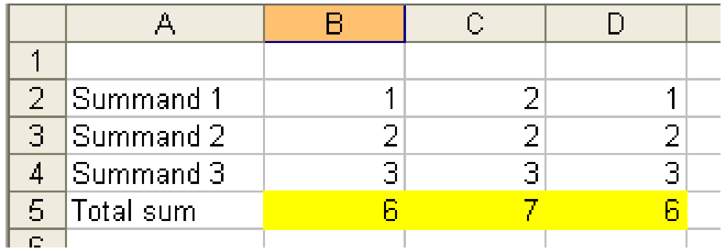
Principles to Support Usability Dix et al.

The multiplicity of ways the user and system exchange information.

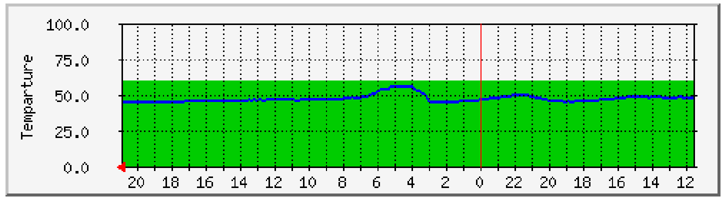
- Dialogue initiative
- Multithreading
- Task migratability
- **Substitutivity**
  - allowing equivalent values of input and output to be substituted for each other
  - representation multiplicity
  - equal opportunity: blurs the distinction between input and output
- Customizability



Scale 61,6%
Opacity 100%
Rotation:
Z 0,0°
Y 0,0°
X 0,0°



	A	B	C	D
1				
2	Summand 1	1	2	1
3	Summand 2	2	2	2
4	Summand 3	3	3	3
5	Total sum	6	7	6



Slide adapted from Dr. Paul Holleis

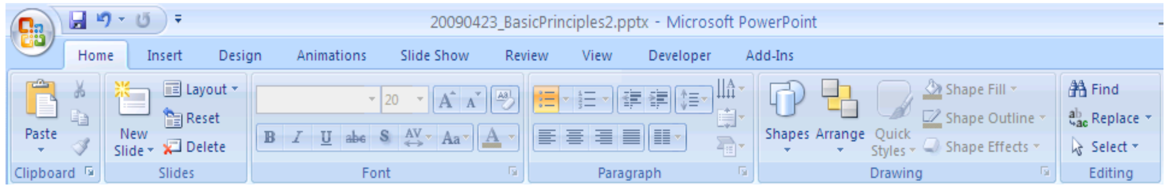
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

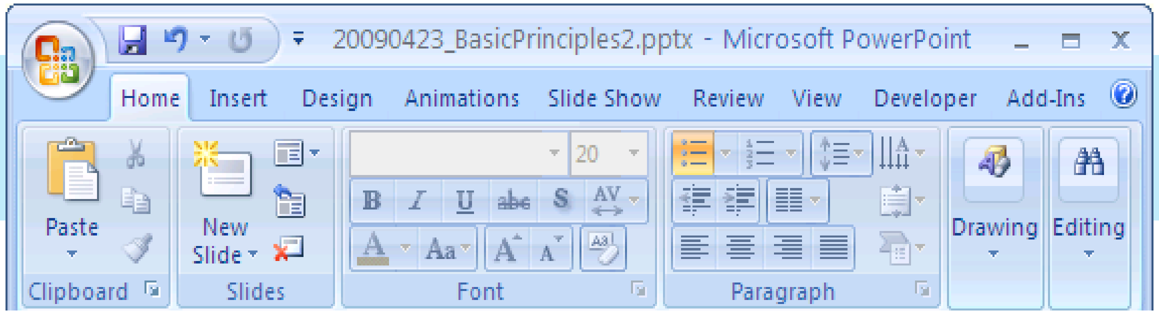
Principle 2: Flexibility

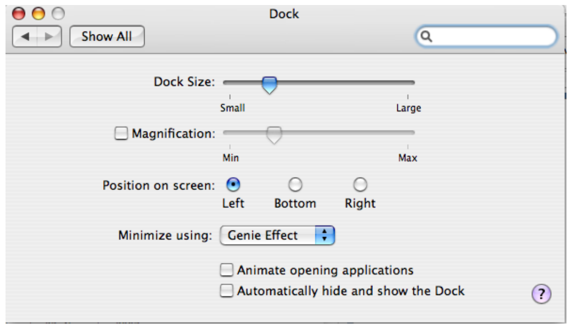
Principles to Support Usability Dix et al.

The multiplicity of ways the user and system exchange information.

- Dialogue initiative
- Multithreading
- Task migratability
- Substitutivity
- **Customizability**
  - modifiability of the user interface by the user (adaptability) or system (adaptivity)
  - <u>adaptability (*anpassbar*)</u>: users ability to adjust the form of input and output
  - adaptivity (*adaptive*): automatic customization of the user interface by the system






Dock

Show All

Dock Size: Small — Large

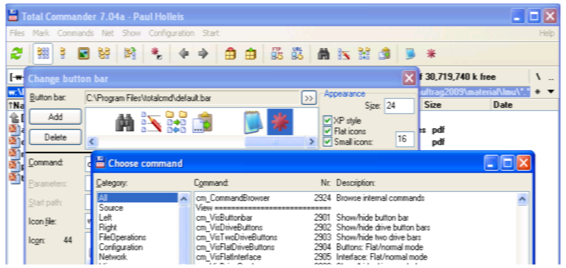
Magnification: Min — Max

Position on screen: [✓ Left] Bottom Right

Minimize using: Genie Effect

Animate opening applications

Automatically hide and show the Dock



Slide adapted from Dr. Paul Holleis

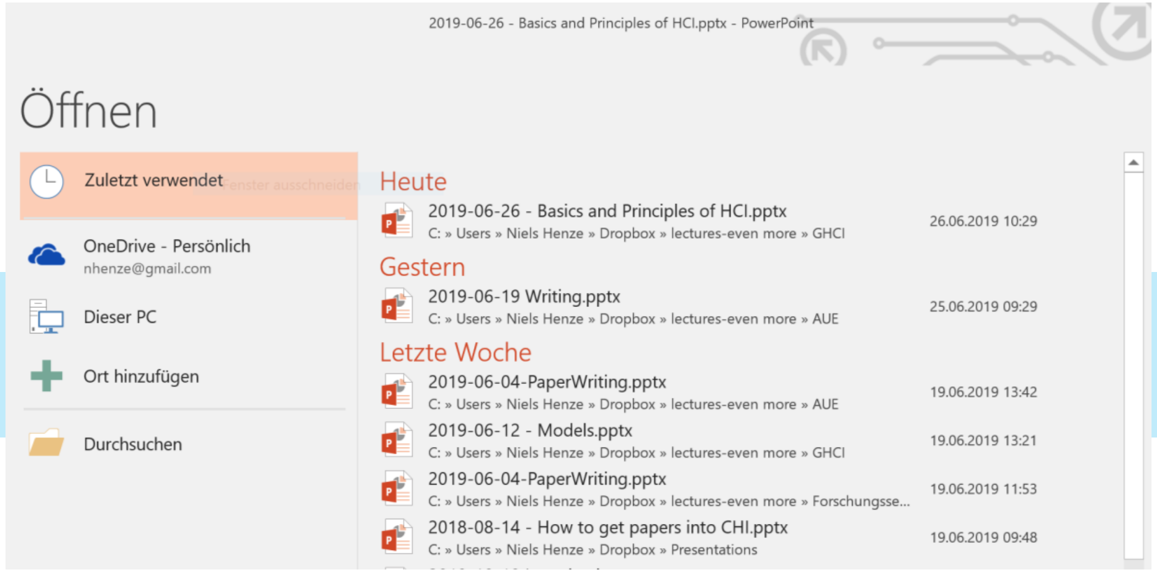
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 2: Flexibility

Principles to Support Usability Dix et al.

The multiplicity of ways the user and system exchange information.

- Dialogue initiative
- Multithreading
- Task migratability
- Substitutivity
- **Customizability**
  - modifiability of the user interface by the user (adaptability) or system (adaptivity)
  - adaptability (*anpassbar*): users ability to adjust the form of input and output
  - adaptivity (*adaptive*): automatic customization of the user interface by the system

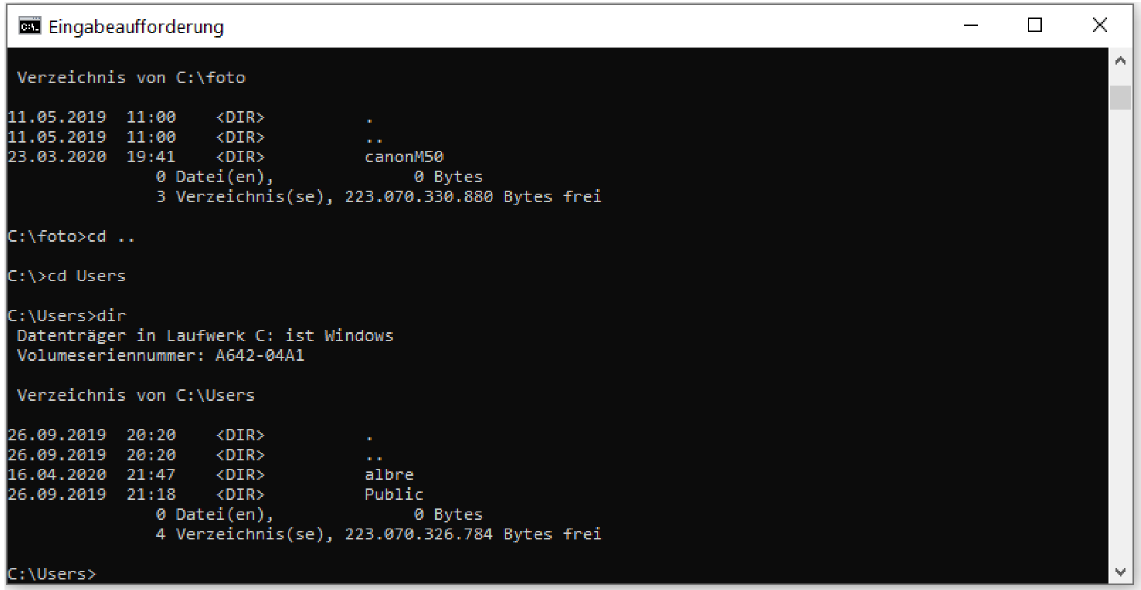

2019-06-26 - Basics and Principles of HCI.pptx - PowerPoint

Öffnen

Heute
2019-06-26 - Basics and Principles of HCI.pptx — C: » Users » Niels Henze » Dropbox » lectures-even more » GHCI — 26.06.2019 10:29

Gestern
2019-06-19 Writing.pptx — C: » Users » Niels Henze » Dropbox » lectures-even more » AUE — 25.06.2019 09:29

Letzte Woche
2019-06-04-PaperWriting.pptx — C: » Users » Niels Henze » Dropbox » lectures-even more » AUE — 19.06.2019 13:42
2019-06-12 - Models.pptx — C: » Users » Niels Henze » Dropbox » lectures-even more » GHCI — 19.06.2019 13:21
2019-06-04-PaperWriting.pptx — C: » Users » Niels Henze » Dropbox » lectures-even more » Forschungsse... — 19.06.2019 11:53
2018-08-14 - How to get papers into CHI.pptx — C: » Users » Niels Henze » Dropbox » Presentations — 19.06.2019 09:48


Eingabeaufforderung

```
 Verzeichnis von C:\foto

11.05.2019  11:00    <DIR>          .
11.05.2019  11:00    <DIR>          ..
23.03.2020  19:41    <DIR>          canonM50
               0 Datei(en),              0 Bytes
               3 Verzeichnis(se), 223.070.330.880 Bytes frei

C:\foto>cd ..

C:\>cd Users

C:\Users>dir
 Datenträger in Laufwerk C: ist Windows
 Volumeseriennummer: A642-04A1

 Verzeichnis von C:\Users

26.09.2019  20:20    <DIR>          .
26.09.2019  20:20    <DIR>          ..
16.04.2020  21:47    <DIR>          albre
26.09.2019  21:18    <DIR>          Public
               0 Datei(en),              0 Bytes
               4 Verzeichnis(se), 223.070.326.784 Bytes frei

C:\Users>
```

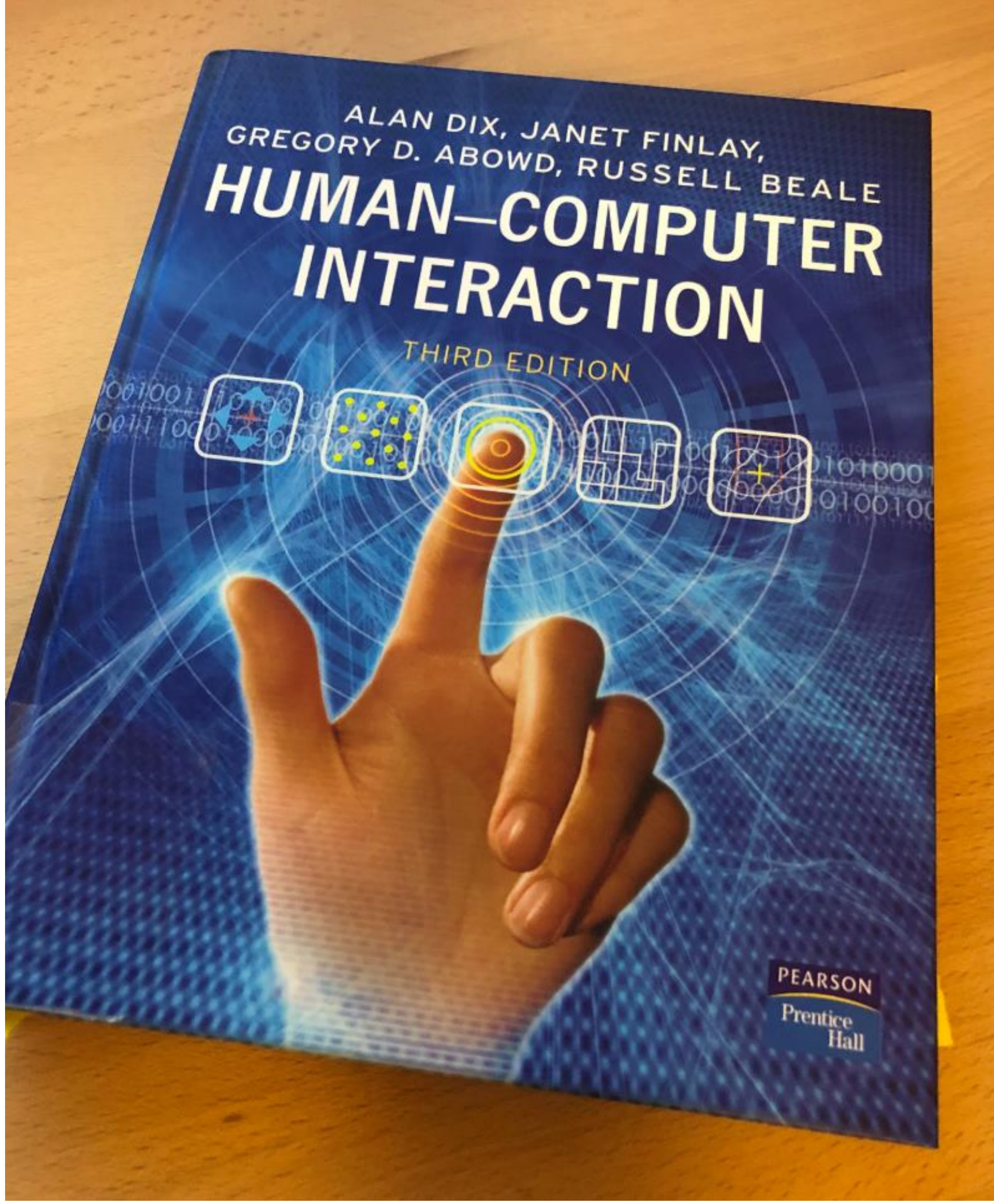
Slide adapted from Dr. Paul Holleis

Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principles to Support Usability

By Dix et al.

- Principle 1: Learnability
- Principle 2: Flexibility
- **Principle 3: Robustness**



Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education, p260ff, https://hcibook.com/.

Principle 3: Robustness

Principles to Support Usability Dix et al.

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

- Observability
- Recoverability
- Task conformance
- Responsiveness

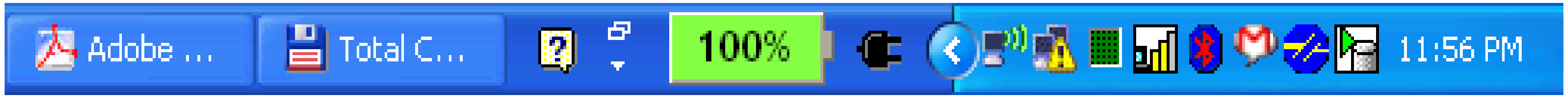
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 3: Robustness

Principles to Support Usability Dix et al.

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

- **Observability**
  - ability of the user to evaluate the internal state of the system from its perceivable representation
- Recoverability
- Task conformance
- Responsiveness

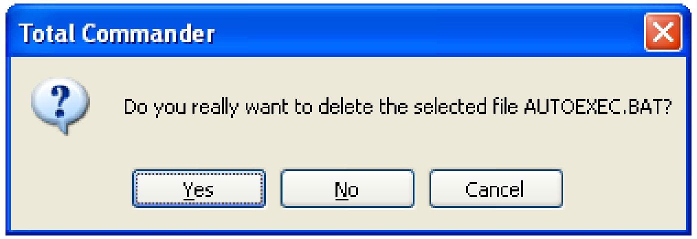


Slide adapted from Dr. Paul Holleis

Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 3: Robustness

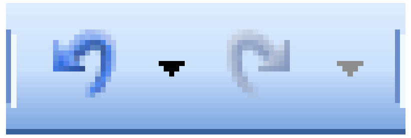
Principles to Support Usability Dix et al.


Total Commander

Do you really want to delete the selected file AUTOEXEC.BAT?

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

- Observability
- **Recoverability**
  - ability of the user to correct a recognized error
  - reachability (states): forward (redo) / backward (undo) recovery
  - commensurate effort (more effort / steps for deleting a file than for moving it)
- Task conformance
- Responsiveness



Slide adapted from Dr. Paul Holleis

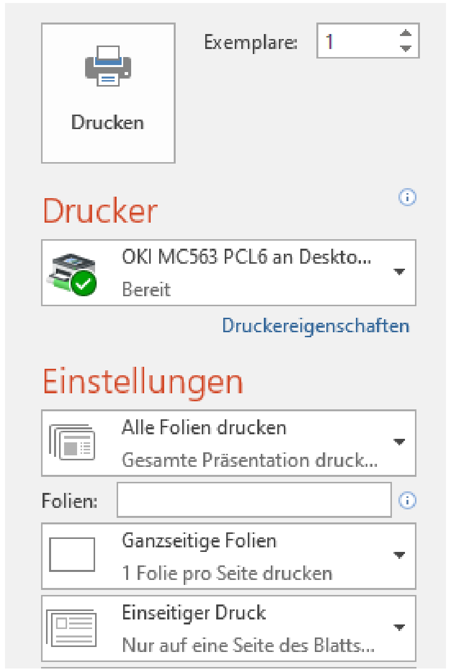
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 3: Robustness

Principles to Support Usability Dix et al.

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

- Observability
- Recoverability
- **Task conformance**
  - degree to which system services support all of the user's tasks
  - task completeness
  - task adequacy
- Responsiveness


Drucken
Exemplare: 1
Drucker
OKI MC563 PCL6 an Deskto...
Bereit
Druckereigenschaften
Einstellungen
Alle Folien drucken
Gesamte Präsentation druck...
Folien:
Ganzseitige Folien
1 Folie pro Seite drucken
Einseitiger Druck
Nur auf eine Seite des Blatts...

Slide adapted from Dr. Paul Holleis

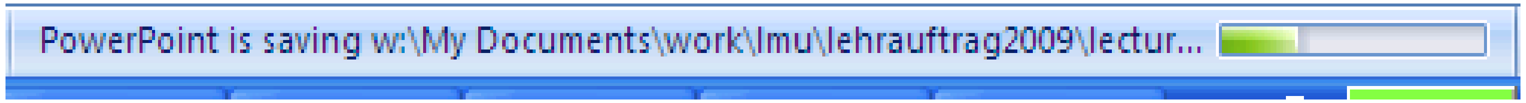
Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

Principle 3: Robustness

Principles to Support Usability Dix et al.

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

- Observability
- Recoverability
- Task conformance
- Responsiveness
  - how the user perceives the rate of communication with the system
  - preferred: short durations and instantaneous responses
  - stability and indication of response time


PowerPoint is saving w:\My Documents\work\lmu\lehrauftrag2009\lectur...

Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education https://hcibook.com/.

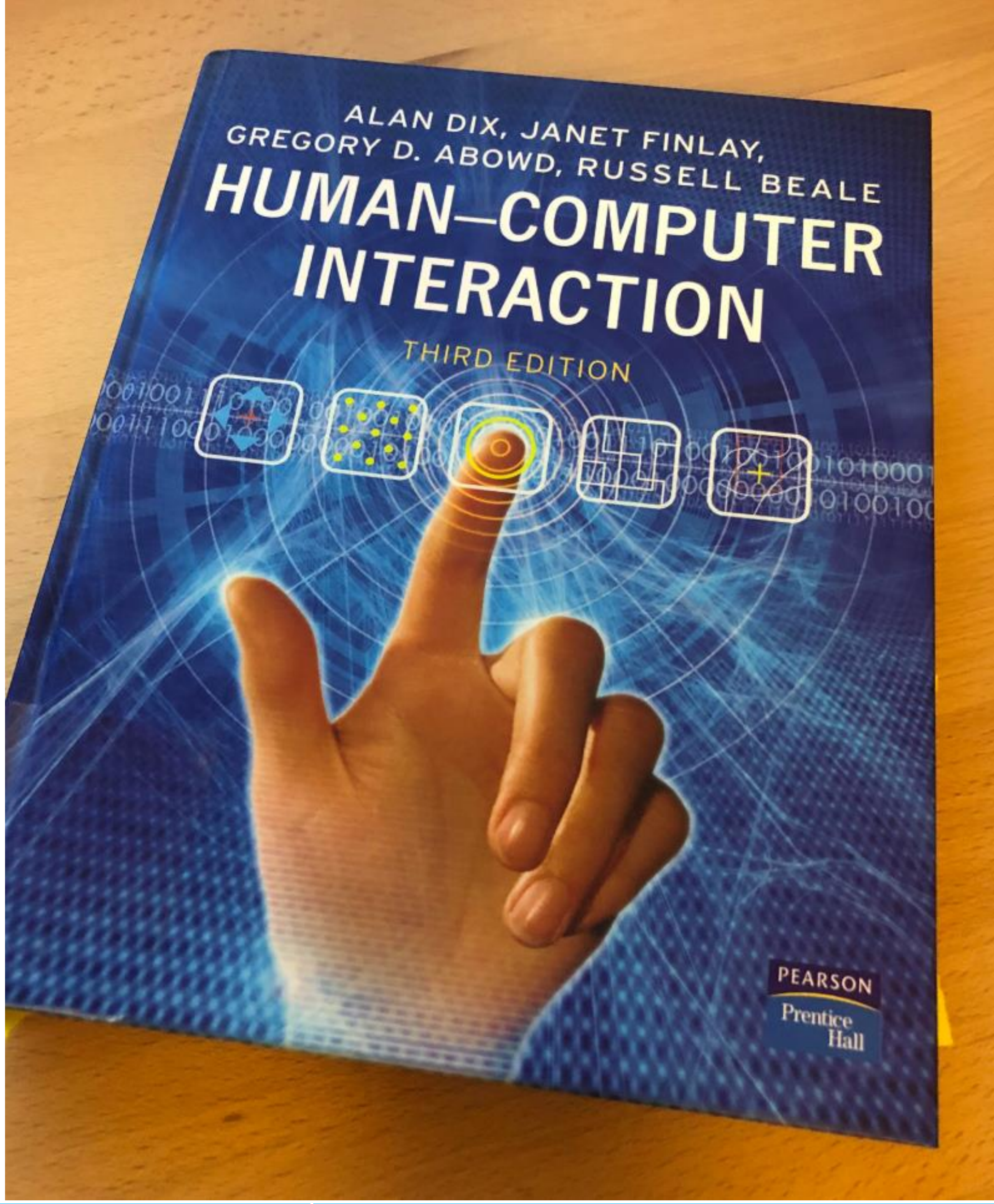
Slide adapted from Dr. Paul Holleis

Principles to Support Usability

By Dix et al.

- Learnability
  - Predictability
  - Synthesizability
  - Familiarity
  - Generalizability
  - Consistency
- Flexibility
  - Dialogue initiative
  - Multithreading
  - Task migratability
  - Substitutivity
  - Customizability
- Robustness
  - Observability
  - Recoverability
  - Responsiveness
  - Task conformance

Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. Pearson Education, p260ff, https://hcibook.com/.



Conversation with Alan Dix

https://youtu.be/n8S6ZgZzgbo

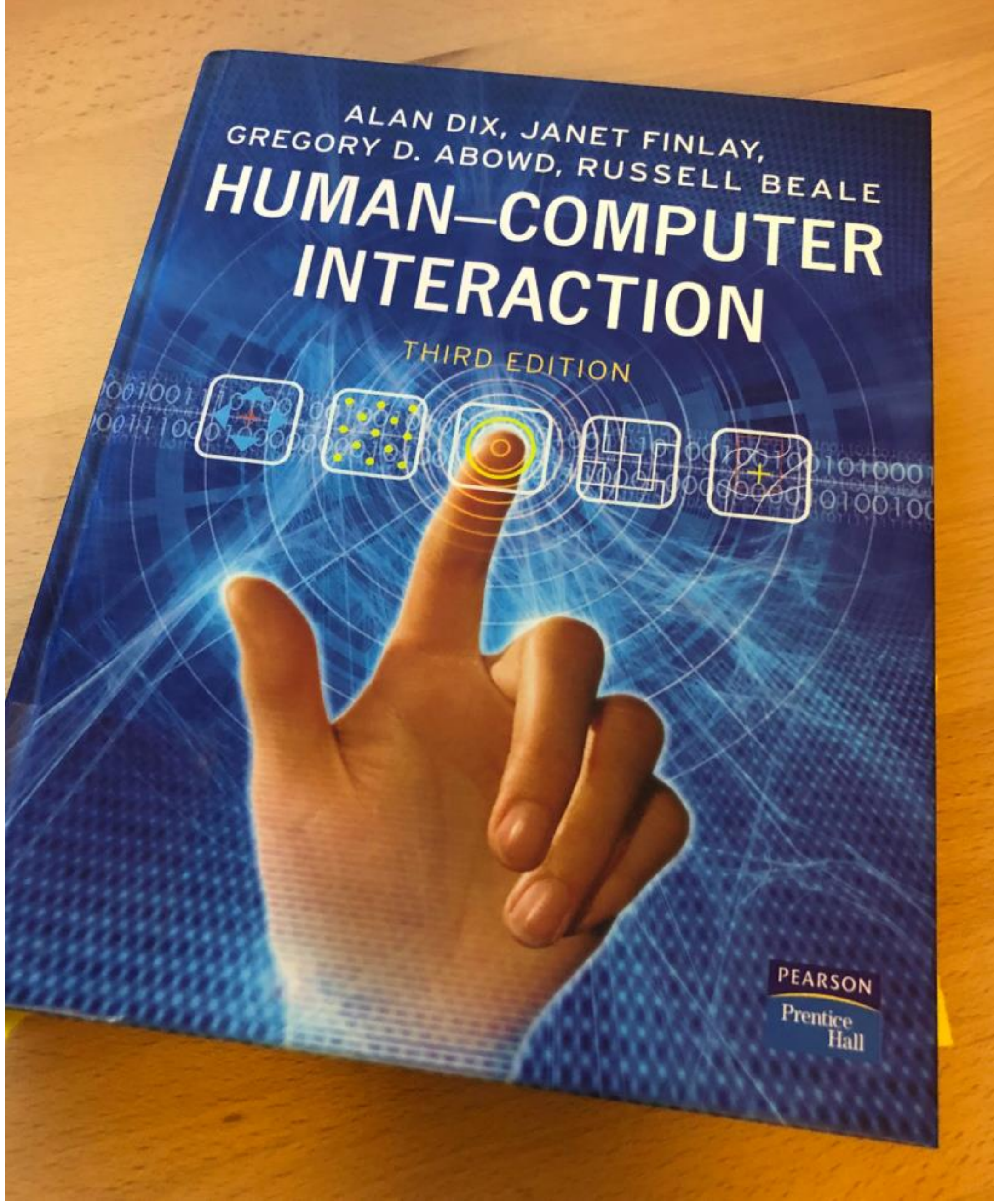
Did you understand this block?

Can you answer these questions?

- What are the 3 principles to support usability according to Alan Dix et al.?
- Give an example for robustness?
- How did our expectations for learnability change over the last 20 years?
- How does recoverability contribute to robustness?
- What two forms of customizability can be discriminated? Give an example for each.
- How does predictability improve learnability? How is predictability achieved?

Reference

- Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). Human-computer interaction. Pearson Education, pp.260ff https://hcibook.com/



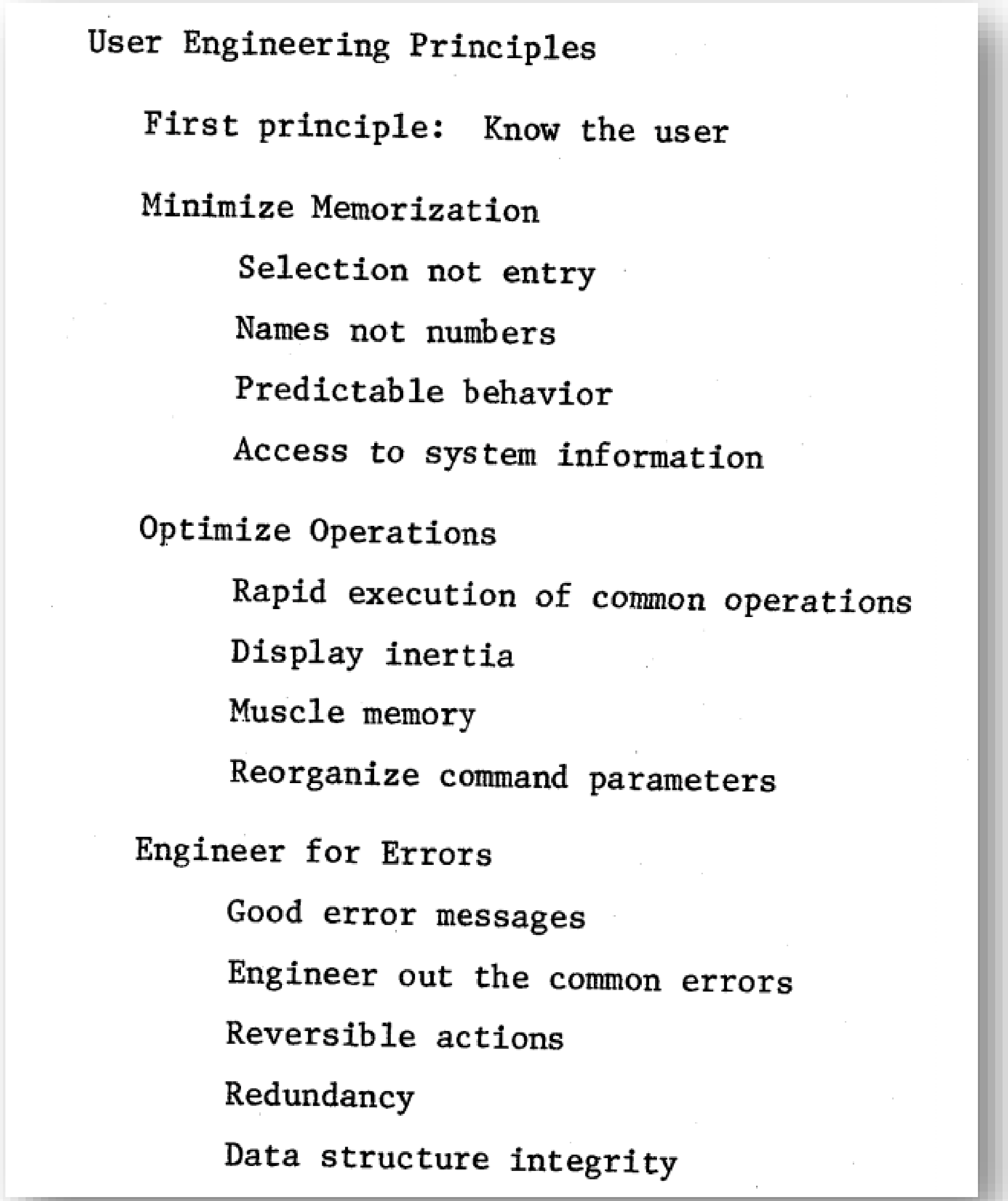
Know the User and Tasks

Learning Goals

- Understand …
  - Why it is important to know the user and their tasks
  - How a task frequency analysis works and what it is good for
  - The concept of personas and their value for designing interacive systems
- Be able to …
  - explain why it is important to determine the supported task in the process before the system is designed
  - explain a hierarchical task analysis for a given example
  - conduct a basic stakeholder analysis

User engineering principles for interactive systems

Wilfred J. Hansen. 1971



User Engineering Principles

First principle: Know the user

Minimize Memorization

Selection not entry

Names not numbers

Predictable behavior

Access to system information

Optimize Operations

Rapid execution of common operations

Display inertia

Muscle memory

Reorganize command parameters

Engineer for Errors

Good error messages

Engineer out the common errors

Reversible actions

Redundancy

Data structure integrity

Wilfred J. Hansen. 1971. User engineering principles for interactive systems. In Proceedings of the fall joint computer conference. http://doi.acm.org/10.1145/1479064.1479159

Who is my user?

Usage Profiles "Know Thy User"

- Different people have different requirements for their interaction with computers.
- Issues to take into account:
  - goals, motivation, personality
  - education, cultural background, training
  - age, gender, physical abilities, …
- Experience:
  - Novice users
  - Knowledgeable intermittent users
  - Expert frequent users

Wilfred J. Hansen. 1971. User engineering principles for interactive systems. In Proceedings of the fall joint computer conference. http://doi.acm.org/10.1145/1479064.1479159

User-Needs and Task Profiles

What is the system for?

- Find out what the user is trying to do!
  - **the goal**
  - what their **needs** are
  - resulting **tasks**
- Supported tasks should be determined before the design starts
- Functionality should only be added if identified to help solving tasks
  - Temptation: If additional functionality is cheap to include it is often done – this can seriously compromise the user interface concept (and potentially the whole software system)!
- Frequency of tasks related to user profiles

User Diversity

One size fits all?

- Example: flight booking webpage
  - Travel agent booking many flights a day – everyday
  - A teacher organizing a field trip (once a year) and making bookings for a large group
  - A business person changing bookings while travelling
  - A family looking for a package holiday
- Basic concepts to structure the problem
  - Usage profiles
  - Task profiles

Task Frequency Analysis

Hypothetical Analysis – travel web site

- Will one website fit all the users?

Task / *Persona*	Group reservation	Change of itinerary	Booking child care	Comparing sales agent performance
Sales agent	0.2	0.1	0.1	0
Manager	0	0	0	0.3
Family	0.05	0.05	0.3	0
Business traveler	0.01	0.2	0.01	0

Task Frequency Analysis

Mini Exercise: Instagram, YouTube, EBay, WhatsApp

- Pick one Application, think of typical personas and their tasks, fill in hypothetical task frequencies

Task *Persona*				

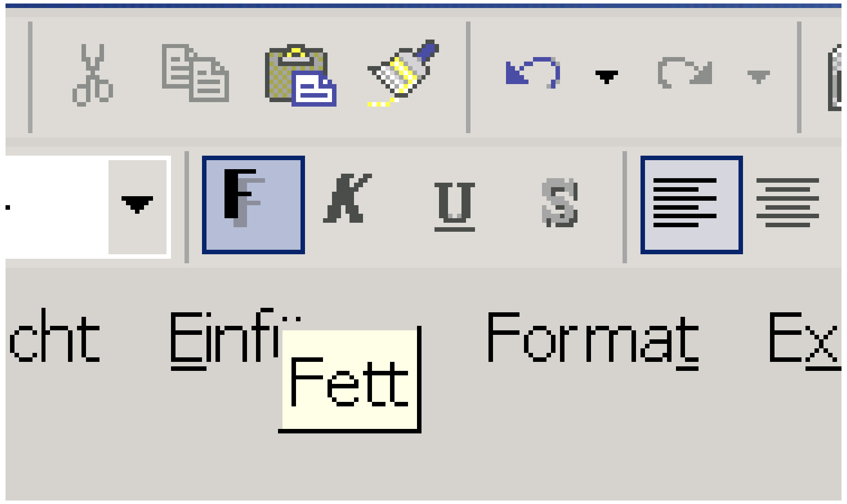
Task Frequency Analysis

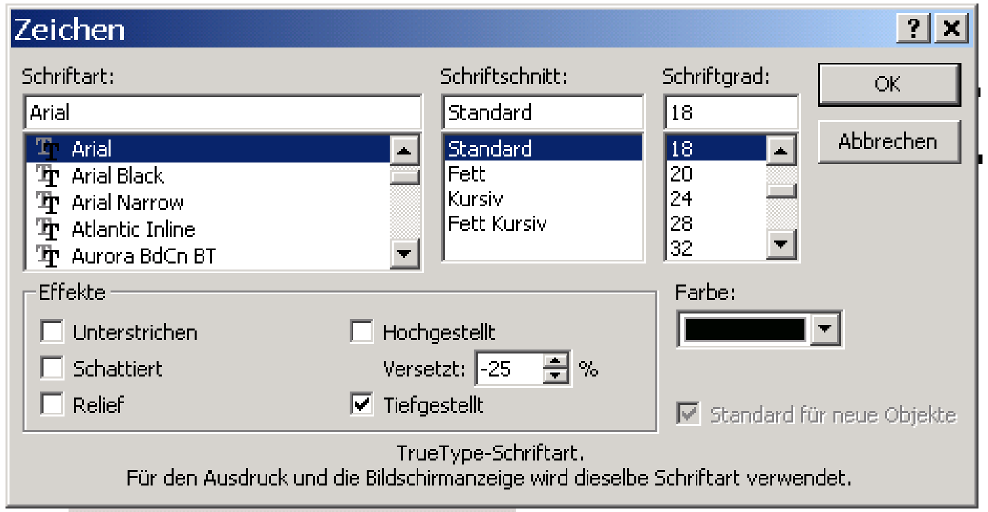
What is it good for?

- Helps to shape the interaction design
  - Frequent action should be simple and quick to carry out
  - Infrequent action may take longer
  - Menu structure should reflect this
- Example
  - Frequent actions: Toolbar or special key
  - Intermediate frequent actions:
    ribbon menu, pull-down menu, key combination (Ctrl+S)
  - Infrequent actions: Sequence of menus or dialogs
- Problem – if many (all) actions occur with very similar relative frequency…

Task Frequency Analysis

Example: Bold is more commonly used than subscript





Schriftart: Arial
- Arial
- Arial Black
- Arial Narrow
- Atlantic Inline
- Aurora BdCn BT

Schriftschnitt: Standard
- Standard
- Fett
- Kursiv
- Fett Kursiv

Schriftgrad: 18
- 18
- 20
- 24
- 28
- 32

Effekte
- [ ] Unterstrichen
- [ ] Schattiert
- [ ] Relief
- [ ] Hochgestellt
- Versetzt: -25 %
- [x] Tiefgestellt

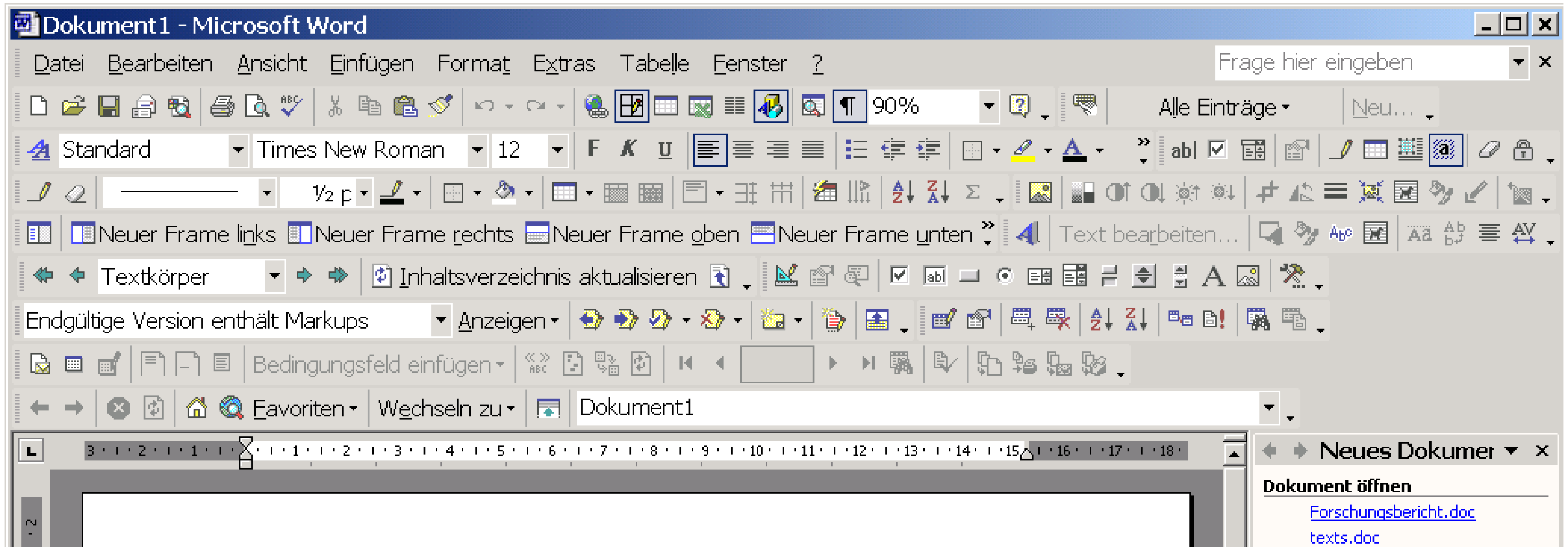
Farbe:

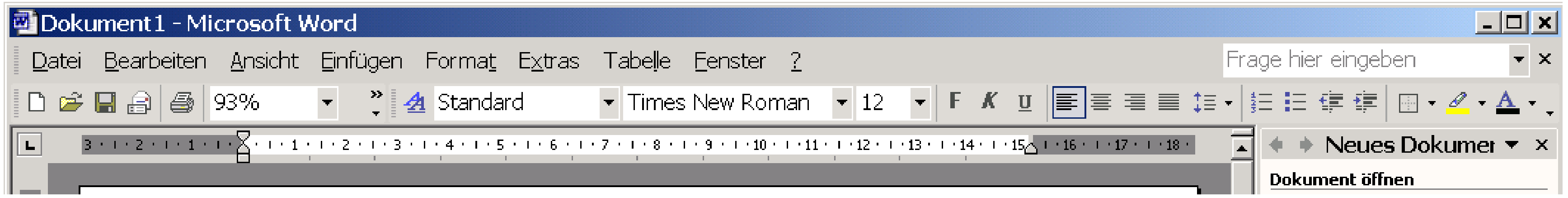
- [x] Standard für neue Objekte

TrueType-Schriftart.
Für den Ausdruck und die Bildschirmanzeige wird dieselbe Schriftart verwendet.

Task Frequency Analysis

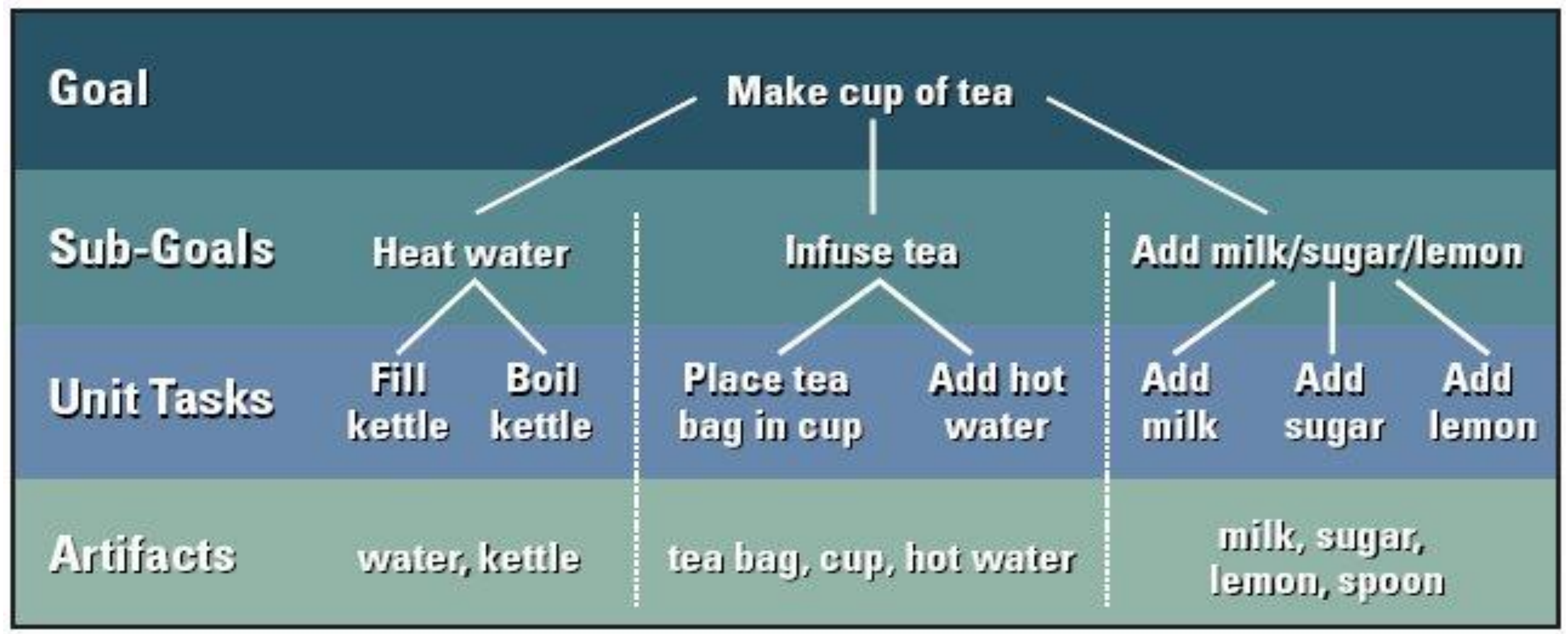
Trade-off between quick access and over-crowed interface




Dokument1 - Microsoft Word
Datei Bearbeiten Ansicht Einfügen Format Extras Tabelle Fenster ?
Frage hier eingeben
93%
Standard
Times New Roman
12
Neues Dokument
Dokument öffnen

Hierarchical Task Analysis

Understanding what it takes to complete a task



William Hudson. HCI and the Web: a Tale of Two Tutorials: a Cognitive Approach to Interactive System Design and Interaction Design Meets Agility. *ACM interactions* 12(1), 2005, 49-51

Where is the Average User?

- There is no such thing as an average user!
- Usage of Persona
  - Typical users but not the average

- Number game…
  - Creating a software that is appropriate for a very specific target group (e.g. 0,1% of the population) may still find a large user base, e.g. in Europe and the US this may be more than half a million people
  - Designing for 90% of the users will leave alone in Germany 8 million users out

Persona

Creating realistic and prototypical users

- “Software today is too often designed **to please too many users**, resulting in **low user satisfaction**.”
- “choose the right individuals to design for […] and then to prioritize these individuals so that the needs of the most important users are met”
- “Personas provide a powerful tool for communicating about **different types of users and their needs**, then deciding which users are the most important to target in the design of form and behavior”
- “Personas represent groups of users”
- Typically 3 to 12 personas

Cooper, A., Reimann, R., & Cronin, D. (2007). About face 3: the essentials of interaction design. John Wiley & Sons. p.80ff

Why Persona

Creating realistic and prototypical users

- “Avoiding the “elastic user”
  - If you do not specify the user you can change their abilities to support a design decision made = “elastic user”
- Avoiding self-referential design
  - The designer or developer often assumes (implicitly) that users have his goals and his skills and abilities.
- Generally, make requirements concrete
  - Seemingly unnecessary detail helps in making the requirements accessible and understandable for a large audience (users, managers, developers)

Cooper, A., Reimann, R., & Cronin, D. (2007). About face 3: the essentials of interaction design. John Wiley & Sons. p.80ff

https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/personas

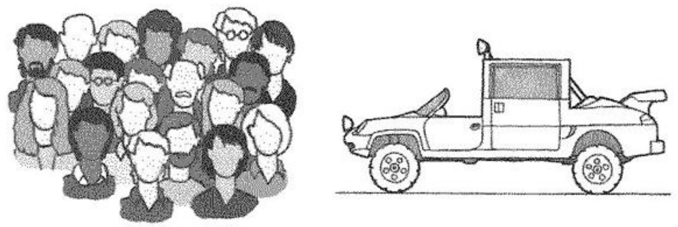
Personas for Car Design

Example from A. Cooper et al.

- Personas and goals
  - Alesandro (fast and fun)
  - Marge (safe, comfortable)
  - Dale (big loads, reliable)
- Results in 3 designs that differ, e.g.
  - Porsche
  - City SUV
  - Pick-up

- Average

  A car that is:
  - fast, fun, safe, comfortable, for big loads, and reliable
- The car that fulfils all requirements, will not fulfil them well…

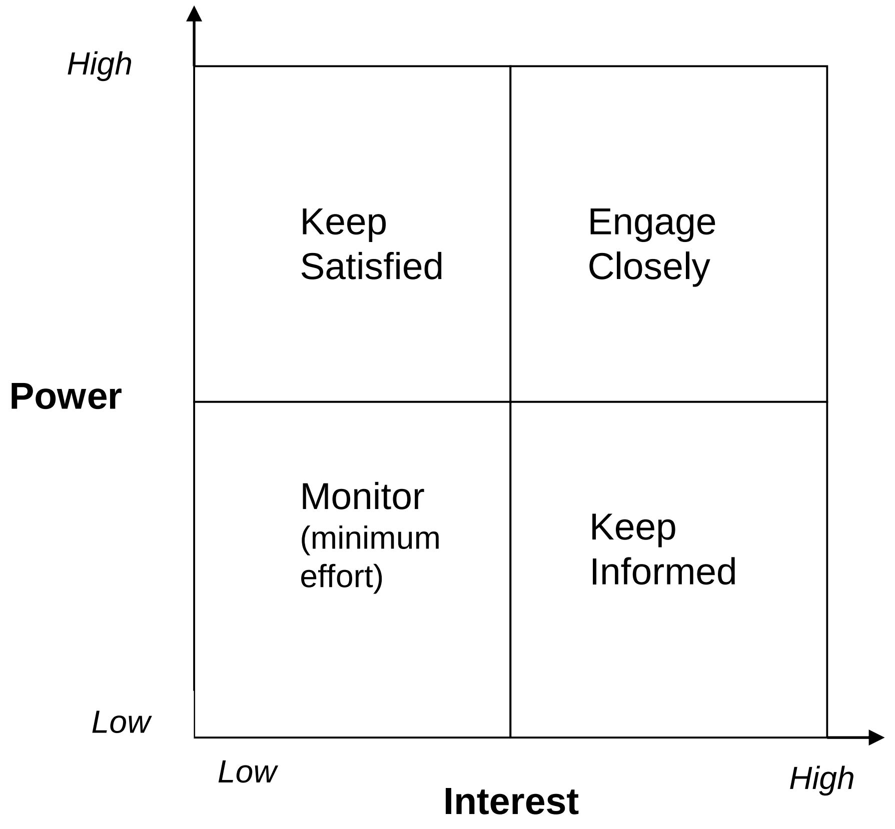


Cooper, A., Reimann, R., & Cronin, D. (2007). About face 3: the essentials of interaction design. John Wiley & Sons. p.80ff

Stakeholder Analysis

Who is interested in the system?

- Identify stakeholders, e.g.
  - Customers, team members, investors, management, suppliers, public, press, shareholders, government, community, sales partners, family, …
- Categorise stakeholders
  - Interest in the project?
  - Influence on the team and project (Power)
  - Attitude (positive / negative)
  - Reasons for attitude
- Force-field analysis
  - Place people in the diagram
  - Revisit throughout the project



https://www.mindtools.com/pages/article/newPPM_07.htm
Mendelow, Aubrey L. "Environmental Scanning-The Impact of the Stakeholder Concept." ICIS. 1981.

Did you understand this block?

Can you answer these questions?

- Why it is important to know the goals and tasks of the users?
- Alan Cooper argues “Software today is too often designed to please too many users, resulting in low user satisfaction.”. How can Persons in the design process help to address this problem?
- Conduct a task frequency analysis for a smart TV that is shared by a family
- Explain what personas are.
- Why is it not useful to design for an average user?
- Conduct a hierarchical task analysis for a taking a photo with your phone and posting it on Instagram.

Reference

- Wilfred J. Hansen. 1971. User engineering principles for interactive systems. In Proceedings of the fall joint computer conference. http://doi.acm.org/10.1145/1479064.1479159
- William Hudson. HCI and the Web: a Tale of Two Tutorials: a Cognitive Approach to Interactive System Design and Interaction Design Meets Agility. ACM interactions 12(1), 2005, 49-51
- Cooper, A., Reimann, R., & Cronin, D. (2007). About face 3: the essentials of interaction design. John Wiley & Sons.
- https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/personas
- https://www.mindtools.com/pages/article/newPPM_07.htm
- Mendelow, Aubrey L. "Environmental Scanning-The Impact of the Stakeholder Concept." ICIS. 1981.

Eight Golden Rules

according to Ben Shneiderman

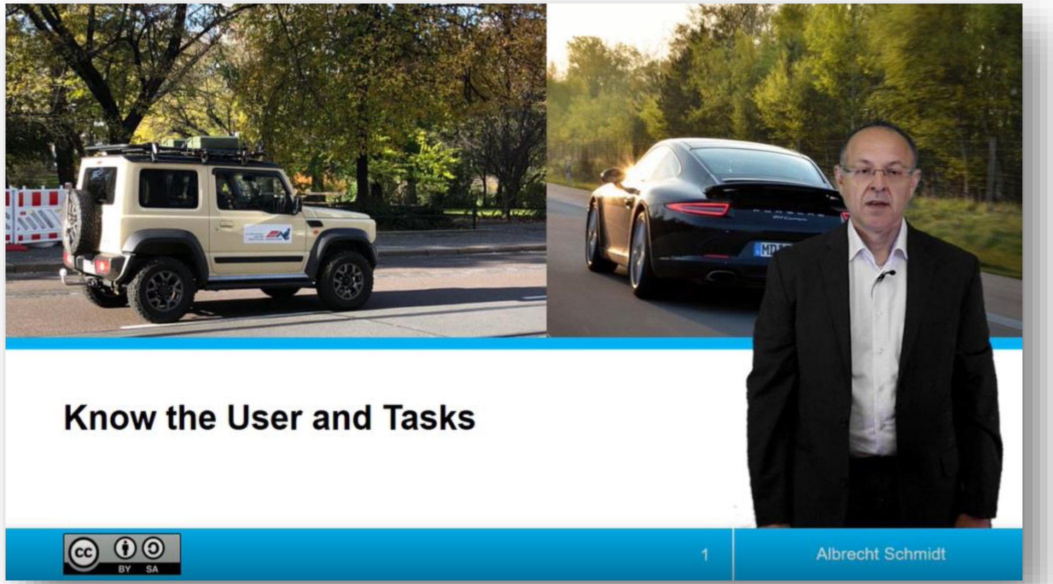
Learning Goals

- Know the Golden Rules according to B. Sheiderman
- Understand …
  - The meaning of the Golden Rules
  - How these Golden Rules help to improve the quality of user interfaces
- Be able to …
  - explain the Golden Rules and give examples
  - discuss user interface designs with regard to these Golden Rules

Principles for UI-Design

By by Ben Shneiderman et al.

- **Principle 1: Recognize User Diversity**
- Principle 2: Follow the Eight Golden Rules
- Principle 3: Prevent Errors



Know the User and Tasks

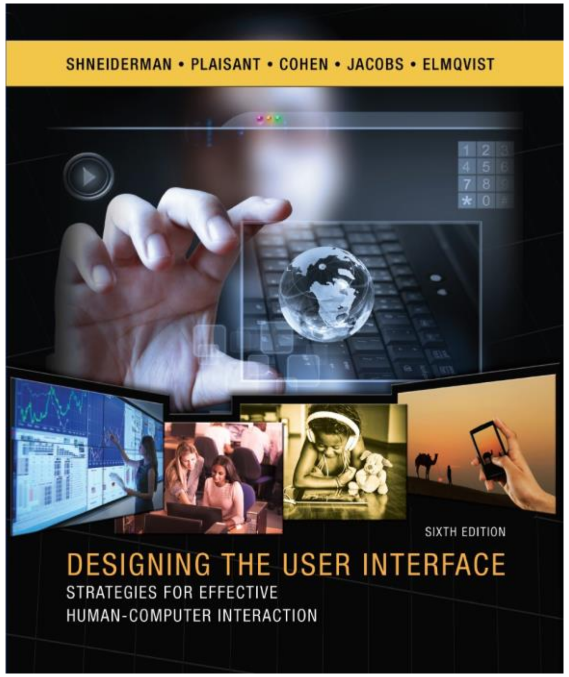
1

Albrecht Schmidt

Principles for UI-Design

By by Ben Shneiderman et al.

- Principle 1: Recognize User Diversity
- **Principle 2: Follow the Eight Golden Rules**
- Principle 3: Prevent Errors


SHNEIDERMAN • PLAISANT • COHEN • JACOBS • ELMQVIST
SIXTH EDITION
DESIGNING THE USER INTERFACE
STRATEGIES FOR EFFECTIVE
HUMAN-COMPUTER INTERACTION

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Follow the 8 Golden Rules*

[*we list 9 here, there are different versions]

1. Strive for consistency
2. Seek universal usability
3. Offer informative feedback
4. Design dialogues to yield closure
5. Prevent Errors
6. Permit easy reversal of actions
7. Keep users in control
8. Reduce short-term memory load

- Enable frequent users to use shortcuts (was 2.)

https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Strive for consistency

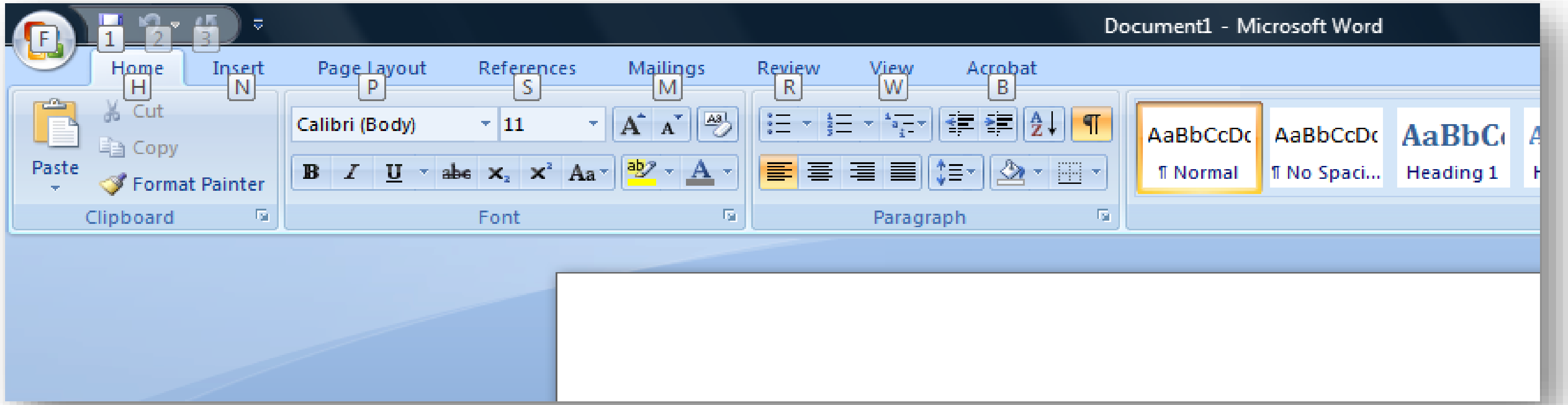
- “Consistent **sequences of actions** should be required in similar situations;
- identical **terminology** should be used in prompts, menus, and help screens;
- and consistent **color, layout, capitalization**, fonts, and so on, should be employed throughout.
- **Exceptions**, such as required confirmation of the delete command or no echoing of passwords, should be comprehensible and limited in number”

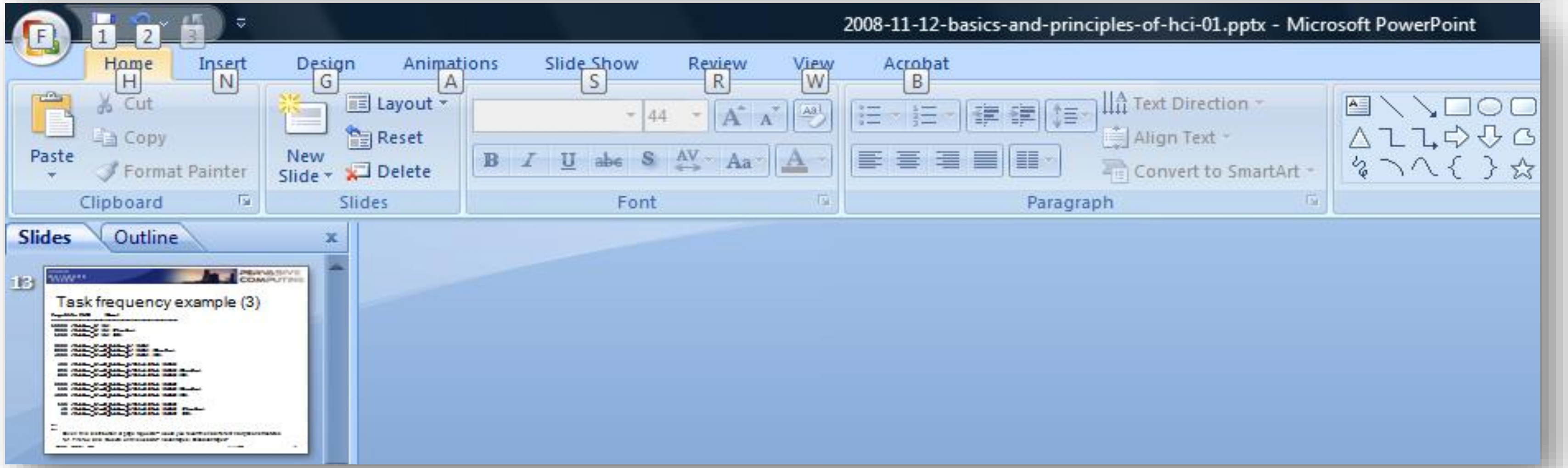
https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Strive for consistency

Example: Consistency across applications

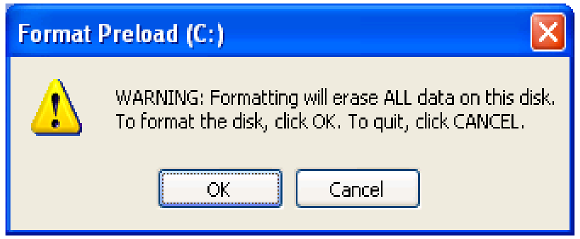




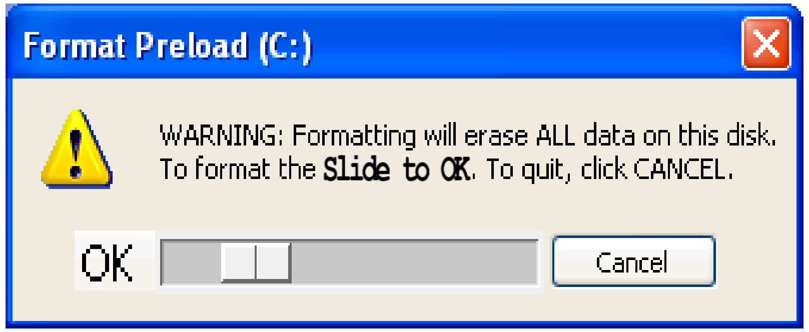
Inconsistencies

Always strive for consistency?

- Dragging file operations?
  - folder on same disk vs. folder on different disk
  - file to trash can vs. disk to trash can
- Fitts' Law suggests bigger buttons for more often used operations
- Inconsistency across platforms, e.g.
  - MacOS vs. Windows
  - Websites on different platforms
  - Mobile device vs. TV vs. PC
- Inconsistency could be used for getting attention
  - Mock-up example


Format Preload (C:)

WARNING: Formatting will erase ALL data on this disk.
To format the disk, click OK. To quit, click CANCEL.


Format Preload (C:)

WARNING: Formatting will erase ALL data on this disk.
To format the **Slide to OK**. To quit, click CANCEL.

Seek universal usability

… see Principle 1

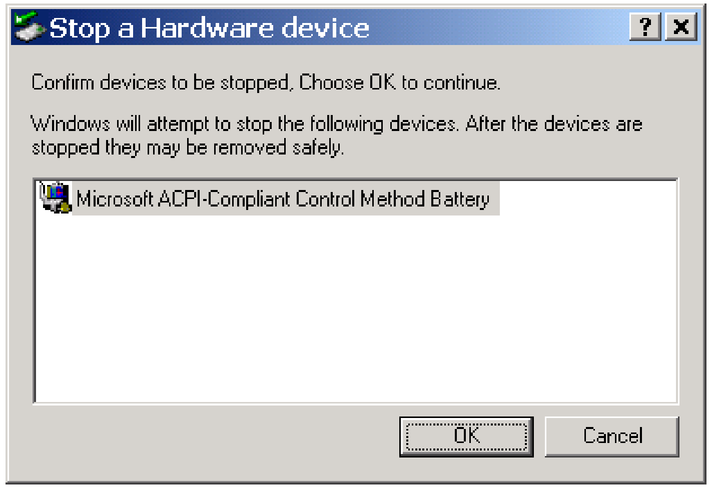
- “Recognize the needs of **diverse users** and design for plasticity, facilitating **transformation of content**.
  - **Novice to expert** differences,
  - **age** ranges, **disabilities**, **international** variations,
  - **technological** diversity
- […] **spectrum of requirements** that guides design.”

https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Offer informative feedback

- “For **every user action**, there should be an **interface feedback**.
- For **frequent** and **minor** actions, the response can be **modest**, whereas
- for **infrequent** and **major** actions, the response should be more **substantial**.”

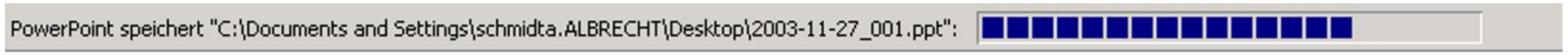


Stop a Hardware device

Confirm devices to be stopped, Choose OK to continue.

Windows will attempt to stop the following devices. After the devices are stopped they may be removed safely.

Microsoft ACPI-Compliant Control Method Battery

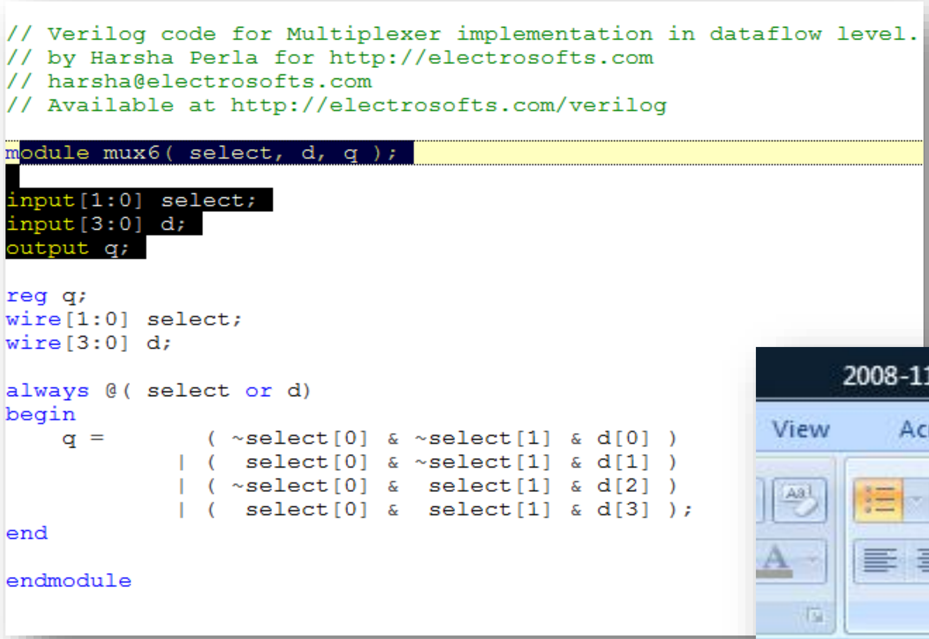


PowerPoint speichert "C:\Documents and Settings\schmidta.ALBRECHT\Desktop\2003-11-27_001.ppt":

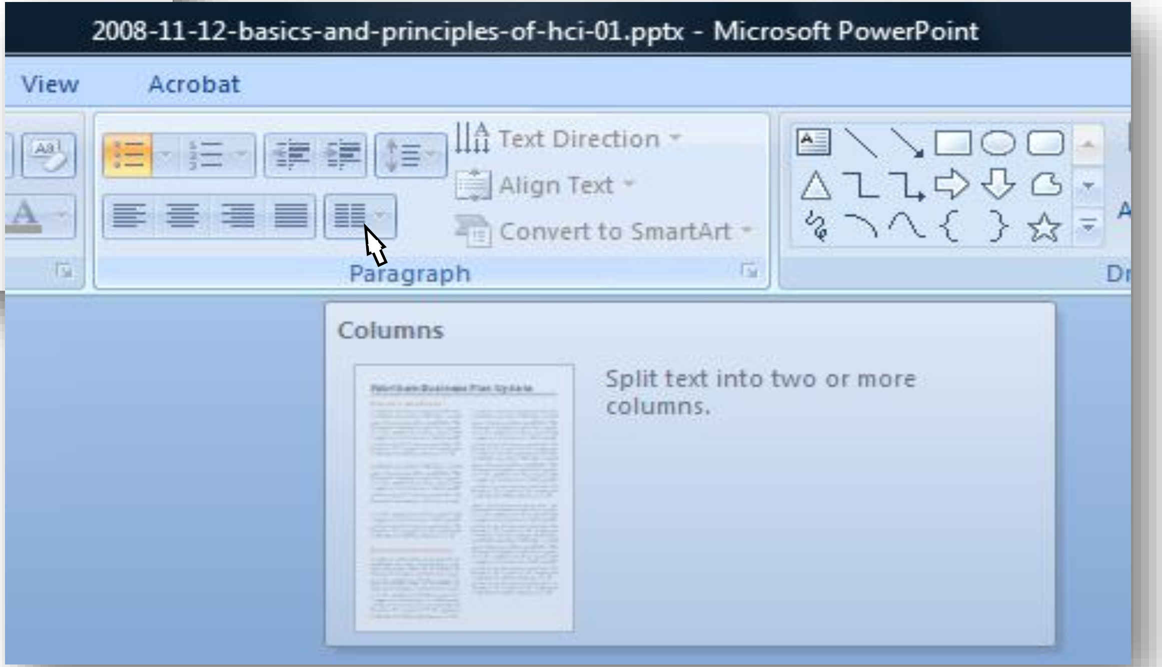
https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Offer informative feedback



```
// Verilog code for Multiplexer implementation in dataflow level.
// by Harsha Perla for http://electrosofts.com
// harsha@electrosofts.com
// Available at http://electrosofts.com/verilog

module mux6( select, d, q );

input[1:0] select;
input[3:0] d;
output q;

reg q;
wire[1:0] select;
wire[3:0] d;

always @( select or d)
begin
    q =       ( ~select[0] & ~select[1] & d[0] )
            | (  select[0] & ~select[1] & d[1] )
            | ( ~select[0] &  select[1] & d[2] )
            | (  select[0] &  select[1] & d[3] );
end

endmodule
```



2008-11-12-basics-and-principles-of-hci-01.pptx - Microsoft PowerPoint

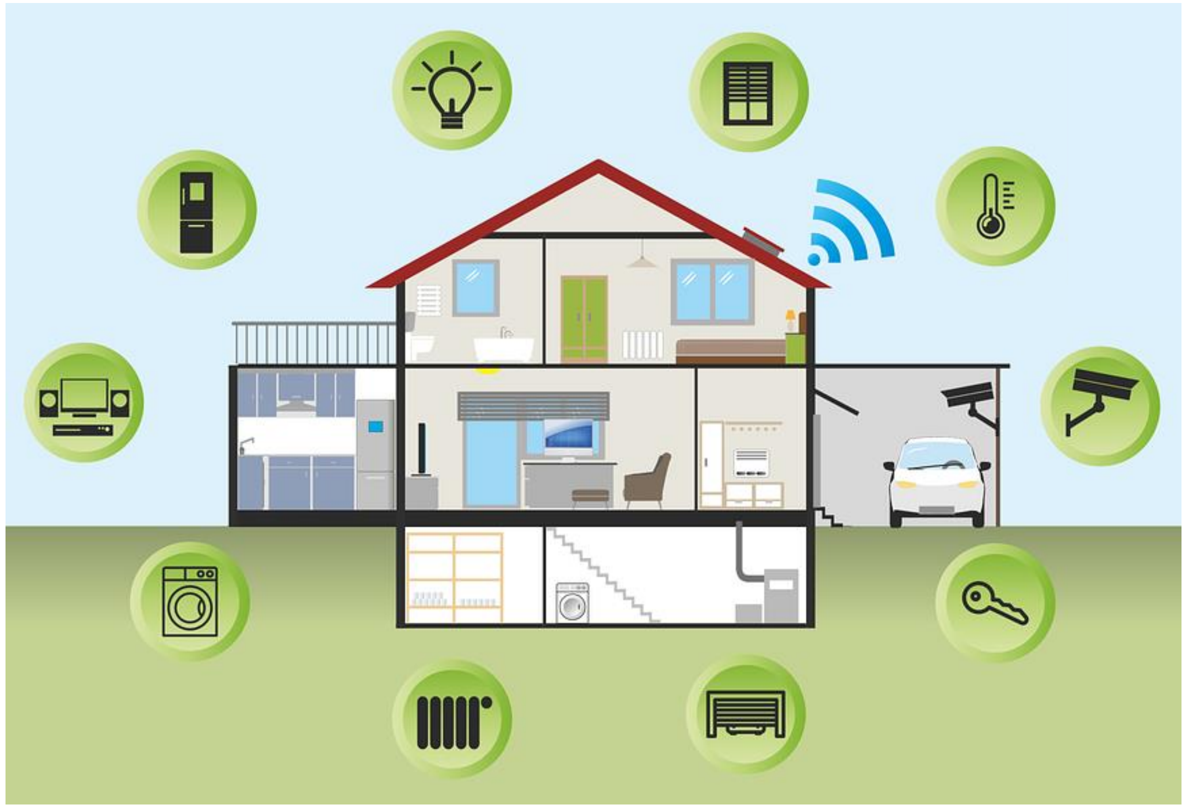
Columns

Split text into two or more columns.

Offer informative feedback

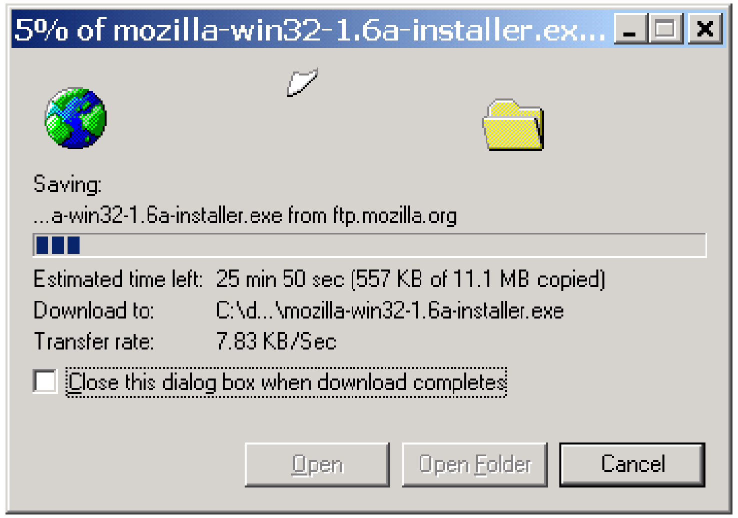
Will ubiquitous computing change this?

Feedback vs. Calm Computing



Design dialogues to yield closure

- “**Sequences of actions** should be organized into groups with a **beginning**, **middle**, and **end**.
- Informative **feedback at the completion** of a group of actions gives users the satisfaction of accomplishment, a sense of relief […]
- For example, e-commerce websites move users from selecting products to the checkout, ending with a clear confirmation page that completes the transaction.”
- Important for actions that are **not immediate** and span over a longer time or multiple steps


5% of mozilla-win32-1.6a-installer.ex...

Saving:
...a-win32-1.6a-installer.exe from ftp.mozilla.org

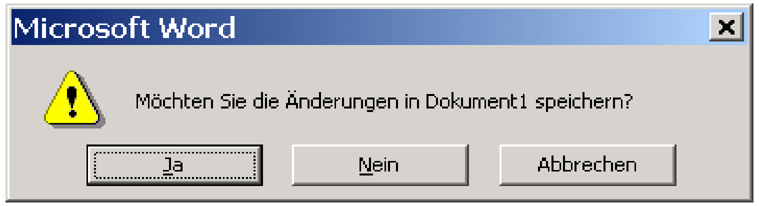
Estimated time left: 25 min 50 sec (557 KB of 11.1 MB copied)
Download to: C:\d...\mozilla-win32-1.6a-installer.exe
Transfer rate: 7.83 KB/Sec

☐ Close this dialog box when download completes

https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Prevent Errors


Microsoft Word

Möchten Sie die Änderungen in Dokument1 speichern?

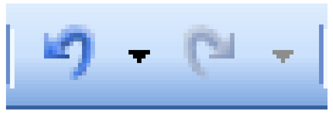
- As much as possible, **design** the interface so that **users cannot make serious errors** […]
- If users make an error, the interface should offer simple, constructive, and specific instructions for **recovery**“
- Detecting errors
- Different options how to handle it:
  - Involve the user with dialogs (current practice)
  - Prevent the error **or its consequences on system level** (e.g. create backups/versions when a file is overwritten, keep all files that have been created by the user)

https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Permit easy reversal of actions

- As much as possible, **actions should be reversible**.
- This feature **relieves anxiety**, since users know that errors can be undone, and **encourages exploration** […]
- The units of reversibility may be a single action, a data-entry task, or a complete group of actions"
- Providing **UNDO functions**
  - Possibly with infinite depth
  - Over sessions
- Not trivial (conceptually as well as technically)
  - *write a text, copy it into the clipboard, undo the writing, the text is still in the clipboard…*



https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Permit easy reversal of actions

Discussion: Why is this not simple?

- What is the cost?
  - Memory?
  - Complexity?
- Why is UNDO sometimes not easy to implement and it may be even impossible?

https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Permit easy reversal of actions

In Ubiquitous Computing?

- As a basic rule – all actions should be reversible
- When is this not possible?
  - Communication applications (e.g. email)
  - Smart environments
  - Machines
  - Cars
- In certain settings processes and basic **physical laws prevent reversal of actions**. Here an interaction layer (buffering user interaction) may be possible – but not always (e.g. breaks, emergency stop)

Keep users in control

- “Experienced users strongly desire the sense that they are **in charge of the interface** and that the interface responds to their actions.
- They **don’t want surprises** or changes in familiar behavior”
- Avoid non-causality
- The system should be predictable
- Current developments (AI, Ubicomp) are in contrast:
  - Intelligent agents
  - Smart environments

https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Or just feeling in control

Keep users in control

- People have to complete tasks under noisy conditions
  - Group A can switch off the noise (remark: if the switch is used, the study results cannot be used)
  - Group B has no influence over the noise
- What happens?

Urban Stress: Experiments on Noise and Social Stressors. DC Glass, JE Singer - 1972 - Academic Press

Or just feeling in control

Keep users in control

- People have to complete tasks under noisy conditions
  - Group A can switch off the noise (remark: if the switch is used, the study results cannot be used)
  - Group B has no influence over the noise

- What happens?
  - Group A performed significantly better than the group B
  - Feeling in control helps

Urban Stress: Experiments on Noise and Social Stressors. DC Glass, JE Singer - 1972 - Academic Press

Reduce short-term memory load

- “Humans’ **limited capacity for information processing** in short-term memory requires that designers avoid interfaces in which users must remember information from one display and then use that information on another display.”
- “rule of thumb is that people can remember ‘**seven plus or minus two chunks**’ of information”
- The system should remember, not the user
  - Make information that is required visible
  - Recognition is easier than recall, use memory aids (visual or audio)

https://www.cs.umd.edu/~ben/goldenrules.html

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

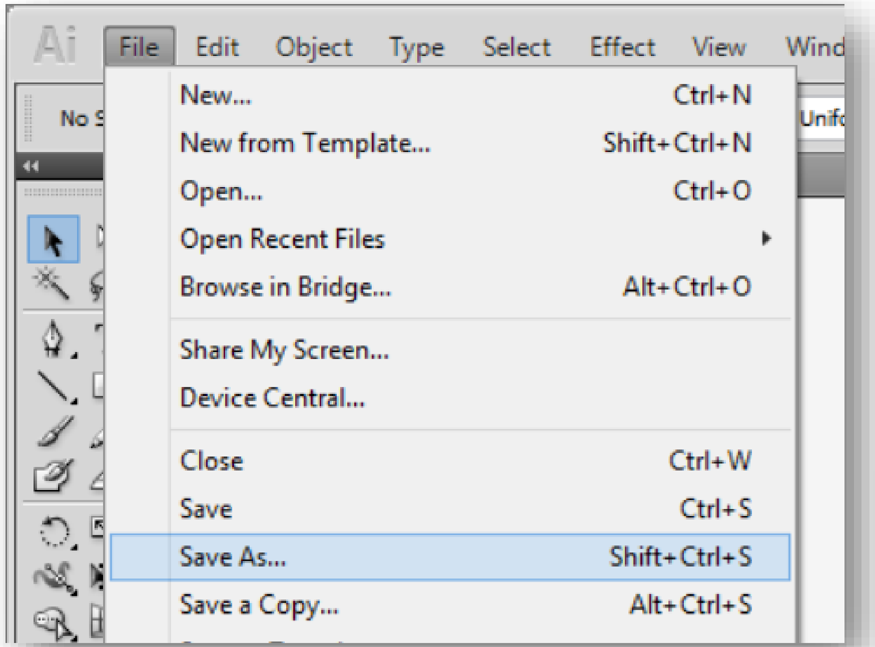
Enable frequent users to use shortcuts

- Improves **speed** for **experienced users**
- Shortcuts on different levels
  - Access to **single commands**, e.g. CTRL+S or toolbar
  - **Customizing** of commands and environments, e.g. printer pre-set (duplex, A4, …)
  - **Reusing actions** performed, e.g. history in command lines, macro functionality
- Shortcuts to single commands are related to consistency
  - CTRL+X, CTRL+C, CTRL+V in Microsoft applications for cut, copy and paste
  - However CTRL+S (saving a document) is only implemented in some applications…

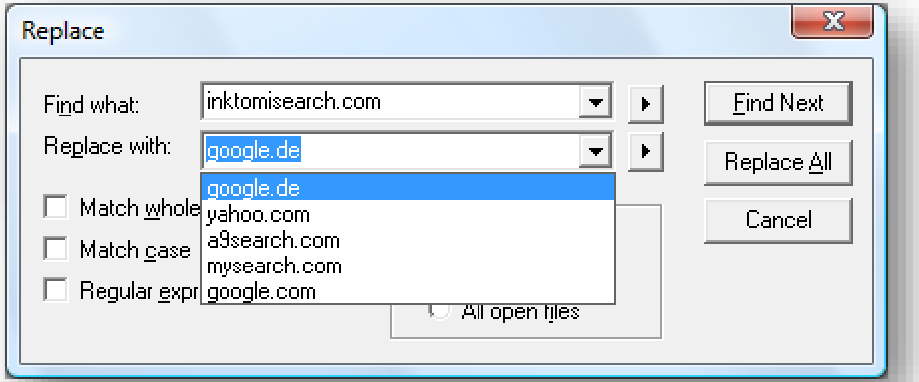
https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces

Keyboard shortcuts and History

Example: Enable frequent users to use shortcuts



New...	Ctrl+N
New from Template...	Shift+Ctrl+N
Open...	Ctrl+O
Open Recent Files	
Browse in Bridge...	Alt+Ctrl+O
Share My Screen...	
Device Central...	
Close	Ctrl+W
Save	Ctrl+S
Save As...	Shift+Ctrl+S
Save a Copy...	Alt+Ctrl+S



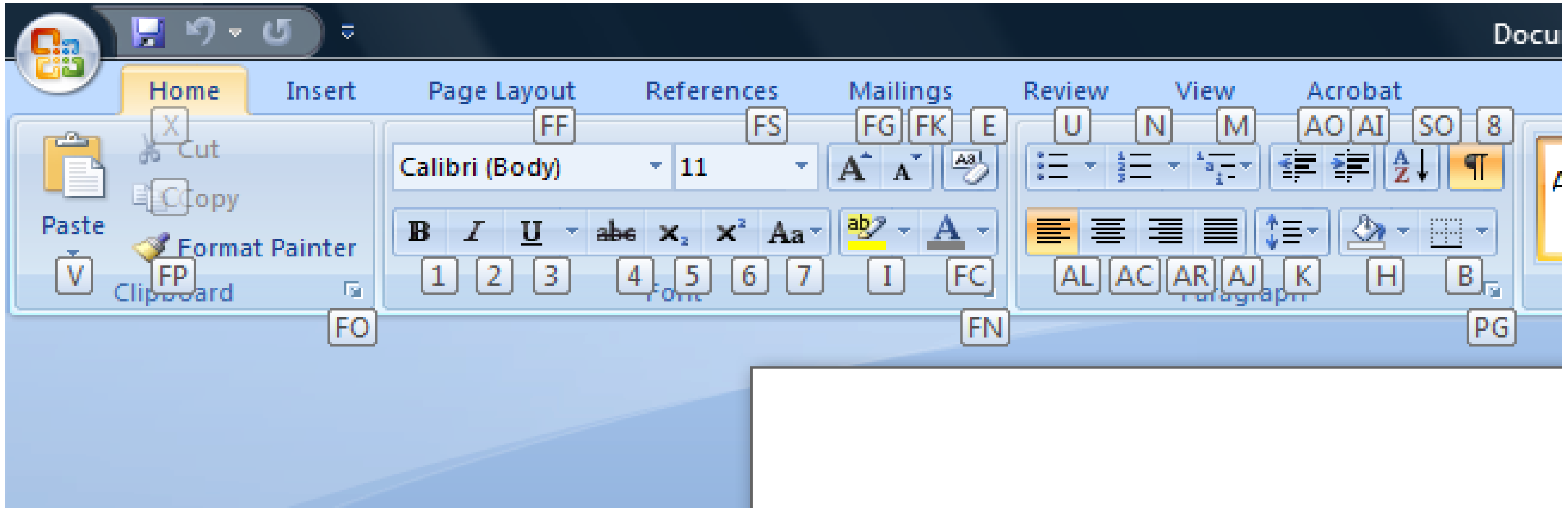
Find what: inktomisearch.com

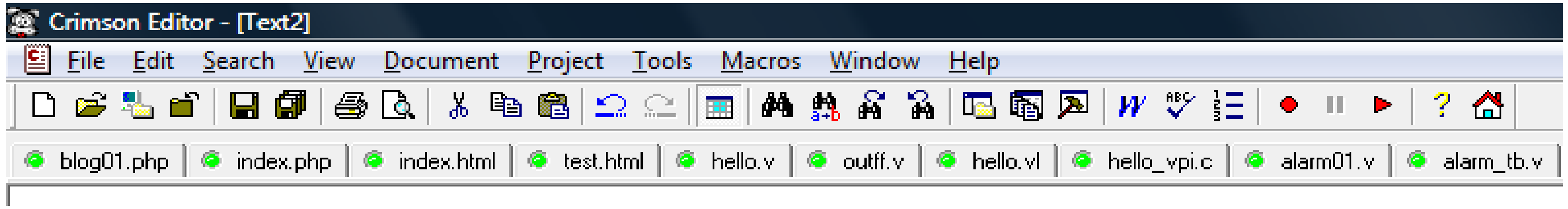
Replace with: google.de

google.de
yahoo.com
a9search.com
mysearch.com
google.com

Keyboard shortcuts and Toolbars

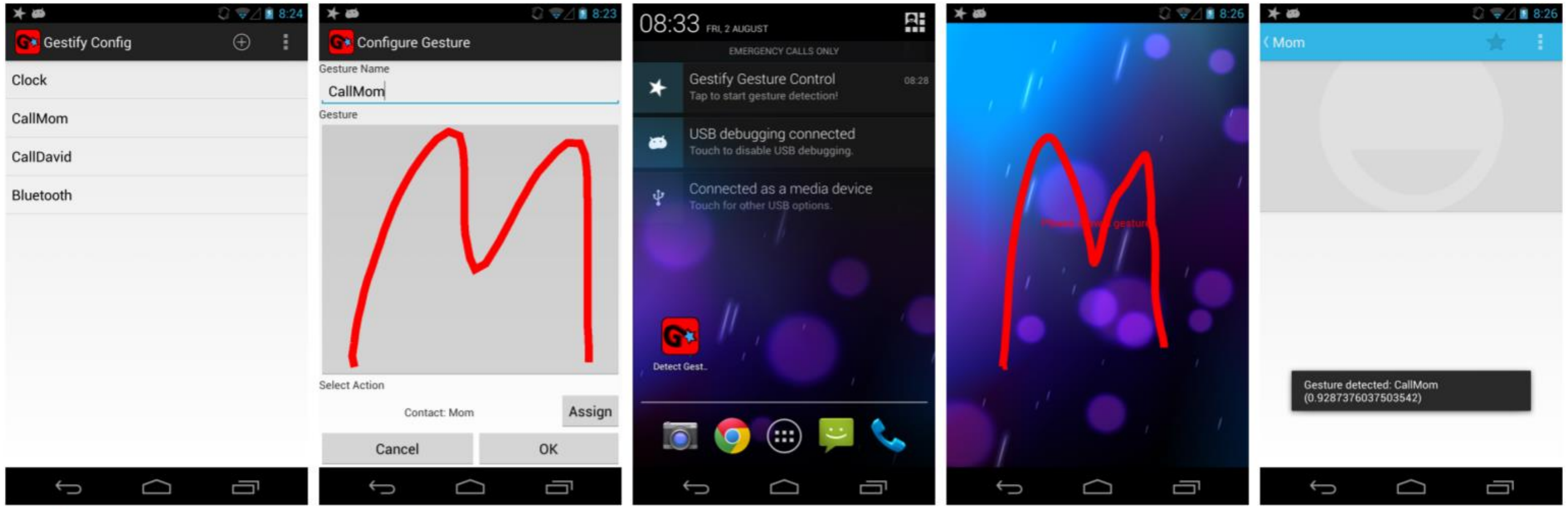
Example: Enable frequent users to use shortcuts





Shortcut Gestures on Smartphones

Example: Enable frequent users to use shortcuts



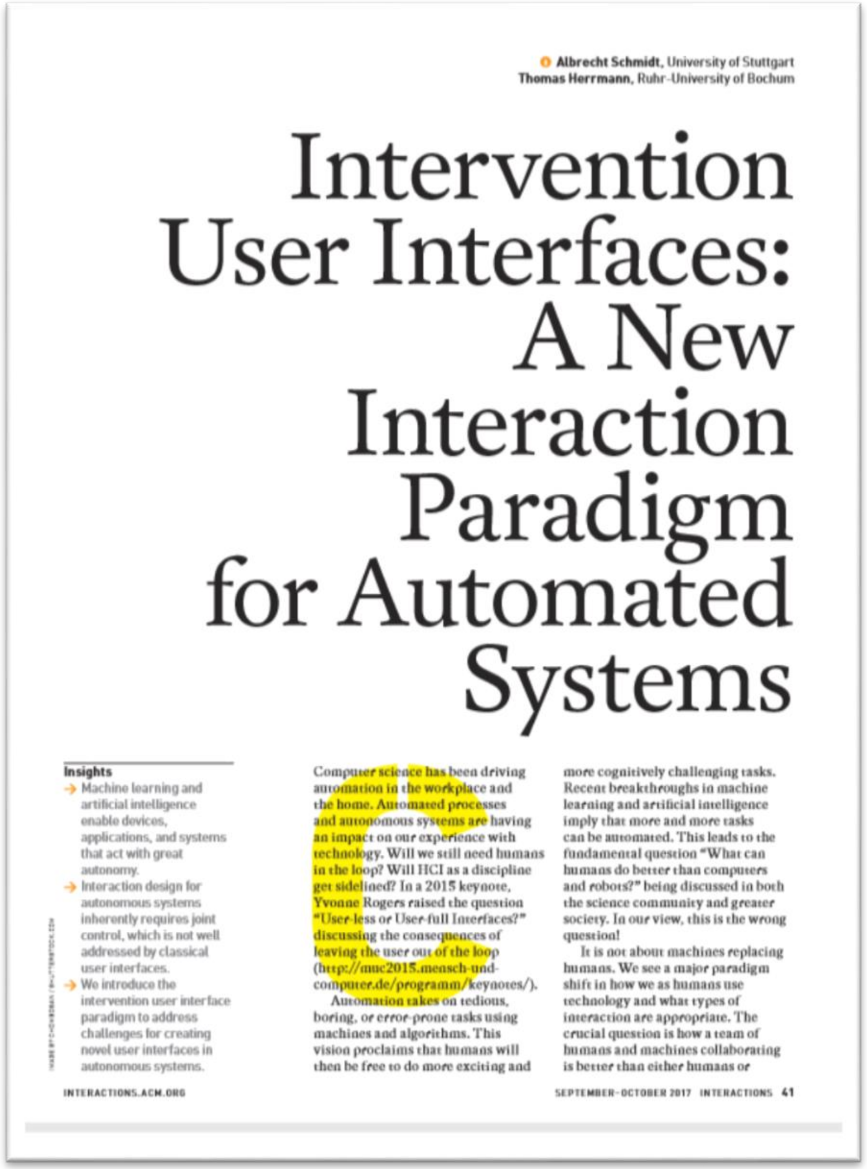
(a) List of existing gestures. (b) Add a new gesture. (c) Trigger detection through one of the provided activators, e.g., notification bar. (d) Draw gesture in recognition activity. (e) A pop-up indicates that the gesture is detected.

Figure 1. Gestify is an Android application that allows users to define and execute their own shortcut gestures. If gestures were trained to the system they can be recognized via different activators, i.e., through the lock screen, the notification bar, the wallpaper, and through a separate activity. If a gesture is detected, a toast is shown and the linked action is executed.

Poppinga, B., Sahami Shirazi, A., Henze, N., Heuten, W., & Boll, S. (2014, September). Understanding shortcut gestures on mobile touch devices. In Proc. of the 16th int. conf. on Human-computer interaction with mobile devices & services (pp. 173-182). ACM.

How can humans stay in control?

- "In the future, we believe that a large class of **automated and autonomous systems** allow for **joint control**, where the majority of decisions are automated but **where users can intervene.**



Albrecht Schmidt, University of Stuttgart
Thomas Herrmann, Ruhr-University of Bochum

Intervention User Interfaces: A New Interaction Paradigm for Automated Systems

Insights

- Machine learning and artificial intelligence enable devices, applications, and systems that act with great autonomy.
- Interaction design for autonomous systems inherently requires joint control, which is not well addressed by classical user interfaces.
- We introduce the intervention user interface paradigm to address challenges for creating novel user interfaces in autonomous systems.

Computer science has been driving automation in the workplace and the home. Automated processes and autonomous systems are having an impact on our experience with technology. Will we still need humans in the loop? Will HCI as a discipline get sidelined? In a 2015 keynote, Yvonne Rogers raised the question "User-less or User-full Interfaces?" discussing the consequences of leaving the user out of the loop (http://muc2015.mensch-und-computer.de/programm/keynotes/).

Automation takes on tedious, boring, or error-prone tasks using machines and algorithms. This vision proclaims that humans will then be free to do more exciting and more cognitively challenging tasks. Recent breakthroughs in machine learning and artificial intelligence imply that more and more tasks can be automated. This leads to the fundamental question "What can humans do better than computers and robots?" being discussed in both the science community and greater society. In our view, this is the wrong question!

It is not about machines replacing humans. We see a major paradigm shift in how we as humans use technology and what types of interaction are appropriate. The crucial question is how a team of humans and machines collaborating is better than either humans or

INTERACTIONS.ACM.ORG

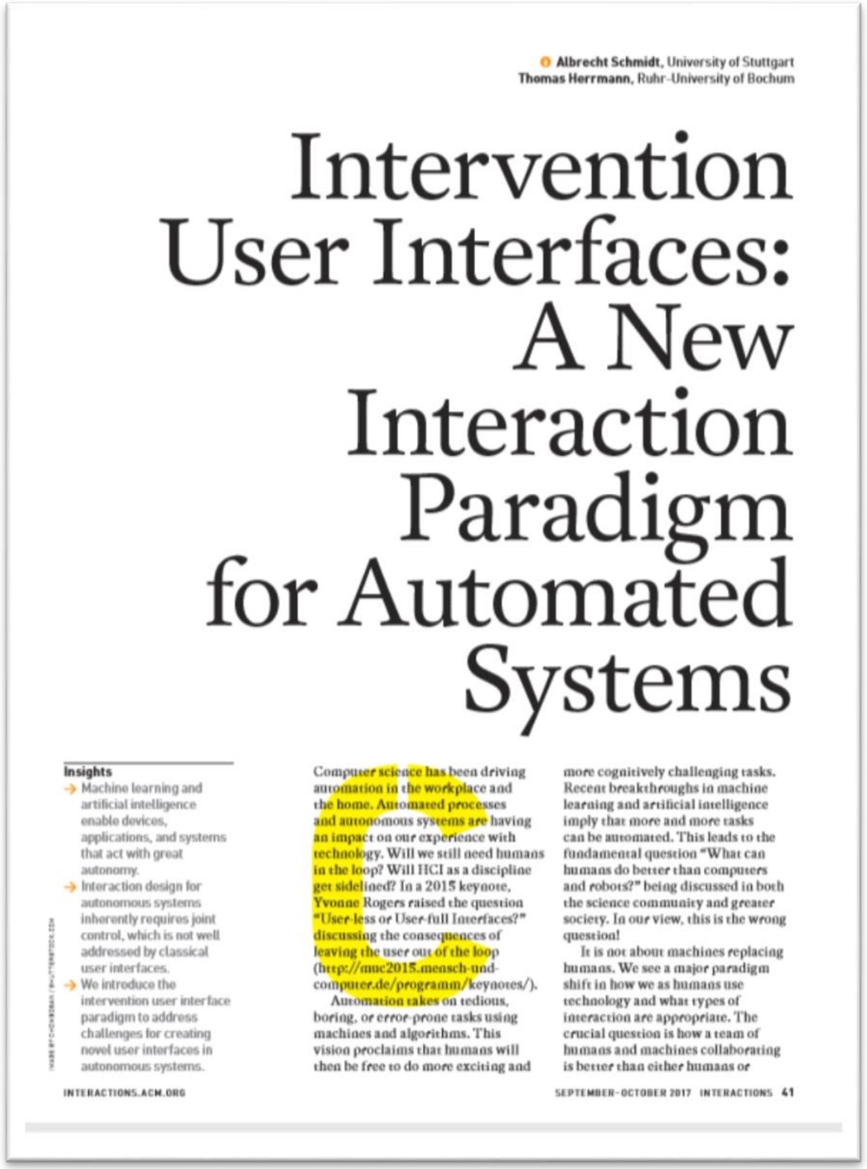
SEPTEMBER-OCTOBER 2017 INTERACTIONS 41

Schmidt, A., & Herrmann, T. (2017). Intervention user interfaces: a new interaction paradigm for automated systems. *interactions*, *24*(5), 40-45.

How can humans stay in control?

Design Principles for Intervention user interfaces

- Ensure expectability and predictability.
- Communicate options for interventions.
- Allow easy exploration of interventions.
- Easy reversal of automated and intervention actions.
- Minimize required attention.
- Communicate how control is shared.



Albrecht Schmidt, University of Stuttgart
Thomas Herrmann, Ruhr-University of Bochum

Intervention User Interfaces: A New Interaction Paradigm for Automated Systems

Insights

- Machine learning and artificial intelligence enable devices, applications, and systems that act with great autonomy.
- Interaction design for autonomous systems inherently requires joint control, which is not well addressed by classical user interfaces.
- We introduce the intervention user interface paradigm to address challenges for creating novel user interfaces in autonomous systems.

Computer science has been driving automation in the workplace and the home. Automated processes and autonomous systems are having an impact on our experience with technology. Will we still need humans in the loop? Will HCI as a discipline get sidelined? In a 2015 keynote, Yvonne Rogers raised the question "User-less or User-full Interfaces?" discussing the consequences of leaving the user out of the loop (http://muc2015.mensch-und-computer.de/programm/keynotes/).

Automation takes on tedious, boring, or error-prone tasks using machines and algorithms. This vision proclaims that humans will then be free to do more exciting and more cognitively challenging tasks. Recent breakthroughs in machine learning and artificial intelligence imply that more and more tasks can be automated. This leads to the fundamental question "What can humans do better than computers and robots?" being discussed in both the science community and greater society. In our view, this is the wrong question!

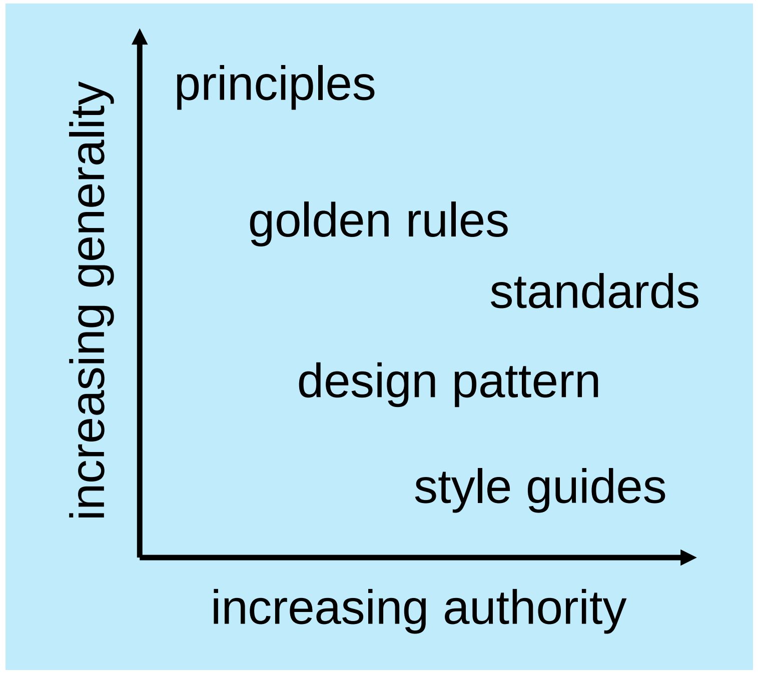
It is not about machines replacing humans. We see a major paradigm shift in how we as humans use technology and what types of interaction are appropriate. The crucial question is how a team of humans and machines collaborating is better than either humans or

INTERACTIONS.ACM.ORG SEPTEMBER-OCTOBER 2017 INTERACTIONS 41

Schmidt, A., & Herrmann, T. (2017). Intervention user interfaces: a new interaction paradigm for automated systems. *interactions*, *24*(5), 40-45.

Types of Design Rules

- **Principles**
  - abstract design rules
- **Golden rules and heuristics**
  - more concrete than principles
- **Standards**
  - (very) detailed design rules
- **Design patterns**
  - generic solution for a specific problem
- **Style guides**
  - provided for devices, operating systems, widget libraries



- **Authority**: whether or not a rule must be followed or whether it is just suggested
- **Generality**: applied to many design situations or focused on specific application situation.

Did you understand this block?

Can you answer these questions?

- Name the Eight Golden Rules?
- Give an example for offering informative feedback.
- In which contexts may offering feedback be problematic?
- Where is easy reversal of action on principle not possible?
- How can consistency help with the learnability of systems?
- Name three areas in the design of user interfaces where consistency should be realized?

Reference

- Shneiderman, B. The Eight Golden Rules of Interface Design. https://www.cs.umd.edu/~ben/goldenrules.html
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/
- Urban Stress: Experiments on Noise and Social Stressors. DC Glass, JE Singer - 1972 - Academic Press
- Euphemia Wong. Shneiderman's Eight Golden Rules Will Help You Design Better Interfaces. https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces
- Poppinga, B., Sahami Shirazi, A., Henze, N., Heuten, W., & Boll, S. (2014, September). Understanding shortcut gestures on mobile touch devices. In Proc. of the 16th int. conf. on Human-computer interaction with mobile devices & services (pp. 173-182). ACM.
- Schmidt, A., & Herrmann, T. (2017). Intervention user interfaces: a new interaction paradigm for automated systems. *interactions*, *24*(5), 40-45.

License

Human Error

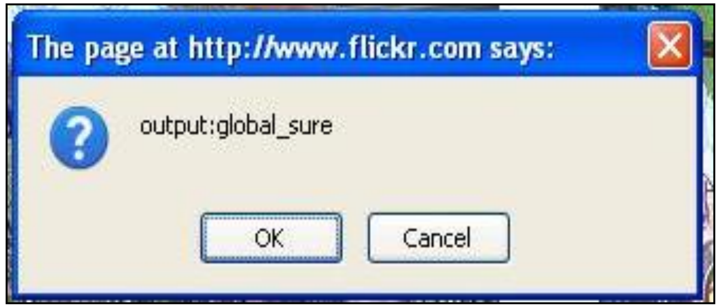
Learning Goals

- Understand …
  - When and how errors should be communicated
  - How human error and design are not independent
  - The difference between mistakes and slips
  - The concept of constraints and how they can help to reduce errors
- Be able to …
  - explain the assumptions that are made about what errors users make
  - discuss different types of slips and give examples
  - Discuss how a user interface designs can be improved to prevent errors

Communicating Systems Errors

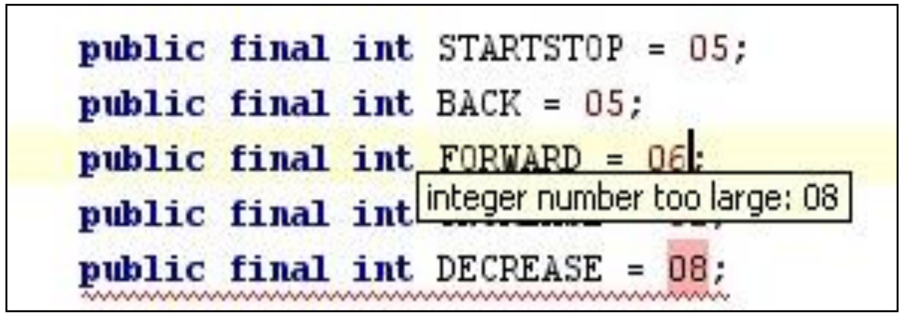
- What to do, if an error in the system occurs?
- Will the user benefit from knowing about the error?
- Can the user do something about the error?
- What other solutions are available?

- If the error is provided to the user it must be
  - Understandable (the user gets what the problem is)
  - Actionable (the user gets options to do things)



The page at http://www.flickr.com says:

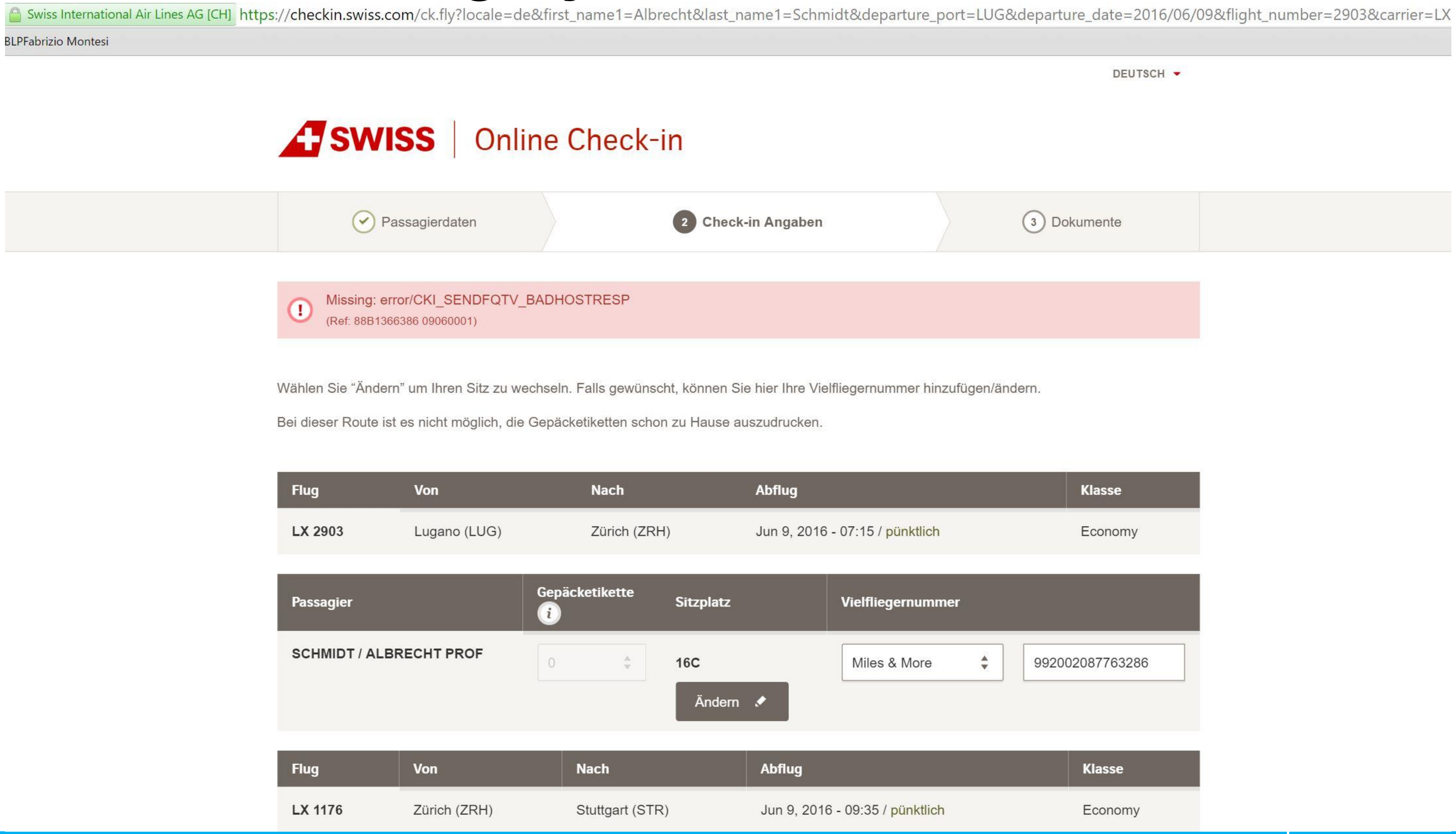
output:global_sure



```
public final int STARTSTOP = 05;
public final int BACK = 05;
public final int FORWARD = 06;
public final int
public final int DECREASE = 08;
```

integer number too large: 08

Communicating Systems Errors



Swiss International Air Lines AG [CH] https://checkin.swiss.com/ck.fly?locale=de&first_name1=Albrecht&last_name1=Schmidt&departure_port=LUG&departure_date=2016/06/09&flight_number=2903&carrier=LX

BLPFabrizio Montesi

Missing: error/CKI_SENDFQTV_BADHOSTRESP
(Ref: 88B1366386 09060001)

Wählen Sie "Ändern" um Ihren Sitz zu wechseln. Falls gewünscht, können Sie hier Ihre Vielfliegernummer hinzufügen/ändern.

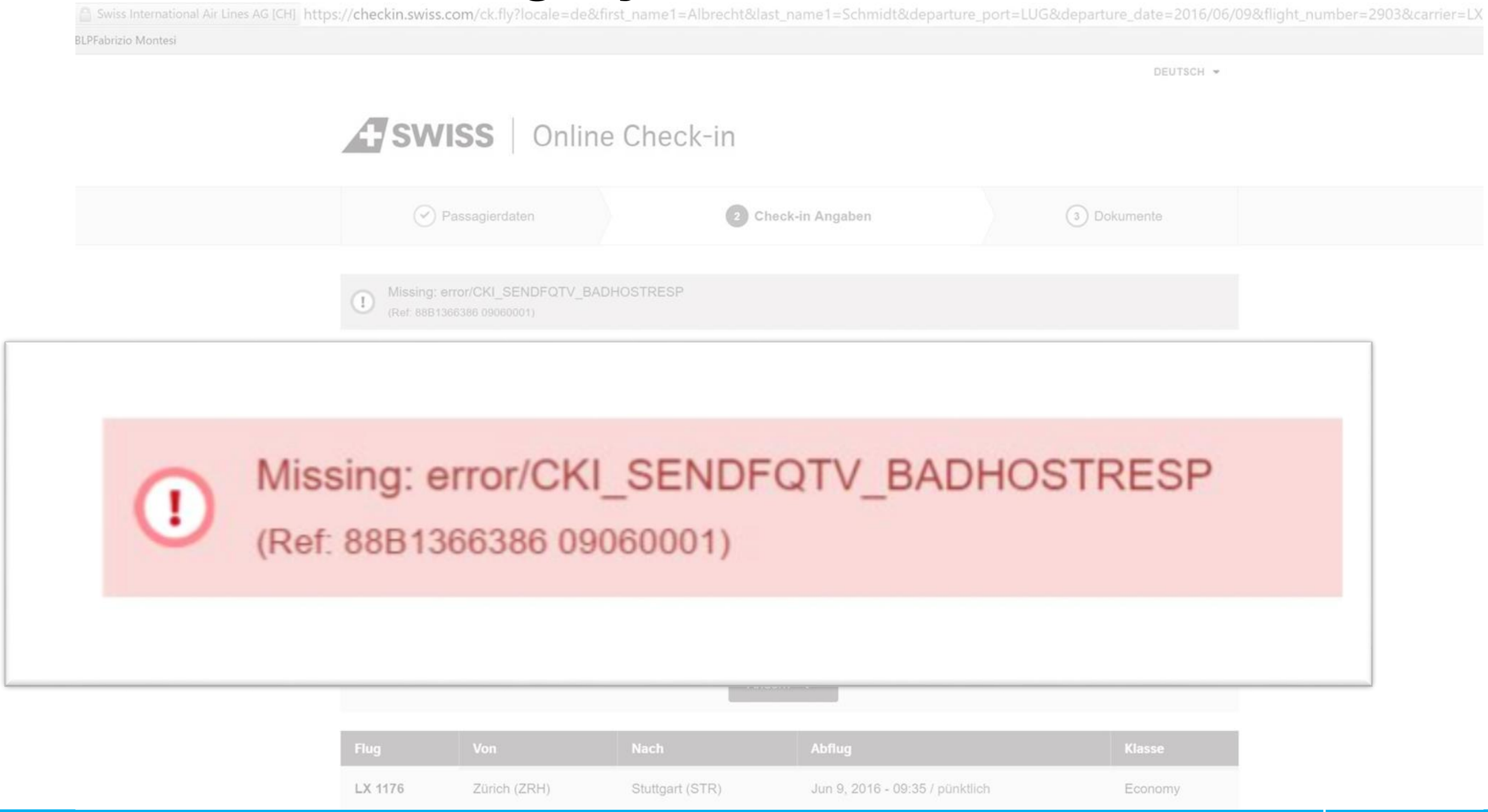
Bei dieser Route ist es nicht möglich, die Gepäcketiketten schon zu Hause auszudrucken.

Flug	Von	Nach	Abflug	Klasse
LX 2903	Lugano (LUG)	Zürich (ZRH)	Jun 9, 2016 - 07:15 / pünktlich	Economy

Passagier	Gepäcketikette	Sitzplatz	Vielfliegernummer	
SCHMIDT / ALBRECHT PROF	0	16C Ändern	Miles & More	992002087763286

Flug	Von	Nach	Abflug	Klasse
LX 1176	Zürich (ZRH)	Stuttgart (STR)	Jun 9, 2016 - 09:35 / pünktlich	Economy

Communicating Systems Errors



Swiss International Air Lines AG [CH] https://checkin.swiss.com/ck.fly?locale=de&first_name1=Albrecht&last_name1=Schmidt&departure_port=LUG&departure_date=2016/06/09&flight_number=2903&carrier=LX

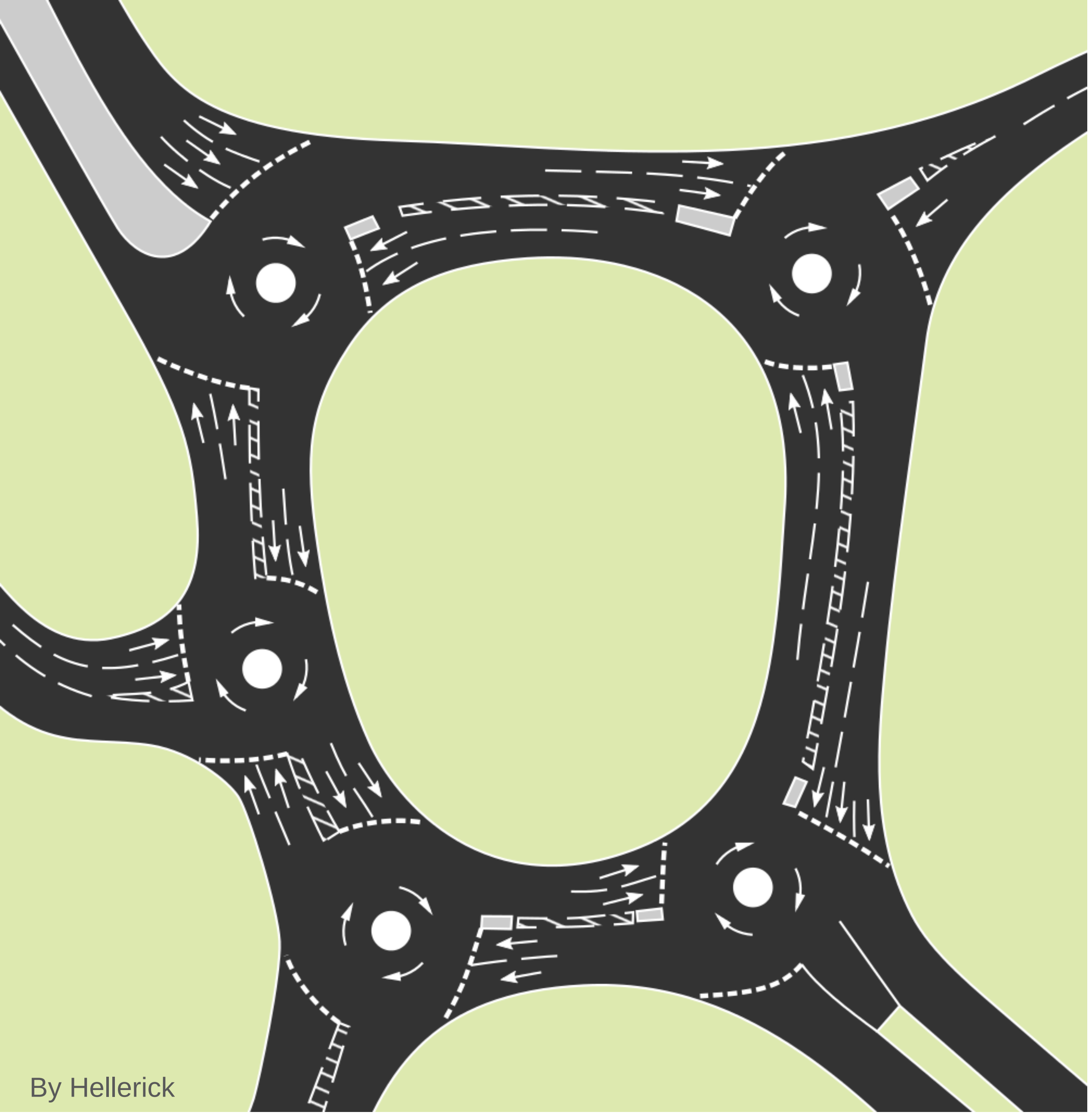
BLPFabrizio Montesi

Missing: error/CKI_SENDFQTV_BADHOSTRESP
(Ref: 88B1366386 09060001)

Missing: error/CKI_SENDFQTV_BADHOSTRESP

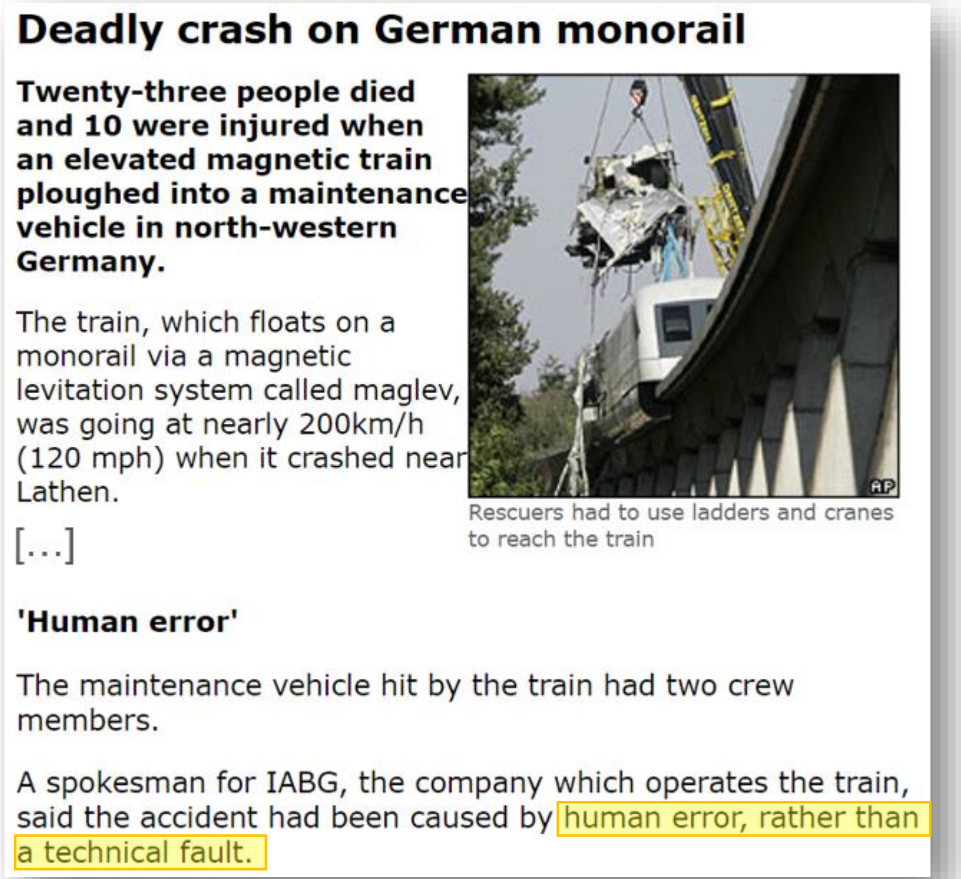
(Ref: 88B1366386 09060001)

Flug	Von	Nach	Abflug	Klasse
LX 1176	Zürich (ZRH)	Stuttgart (STR)	Jun 9, 2016 - 09:35 / pünktlich	Economy



Who’s fault is it,
if an accident happens

Human Error as the Ultimate Explanation?



Deadly crash on German monorail

Twenty-three people died and 10 were injured when an elevated magnetic train ploughed into a maintenance vehicle in north-western Germany.

The train, which floats on a monorail via a magnetic levitation system called maglev, was going at nearly 200km/h (120 mph) when it crashed near Lathen.

Rescuers had to use ladders and cranes to reach the train

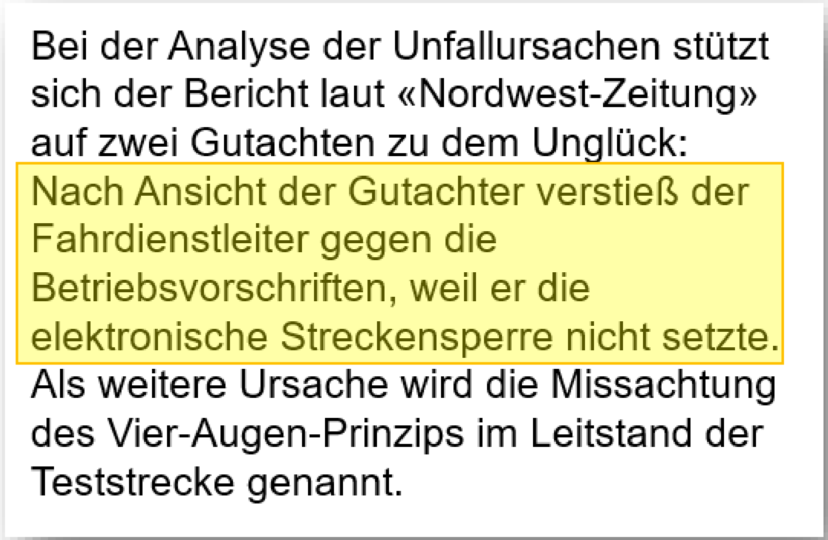
[…]

'Human error'

The maintenance vehicle hit by the train had two crew members.

A spokesman for IABG, the company which operates the train, said the accident had been caused by human error, rather than a technical fault.

http://news.bbc.co.uk/1/hi/world/europe/5370564.stm



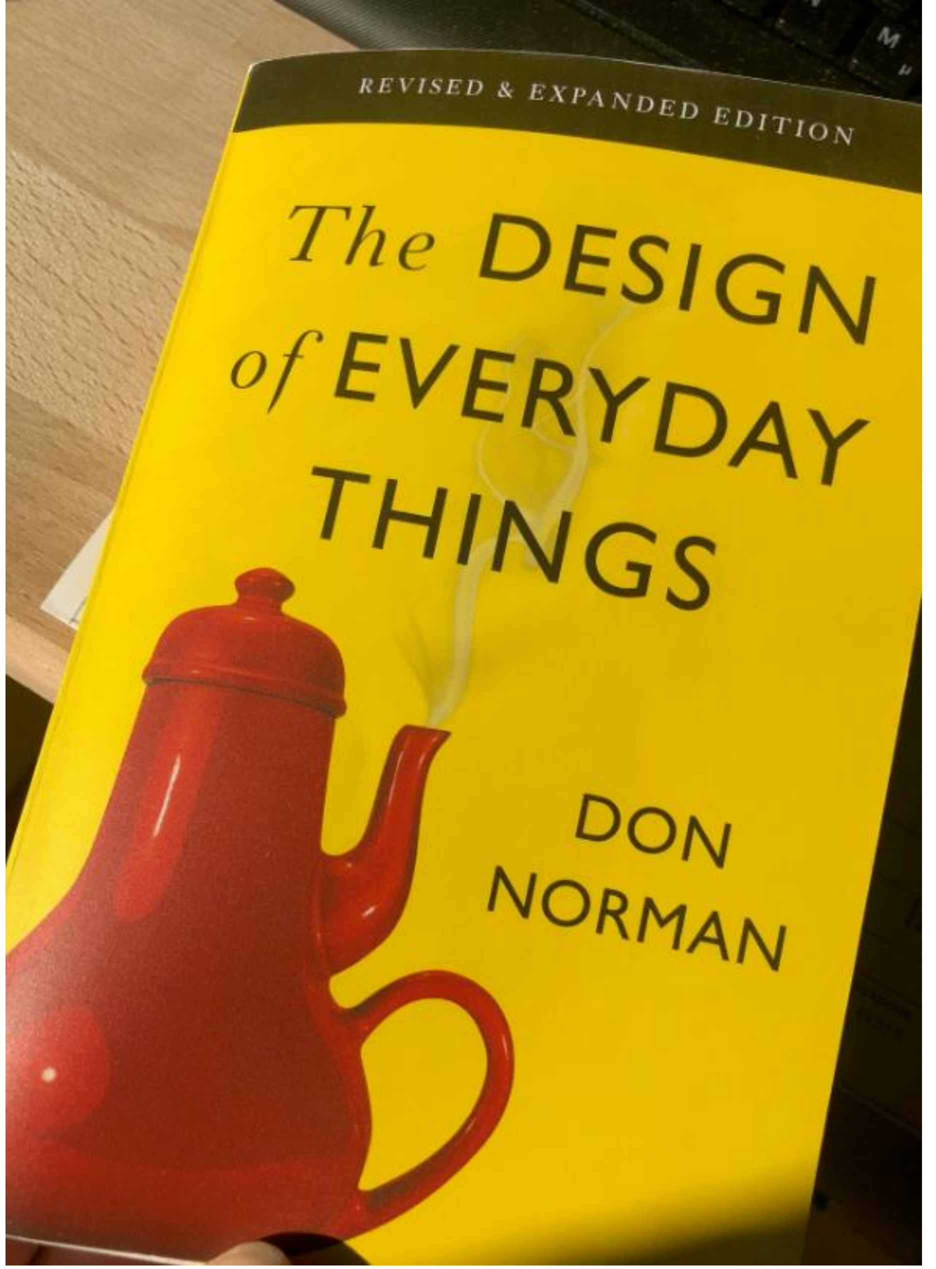
Bei der Analyse der Unfallursachen stützt sich der Bericht laut «Nordwest-Zeitung» auf zwei Gutachten zu dem Unglück: Nach Ansicht der Gutachter verstieß der Fahrdienstleiter gegen die Betriebsvorschriften, weil er die elektronische Streckensperre nicht setzte. Als weitere Ursache wird die Missachtung des Vier-Augen-Prinzips im Leitstand der Teststrecke genannt.

http://www.netzeitung.de/politik/deutschland/720674.html

About (Human) Errors…

… and implications for user interface design

- **“If an error is possible, someone will make it”** (Norman)
- “Human Error” are a starting point to look for design problems.



Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

About (Human) Errors…

… and implications for user interface design

- Design implications
  - Assume **all possible errors will be made**
  - **Minimize the chance** to make errors (constraints)
  - **Minimize the effect** that errors have (is difficult!)
  - Include mechanism to **detect errors**
  - Attempt to make actions **reversible**
- Prevent that users make errors in the first place
  - Make it **impossible to enter wrong commands**
  - Ensure that users **can always recover**

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Understanding Errors

- Errors are routinely made
  - Communication and language is used between people to clarify – more often than one imagines
  - Common understanding of goals and intentions between people helps to overcome errors
- Two fundamental categories
  - Mistakes = wrong goal
    - overgeneralization
    - wrong conclusions
  - Slips = right goal but wrong action
    - Result of “automatic” behaviour
    - Appropriate goal but performance/action is wrong

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Understanding Errors

- **Errors are routinely made**
  - Communication and **language** is used between people to **clarify** – more often than one imagines
  - **Common understanding** of goals and intentions between people helps to overcome errors
- Two fundamental categories
  - **Mistakes = wrong goal**
    - overgeneralization
    - wrong conclusions
  - **Slips = right goal but wrong action**
    - Result of “automatic” behaviour
    - Appropriate goal but performance/action is wrong

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Understanding the types of Slips Users Make

- Capture errors
- Description errors
- Data driven errors
- Associate action errors
- Loss-of-Activation error ~ forgetting
- Mode error

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Capture errors

Understanding the types of Slips Users Make

- **Capture errors**
  - Two actions with common start point, the more familiar one captures the unusual (driving to work on Saturday instead of the supermarket)
- Description errors
- Data driven errors
- Associate action errors
- Loss-of-Activation error ~ forgetting
- Mode error

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Description errors

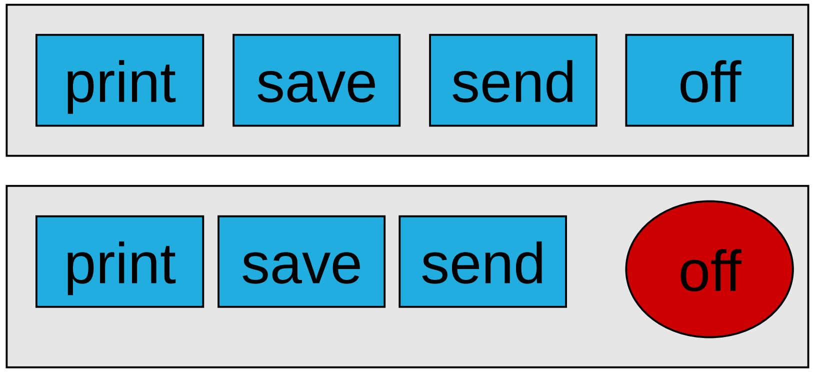
Understanding the types of Slips Users Make

- Capture errors
- **Description errors**
  - Performing an action that is close to the action that one wanted to perform (putting the cutlery in the bin instead of the sink)
- Data driven errors
- Associate action errors
- Loss-of-Activation error ~ forgetting
- Mode error

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

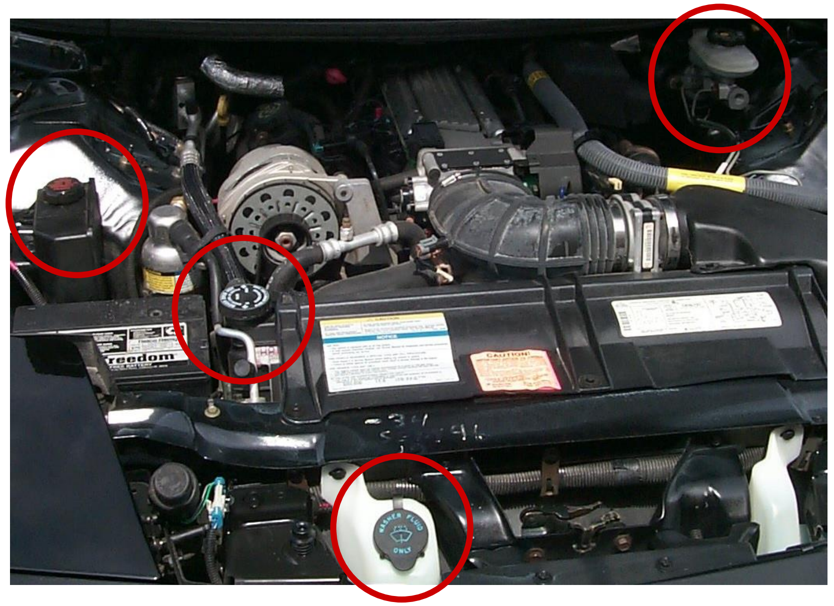
Description errors - Example

Understanding the types of Slips Users Make



- Related to Gestalt theory

- Example Car
  - Different openings for fluids, e.g. oil, water, break, …
  - Openings differ in
    - Size
    - Position
    - Mechanism to open
    - Color



- Design recommendations
  - Make controls for different actions look different

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Data driven errors

Understanding the types of Slips Users Make

- Capture errors
- Description errors
- **Data driven errors**
  - Using data that is visible in a particular moment instead of the data that is well-known (calling the room number you see instead of the phone number you know by heart)
- Associate action errors
- Loss-of-Activation error ~ forgetting
- Mode error

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Associate action errors

Understanding the types of Slips Users Make

- Capture errors
- Description errors
- Data driven errors
- **Associate action errors**
  - You think of something and that influences your action. (e.g. saying come in after picking up the phone)
- Loss-of-Activation error ~ forgetting
- Mode error

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Loss-of-Activation error ~ forgetting

Understanding the types of Slips Users Make

- Capture errors
- Description errors
- Data driven errors
- Associate action errors
- **Loss-of-Activation error ~ forgetting**
  - In a given environment you decided to do something but when leaving then you forgot what you wanted to do. Going back to the start place you remember.
- Mode error

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Mode error

Understanding the types of Slips Users Make

- Capture errors
- Description errors
- Data driven errors
- Associate action errors
- Loss-of-Activation error ~ forgetting
- **Mode error**
  - You forget that you are in a mode that does not allow a certain action or where a action has a different effect

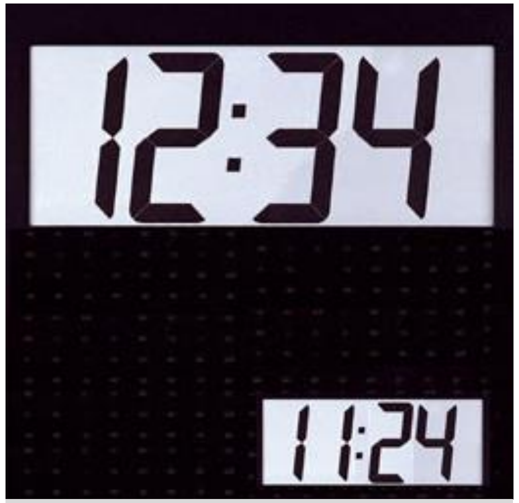
Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Mode error - Example

Understanding the types of Slips Users Make

- Why use modes in the first place?
  - User interface trade-off, e.g.
    - number of buttons needed can be reduced
- Design recommendations
  - Minimize number of modes
  - Make modes always visible
- Example alarm clock
  - Mode vs. mode free
  - Visualization of mode
- What is your solution?
  - Draw the control elements
  - Provide labels

Setting time and alarm with mode?



Setting time and alarm without mode?

Correcting Errors

Actions on different level

- If something goes wrong, we attempt corrections on the lowest level

- A task includes action on different levels
  - Drive to University
  - Get into the car
  - Open the car door
  - Insert car key and turn
  - Apply pressure to the key
  - …

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Preventing Errors

Confirmation is unlikely to prevent Errors

- Example
  - User: “remove the file ‘most-important-work.txt’”
  - computer: “**`are you sure that you want to remove the file ‘most-important-work.txt`**’?”
  - User: “yes”
  - Computer: “**`are you certain`**?”
  - User: “yes of course”
  - Computer: “**`the file ‘most-important-work.txt’ has been removed`**”
  - User: Oops, damm
- The user is not reconsidering the overall action – it only prompts to think about the immediate action (clicking)
- A solution is to make the action reversible

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Detecting Errors

- When “human” errors are detected get into a understandable dialog with the user

Forcing Function

- **Interlock** (e.g. functions can only be done in a certain order)
- **Lock-Ins** (e.g. you can not leave, before you have not done something)
- **Lock-Outs** (e.g. you can get in, before you have not done something)

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

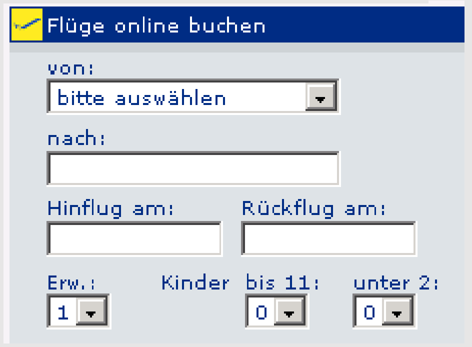
Constraints to prevent errors

- Physical constraints
  - Basic physical limitations
- Semantic constraints
  - Assumption to create something meaningful
- Cultural constraints
  - Borders and context provided by cultural conventions
- Logical constraints
  - Restrictions due to reasoning

- Applying constraints is a design decision!
  - Practical way to realize the principle "prevent errors"

GUI Example

Date unconstrained



Flüge online buchen

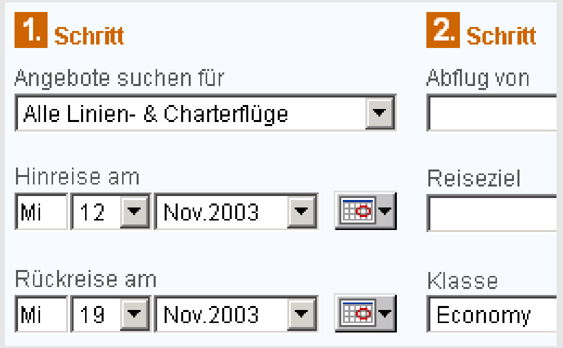
von: bitte auswählen

nach:

Hinflug am: Rückflug am:

Erw.: 1 Kinder bis 11: 0 unter 2: 0

Date constrained



1. Schritt

Angebote suchen für: Alle Linien- & Charterflüge

Hinreise am: Mi 12 Nov.2003

Rückreise am: Mi 19 Nov.2003

2. Schritt

Abflug von

Reiseziel

Klasse: Economy

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

Sketching a Form

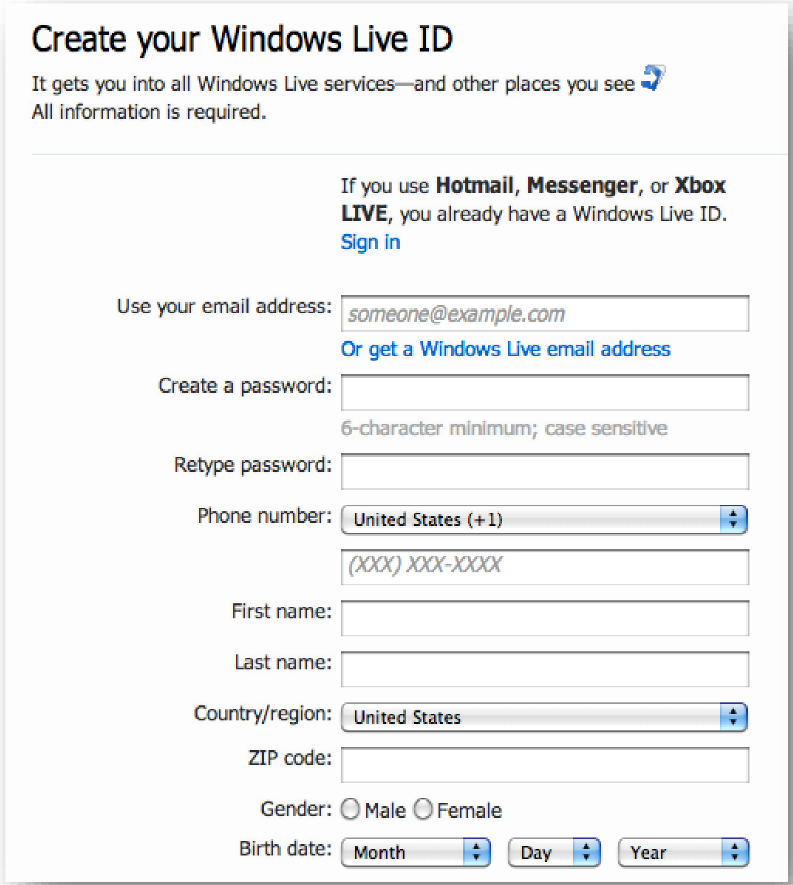
Mini-Exercise: Preventing Errors

- Design a Webform for inputting the following information:
  - Family Name, First Name
  - Country
  - Town, post code, street name and number
  - Email address
  - Gender
  - Birthday incl. year
  - Phone number

- What typical errors do you expect when people fill in the form?
- How to minimize the possibility of errors?
- How to minimize the effect of errors?

What errors do you expect?

Mini-Exercise: How would you prevent them?

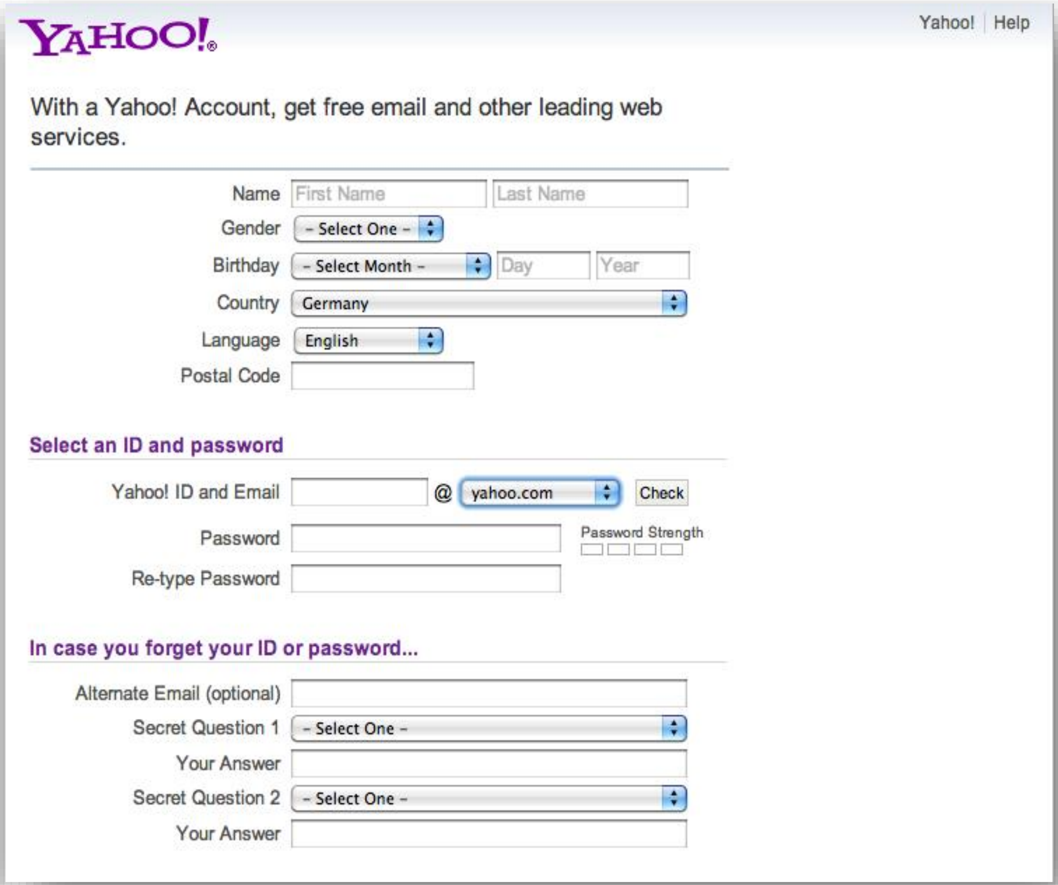

Create your Windows Live ID

It gets you into all Windows Live services—and other places you see

All information is required.

If you use **Hotmail**, **Messenger**, or **Xbox LIVE**, you already have a Windows Live ID.
Sign in

Use your email address:	someone@example.com
	Or get a Windows Live email address
Create a password:	
	6-character minimum; case sensitive
Retype password:	
Phone number:	United States (+1)
	(XXX) XXX-XXXX
First name:	
Last name:	
Country/region:	United States
ZIP code:	
Gender:	Male Female
Birth date:	Month Day Year


Yahoo! | Help

With a Yahoo! Account, get free email and other leading web services.

Name	First Name Last Name
Gender	- Select One -
Birthday	- Select Month - Day Year
Country	Germany
Language	English
Postal Code	

Select an ID and password

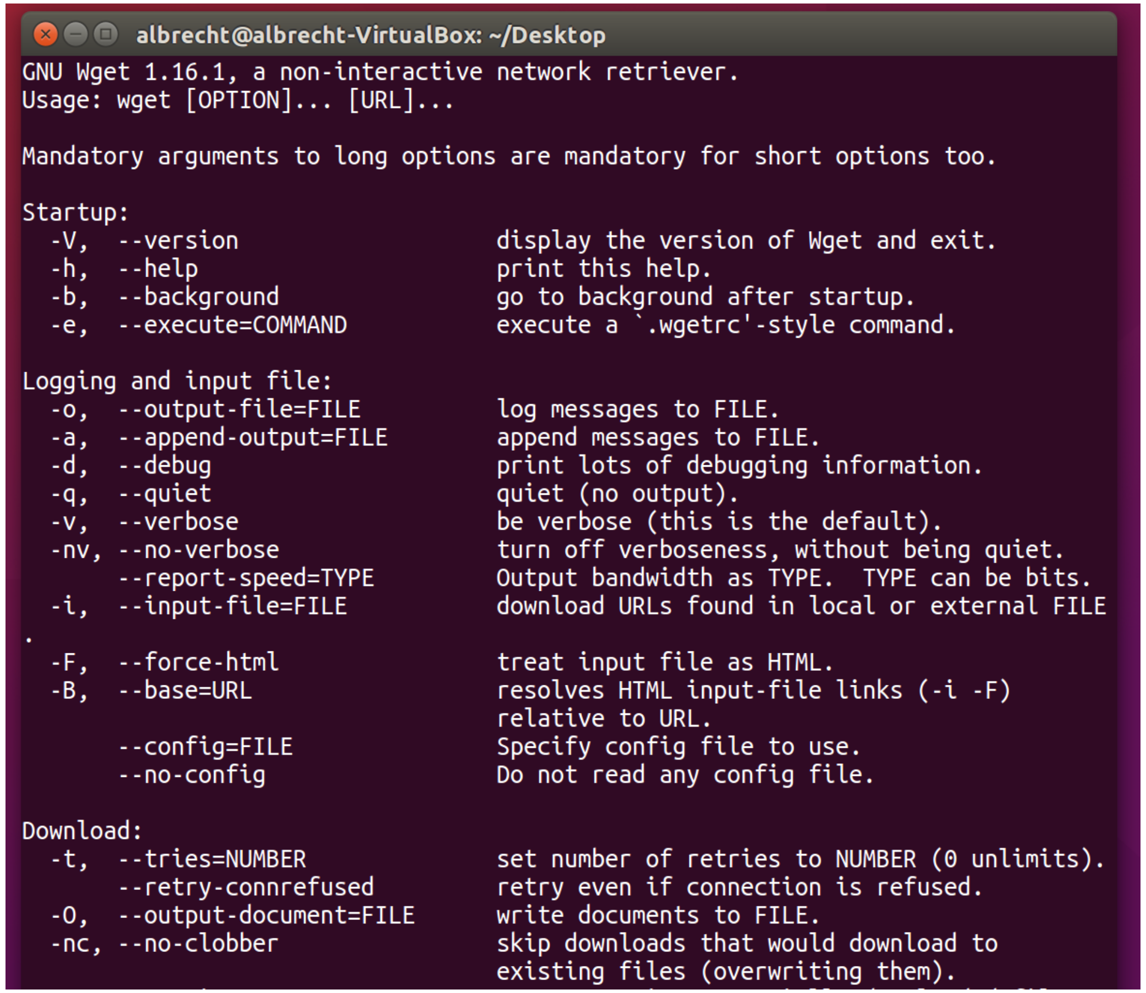
Yahoo! ID and Email	@ yahoo.com Check
Password	Password Strength
Re-type Password	

In case you forget your ID or password...

Alternate Email (optional)	
Secret Question 1	- Select One -
Your Answer	
Secret Question 2	- Select One -
Your Answer	

What errors do you expect?

Mini-Exercise: How would you prevent them?

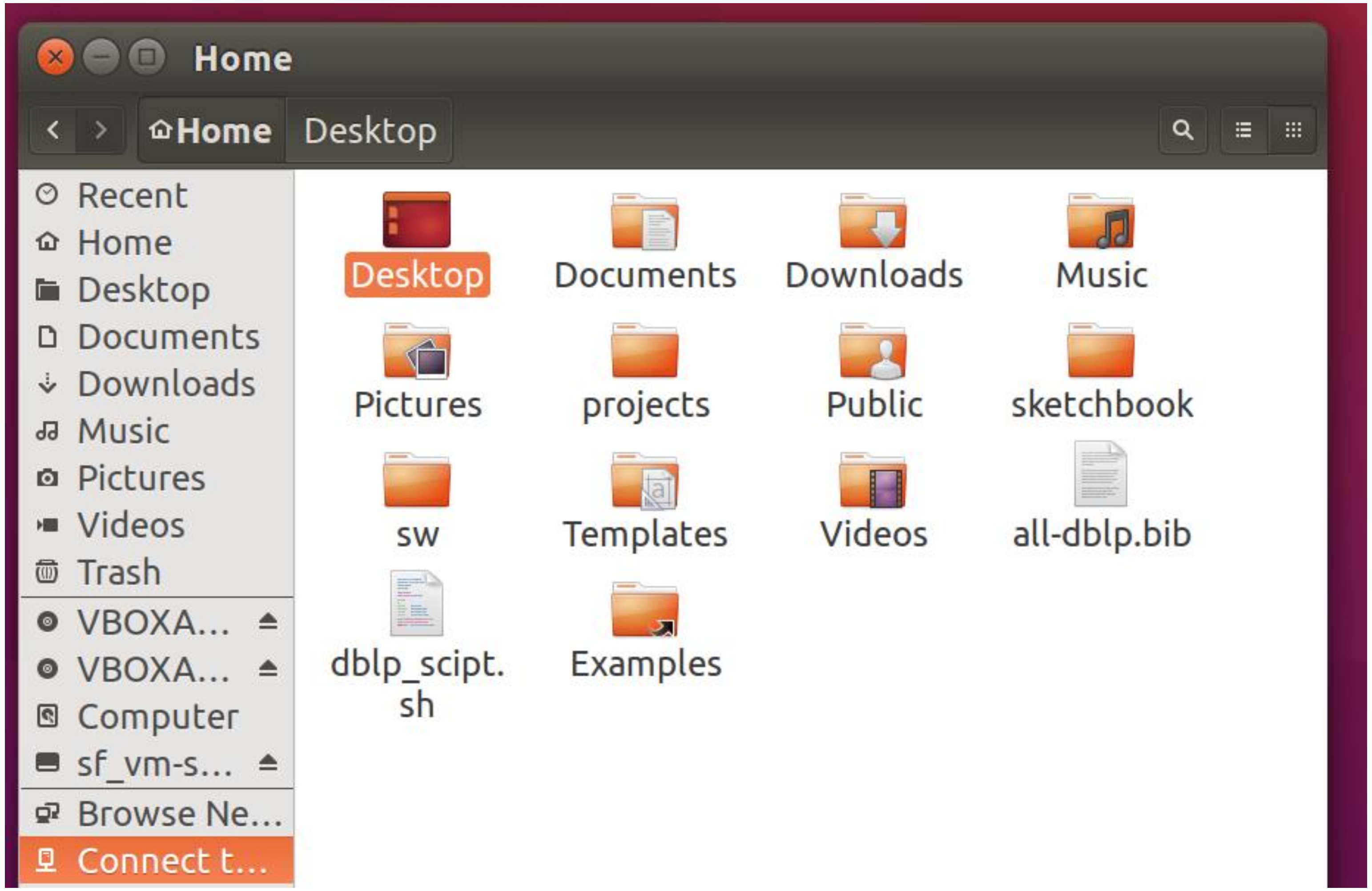


albrecht@albrecht-VirtualBox: ~/Desktop

```
GNU Wget 1.16.1, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...

Mandatory arguments to long options are mandatory for short options too.

Startup:
  -V,  --version                   display the version of Wget and exit.
  -h,  --help                      print this help.
  -b,  --background                go to background after startup.
  -e,  --execute=COMMAND           execute a `.wgetrc'-style command.

Logging and input file:
  -o,  --output-file=FILE          log messages to FILE.
  -a,  --append-output=FILE        append messages to FILE.
  -d,  --debug                     print lots of debugging information.
  -q,  --quiet                     quiet (no output).
  -v,  --verbose                   be verbose (this is the default).
  -nv, --no-verbose                turn off verboseness, without being quiet.
       --report-speed=TYPE         Output bandwidth as TYPE.  TYPE can be bits.
  -i,  --input-file=FILE           download URLs found in local or external FILE
.
  -F,  --force-html                treat input file as HTML.
  -B,  --base=URL                  resolves HTML input-file links (-i -F)
                                   relative to URL.
       --config=FILE               Specify config file to use.
       --no-config                 Do not read any config file.

Download:
  -t,  --tries=NUMBER              set number of retries to NUMBER (0 unlimits).
       --retry-connrefused         retry even if connection is refused.
  -O,  --output-document=FILE      write documents to FILE.
  -nc, --no-clobber                skip downloads that would download to
                                   existing files (overwriting them).
```

What errors do you expect?

Mini-Exercise: How would you prevent them?



Learning Goals

- Understand …
  - When and how errors should be communicated
  - How human error and design are not independent
  - The difference between mistakes and slips
  - The concept of constraints and how they can help to reduce errors
- Be able to …
  - explain the assumptions that are made about what errors users make
  - discuss different types of slips and give examples
  - Discuss how a user interface designs can be improved to prevent errors

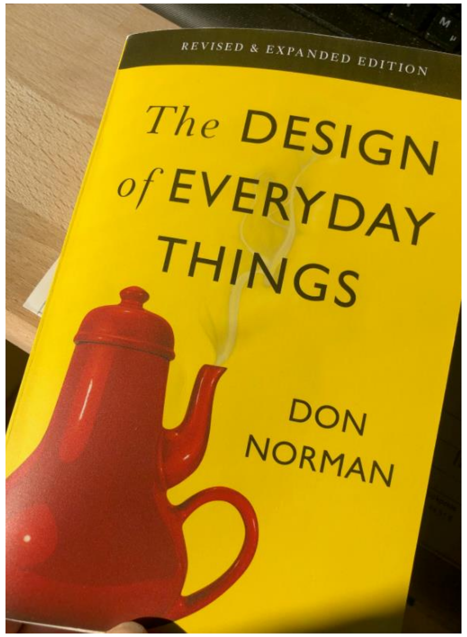
Did you understand this block?

Can you answer these questions?

- When should you not communicate a system error to the user?
- Given a Webform – discuss the statement “All possible errors will be made.”
- Explain the difference between mistakes and slips
- What is a capture error? Give an example.
- What is a data driven error? Give an example.
- Explain physical constraints on the example of a Micro-USB and USB-C connector
- Explain the concept of constraints using the example of a Date-Picker
- Discuss how a user interface designs can be improved to prevent errors

Reference

- Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson. http://www.cs.umd.edu/hcil/DTUI6/



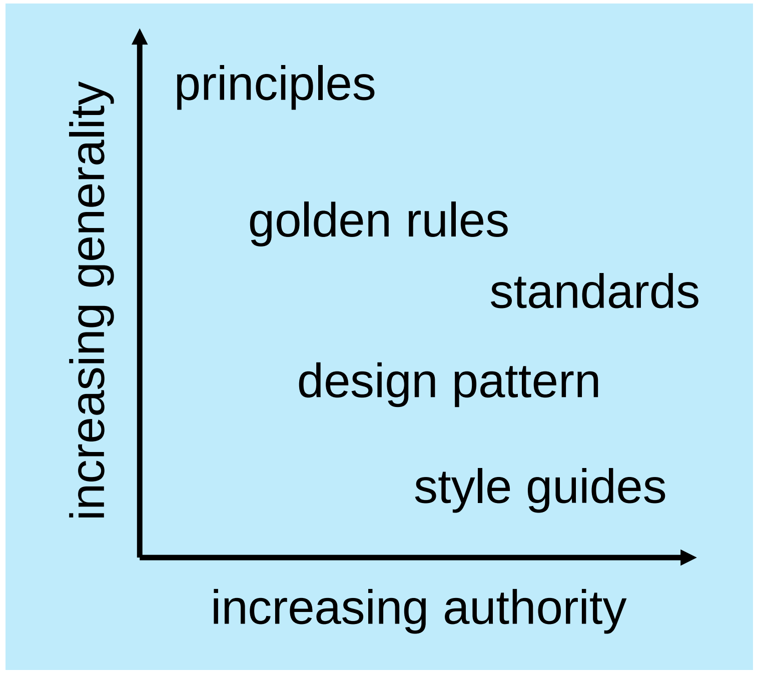
Style Guides and UI Guidelines

Learning Goals

- Know about standards, recommendations, regulations and laws applicable to user interface design
- Understand …
  - The basic structure of a user interface guideline
  - What is described in a user interface guideline
- Be able to …
  - find for specific user interface elements, how the should be designed and behave on a given platform (e.g. where to find, how to design a switch in iOS)

Types of Design Rules

- **Principles**
  - abstract design rules
- **Golden rules and heuristics**
  - more concrete than principles
- **Standards**
  - (very) detailed design rules
- **Design patterns**
  - generic solution for a specific problem
- **Style guides**
  - provided for devices, operating systems, widget libraries



- **Authority**: whether or not a rule must be followed or whether it is just suggested
- **Generality**: applied to many design situations or focused on specific application situation.

(more) Guidelines

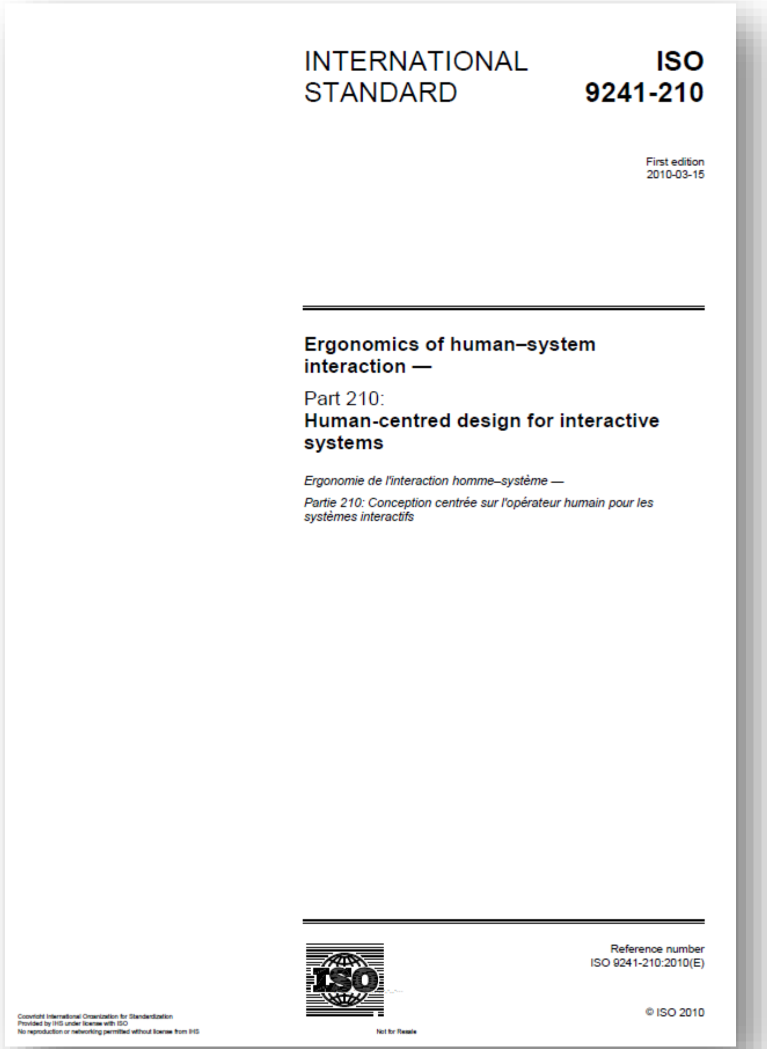
Hix and Hartson, Developing User Interfaces, Wiley, 1993

- User centered design
- Know the user
- Involve the user
- Prevent user errors
- Optimize user operation
- Keep control with the user
- Help the user to get started
- Give a task-based mental model
- Be consistent
- Keep it simple
- Design for memory limitations
- Use recognition rather recall
- Use cognitive directness
- Draw on real world analogies
- Use informative feedback
- Give status indicators
- Use user-centred wording
- Use non-threatening wording
- Use specific constructive advice
- Make the system take the blame
- Do not anthropomorphise
- Use modes cautiously
- Make user action reversible
- Get attention judiciously
- Maintain display inertia
- Organize screen to manage complexity
- Accommodate individual difference

ISO 9241
Ergonomics of human–system interaction

Many parts with specific focus

- Part 1: General introduction
- Part 2: Guidance on task requirements
- Part 3: Visual display requirements
- Part 4: Keyboard requirements
- …
- Part 110: **Dialogue principles**
- Part 151: Guidance on **World Wide Web** user interfaces
- Part 171: Guidance on software **accessibility**
- Part 210: Human-centered design for interactive systems
- …
- Part 420: Selection procedures for physical input devices
- Part 910: Framework for tactile and haptic interaction
- Part 920: Guidance on tactile and haptic interactions



INTERNATIONAL STANDARD

ISO 9241-210

First edition
2010-03-15

Ergonomics of human–system interaction —

Part 210:
Human-centred design for interactive systems

Ergonomie de l'interaction homme–système —

Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs

Reference number
ISO 9241-210:2010(E)

© ISO 2010

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9241 Ergonomics of human–system interaction

ISO 9241
Ergonomics of human–system interaction

Part 210: Human-centered design for interactive systems

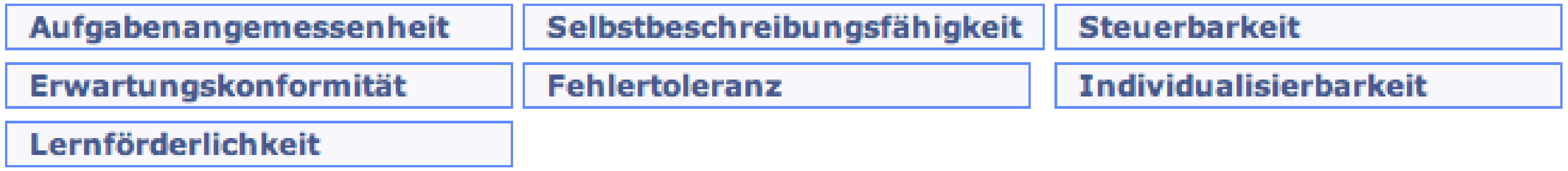
Human Centered Design

ISO 9241-210

ISO 9241 Ergonomics of human–system interaction

ISO 9241
Ergonomics of human–system interaction
Part 110 Dialogue Principles

- Suitability for the task
- Self-descriptiveness
- Controllability
- Conformity with user expectations
- Error tolerance
- Suitability for individualisation
- Suitability for learning



Aufgabenangemessenheit	Selbstbeschreibungsfähigkeit	Steuerbarkeit
Erwartungskonformität	Fehlertoleranz	Individualisierbarkeit
Lernförderlichkeit		

ISO 9241 Ergonomics of human–system interaction

Recommendations, Regulations & Laws

Verordnung über Arbeitsstätten (Arbeitsstättenverordnung - ArbStättV)

1. Allgemeine Anforderungen
2. Maßnahmen zum Schutz vor besonderen Gefahren
3. Arbeitsbedingungen
4. Sanitär-, Pausen- und Bereitschaftsräume, Kantinen, Erste-Hilfe-Räume und Unterkünfte
5. Ergänzende Anforderungen und Maßnahmen für besondere Arbeitsstätten und Arbeitsplätze
6. **Maßnahmen zur Gestaltung von Bildschirmarbeitsplätzen**
    1. Allgemeine Anforderungen an Bildschirmarbeitsplätze
    2. Allgemeine Anforderungen an Bildschirme und Bildschirmgeräte
    3. Anforderungen an Bildschirmgeräte und Arbeitsmittel für die ortsgebundene Verwendung an Arbeitsplätzen
    4. Anforderungen an tragbare Bildschirmgeräte für die ortsveränderliche Verwendung an Arbeitsplätzen
    5. Anforderungen an die Benutzerfreundlichkeit von Bildschirmarbeitsplätzen

https://www.gesetze-im-internet.de/arbst_ttv_2004/anhang.html

Recommendations, Regulations & Laws

Verordnung über Arbeitsstätten (Arbeitsstättenverordnung - ArbStättV)

6.5 Anforderungen an die **Benutzerfreundlichkeit** von Bildschirmarbeitsplätzen

(1) Beim Betreiben der Bildschirmarbeitsplätze hat der Arbeitgeber dafür zu sorgen, dass der Arbeitsplatz den Arbeitsaufgaben **angemessen** gestaltet ist. Er hat insbesondere **geeignete Softwaresysteme** bereitzustellen.

(2) Die Bildschirmgeräte und die Software müssen **entsprechend den Kenntnissen und Erfahrungen der Beschäftigten** im Hinblick auf die jeweilige **Arbeitsaufgabe angepasst** werden können.

(3) Das Softwaresystem muss den Beschäftigten **Angaben über die jeweiligen Dialogabläufe** machen.

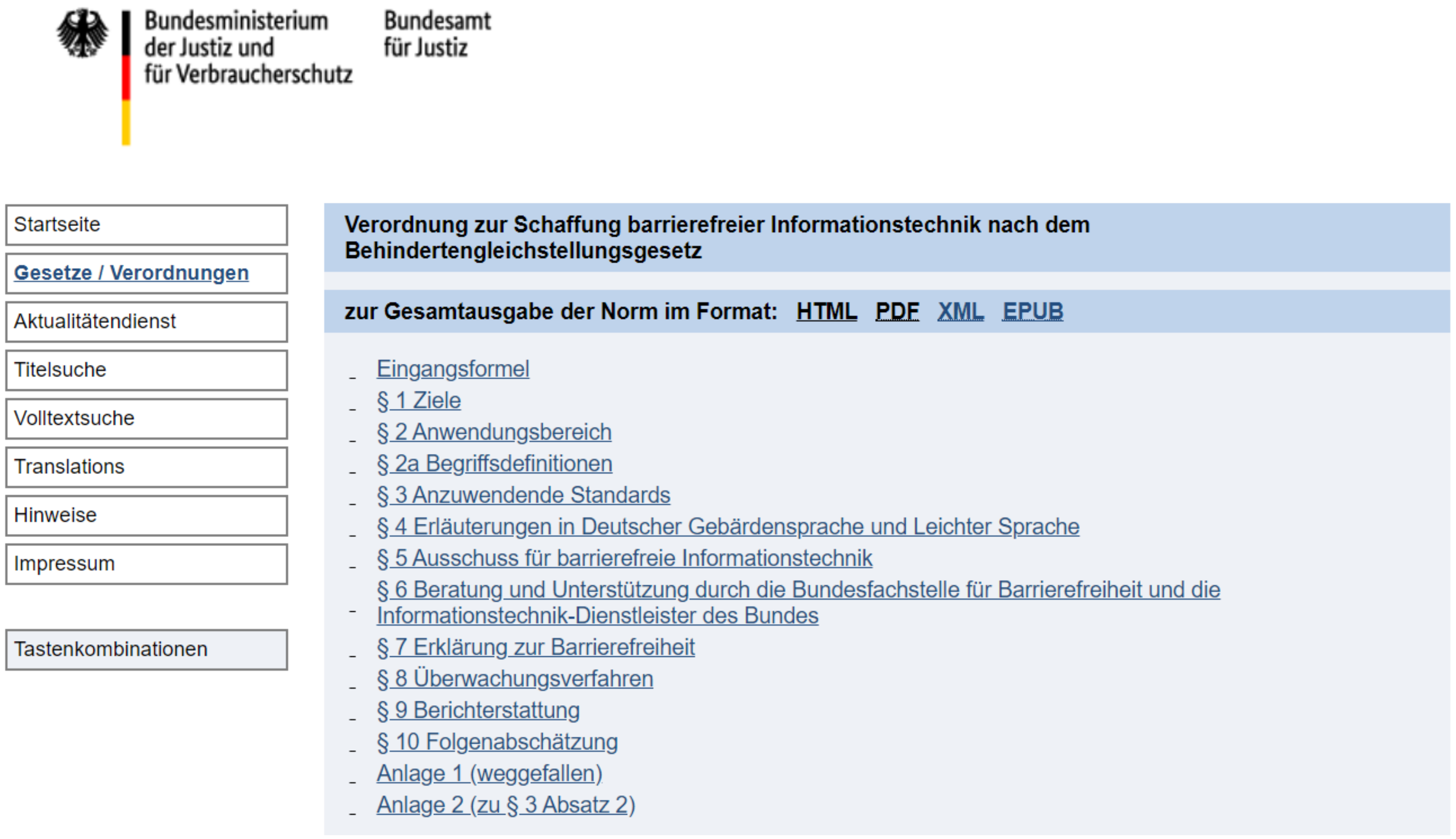
(4) Die Bildschirmgeräte und die Software müssen es den Beschäftigten ermöglichen, die **Dialogabläufe zu beeinflussen**. Sie müssen eventuelle Fehler bei der Handhabung beschreiben und eine **Fehlerbeseitigung** mit begrenztem Arbeitsaufwand erlauben.

(5) Eine Kontrolle der Arbeit hinsichtlich der qualitativen oder quantitativen Ergebnisse darf ohne Wissen der Beschäftigten nicht durchgeführt werden.

https://www.gesetze-im-internet.de/arbst_ttv_2004/anhang.html

Recommendations, Regulations & Laws

Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz



Bundesministerium der Justiz und für Verbraucherschutz

Bundesamt für Justiz

Startseite
Gesetze / Verordnungen
Aktualitätendienst
Titelsuche
Volltextsuche
Translations
Hinweise
Impressum
Tastenkombinationen

Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz

zur Gesamtausgabe der Norm im Format: HTML PDF XML EPUB

https://www.gesetze-im-internet.de/bitv_2_0/index.html

Recommendations, Regulations & Laws

Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz

§ 1 Ziele

- (1) Die Barrierefreie-Informationstechnik-Verordnung dient dem Ziel, eine umfassend und grundsätzlich uneingeschränkt **barrierefreie Gestaltung moderner Informations- und Kommunikationstechnik** zu ermöglichen und zu gewährleisten.
- (2) Informationen und Dienstleistungen öffentlicher Stellen, die elektronisch zur Verfügung gestellt werden, sowie elektronisch unterstützte Verwaltungsabläufe mit und innerhalb der Verwaltung, einschließlich der Verfahren zur elektronischen Aktenführung und zur elektronischen Vorgangsbearbeitung, sind **für Menschen mit Behinderungen zugänglich und nutzbar zu gestalten**

https://www.gesetze-im-internet.de/bitv_2_0/index.html

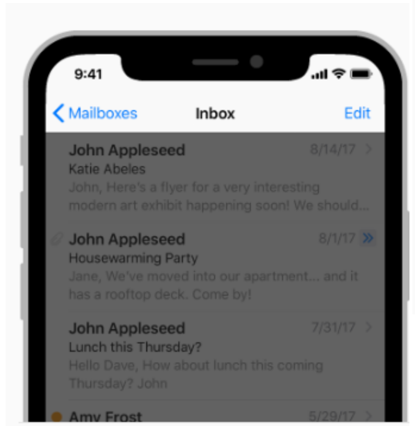
Web Content Accessibility Guidelines (WCAG) 2.1

WCAG 2.1 at a Glance, Summary, www.w3.org/TR/WCAG21

- “Perceivable
  - Provide **text alternatives** for non-text content,
  - Provide captions and other **alternatives for multimedia**.
  - Create content that can be **presented in different ways**, including by assistive technologies, without losing meaning.
  - Make it easier for users to **see and hear** content.
- Operable
  - Make all functionality available from a **keyboard**.
  - Give users **enough time** to read and use content.
  - **Do not use content that causes seizures or physical reactions**.
  - Help users **navigate** and find content.
  - Make it easier to use **inputs other** than keyboard.
- Understandable
  - Make text **readable and understandable**.
  - Make content appear and operate in **predictable ways**.
  - Help users **avoid and correct mistakes**.
- Robust
  - Maximize **compatibility** with current and future user tools.”

User Interface Guidelines

- Define how the user interface
  - Looks like
  - Is operated
  - Reacts
  - Feels
- Usually most is “encoded” in the libraries, frameworks and UI builder
- Need to learn, how to find it and how to read it



9:41
Mailboxes | Inbox | Edit

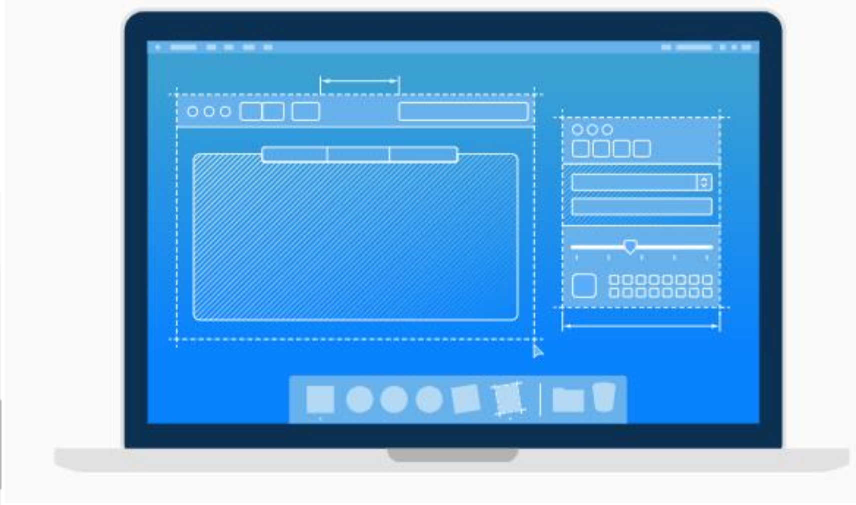
John Appleseed 8/14/17
Katie Abeles
John, Here's a flyer for a very interesting modern art exhibit happening soon! We should...

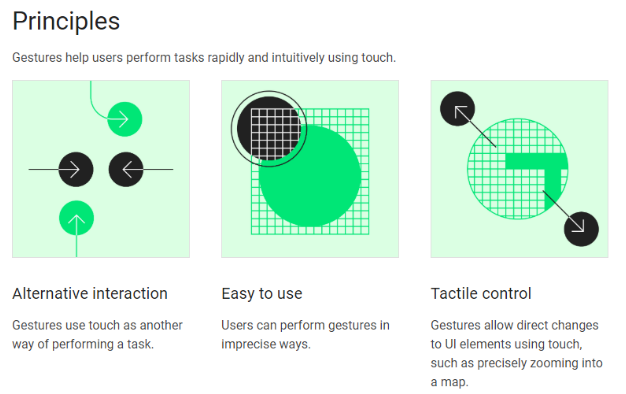
John Appleseed 8/1/17
Housewarming Party
Jane, We've moved into our apartment... and it has a rooftop deck. Come by!

John Appleseed 7/31/17
Lunch this Thursday?
Hello Dave, How about lunch this coming Thursday? John

Amy Frost 5/29/17

Standard title





Principles

Gestures help users perform tasks rapidly and intuitively using touch.

Alternative interaction

Gestures use touch as another way of performing a task.

Easy to use

Users can perform gestures in imprecise ways.

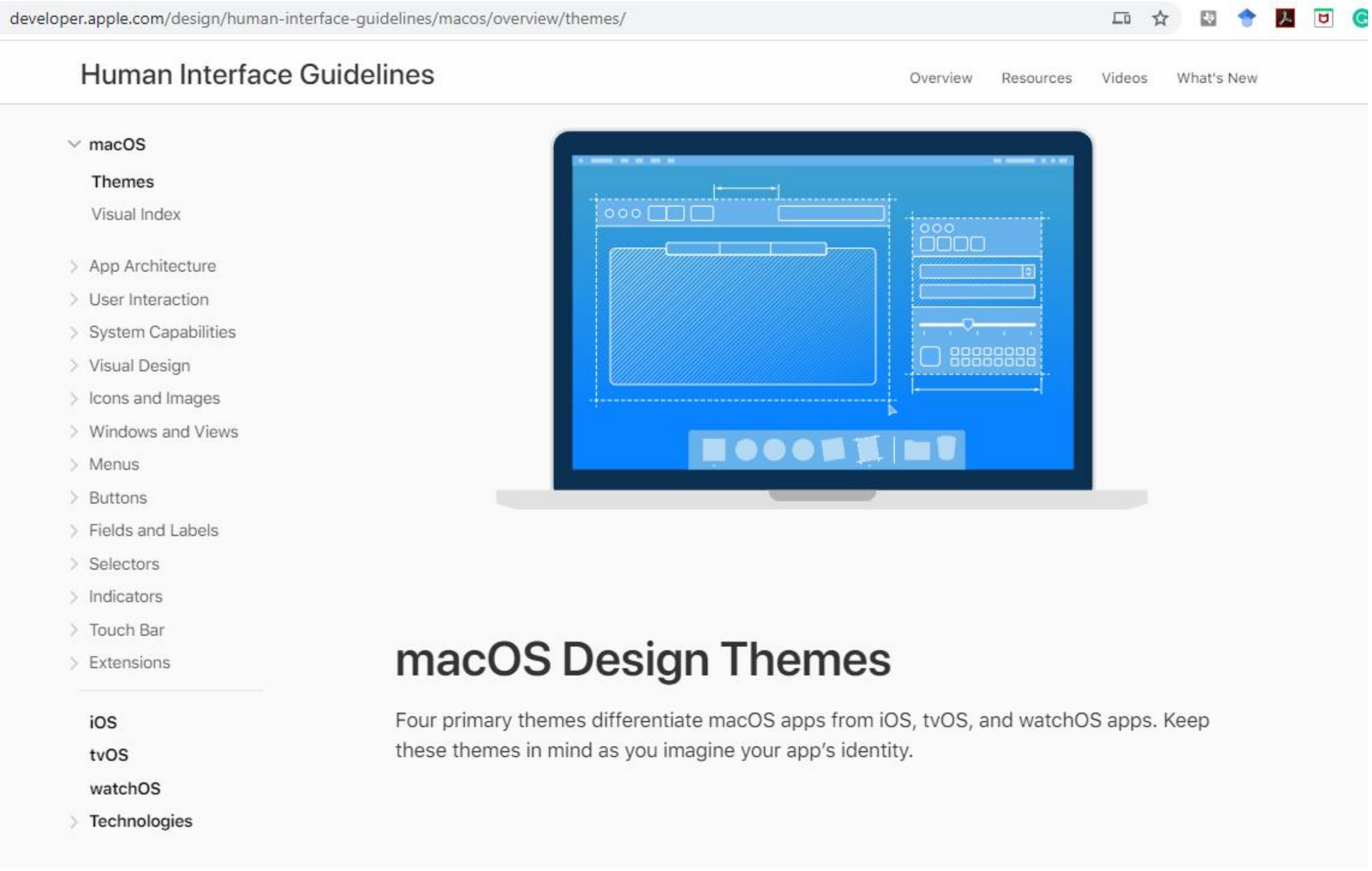
Tactile control

Gestures allow direct changes to UI elements using touch, such as precisely zooming into a map.

Calendar view

September 2014

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4



developer.apple.com/design/human-interface-guidelines/macos/overview/themes/

Human Interface Guidelines

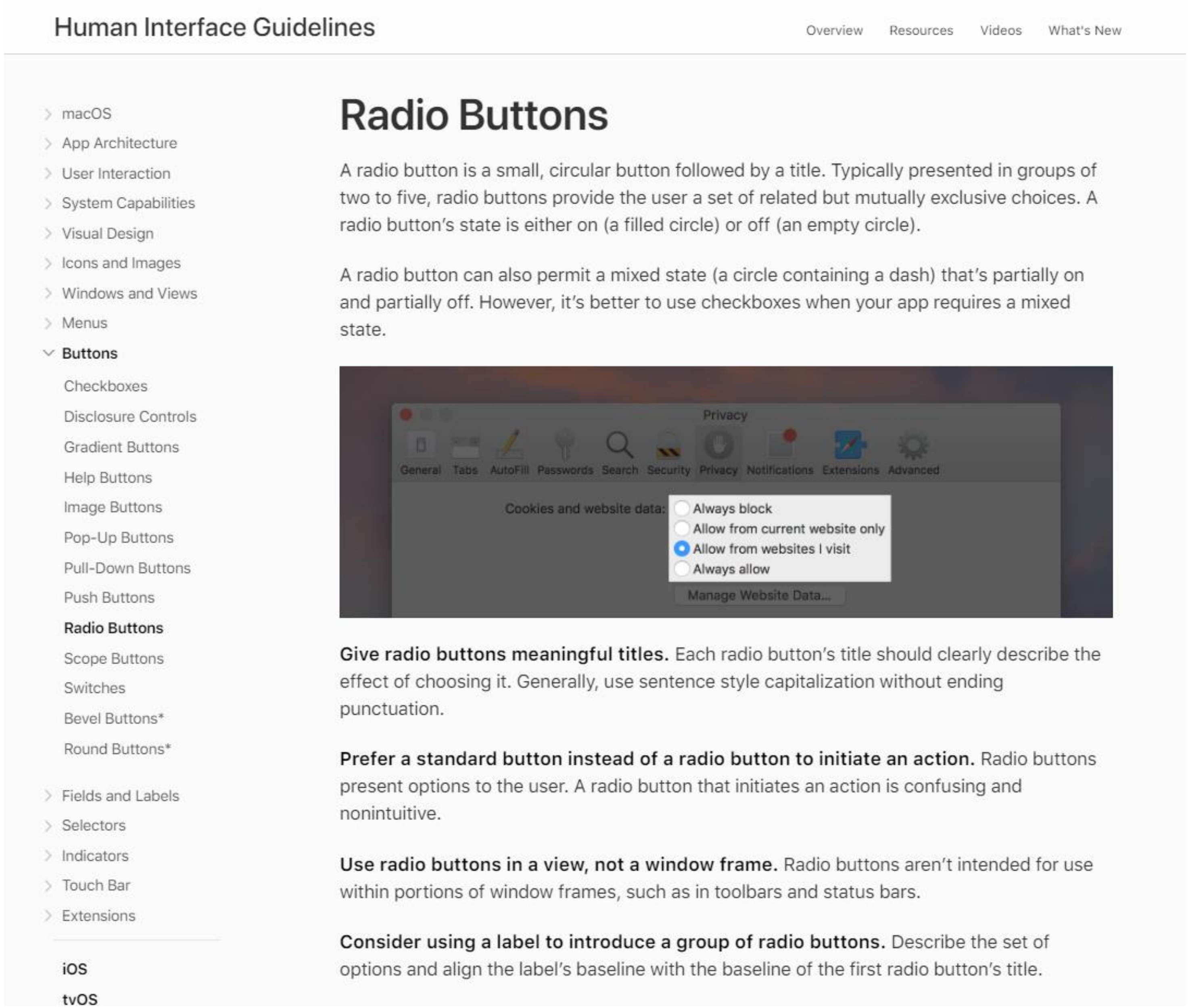
Overview Resources Videos What's New

- macOS
  - Themes
  - Visual Index
- App Architecture
- User Interaction
- System Capabilities
- Visual Design
- Icons and Images
- Windows and Views
- Menus
- Buttons
- Fields and Labels
- Selectors
- Indicators
- Touch Bar
- Extensions
- iOS
- tvOS
- watchOS
- Technologies

macOS Design Themes

Four primary themes differentiate macOS apps from iOS, tvOS, and watchOS apps. Keep these themes in mind as you imagine your app's identity.

https://developer.apple.com/design/human-interface-guidelines/macos/



Human Interface Guidelines

Overview Resources Videos What's New

- macOS
- App Architecture
- User Interaction
- System Capabilities
- Visual Design
- Icons and Images
- Windows and Views
- Menus
- Buttons
  - Checkboxes
  - Disclosure Controls
  - Gradient Buttons
  - Help Buttons
  - Image Buttons
  - Pop-Up Buttons
  - Pull-Down Buttons
  - Push Buttons
  - **Radio Buttons**
  - Scope Buttons
  - Switches
  - Bevel Buttons*
  - Round Buttons*
- Fields and Labels
- Selectors
- Indicators
- Touch Bar
- Extensions

iOS

tvOS

Radio Buttons

A radio button is a small, circular button followed by a title. Typically presented in groups of two to five, radio buttons provide the user a set of related but mutually exclusive choices. A radio button's state is either on (a filled circle) or off (an empty circle).

A radio button can also permit a mixed state (a circle containing a dash) that's partially on and partially off. However, it's better to use checkboxes when your app requires a mixed state.

Cookies and website data:
- ○ Always block
- ○ Allow from current website only
- ● Allow from websites I visit
- ○ Always allow

Manage Website Data...

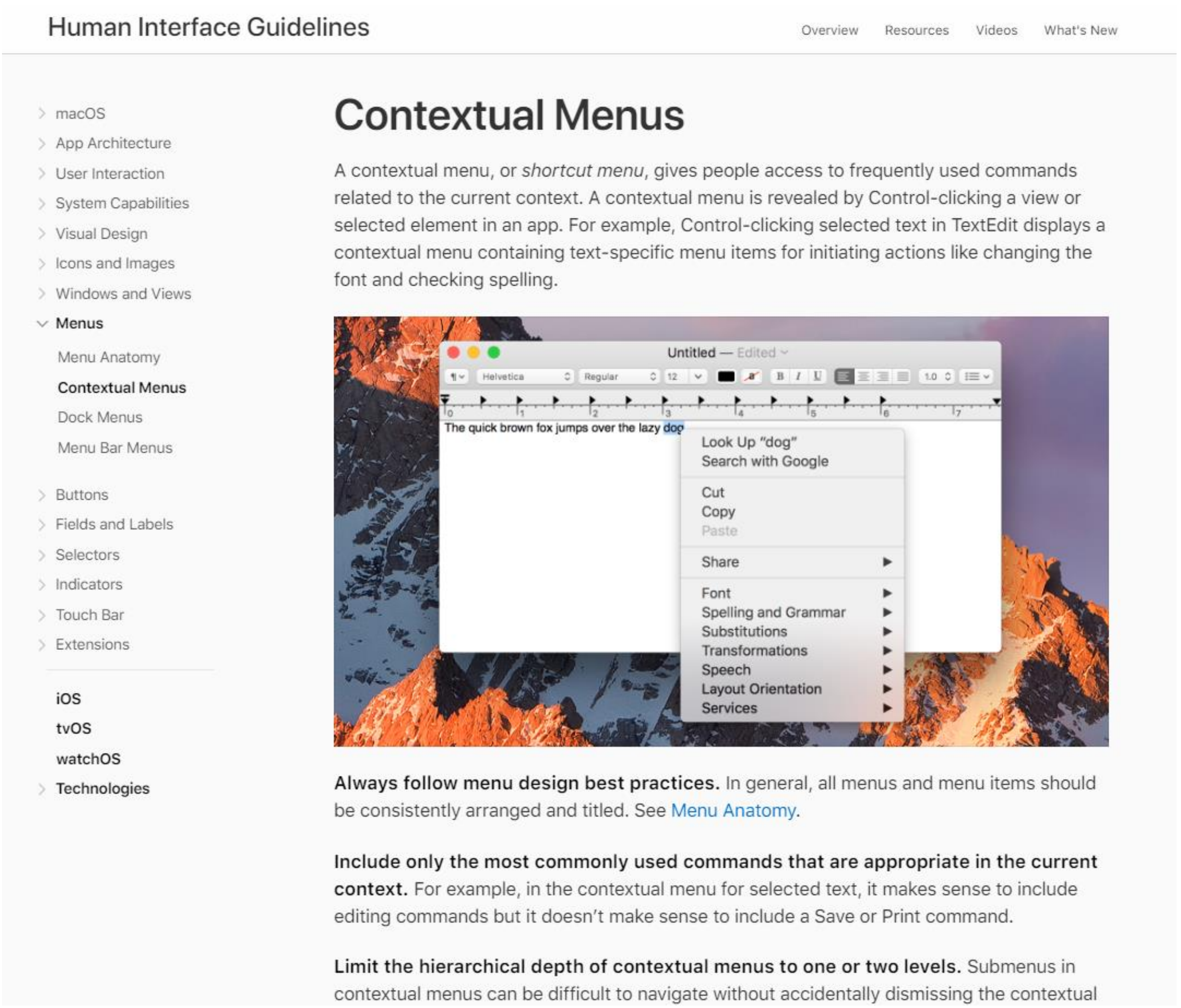
Give radio buttons meaningful titles. Each radio button's title should clearly describe the effect of choosing it. Generally, use sentence style capitalization without ending punctuation.

Prefer a standard button instead of a radio button to initiate an action. Radio buttons present options to the user. A radio button that initiates an action is confusing and nonintuitive.

Use radio buttons in a view, not a window frame. Radio buttons aren't intended for use within portions of window frames, such as in toolbars and status bars.

Consider using a label to introduce a group of radio buttons. Describe the set of options and align the label's baseline with the baseline of the first radio button's title.

https://developer.apple.com/design/human-interface-guidelines/macos/



Human Interface Guidelines

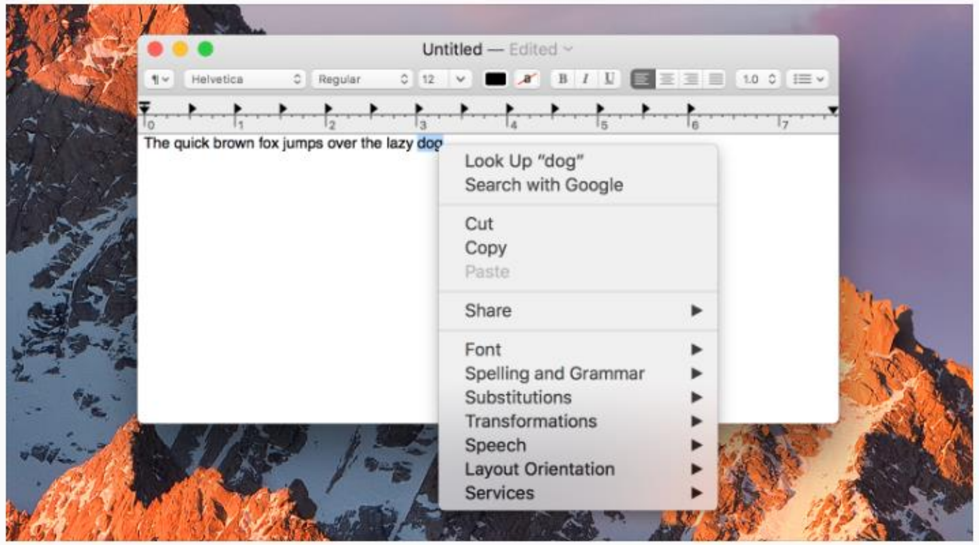
Overview Resources Videos What's New

- macOS
- App Architecture
- User Interaction
- System Capabilities
- Visual Design
- Icons and Images
- Windows and Views
- Menus
  - Menu Anatomy
  - Contextual Menus
  - Dock Menus
  - Menu Bar Menus
- Buttons
- Fields and Labels
- Selectors
- Indicators
- Touch Bar
- Extensions

- iOS
- tvOS
- watchOS
- Technologies

Contextual Menus

A contextual menu, or *shortcut menu*, gives people access to frequently used commands related to the current context. A contextual menu is revealed by Control-clicking a view or selected element in an app. For example, Control-clicking selected text in TextEdit displays a contextual menu containing text-specific menu items for initiating actions like changing the font and checking spelling.

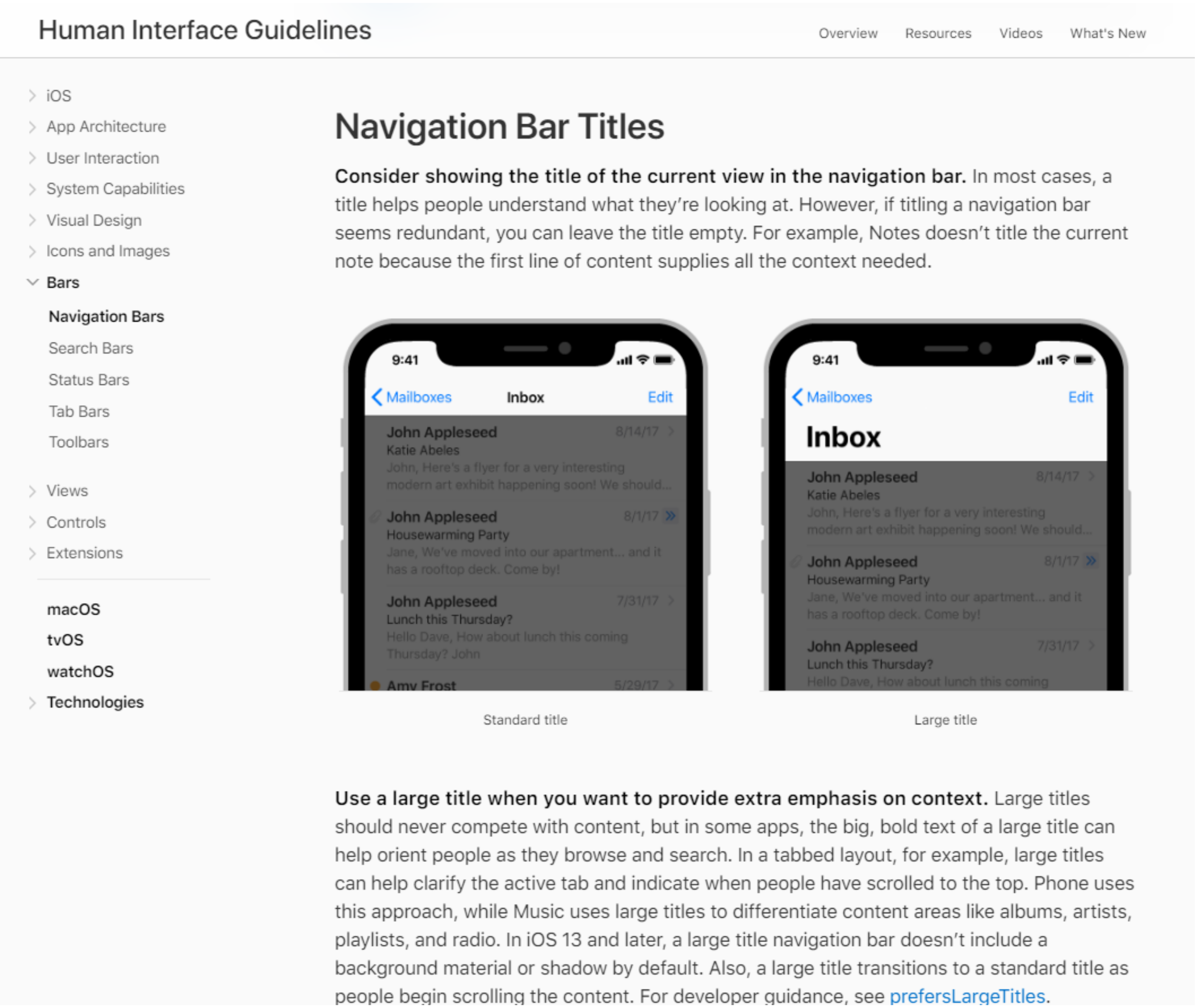


Always follow menu design best practices. In general, all menus and menu items should be consistently arranged and titled. See Menu Anatomy.

Include only the most commonly used commands that are appropriate in the current context. For example, in the contextual menu for selected text, it makes sense to include editing commands but it doesn't make sense to include a Save or Print command.

Limit the hierarchical depth of contextual menus to one or two levels. Submenus in contextual menus can be difficult to navigate without accidentally dismissing the contextual

https://developer.apple.com/design/human-interface-guidelines/macos/



Human Interface Guidelines

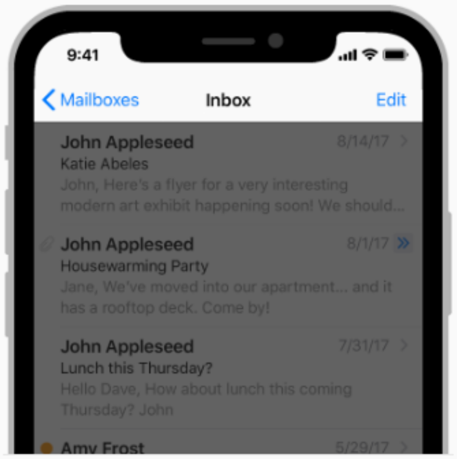
Overview Resources Videos What's New

- iOS
- App Architecture
- User Interaction
- System Capabilities
- Visual Design
- Icons and Images
- Bars
  - Navigation Bars
  - Search Bars
  - Status Bars
  - Tab Bars
  - Toolbars
- Views
- Controls
- Extensions

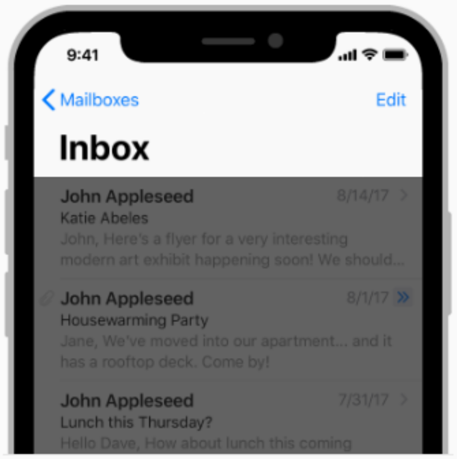
- macOS
- tvOS
- watchOS
- Technologies

Navigation Bar Titles

Consider showing the title of the current view in the navigation bar. In most cases, a title helps people understand what they're looking at. However, if titling a navigation bar seems redundant, you can leave the title empty. For example, Notes doesn't title the current note because the first line of content supplies all the context needed.


9:41
Mailboxes
Inbox
Edit
John Appleseed 8/14/17
Katie Abeles
John, Here's a flyer for a very interesting modern art exhibit happening soon! We should...
John Appleseed 8/1/17
Housewarming Party
Jane, We've moved into our apartment... and it has a rooftop deck. Come by!
John Appleseed 7/31/17
Lunch this Thursday?
Hello Dave, How about lunch this coming Thursday? John
Amy Frost 5/29/17

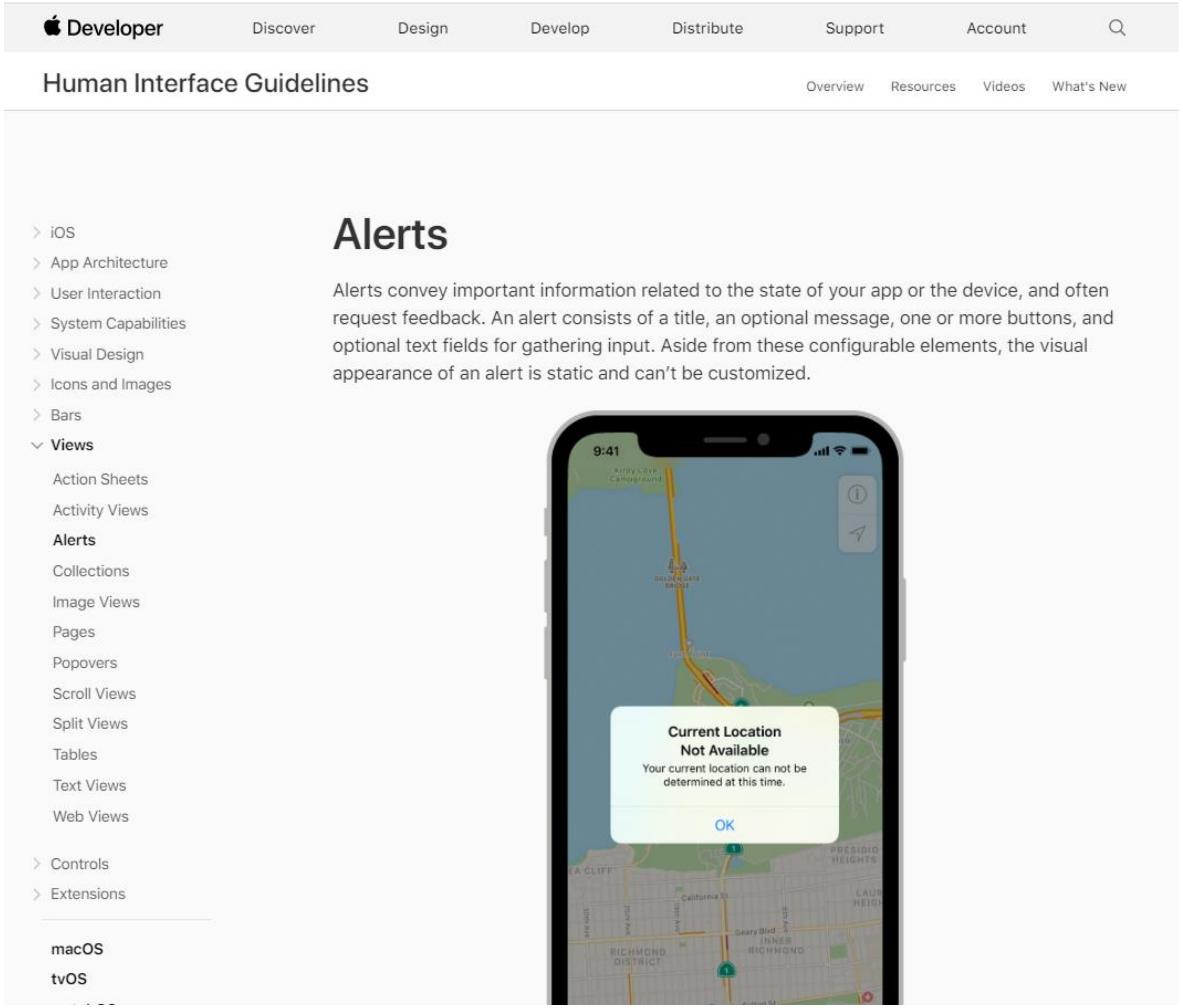
Standard title


9:41
Mailboxes
Edit
Inbox
John Appleseed 8/14/17
Katie Abeles
John, Here's a flyer for a very interesting modern art exhibit happening soon! We should...
John Appleseed 8/1/17
Housewarming Party
Jane, We've moved into our apartment... and it has a rooftop deck. Come by!
John Appleseed 7/31/17
Lunch this Thursday?
Hello Dave, How about lunch this coming

Large title

Use a large title when you want to provide extra emphasis on context. Large titles should never compete with content, but in some apps, the big, bold text of a large title can help orient people as they browse and search. In a tabbed layout, for example, large titles can help clarify the active tab and indicate when people have scrolled to the top. Phone uses this approach, while Music uses large titles to differentiate content areas like albums, artists, playlists, and radio. In iOS 13 and later, a large title navigation bar doesn't include a background material or shadow by default. Also, a large title transitions to a standard title as people begin scrolling the content. For developer guidance, see prefersLargeTitles.

https://developer.apple.com/design/human-interface-guidelines/ios/views/


 Developer Discover Design Develop Distribute Support Account

Human Interface Guidelines Overview Resources Videos What's New

- iOS
- App Architecture
- User Interaction
- System Capabilities
- Visual Design
- Icons and Images
- Bars
- Views
  - Action Sheets
  - Activity Views
  - Alerts
  - Collections
  - Image Views
  - Pages
  - Popovers
  - Scroll Views
  - Split Views
  - Tables
  - Text Views
  - Web Views
- Controls
- Extensions

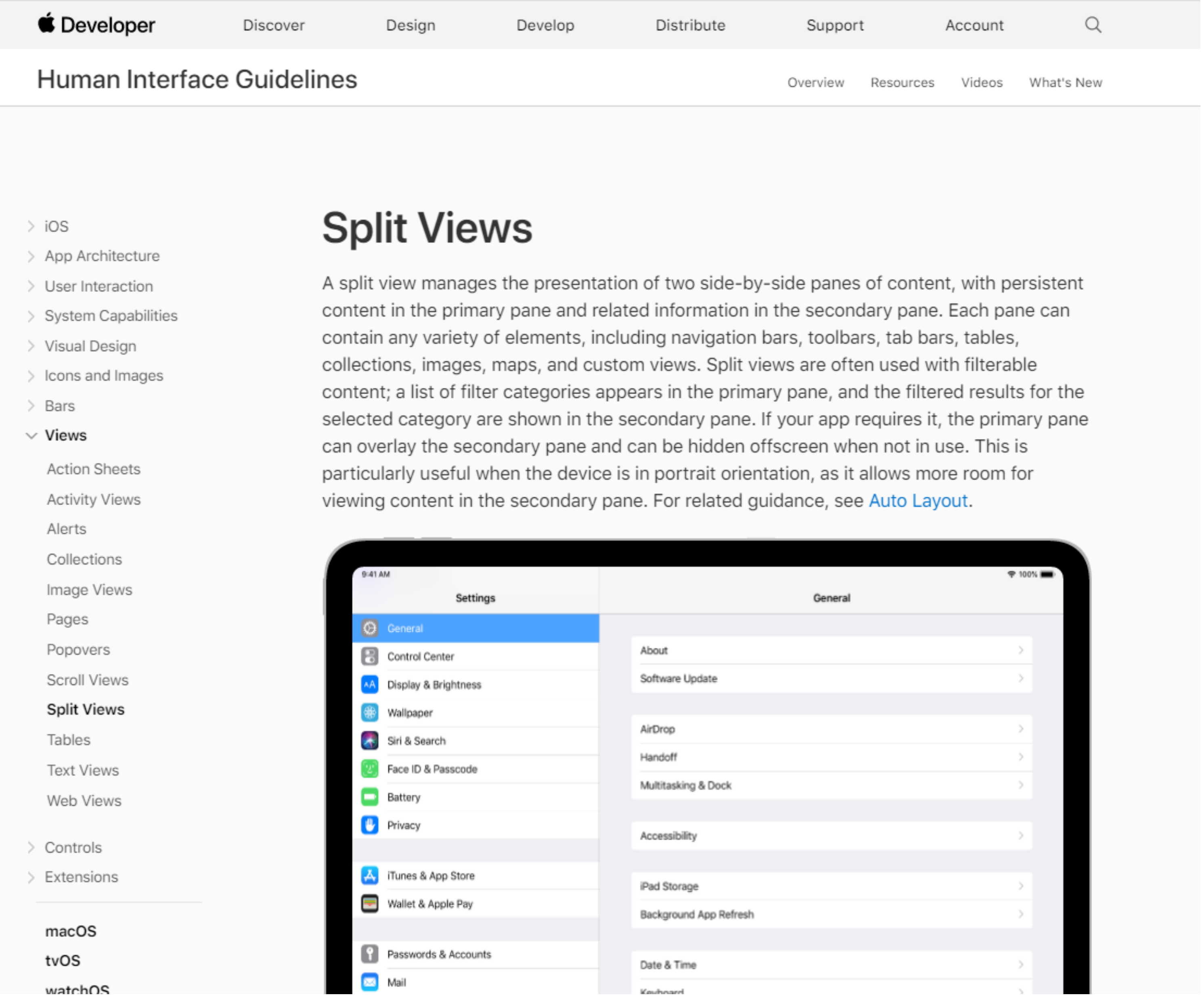
macOS
tvOS

Alerts

Alerts convey important information related to the state of your app or the device, and often request feedback. An alert consists of a title, an optional message, one or more buttons, and optional text fields for gathering input. Aside from these configurable elements, the visual appearance of an alert is static and can't be customized.

9:41
Current Location Not Available
Your current location can not be determined at this time.
OK

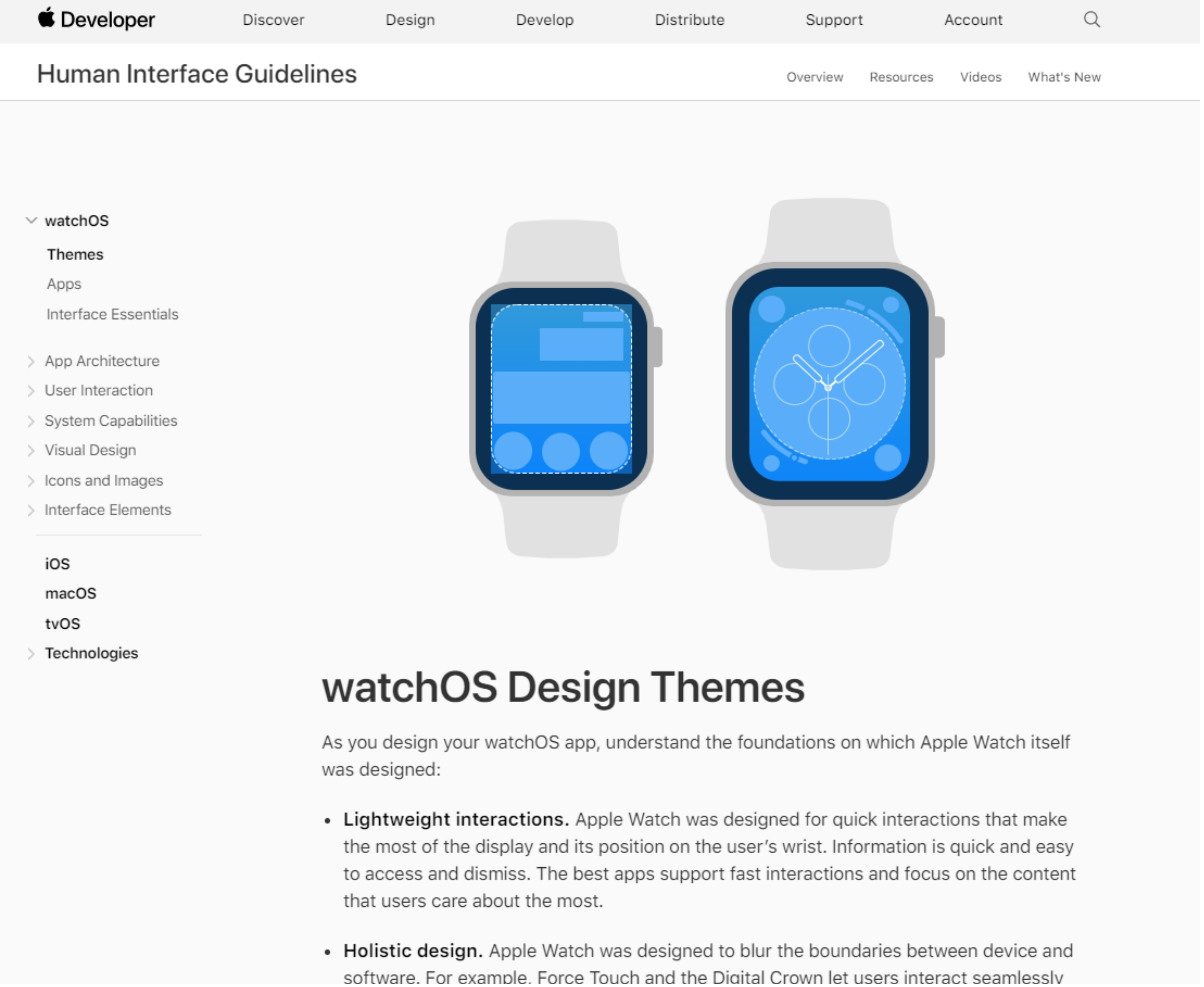
https://developer.apple.com/design/human-interface-guidelines/ios/views/



Split Views

A split view manages the presentation of two side-by-side panes of content, with persistent content in the primary pane and related information in the secondary pane. Each pane can contain any variety of elements, including navigation bars, toolbars, tab bars, tables, collections, images, maps, and custom views. Split views are often used with filterable content; a list of filter categories appears in the primary pane, and the filtered results for the selected category are shown in the secondary pane. If your app requires it, the primary pane can overlay the secondary pane and can be hidden offscreen when not in use. This is particularly useful when the device is in portrait orientation, as it allows more room for viewing content in the secondary pane. For related guidance, see Auto Layout.

https://developer.apple.com/design/human-interface-guidelines/ios/views/

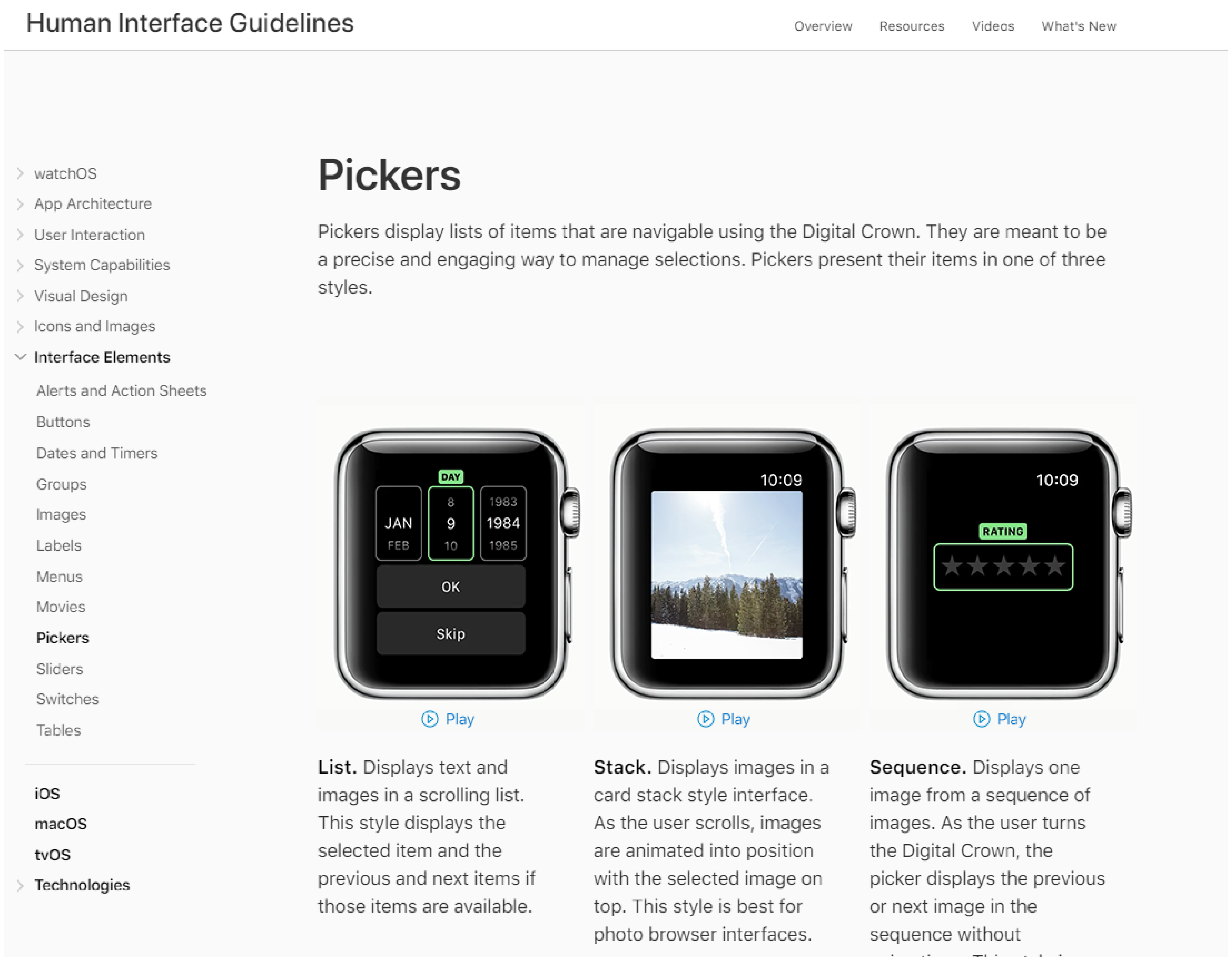


watchOS Design Themes

As you design your watchOS app, understand the foundations on which Apple Watch itself was designed:

- **Lightweight interactions.** Apple Watch was designed for quick interactions that make the most of the display and its position on the user's wrist. Information is quick and easy to access and dismiss. The best apps support fast interactions and focus on the content that users care about the most.
- **Holistic design.** Apple Watch was designed to blur the boundaries between device and software. For example, Force Touch and the Digital Crown let users interact seamlessly

https://developer.apple.com/design/human-interface-guidelines/watchos/overview/themes/



- watchOS
- App Architecture
- User Interaction
- System Capabilities
- Visual Design
- Icons and Images
- Interface Elements
  - Alerts and Action Sheets
  - Buttons
  - Dates and Timers
  - Groups
  - Images
  - Labels
  - Menus
  - Movies
  - Pickers
  - Sliders
  - Switches
  - Tables
- iOS
- macOS
- tvOS
- Technologies

Pickers

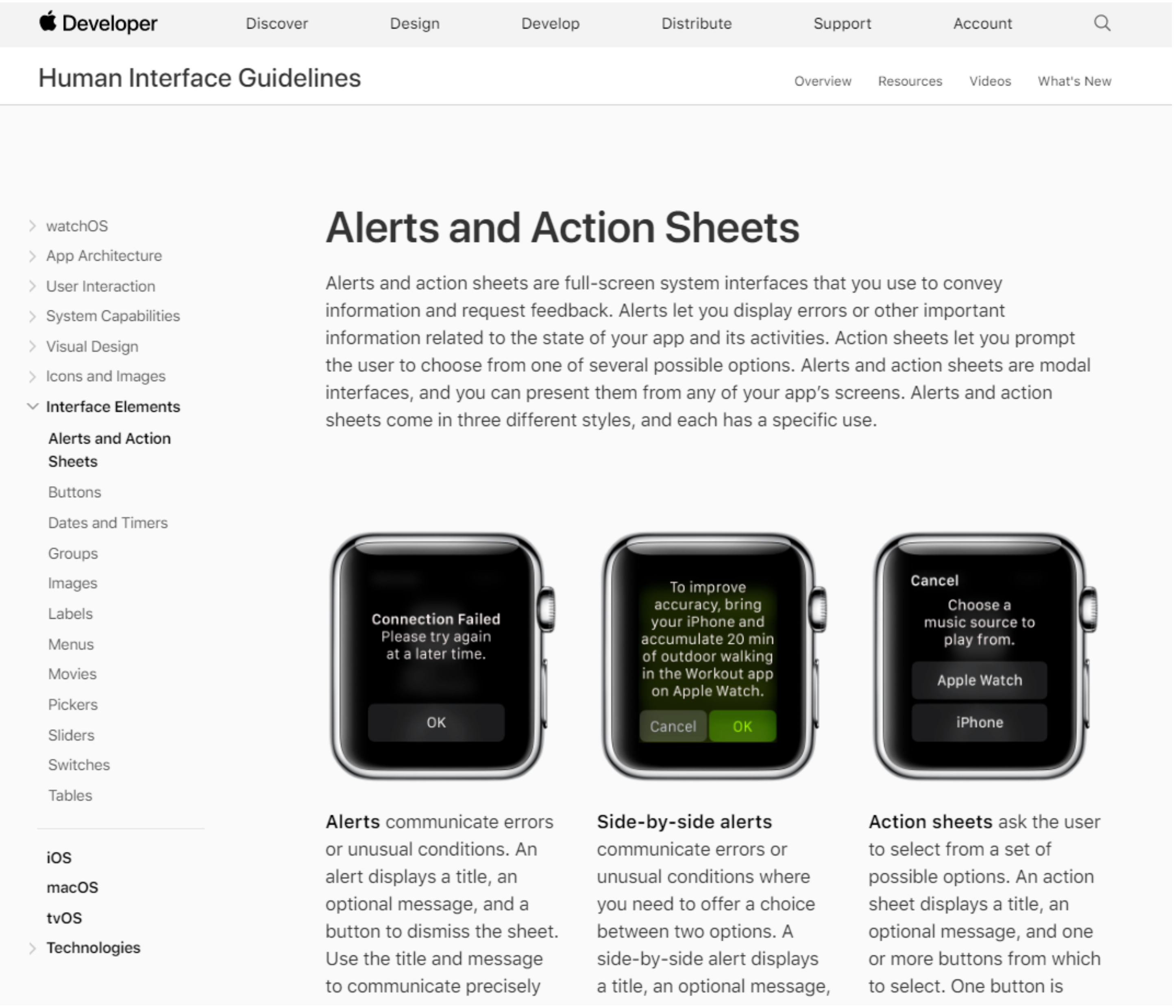
Pickers display lists of items that are navigable using the Digital Crown. They are meant to be a precise and engaging way to manage selections. Pickers present their items in one of three styles.

List. Displays text and images in a scrolling list. This style displays the selected item and the previous and next items if those items are available.

Stack. Displays images in a card stack style interface. As the user scrolls, images are animated into position with the selected image on top. This style is best for photo browser interfaces.

Sequence. Displays one image from a sequence of images. As the user turns the Digital Crown, the picker displays the previous or next image in the sequence without

https://developer.apple.com/design/human-interface-guidelines/watchos/overview/themes/


Developer Discover Design Develop Distribute Support Account

Human Interface Guidelines

Overview Resources Videos What's New

- watchOS
- App Architecture
- User Interaction
- System Capabilities
- Visual Design
- Icons and Images
- Interface Elements
  - Alerts and Action Sheets
  - Buttons
  - Dates and Timers
  - Groups
  - Images
  - Labels
  - Menus
  - Movies
  - Pickers
  - Sliders
  - Switches
  - Tables
- iOS
- macOS
- tvOS
- Technologies

Alerts and Action Sheets

Alerts and action sheets are full-screen system interfaces that you use to convey information and request feedback. Alerts let you display errors or other important information related to the state of your app and its activities. Action sheets let you prompt the user to choose from one of several possible options. Alerts and action sheets are modal interfaces, and you can present them from any of your app's screens. Alerts and action sheets come in three different styles, and each has a specific use.

Connection Failed
Please try again at a later time.
OK

To improve accuracy, bring your iPhone and accumulate 20 min of outdoor walking in the Workout app on Apple Watch.
Cancel OK

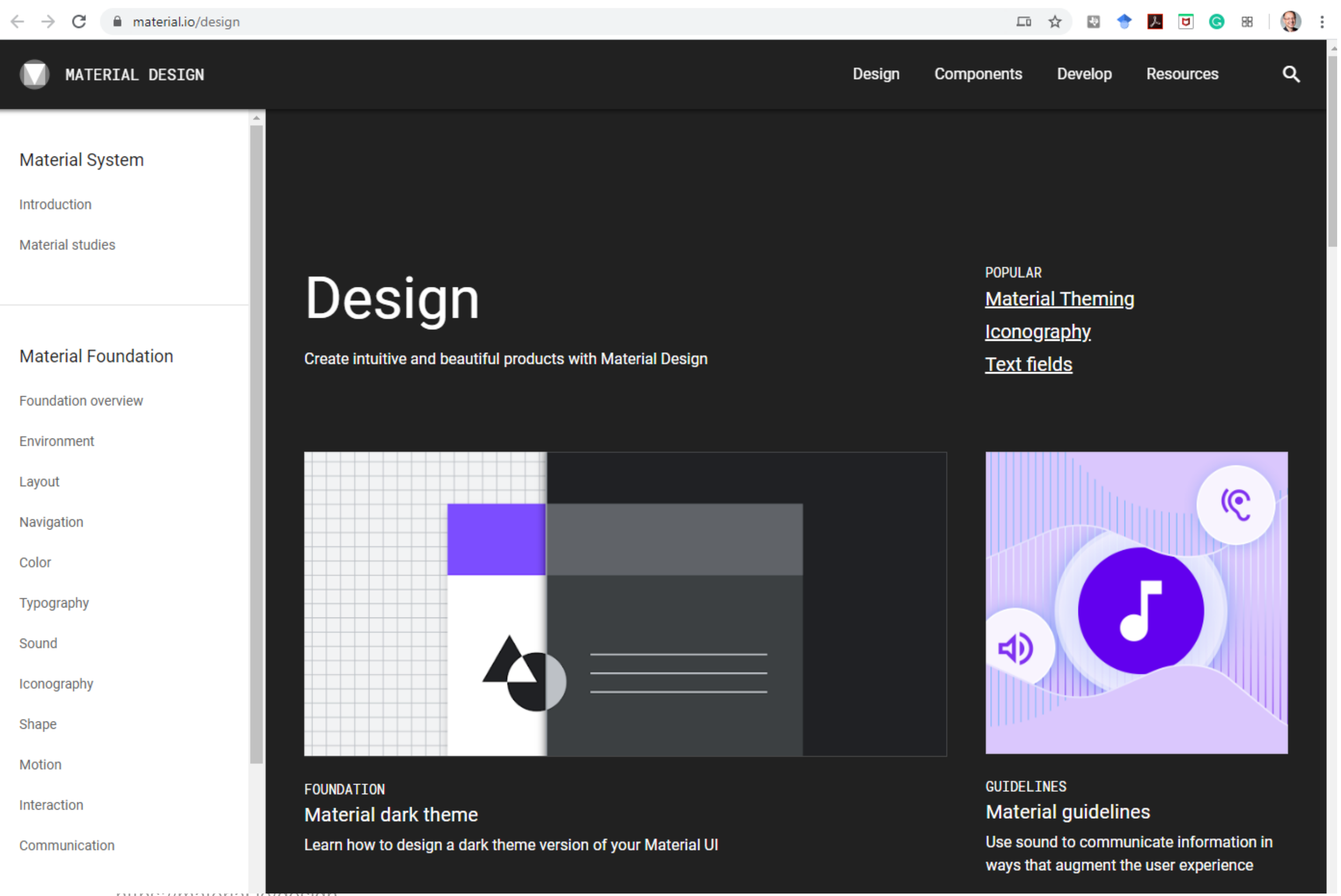
Cancel
Choose a music source to play from.
Apple Watch
iPhone

Alerts communicate errors or unusual conditions. An alert displays a title, an optional message, and a button to dismiss the sheet. Use the title and message to communicate precisely

Side-by-side alerts communicate errors or unusual conditions where you need to offer a choice between two options. A side-by-side alert displays a title, an optional message,

Action sheets ask the user to select from a set of possible options. An action sheet displays a title, an optional message, and one or more buttons from which to select. One button is

https://developer.apple.com/design/human-interface-guidelines/watchos/overview/themes/


material.io/design

MATERIAL DESIGN

Design Components Develop Resources

Material System

Introduction

Material studies

Material Foundation

Foundation overview

Environment

Layout

Navigation

Color

Typography

Sound

Iconography

Shape

Motion

Interaction

Communication

Design

Create intuitive and beautiful products with Material Design

POPULAR

Material Theming

Iconography

Text fields

FOUNDATION

Material dark theme

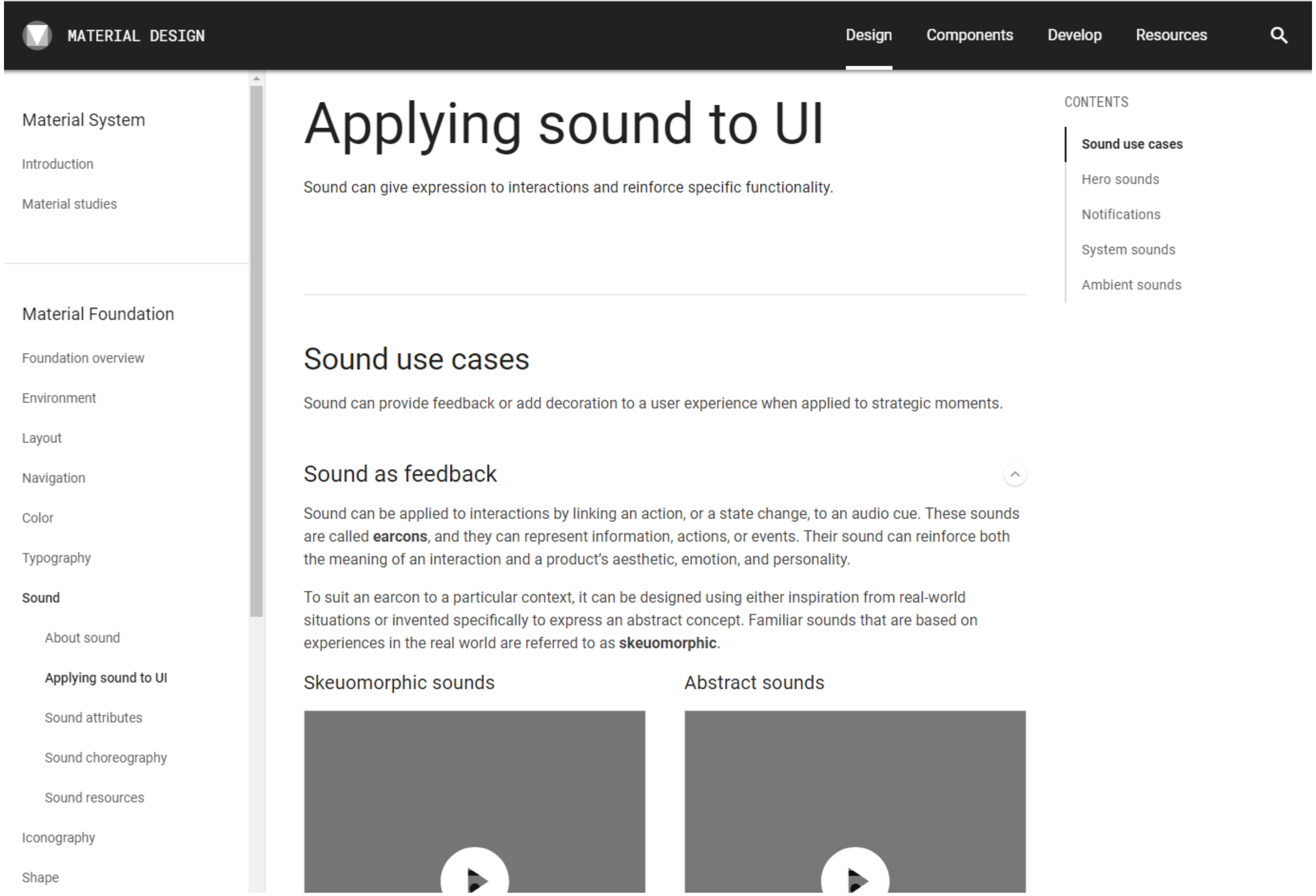
Learn how to design a dark theme version of your Material UI

GUIDELINES

Material guidelines

Use sound to communicate information in ways that augment the user experience

https://material.io/design



Material System

Introduction

Material studies

Material Foundation

Foundation overview

Environment

Layout

Navigation

Color

Typography

Sound

About sound

Applying sound to UI

Sound attributes

Sound choreography

Sound resources

Iconography

Shape

Applying sound to UI

Sound can give expression to interactions and reinforce specific functionality.

Sound use cases

Sound can provide feedback or add decoration to a user experience when applied to strategic moments.

Sound as feedback

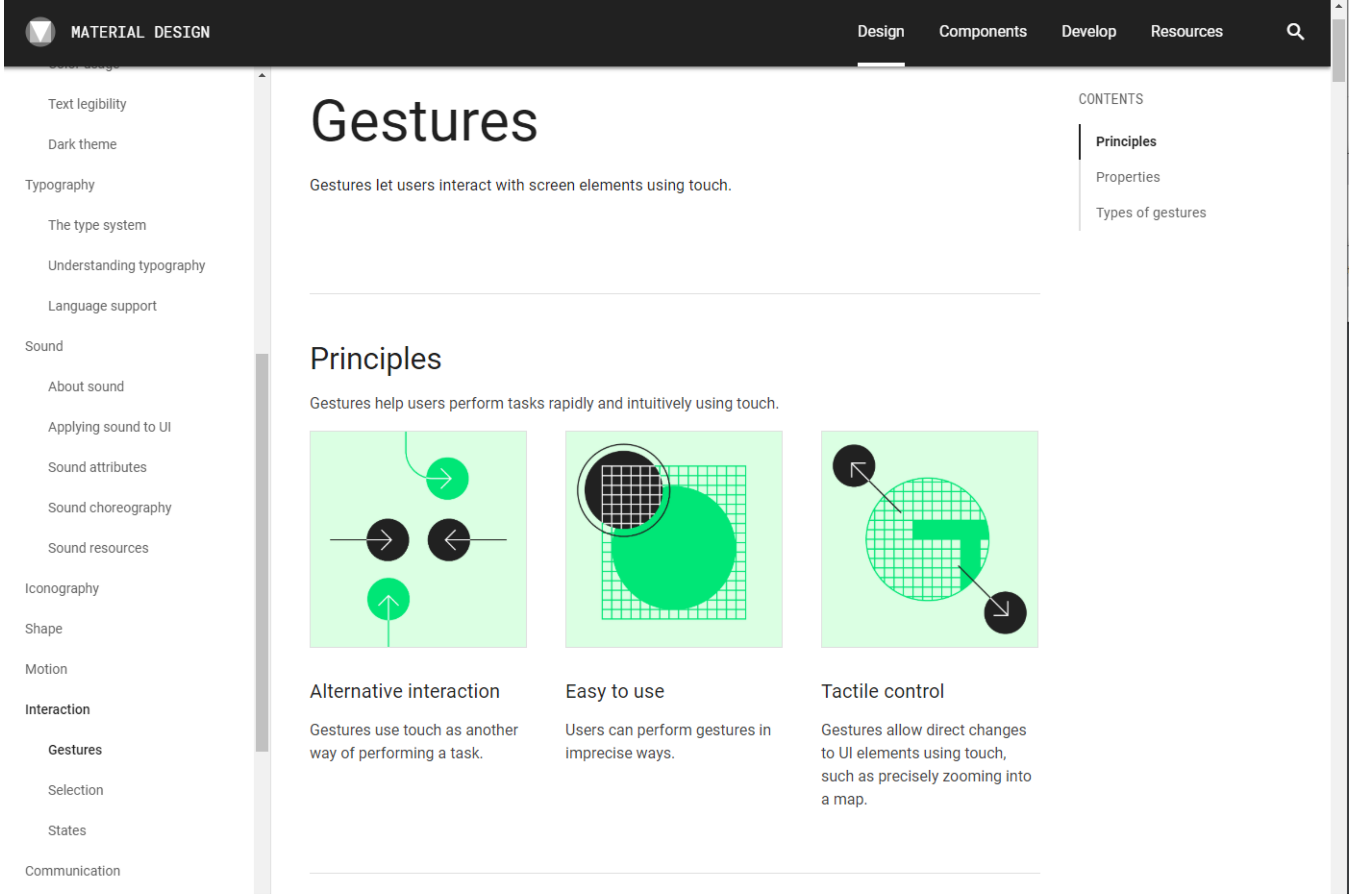
Sound can be applied to interactions by linking an action, or a state change, to an audio cue. These sounds are called **earcons**, and they can represent information, actions, or events. Their sound can reinforce both the meaning of an interaction and a product's aesthetic, emotion, and personality.

To suit an earcon to a particular context, it can be designed using either inspiration from real-world situations or invented specifically to express an abstract concept. Familiar sounds that are based on experiences in the real world are referred to as **skeuomorphic**.

Skeuomorphic sounds

Abstract sounds

https://material.io/design



Gestures

Gestures let users interact with screen elements using touch.

Principles

Gestures help users perform tasks rapidly and intuitively using touch.

Alternative interaction

Gestures use touch as another way of performing a task.

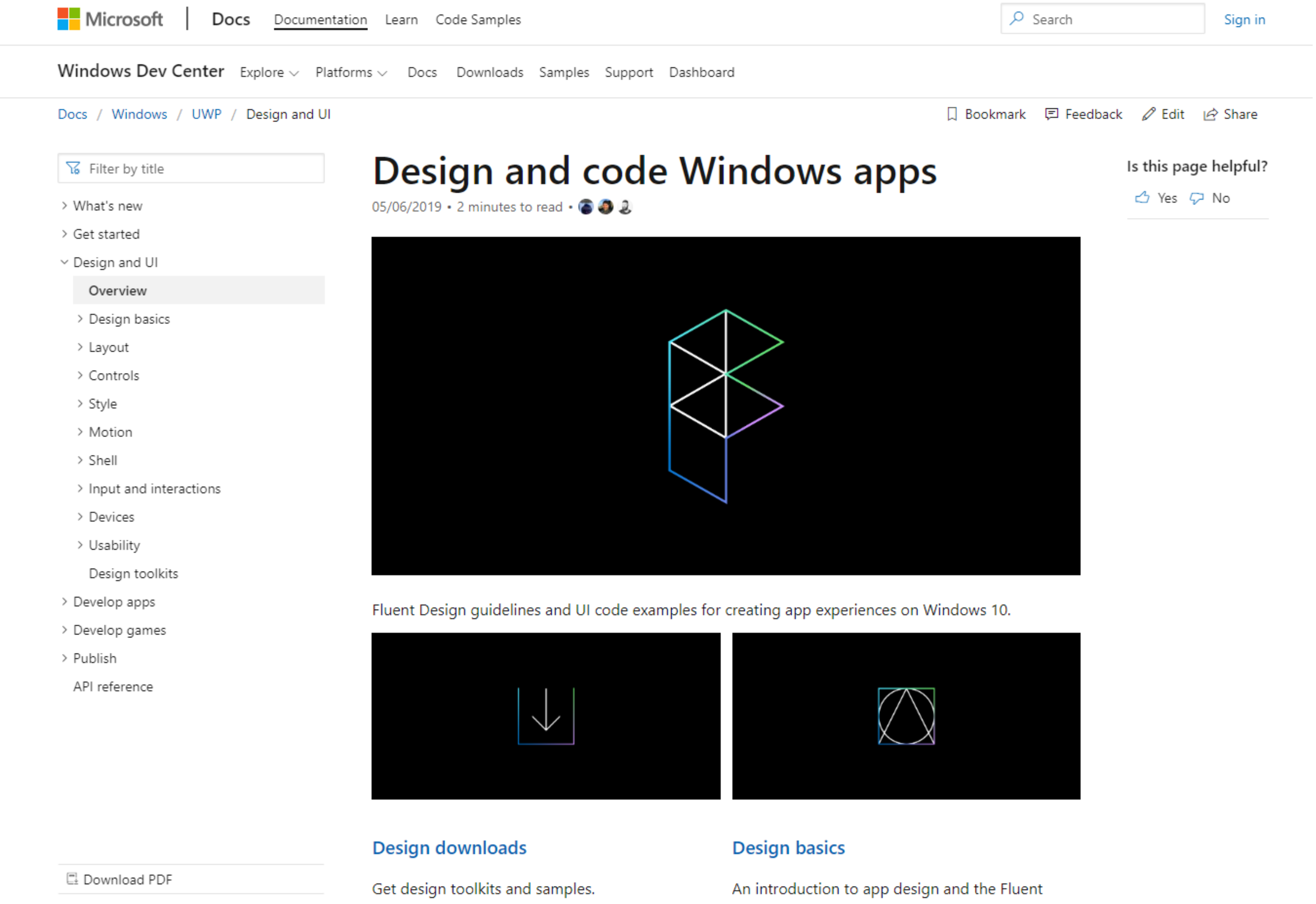
Easy to use

Users can perform gestures in imprecise ways.

Tactile control

Gestures allow direct changes to UI elements using touch, such as precisely zooming into a map.

https://material.io/design


Microsoft | Docs Documentation Learn Code Samples Search Sign in

Windows Dev Center Explore Platforms Docs Downloads Samples Support Dashboard

Docs / Windows / UWP / Design and UI Bookmark Feedback Edit Share

Filter by title

- What's new
- Get started
- Design and UI
  - Overview
  - Design basics
  - Layout
  - Controls
  - Style
  - Motion
  - Shell
  - Input and interactions
  - Devices
  - Usability
  - Design toolkits
- Develop apps
- Develop games
- Publish
- API reference

Download PDF

Design and code Windows apps

05/06/2019 • 2 minutes to read •

Is this page helpful?

Yes No

Fluent Design guidelines and UI code examples for creating app experiences on Windows 10.

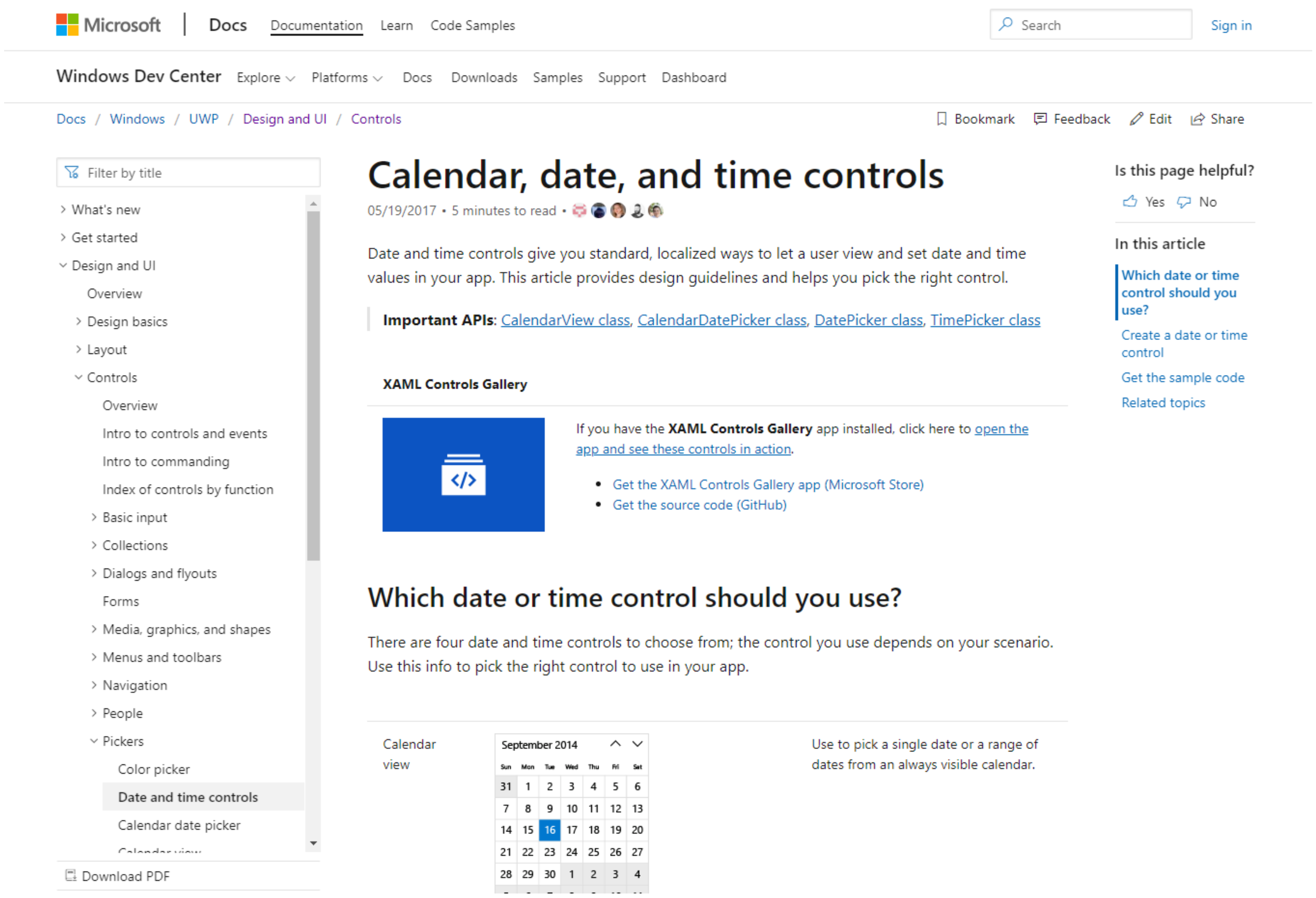
Design downloads

Get design toolkits and samples.

Design basics

An introduction to app design and the Fluent

https://docs.microsoft.com/en-us/windows/uwp/design/



Microsoft | Docs Documentation Learn Code Samples Search Sign in

Windows Dev Center Explore Platforms Docs Downloads Samples Support Dashboard

Docs / Windows / UWP / Design and UI / Controls Bookmark Feedback Edit Share

Filter by title

- What's new
- Get started
- Design and UI
  - Overview
  - Design basics
  - Layout
  - Controls
    - Overview
    - Intro to controls and events
    - Intro to commanding
    - Index of controls by function
    - Basic input
    - Collections
    - Dialogs and flyouts
    - Forms
    - Media, graphics, and shapes
    - Menus and toolbars
    - Navigation
    - People
    - Pickers
      - Color picker
      - Date and time controls
      - Calendar date picker

Download PDF

Calendar, date, and time controls

05/19/2017 • 5 minutes to read •

Date and time controls give you standard, localized ways to let a user view and set date and time values in your app. This article provides design guidelines and helps you pick the right control.

Important APIs: CalendarView class, CalendarDatePicker class, DatePicker class, TimePicker class

XAML Controls Gallery

If you have the **XAML Controls Gallery** app installed, click here to open the app and see these controls in action.

- Get the XAML Controls Gallery app (Microsoft Store)
- Get the source code (GitHub)

Which date or time control should you use?

There are four date and time controls to choose from; the control you use depends on your scenario. Use this info to pick the right control to use in your app.

Calendar view

September 2014

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Use to pick a single date or a range of dates from an always visible calendar.

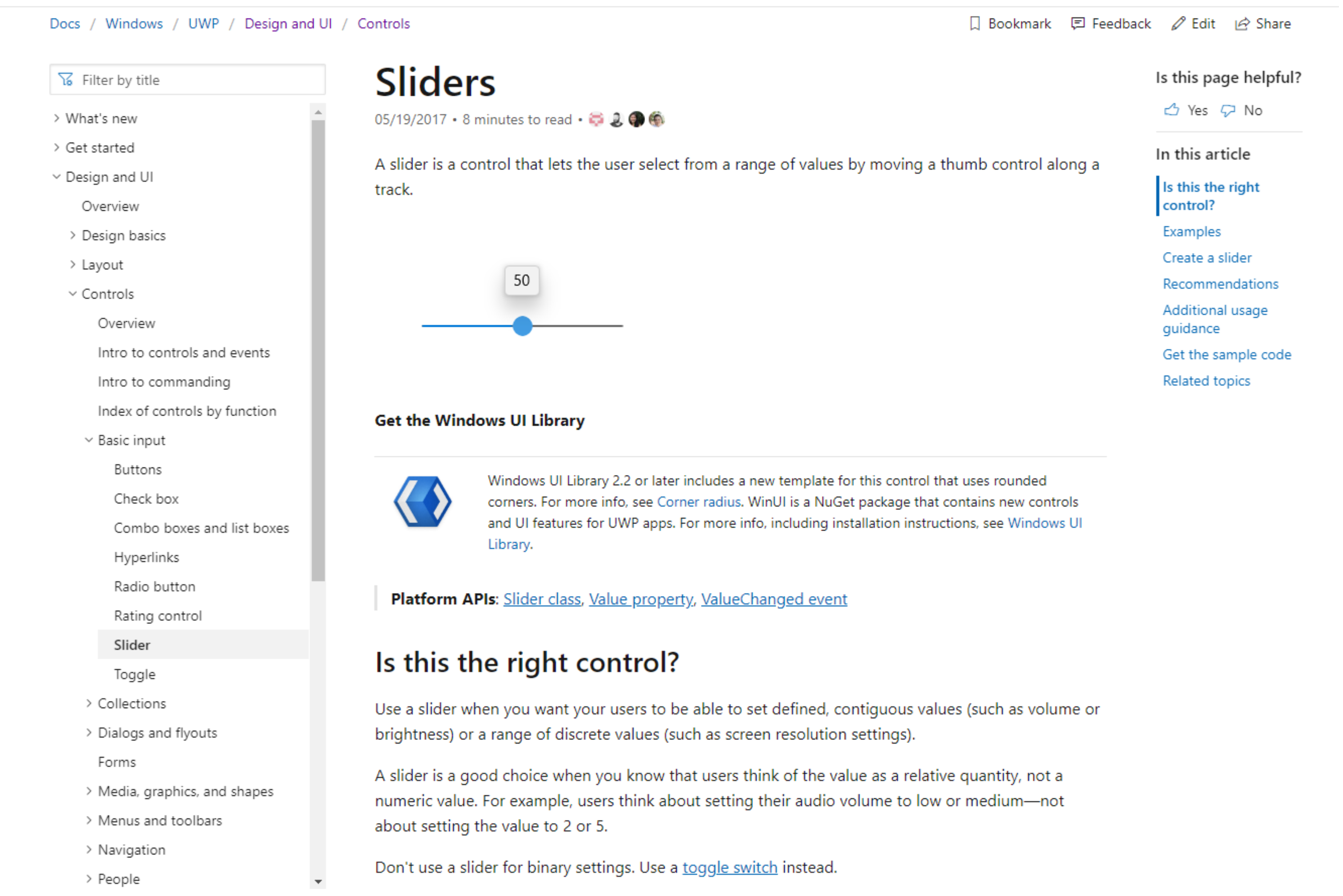
Is this page helpful?

Yes No

In this article

- Which date or time control should you use?
- Create a date or time control
- Get the sample code
- Related topics

https://docs.microsoft.com/en-us/windows/uwp/design/



Docs / Windows / UWP / Design and UI / Controls

Bookmark Feedback Edit Share

Filter by title

- What's new
- Get started
- Design and UI
  - Overview
  - Design basics
  - Layout
  - Controls
    - Overview
    - Intro to controls and events
    - Intro to commanding
    - Index of controls by function
    - Basic input
      - Buttons
      - Check box
      - Combo boxes and list boxes
      - Hyperlinks
      - Radio button
      - Rating control
      - Slider
      - Toggle
    - Collections
    - Dialogs and flyouts
    - Forms
    - Media, graphics, and shapes
    - Menus and toolbars
    - Navigation
    - People

Sliders

05/19/2017 • 8 minutes to read •

A slider is a control that lets the user select from a range of values by moving a thumb control along a track.

50

Get the Windows UI Library

Windows UI Library 2.2 or later includes a new template for this control that uses rounded corners. For more info, see Corner radius. WinUI is a NuGet package that contains new controls and UI features for UWP apps. For more info, including installation instructions, see Windows UI Library.

Platform APIs: Slider class, Value property, ValueChanged event

Is this the right control?

Use a slider when you want your users to be able to set defined, contiguous values (such as volume or brightness) or a range of discrete values (such as screen resolution settings).

A slider is a good choice when you know that users think of the value as a relative quantity, not a numeric value. For example, users think about setting their audio volume to low or medium—not about setting the value to 2 or 5.

Don't use a slider for binary settings. Use a toggle switch instead.

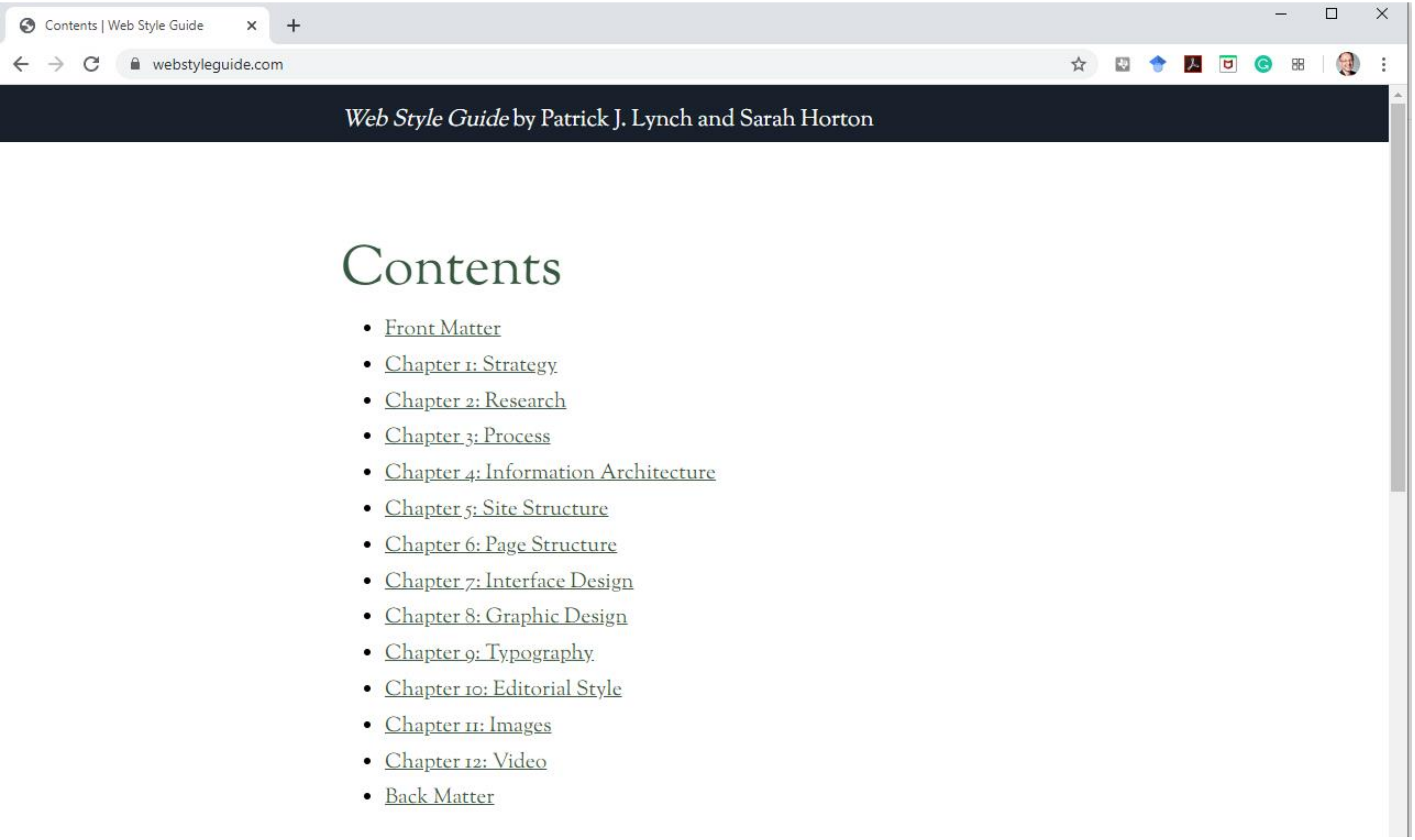
Is this page helpful?

Yes No

In this article

- Is this the right control?
- Examples
- Create a slider
- Recommendations
- Additional usage guidance
- Get the sample code
- Related topics

https://docs.microsoft.com/en-us/windows/uwp/design/


Contents | Web Style Guide

webstyleguide.com

Web Style Guide by Patrick J. Lynch and Sarah Horton

Contents

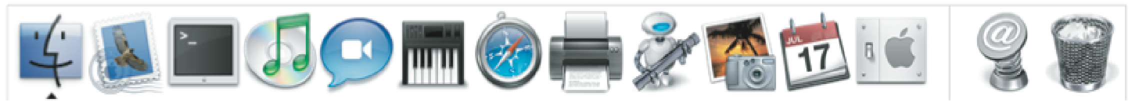
https://webstyleguide.com/

Apple Human Interface Guidelines (2005)

Icon Genres and Families

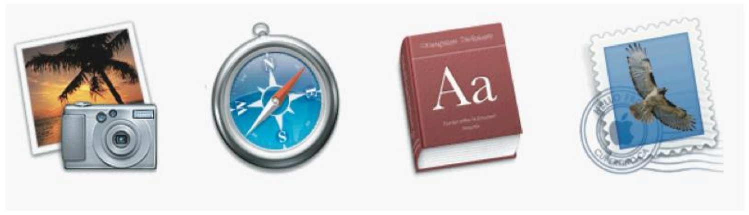
Icon genres help communicate what you can do with an application before you open it. Applications are classified by role—user applications, software utilities, and so on—and each category, or genre, has its own icon style. This differentiation is very important for helping users easily distinguish between types of icons in the Dock.

Figure 10-1 Application icons of different genres—user applications and utilities—shown as they might appear in the Dock

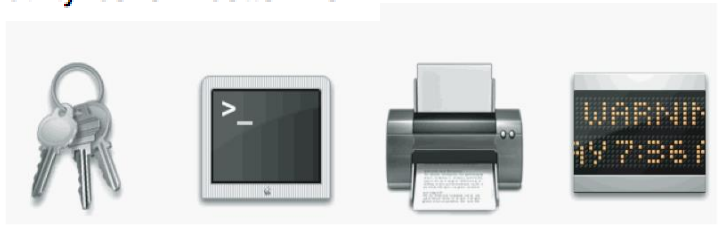


For example, the icons for user applications are colorful and inviting, while utilities have a more serious appearance. Figure 10-2 shows user application icons in the top row and utility icons in the bottom row. These genres are further described in "User Application Icons" (page 125) and "Utility Icons" (page 126).

User application icons in top row

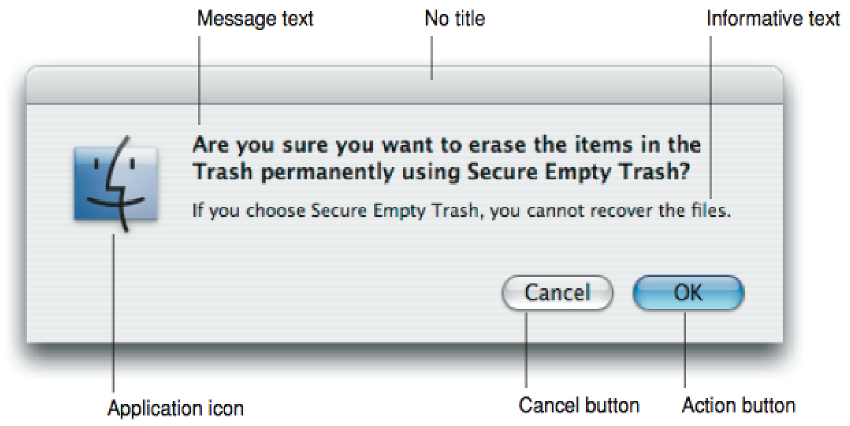


utility icons in bottom row



Apple Human Interface Guidelines (2005), Apple Computer, Inc., p55

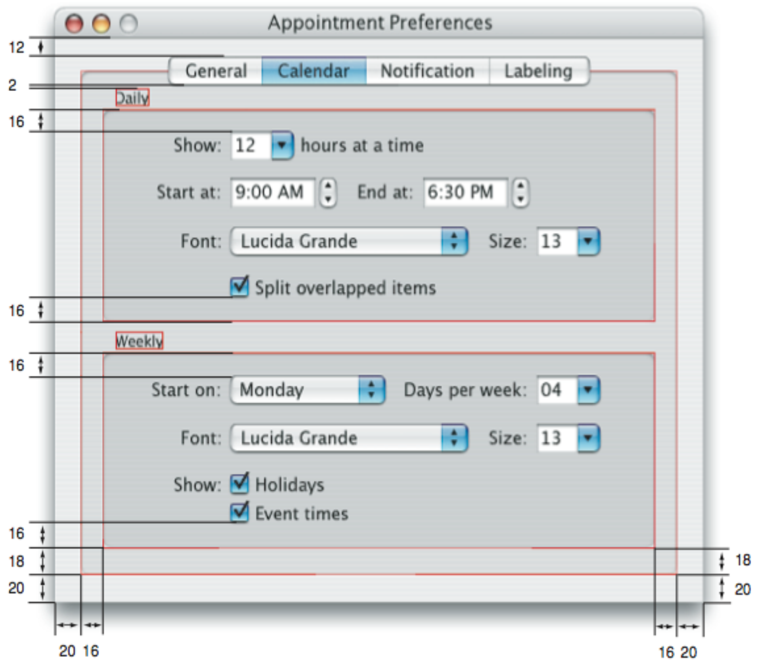
Figure 13-29 A standard alert



Are you sure you want to erase the items in the Trash permanently using Secure Empty Trash?

If you choose Secure Empty Trash, you cannot recover the files.

Figure 15-8 Layout dimensions for a changeable pane dialog



Show: 12 hours at a time

Start at: 9:00 AM End at: 6:30 PM

Font: Lucida Grande Size: 13

☑ Split overlapped items

Start on: Monday Days per week: 04

Font: Lucida Grande Size: 13

Show: ☑ Holidays ☑ Event times

Scrolling List Specifications

Figure 14-51 Scrolling list dimensions

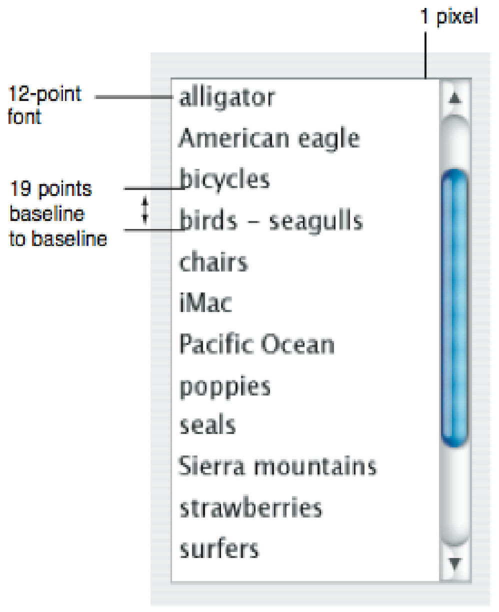
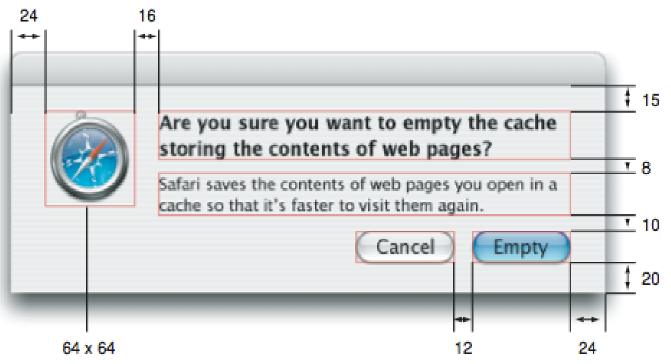


Figure 15-10 Layout dimensions for a standard alert

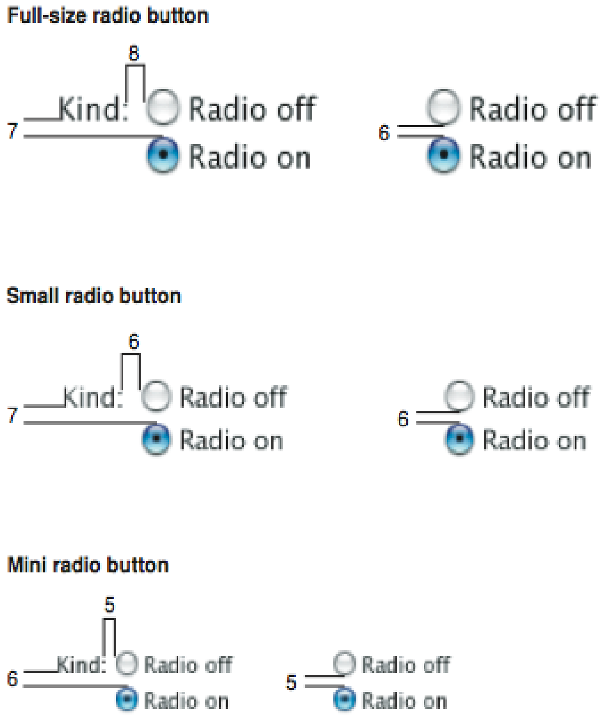


Are you sure you want to empty the cache storing the contents of web pages?

Safari saves the contents of web pages you open in a cache so that it's faster to visit them again.

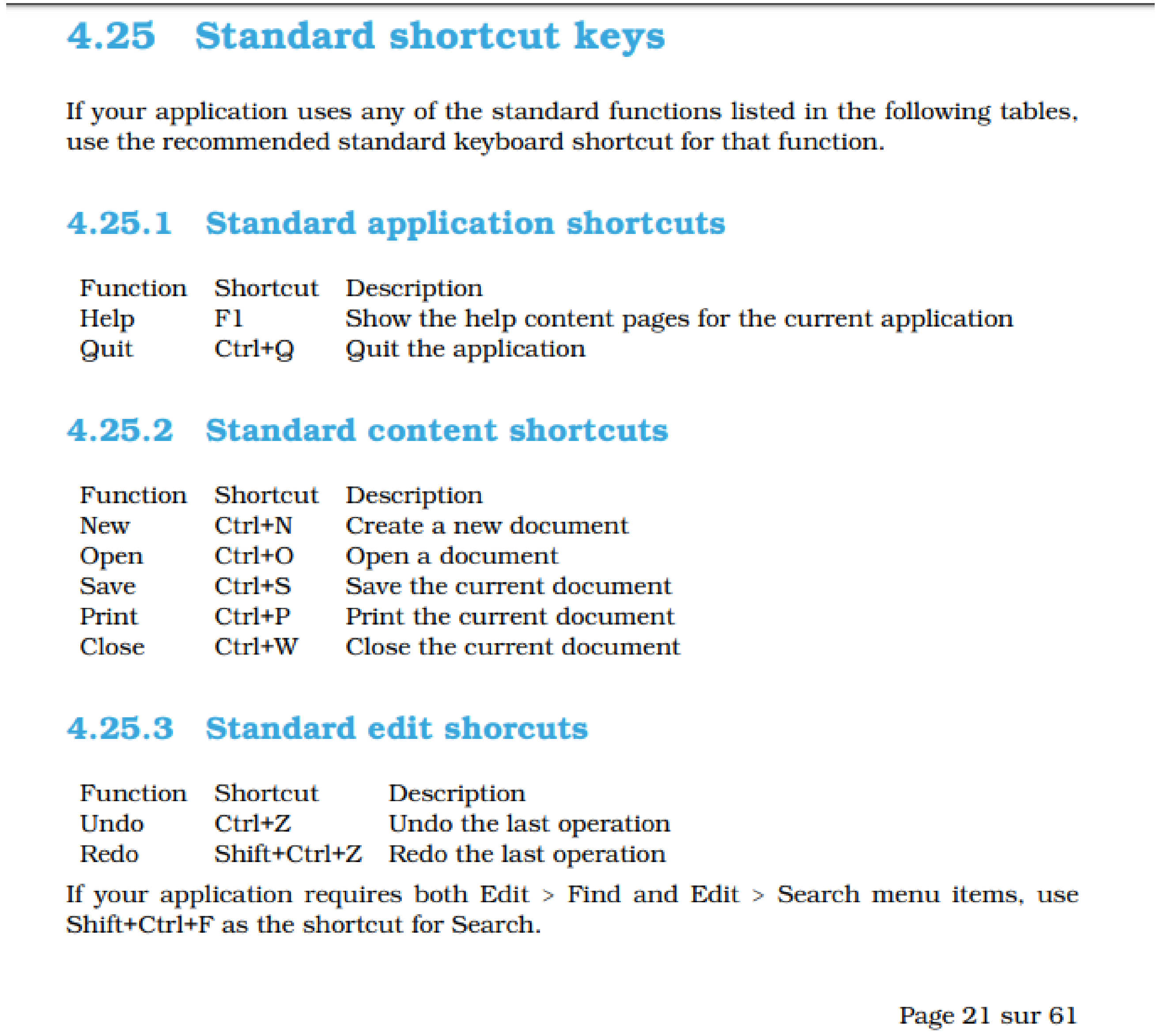
Radio Button Specifications

Figure 14-14 Radio button spacing



Align the baselines of the label and the first button's text.

Apple Human Interface Guidelines (2005), Apple Computer, Inc



4.25 Standard shortcut keys

If your application uses any of the standard functions listed in the following tables, use the recommended standard keyboard shortcut for that function.

4.25.1 Standard application shortcuts

Function	Shortcut	Description
Help	F1	Show the help content pages for the current application
Quit	Ctrl+Q	Quit the application

4.25.2 Standard content shortcuts

Function	Shortcut	Description
New	Ctrl+N	Create a new document
Open	Ctrl+O	Open a document
Save	Ctrl+S	Save the current document
Print	Ctrl+P	Print the current document
Close	Ctrl+W	Close the current document

4.25.3 Standard edit shorcuts

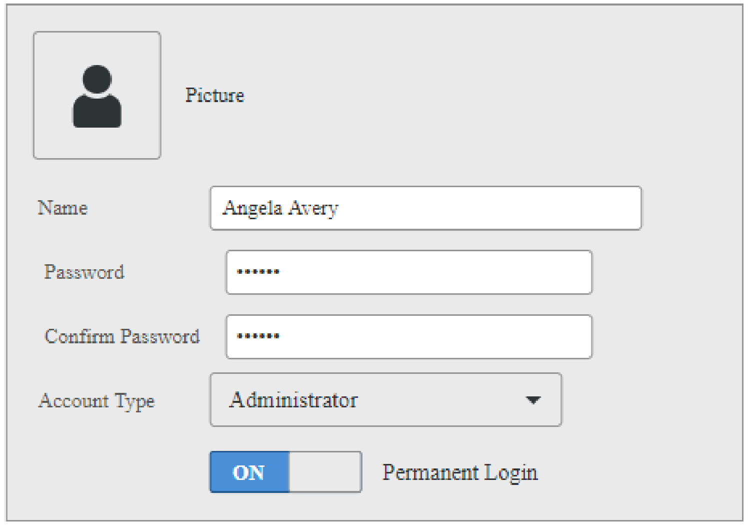
Function	Shortcut	Description
Undo	Ctrl+Z	Undo the last operation
Redo	Shift+Ctrl+Z	Redo the last operation

If your application requires both Edit > Find and Edit > Search menu items, use Shift+Ctrl+F as the shortcut for Search.

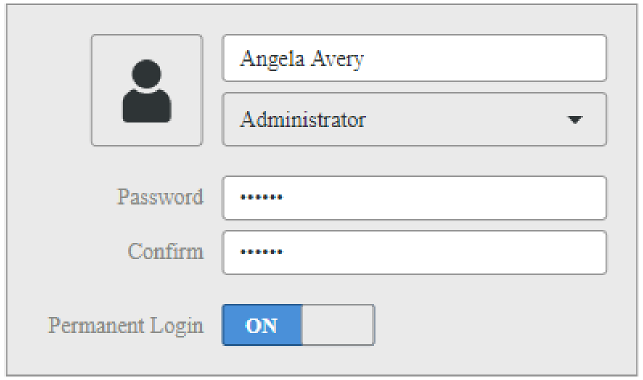
Page 21 sur 61

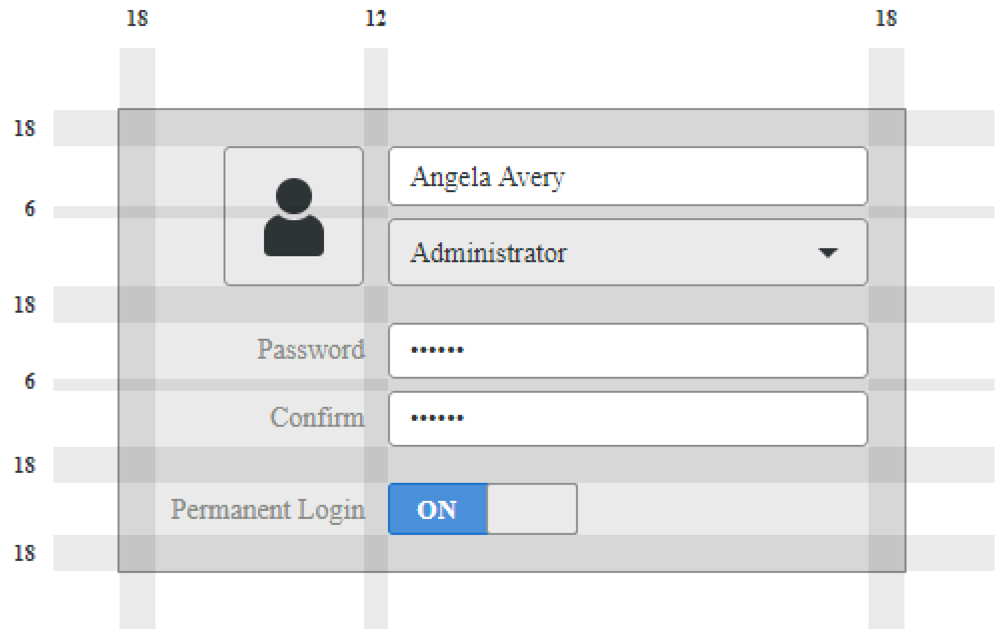
https://people.gnome.org/~fpeters/hig3.pdf

Incorrect spacing and alignment:

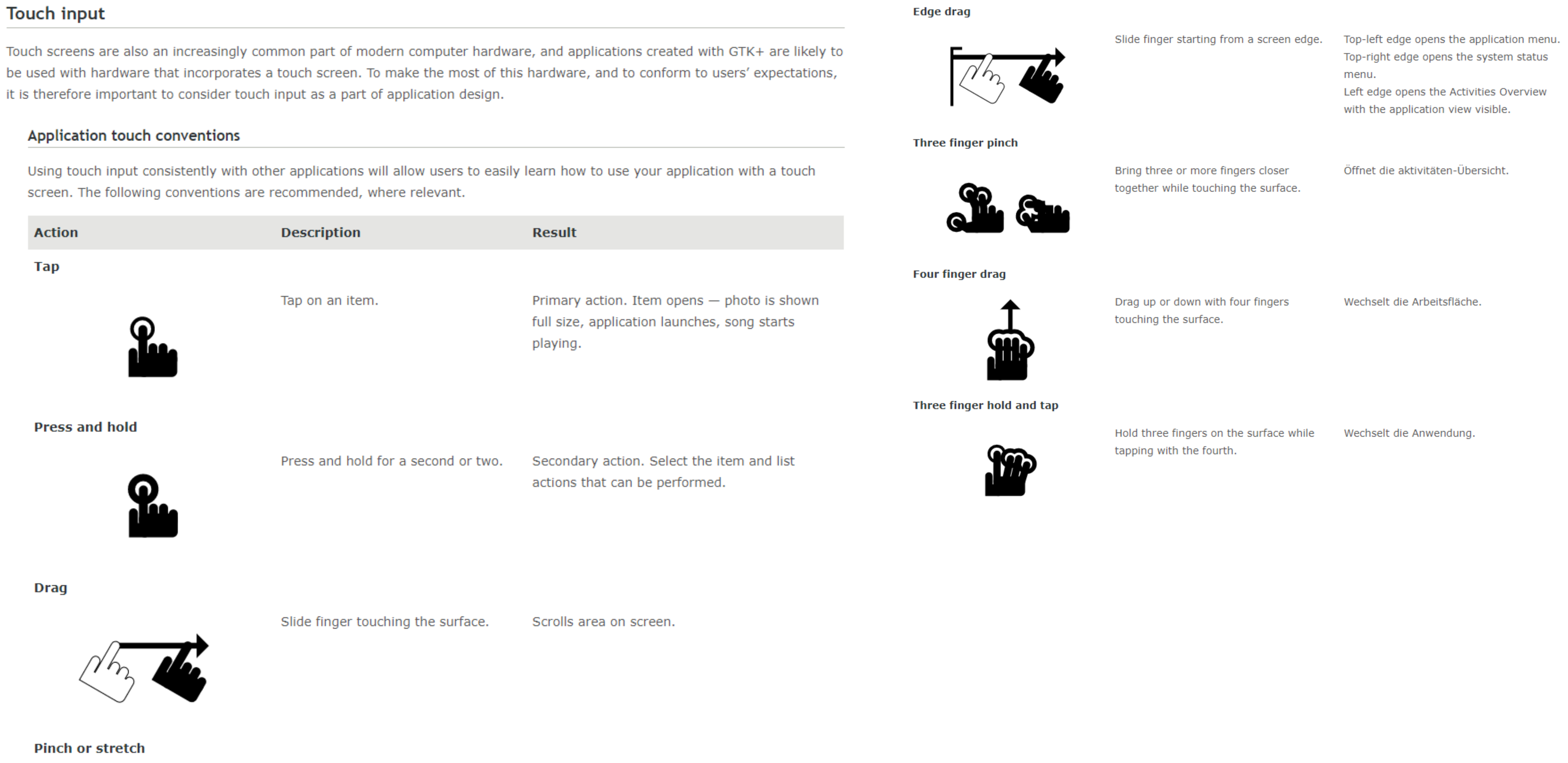


Correct spacing and alignment:





https://developer.gnome.org/hig/stable/visual-layout.html.en



Touch input

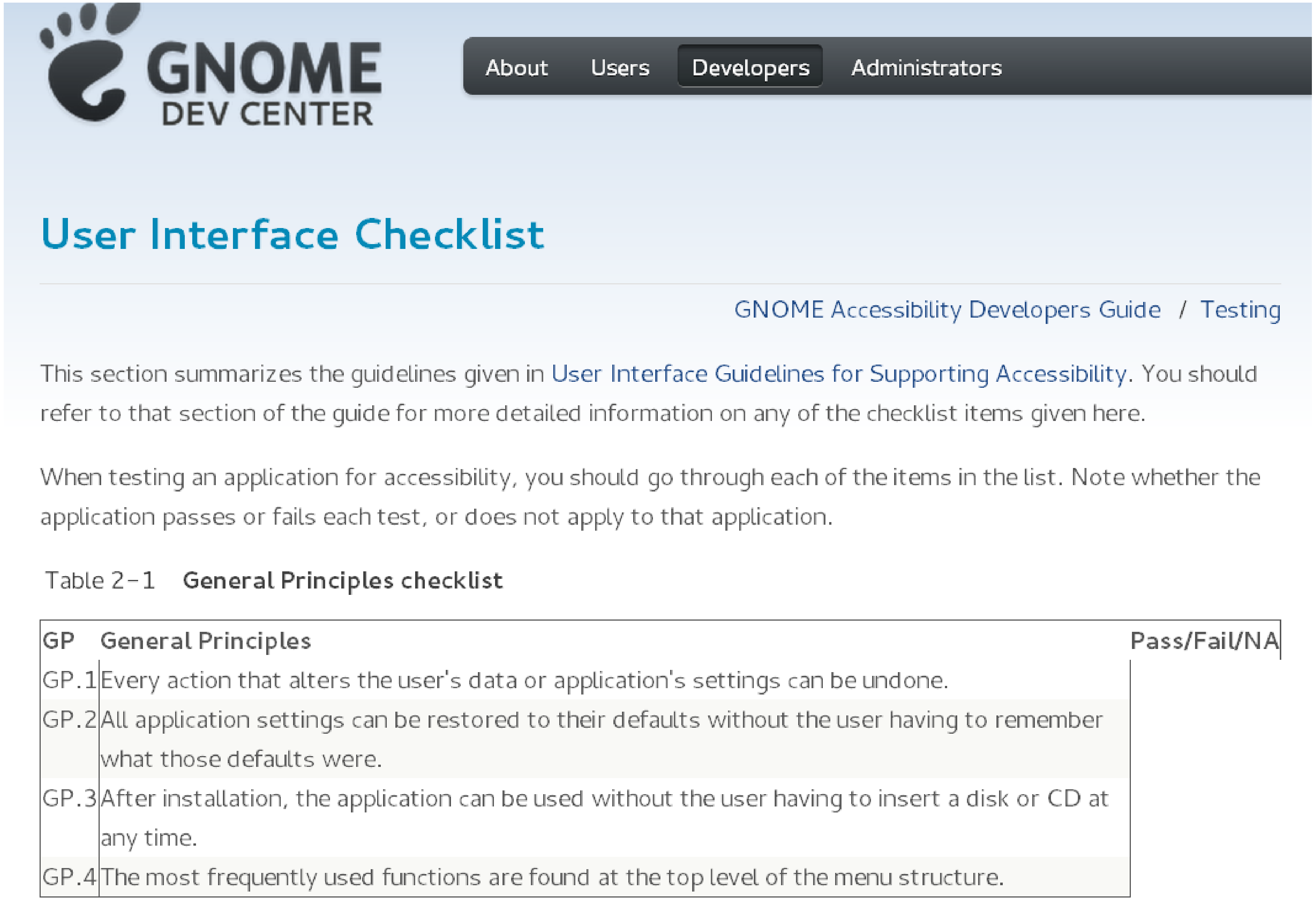
Touch screens are also an increasingly common part of modern computer hardware, and applications created with GTK+ are likely to be used with hardware that incorporates a touch screen. To make the most of this hardware, and to conform to users' expectations, it is therefore important to consider touch input as a part of application design.

Application touch conventions

Using touch input consistently with other applications will allow users to easily learn how to use your application with a touch screen. The following conventions are recommended, where relevant.

Action	Description	Result
Tap	Tap on an item.	Primary action. Item opens — photo is shown full size, application launches, song starts playing.
Press and hold	Press and hold for a second or two.	Secondary action. Select the item and list actions that can be performed.
Drag	Slide finger touching the surface.	Scrolls area on screen.
Pinch or stretch		
Edge drag	Slide finger starting from a screen edge.	Top-left edge opens the application menu. Top-right edge opens the system status menu. Left edge opens the Activities Overview with the application view visible.
Three finger pinch	Bring three or more fingers closer together while touching the surface.	Öffnet die aktivitäten-Übersicht.
Four finger drag	Drag up or down with four fingers touching the surface.	Wechselt die Arbeitsfläche.
Three finger hold and tap	Hold three fingers on the surface while tapping with the fourth.	Wechselt die Anwendung.

https://developer.gnome.org/hig/stable/pointer-and-touch-input.html.en



User Interface Checklist

GNOME Accessibility Developers Guide / Testing

This section summarizes the guidelines given in User Interface Guidelines for Supporting Accessibility. You should refer to that section of the guide for more detailed information on any of the checklist items given here.

When testing an application for accessibility, you should go through each of the items in the list. Note whether the application passes or fails each test, or does not apply to that application.

Table 2-1 **General Principles checklist**

GP	General Principles	Pass/Fail/NA
GP.1	Every action that alters the user's data or application's settings can be undone.	
GP.2	All application settings can be restored to their defaults without the user having to remember what those defaults were.	
GP.3	After installation, the application can be used without the user having to insert a disk or CD at any time.	
GP.4	The most frequently used functions are found at the top level of the menu structure.	

https://developer.gnome.org/accessibility-devel-guide/stable/gad-checklist.html.en

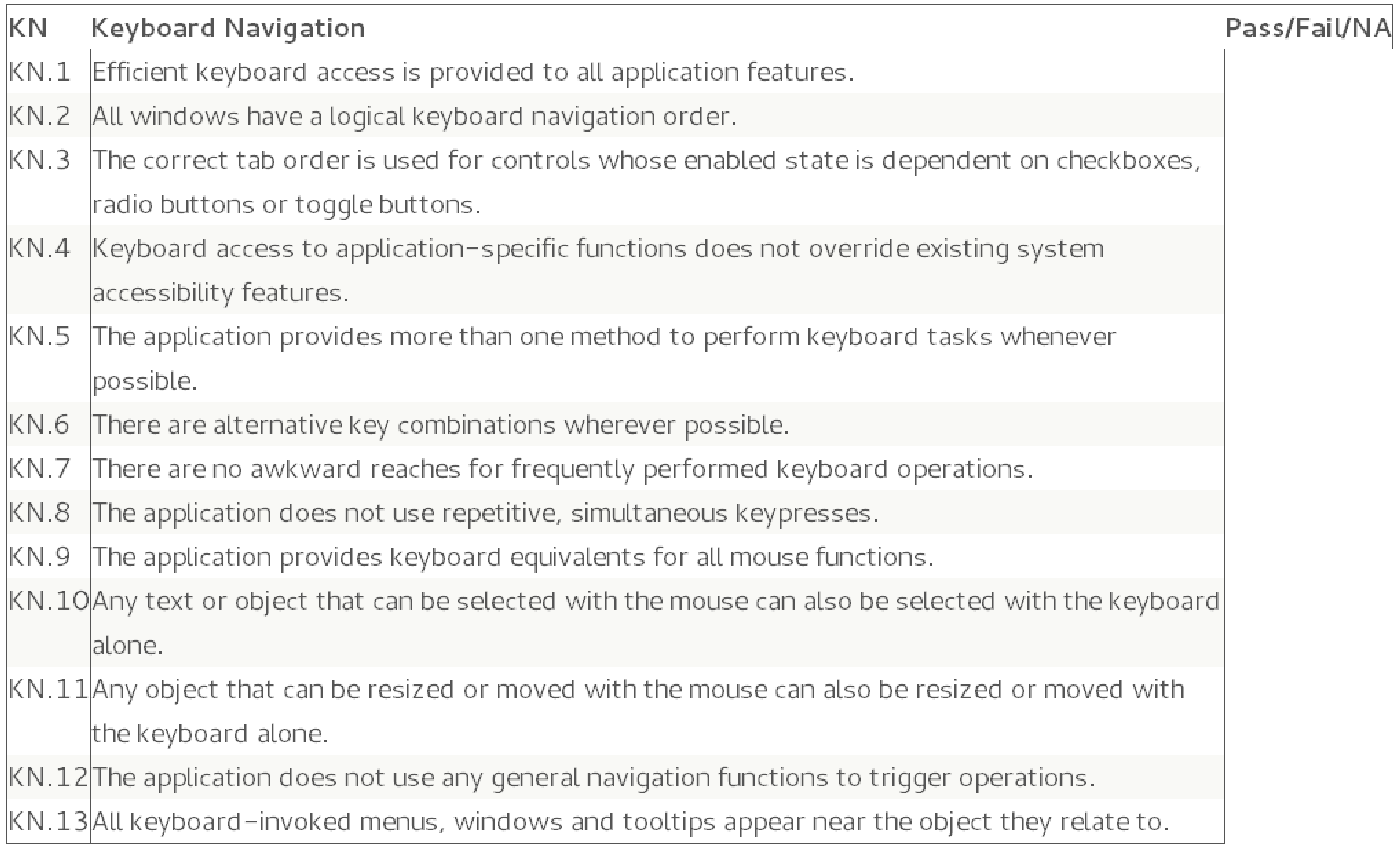


Table 2-2 **Keyboard navigation checklist**

KN	Keyboard Navigation	Pass/Fail/NA
KN.1	Efficient keyboard access is provided to all application features.	
KN.2	All windows have a logical keyboard navigation order.	
KN.3	The correct tab order is used for controls whose enabled state is dependent on checkboxes, radio buttons or toggle buttons.	
KN.4	Keyboard access to application-specific functions does not override existing system accessibility features.	
KN.5	The application provides more than one method to perform keyboard tasks whenever possible.	
KN.6	There are alternative key combinations wherever possible.	
KN.7	There are no awkward reaches for frequently performed keyboard operations.	
KN.8	The application does not use repetitive, simultaneous keypresses.	
KN.9	The application provides keyboard equivalents for all mouse functions.	
KN.10	Any text or object that can be selected with the mouse can also be selected with the keyboard alone.	
KN.11	Any object that can be resized or moved with the mouse can also be resized or moved with the keyboard alone.	
KN.12	The application does not use any general navigation functions to trigger operations.	
KN.13	All keyboard-invoked menus, windows and tooltips appear near the object they relate to.	

https://developer.gnome.org/accessibility-devel-guide/stable/gad-checklist.html.en

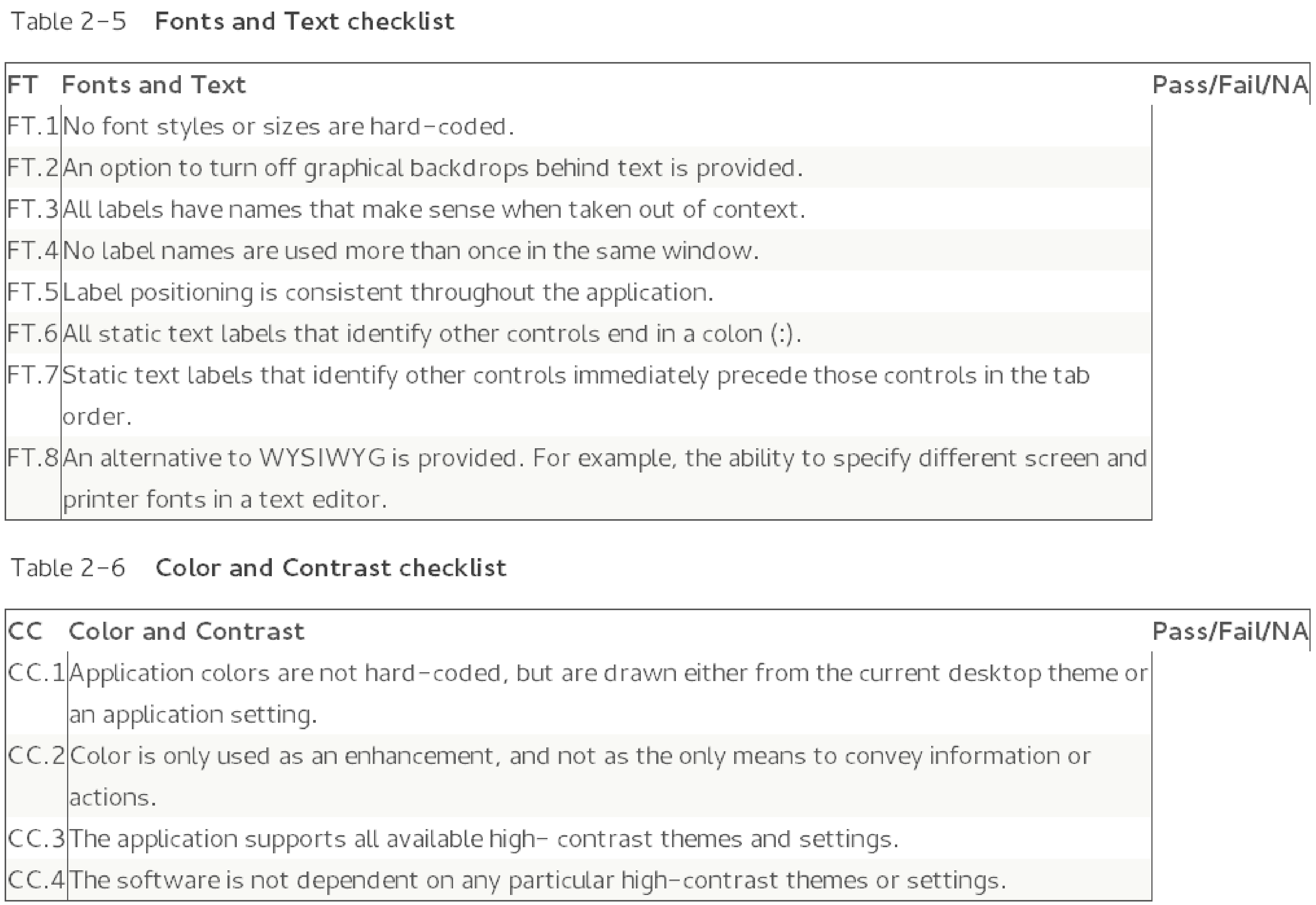


Table 2-5 **Fonts and Text checklist**

FT	Fonts and Text	Pass/Fail/NA
FT.1	No font styles or sizes are hard-coded.	
FT.2	An option to turn off graphical backdrops behind text is provided.	
FT.3	All labels have names that make sense when taken out of context.	
FT.4	No label names are used more than once in the same window.	
FT.5	Label positioning is consistent throughout the application.	
FT.6	All static text labels that identify other controls end in a colon (:).	
FT.7	Static text labels that identify other controls immediately precede those controls in the tab order.	
FT.8	An alternative to WYSIWYG is provided. For example, the ability to specify different screen and printer fonts in a text editor.	

Table 2-6 **Color and Contrast checklist**

CC	Color and Contrast	Pass/Fail/NA
CC.1	Application colors are not hard-coded, but are drawn either from the current desktop theme or an application setting.	
CC.2	Color is only used as an enhancement, and not as the only means to convey information or actions.	
CC.3	The application supports all available high- contrast themes and settings.	
CC.4	The software is not dependent on any particular high-contrast themes or settings.	

https://developer.gnome.org/accessibility-devel-guide/stable/gad-checklist.html.en

Gnome User Interface Checklist

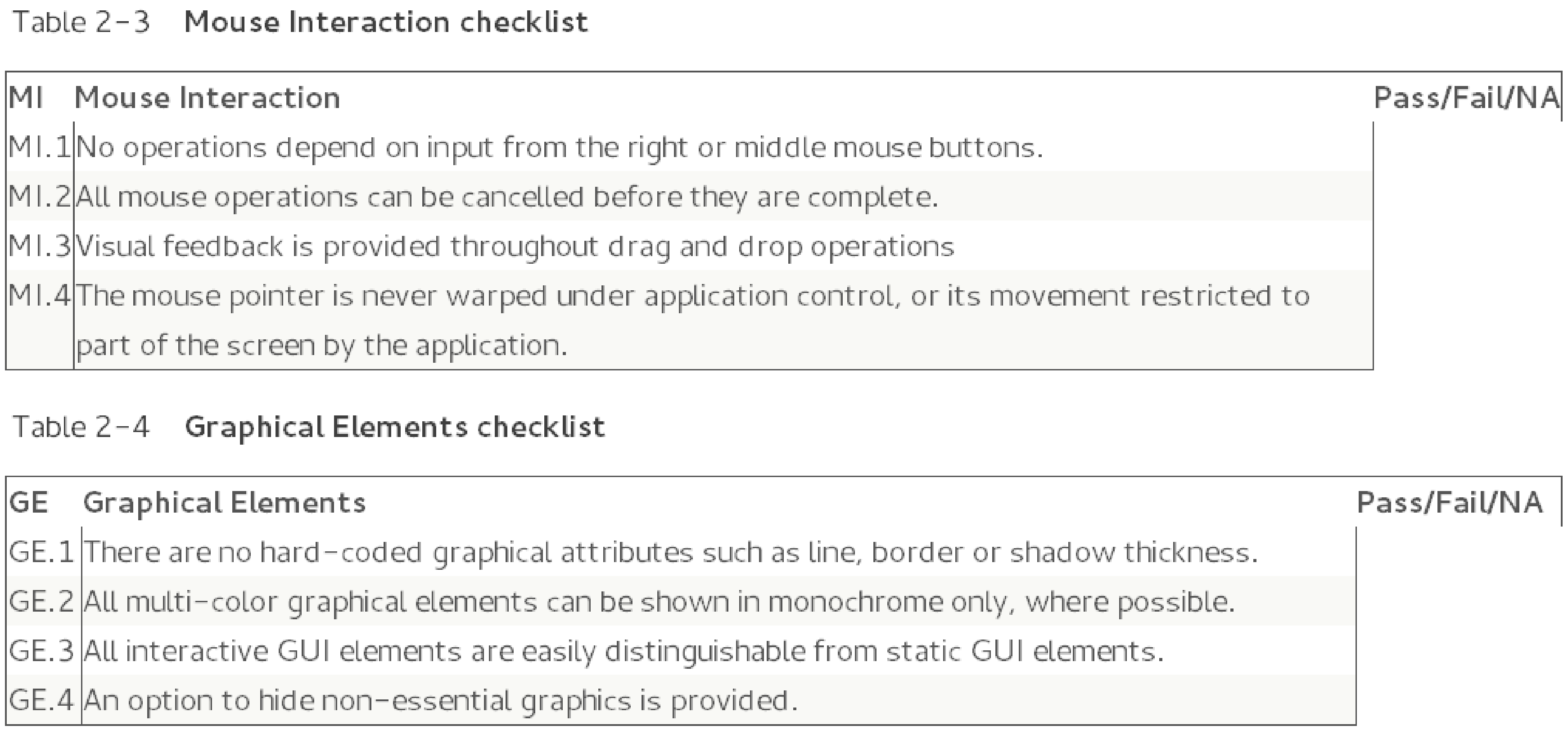


Table 2-3 **Mouse Interaction checklist**

MI	Mouse Interaction	Pass/Fail/NA
MI.1	No operations depend on input from the right or middle mouse buttons.	
MI.2	All mouse operations can be cancelled before they are complete.	
MI.3	Visual feedback is provided throughout drag and drop operations	
MI.4	The mouse pointer is never warped under application control, or its movement restricted to part of the screen by the application.	

Table 2-4 **Graphical Elements checklist**

GE	Graphical Elements	Pass/Fail/NA
GE.1	There are no hard-coded graphical attributes such as line, border or shadow thickness.	
GE.2	All multi-color graphical elements can be shown in monochrome only, where possible.	
GE.3	All interactive GUI elements are easily distinguishable from static GUI elements.	
GE.4	An option to hide non-essential graphics is provided.	

https://developer.gnome.org/accessibility-devel-guide/stable/gad-checklist.html.en

Gnome User Interface Checklist

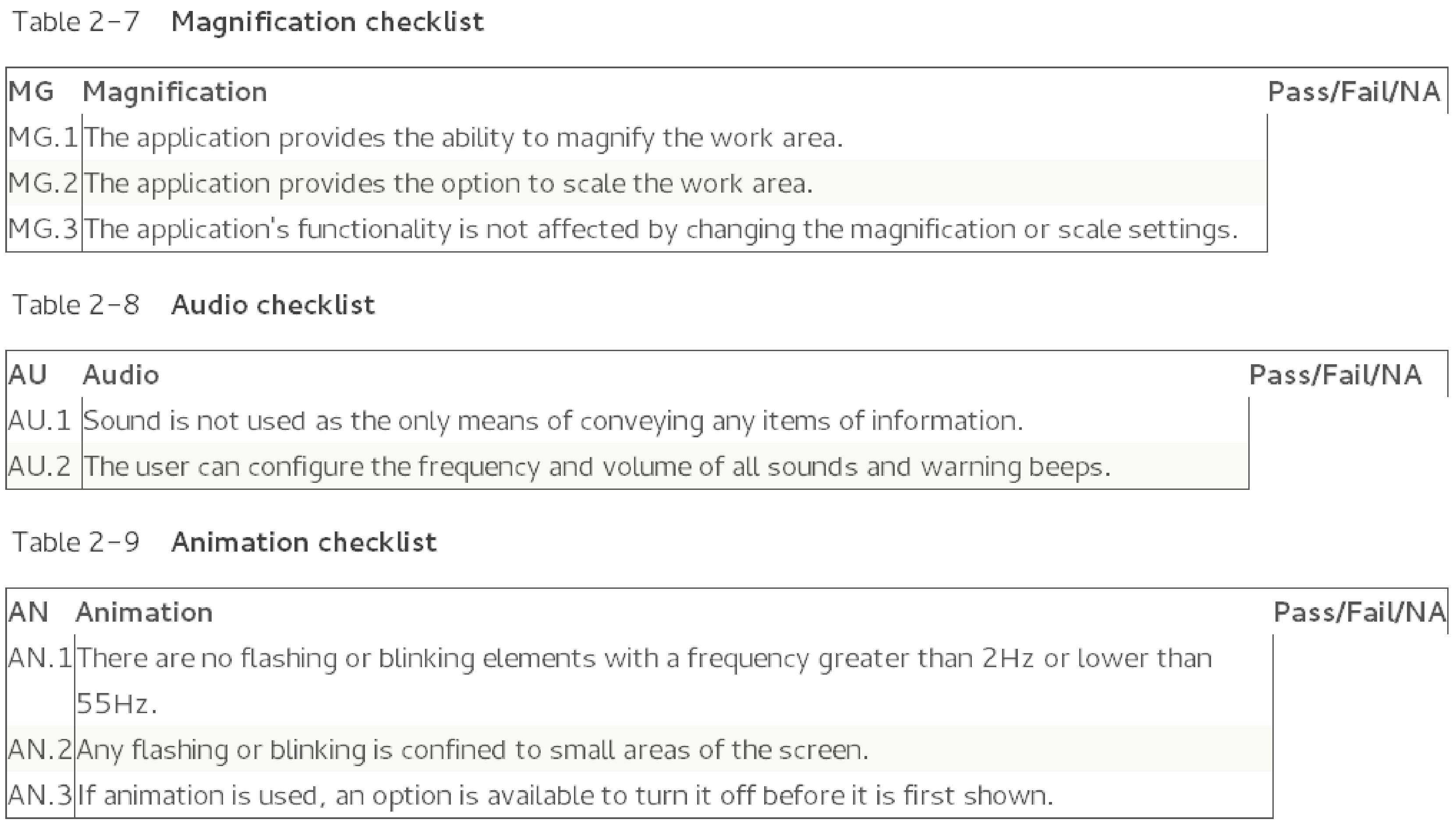


Table 2-7 **Magnification checklist**

MG	Magnification	Pass/Fail/NA
MG.1	The application provides the ability to magnify the work area.	
MG.2	The application provides the option to scale the work area.	
MG.3	The application's functionality is not affected by changing the magnification or scale settings.	

Table 2-8 **Audio checklist**

AU	Audio	Pass/Fail/NA
AU.1	Sound is not used as the only means of conveying any items of information.	
AU.2	The user can configure the frequency and volume of all sounds and warning beeps.	

Table 2-9 **Animation checklist**

AN	Animation	Pass/Fail/NA
AN.1	There are no flashing or blinking elements with a frequency greater than 2Hz or lower than 55Hz.	
AN.2	Any flashing or blinking is confined to small areas of the screen.	
AN.3	If animation is used, an option is available to turn it off before it is first shown.	

https://developer.gnome.org/accessibility-devel-guide/stable/gad-checklist.html.en

Gnome User Interface Checklist

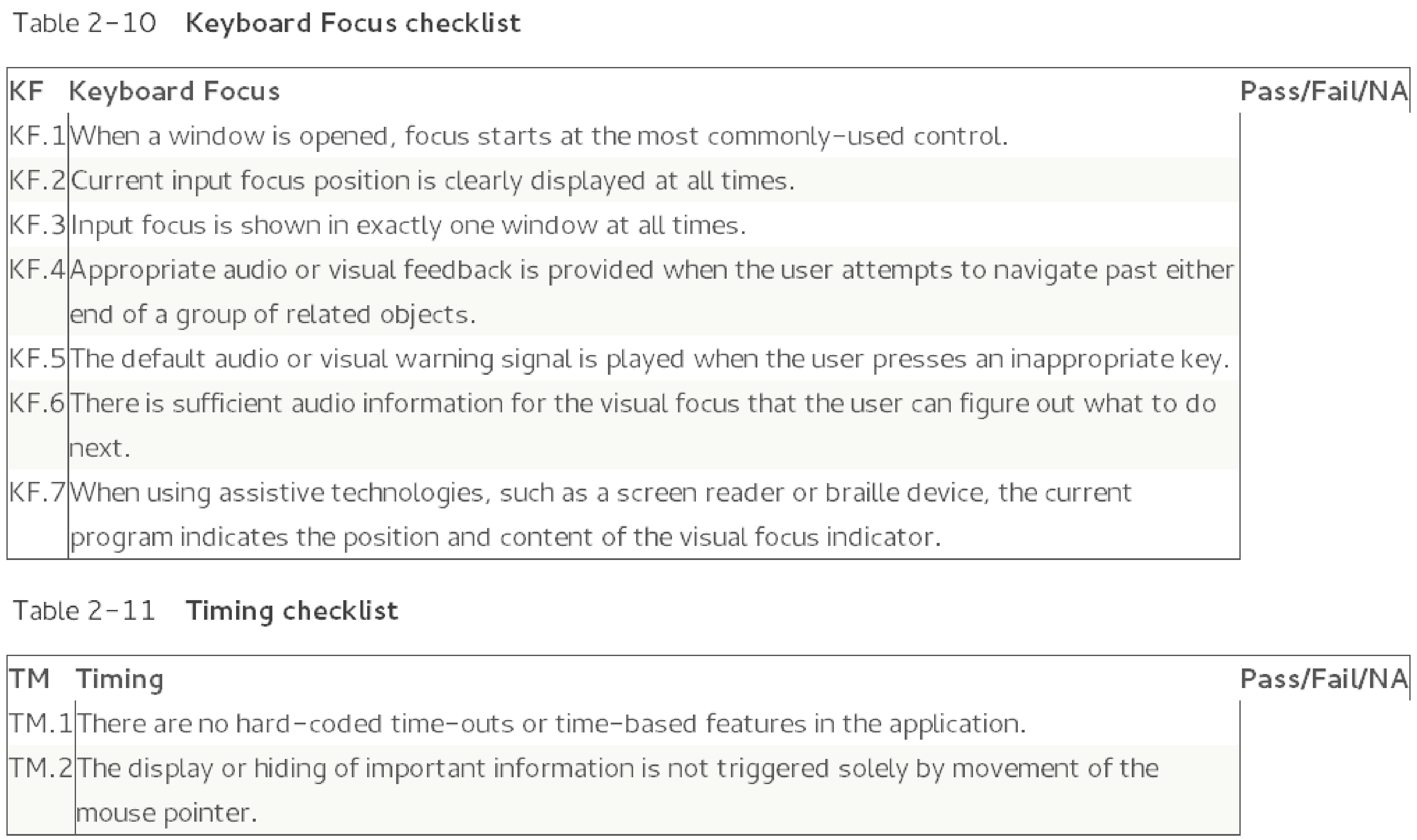


Table 2-10 **Keyboard Focus checklist**

KF	Keyboard Focus	Pass/Fail/NA
KF.1	When a window is opened, focus starts at the most commonly-used control.	
KF.2	Current input focus position is clearly displayed at all times.	
KF.3	Input focus is shown in exactly one window at all times.	
KF.4	Appropriate audio or visual feedback is provided when the user attempts to navigate past either end of a group of related objects.	
KF.5	The default audio or visual warning signal is played when the user presses an inappropriate key.	
KF.6	There is sufficient audio information for the visual focus that the user can figure out what to do next.	
KF.7	When using assistive technologies, such as a screen reader or braille device, the current program indicates the position and content of the visual focus indicator.	

Table 2-11 **Timing checklist**

TM	Timing	Pass/Fail/NA
TM.1	There are no hard-coded time-outs or time-based features in the application.	
TM.2	The display or hiding of important information is not triggered solely by movement of the mouse pointer.	

https://developer.gnome.org/accessibility-devel-guide/stable/gad-checklist.html.en

Did you understand this block?

Can you answer these questions?

- What are the main recommendations in the 4 main recommendations in the Web Content Accessibility Guidelines (WCAG) 2.1
- What is the common structure for a user interface guideline? What are typical parts that are described?
- Find where the following is described described.
  - How do you design buttons on the apple watch?
  - What recommendations are there for icon design for Windows?
  - What are the font recommendations for Material Design?

Reference

- Hix and Hartson, Developing User Interfaces, Wiley, 1993
- ISO 9241 Ergonomics of human–system interaction
- Verordnung über Arbeitsstätten (Arbeitsstättenverordnung - ArbStättV) https://www.gesetze-im-internet.de/arbst_ttv_2004/anhang.html
- Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz https://www.gesetze-im-internet.de/bitv_2_0/index.html
- Web Content Accessibility Guidelines (WCAG) 2.1 www.w3.org/TR/WCAG21
- https://developer.apple.com/design/human-interface-guidelines/macos/
- https://developer.apple.com/design/human-interface-guidelines/ios/views/
- https://developer.apple.com/design/human-interface-guidelines/watchos/overview/themes/
- https://material.io/design
- https://docs.microsoft.com/en-us/windows/uwp/design/
- https://webstyleguide.com/
- Apple Human Interface Guidelines (2005), Apple Computer, Inc
- https://people.gnome.org/~fpeters/hig3.pdf
- https://developer.gnome.org/hig/stable/pointer-and-touch-input.html.en
- https://developer.gnome.org/accessibility-devel-guide/stable/gad-checklist.html.en

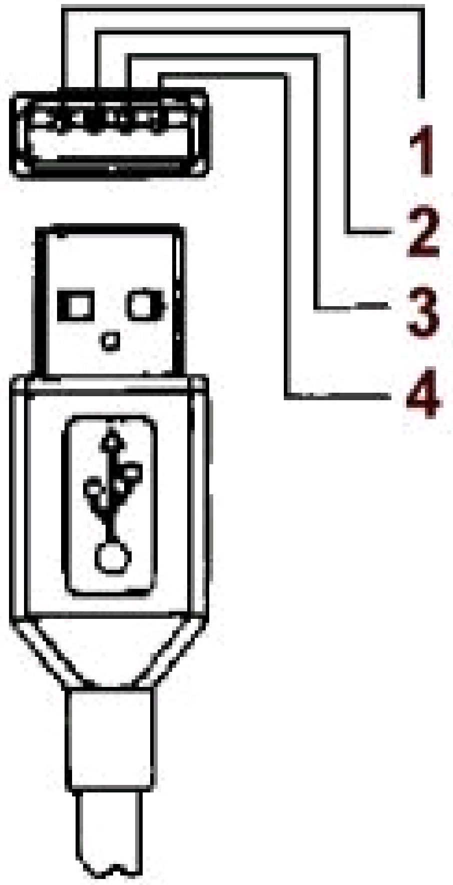
License

Constraints and Mappings

Learning Goals

- Understand …
  - What constraints are in the context of interaction design and user interfaces
  - How mappings impact the easy of use
- Be able to …
  - use mappings and constraints as mechanisms in the design of user interfaces to prevent errors

USB-A-Stecker (Front- und Draufsicht)



1 +5V
2 D-
3 D+
4 Ground

Physical Constraints & Affordances

- USB Memory Stick vs. DVD vs. money
  - How many ways can you put them in?
  - If there is more than one option (physically) cater for these cases

- Dials vs. Buttons vs. Sliders
  - Dials are turned
  - Buttons are pressed
  - Sliders are pushed

Constraints

- Physical constraints
  - Basic physical limitations
- Semantic constraints
  - Assumption to create something meaningful
- Cultural constraints
  - Borders and context provided by cultural conventions
- Logical constraints
  - Restrictions due to reasoning

Applying constraints is a design decision!
A Practical way to realise the principle “prevent errors”

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

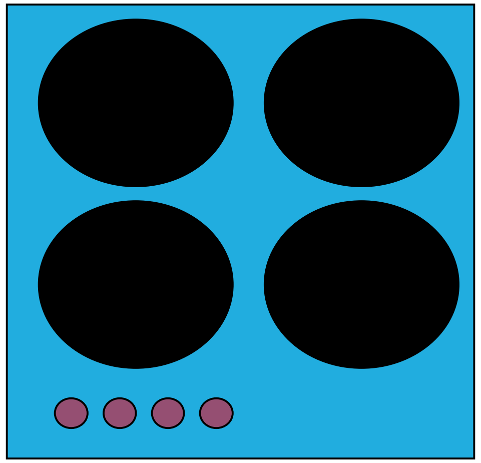
Cultural Constraints

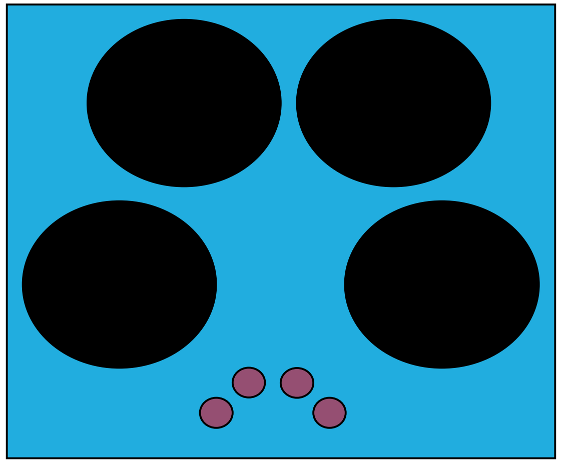
- Universal or culturally specific
- Arbitrary conventions that have been learned
- Users’ expectations build on cultural constraints

- Example Colors
  - Red
  - Green
  - Blue

Mapping

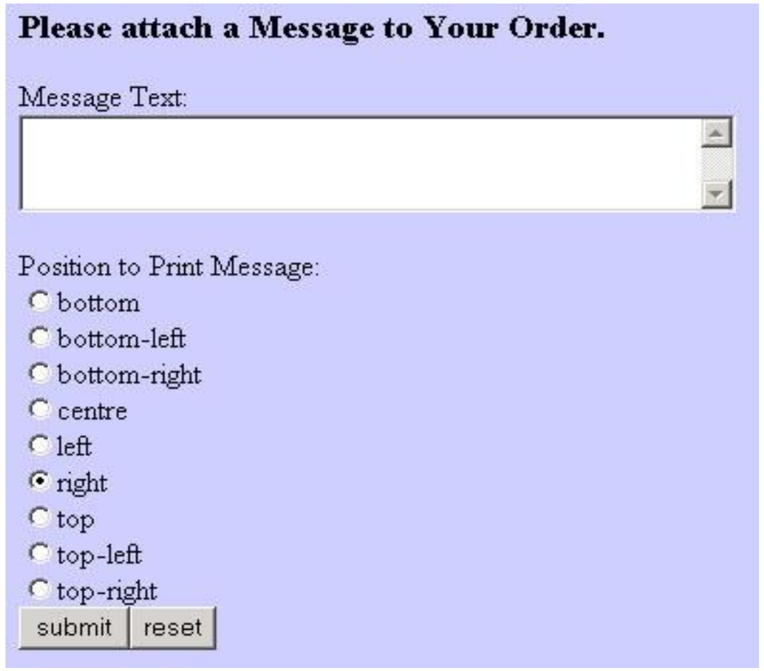
- Relationship between controls and action
- Mappings should be
  - Understandable
    (e.g. moving the mouse up move the slider up)
  - Consistent
  - Recognizable or at least quickly
    learnable and easy to recall
  - Natural, meaning to be consistent
    with knowledge the user already has
- Example: cooker
- For these issues see also Gestalt theory!





Mapping

Example

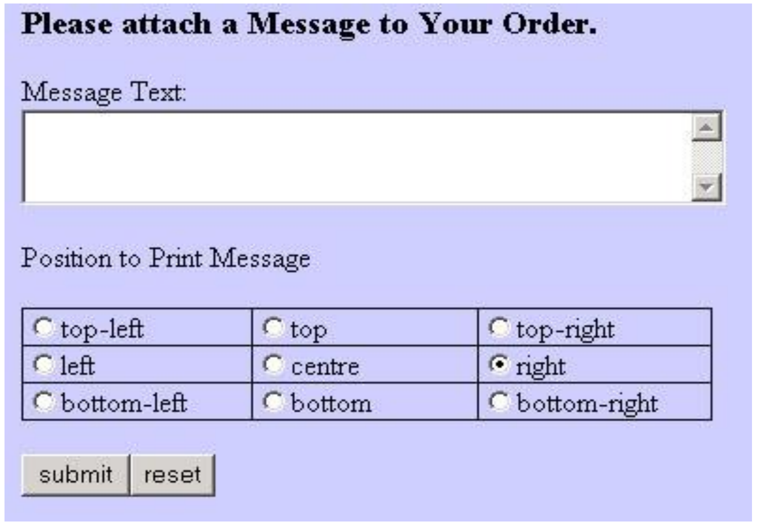


Please attach a Message to Your Order.

Message Text:

Position to Print Message:

- bottom
- bottom-left
- bottom-right
- centre
- left
- [✓ right]
- top
- top-left
- top-right

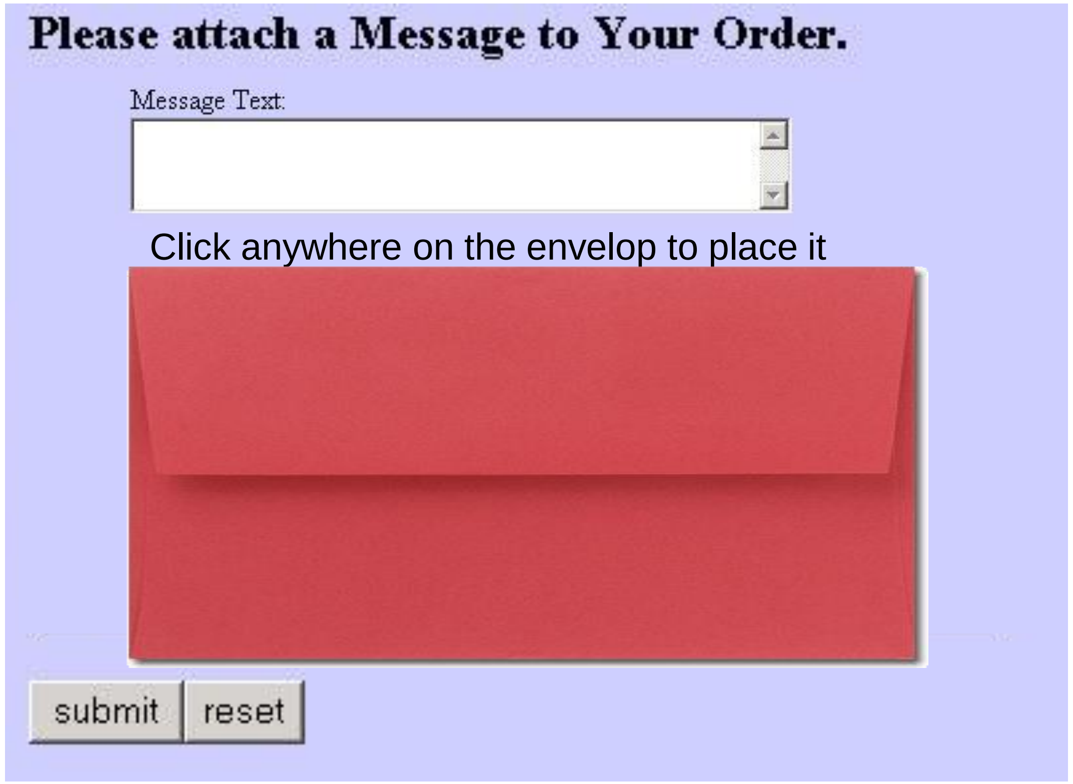


Please attach a Message to Your Order.

Message Text:

Position to Print Message

top-left	top	top-right
left	centre	[✓ right]
bottom-left	bottom	bottom-right



Please attach a Message to Your Order.

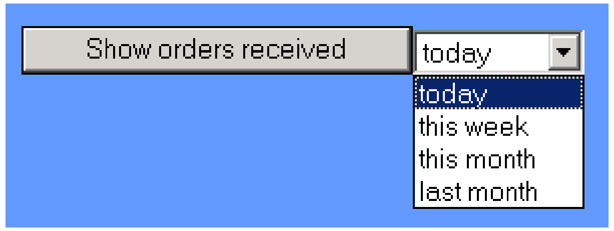
Message Text:

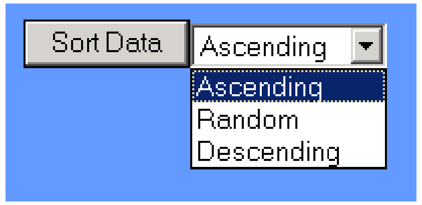
Click anywhere on the envelop to place it

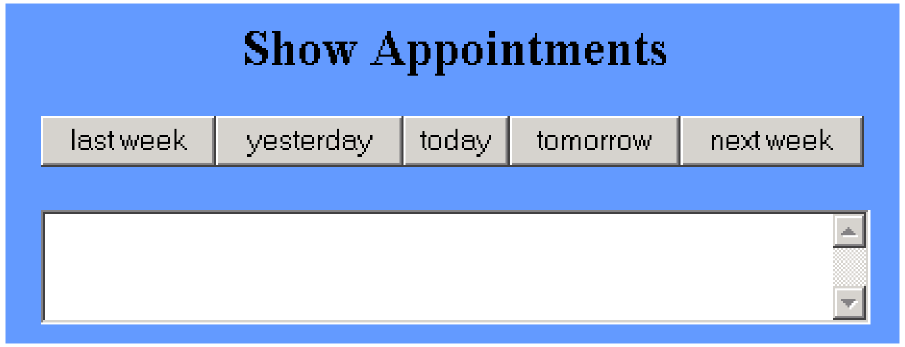
Mapping

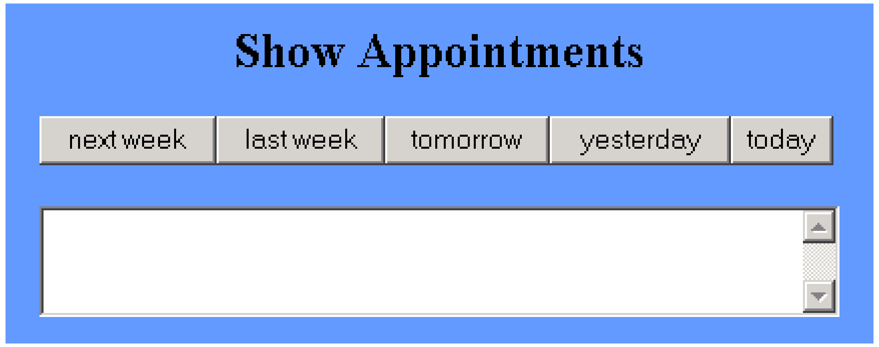
Example

- "Natural" mappings can be found in many areas
- It is not always obvious what the "natural" mapping is
- Correlation with cultural constraints









Did you understand this block?

Can you answer these questions?

- What different constrains are relevant for the design of user interfaces?
- Explain cultural constraints in the context of Uis.
- What is important when we design mappings? How can mappings increase or decrease usability?

Reference

- Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York: Doubleday.

License