Vietnam National University of HCMC

International University

School of Computer Science and Engineering

UI/UX Design & Evaluation
★ Design Theory, Principles and Guidelines ★

Dr Vi Chi Thanh - vcthanh@hcmiu.edu.vn

https://vichithanh.github.io

SCAN ME

Course Overview

1. Introduction to HCI
2. Needfinding
3. Defining needs and tasks
4. Prototyping
5. **Design guidelines, principles, and heuristics**

6. Human abilities and theoretical models
7. Visual design and design patterns
8. Heuristic evaluation
9. Usability testing
10. Advanced interactions

Hall of Fame or Shame?

Did we make you smile?

Based on your shopping experience, how likely are you to recommend us on a scale of 0 - 10?
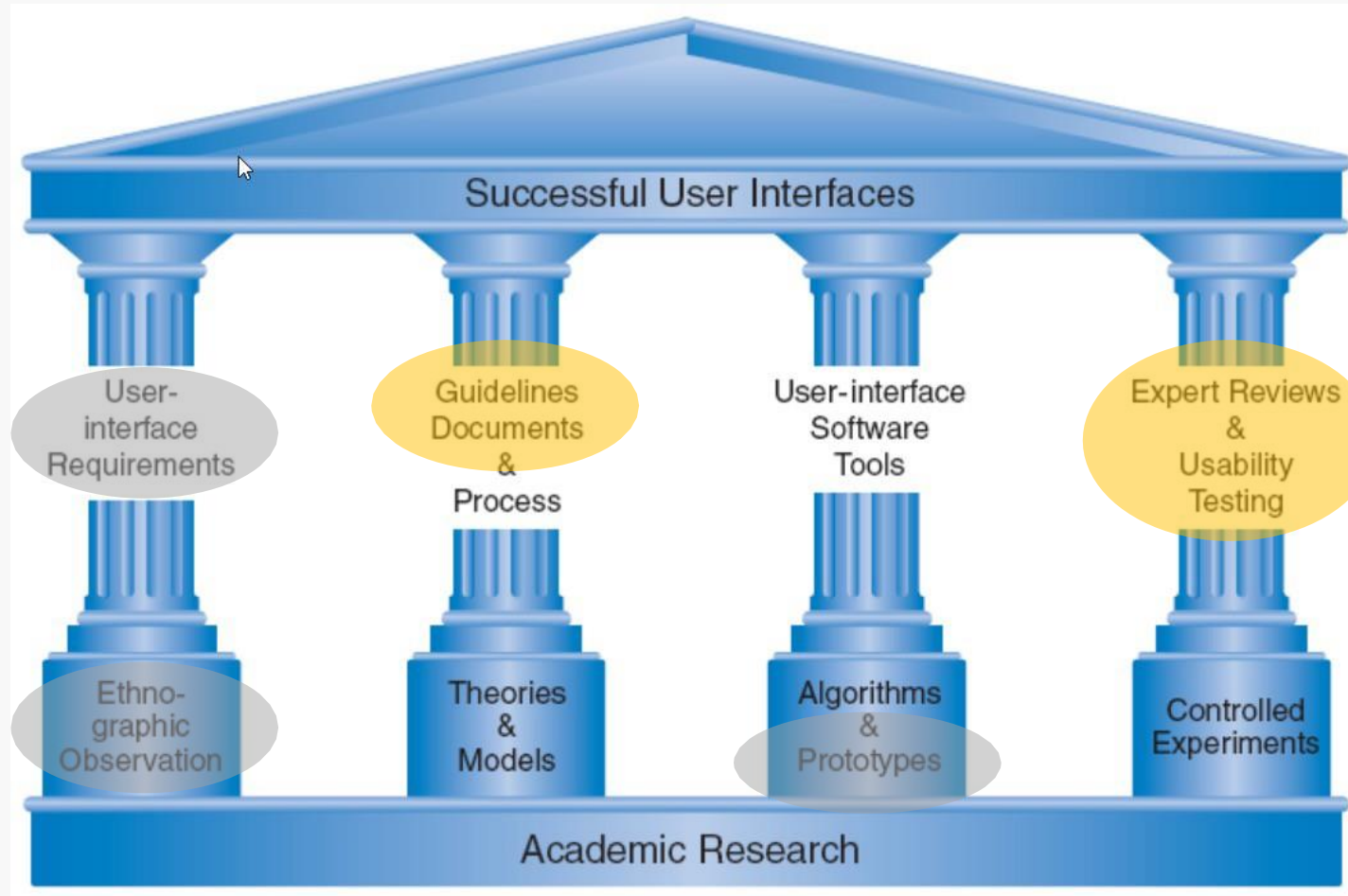
Extremely unlikely

0 1 2 3 4

5 6 7 8 9 10

Extremely likely

The Four Pillars of Design

Ben Shneiderman & Catherine Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction

Goals

- Generating design solutions
  - Guidelines
  - Principles
  - Theories

- Evaluating generated designs
  - Expert reviews and heuristics
  - Usability testing
  - Controlled experiments

Generating Design Solutions

- Guidelines: Low-level focused advice about good practices and cautions against dangers.

- Principles: Mid-level strategies or rules to analyze and compare design alternatives.

- Theories: High-level widely applicable frameworks to draw on during design and evaluation, as well as to support communication and teaching.

Specific cases

General

Practical, operational

Concrete formulation

Abstract, theoretical
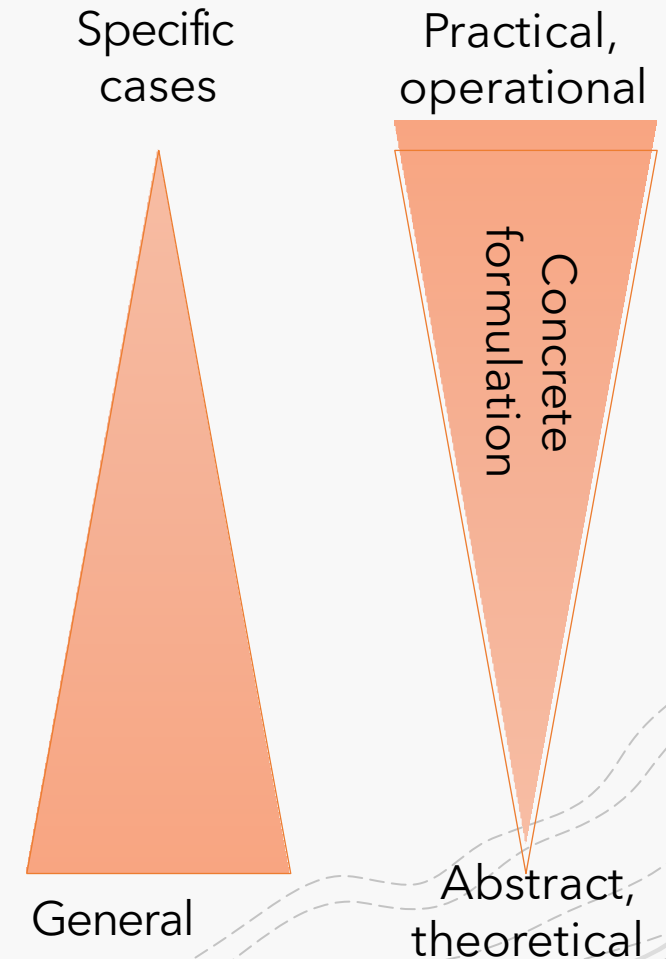
Design Theories

Theoretical frameworks enabling foundational research

The "Why"

Design Theories

Types of theories

- Descriptive
  - UI elements, terminology, semantics
- Explanatory
  - Sequences of events with causal relationships
- Prescriptive
  - Guidelines for designers to make decisions
- Predictive
  - Comparison of design alternatives based on performance figures

Human capacity

- Motor task
  - Skill in pointing, clicking, … movements
- Perceptual
  - Sensory inputs
- Cognitive
  - Problem-solving, short-/long-term memory

Norman's Action Models (Explanatory)
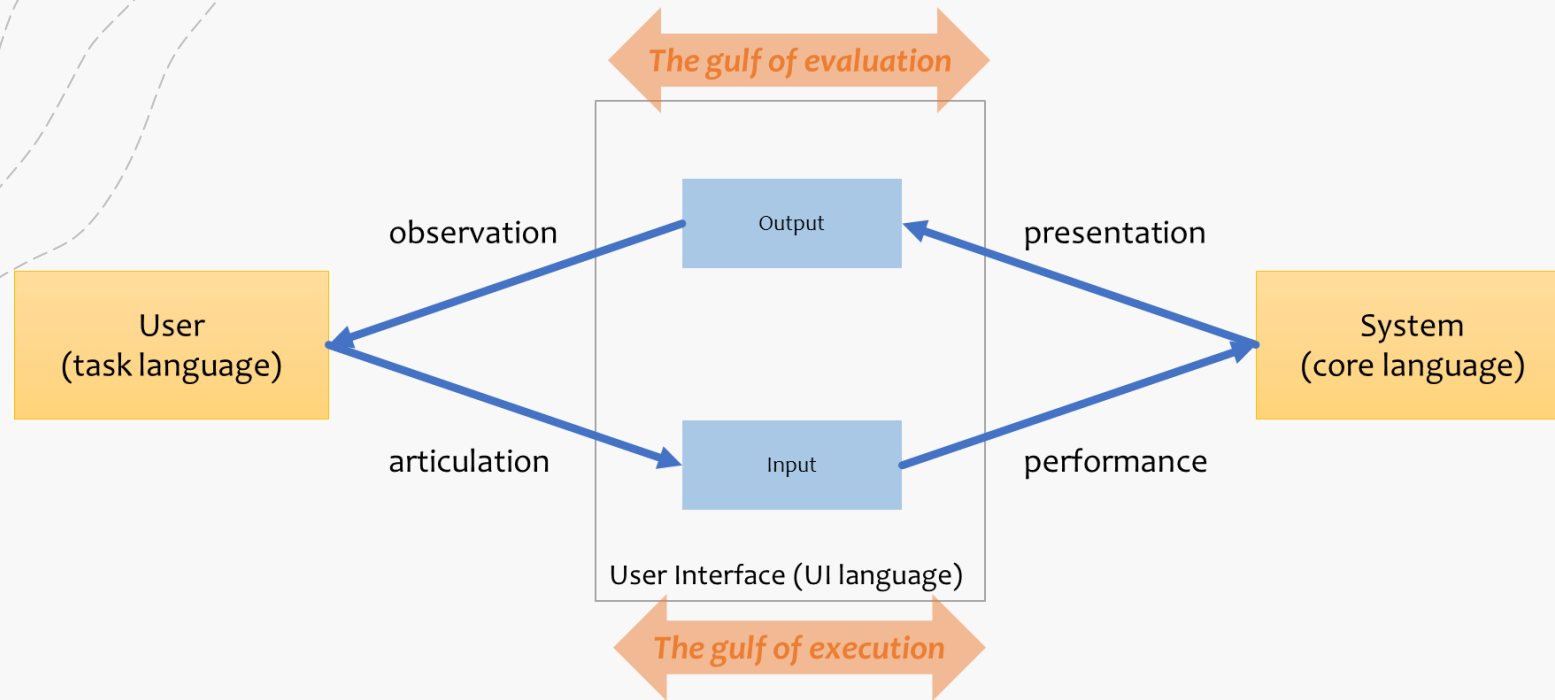
1. **Goal** (form the goal)
2. **Plan** (the action)
3. **Specify** (an action sequence)
4. **Perform** (the action sequence)
5. **Perceive** (the state of the world)
6. **Interpret** (the perception)
7. **Compare** (the outcome with the goal)

Foley and van Dam Four-level Approach (Descriptive)

- Conceptual level
  - User's mental model of the interactive system
- Semantic level
  - Describes the meanings conveyed by the user's command input and by the computer's output display
- Syntactic level
  - Defines how the units (words) that convey semantics are assembled into a complete sentence that instructs the computer to perform a certain task
- Lexical level
  - Deals with device dependencies and with the precise mechanisms by which a user specifies the syntax

Consistency Theories (Prescriptive)

- **Consistency** of nouns (objects) and verbs (actions)
  - Reduces learning time and errors
- Consistency of
  - Color
  - Layout
  - Icons
  - Fonts and Font sizes
  - Button sizes
  - …
- Inconsistencies might be used (sparingly!) for drawing attention

Design Principles

The important aspects that we need to consider when creating a design.

The "What"

Design Principles

- More practical than Theories
- More fundamental, widely applicable, and enduring than Guidelines
- Fundamental principles ($\rightarrow$ from Needfinding)
  - Determine user's skill levels
  - Identify the tasks
- 5 primary interaction styles
- 8 golden rules of interface design
- Prevent errors
- Automation and human control

Interaction Styles

- Direct manipulation

- Menu selection

- Form fill-in

- Command language

- Natural language

Advantages	Disadvantages
Direct manipulation	
Visually presents task concepts	May be hard to program
Allows easy learning	May require graphics display and pointing devices
Allows easy retention	
Allows errors to be avoided	
Encourages exploration	
Affords high subjective satisfaction	
Menu selection	
Shortens learning	Presents danger of many menus
Reduces keystrokes	May slow frequent users
Structures decision making	Consumes screen space
Permits use of dialog-management tools	Requires rapid display rate
Allows easy support of error handling	
Form fill-in	
Simplifies data entry	Consumes screen space
Requires modest training	
Gives convenient assistance	
Permits use of form-management tools	
Command language	
Flexible	Poor error handling
Appeals to "power" users	Requires substantial training and memorization
Supports user initiative	
Allows convenient creation of user-defined macros	
Natural language	
Relieves burden of learning syntax	Requires clarification dialog
	May not show context
	May require more keystrokes
	Unpredictable

Norman's Principles from Action Models

Principles of good design

- State and the action alternatives should be visible

- Should be a good conceptual model with a consistent system image

- Interface should include good mappings that reveal the relationships between stages

- User should receive continuous feedback
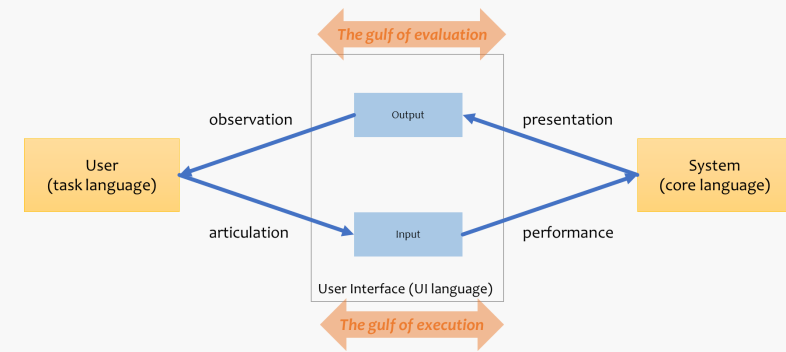
User failures can occur

- Users can form an inadequate goal

- Might not find the correct interface object because of an incomprehensible label or icon

- May not know how to specify or execute a desired action

- May receive inappropriate or misleading feedback

The 8 Golden Rules of Interface Design

- Strive for consistency

- Cater to universal usability

- Offer informative feedback

- Design dialogs to yield closure

- Prevent errors

- Permit easy reversal of actions

- Keep users in control
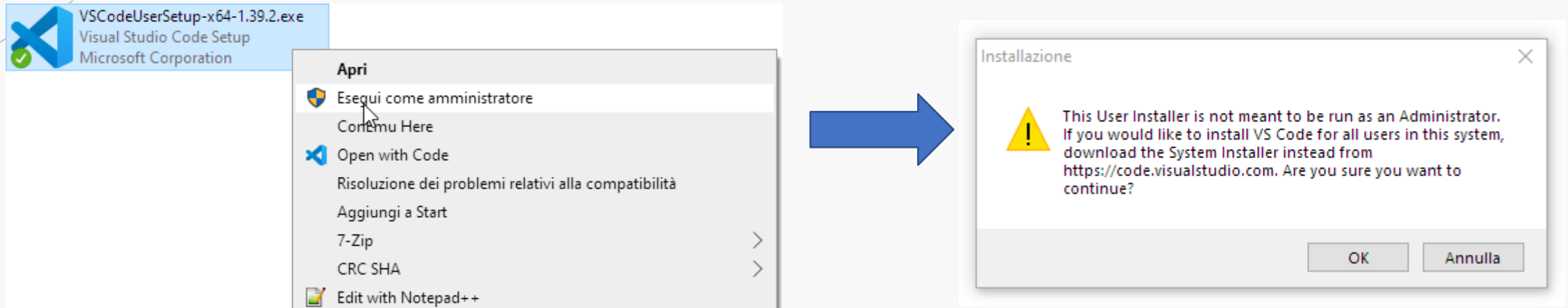
- Reduce short-term memory load

The 8 Golden Rules of Interface Design

- **Strive for consistency**
- Cater to universal usability
- Offer informative feedback
- Design dialogs to yield closure
- Prevent errors
- Permit easy reversal of actions
- Keep users in control
- Reduce short-term memory load

- Similar situations should lead to similar sequences of actions
- Same terminology in prompts, menus, help
- Color, layout, capitalization, fonts,
- …
- Exceptions should be comprehensive and limited
- E.g., delete, password echo

Consistency with mental models

Consistency of Interpretation

- Which one is the selected one?
  - Color codes are ambiguous
  - No further internal clues
  - No external clues
- Does it represent the current status?
- Does it represent the status that we want to achieve?

Inconsistency for Drawing Attention

- The border color and button text color in the "danger zone" are deliberately different than the rest of the page

The 8 Golden Rules of Interface Design

- Strive for consistency
- **Cater to universal usability**
- Offer informative feedback
- Design dialogs to yield closure
- Prevent errors
- Permit easy reversal of actions
- Keep users in control
- Reduce short-term memory load

- Users with different needs: let the interface *adapt*, let content be *transformed*
- Novices vs. experts. Young vs elderly. Web vs. mobile. Users with disabilities (→Accessibility)
- **Responsive** design
- International (and cultural) variations

The 8 Golden Rules of Interface Design

- Strive for consistency
- **Cater to universal usability**
- Offer informative feedback
- Design dialogs to yield closure
- Prevent errors
- Permit easy reversal of actions
- Keep users in control
- Reduce short-term memory load

- For *__every__* human action, there should be an interface feedback
- Frequent and minor actions: light feedback
- Infrequent and major actions: stronger feedback
- Visual presentation of objects helps showing the changes (e.g., dim, highlight, grey out, …)

Example

Example

Try to install VS Code for all users on a computer (install to Program Files rather than user's folders)
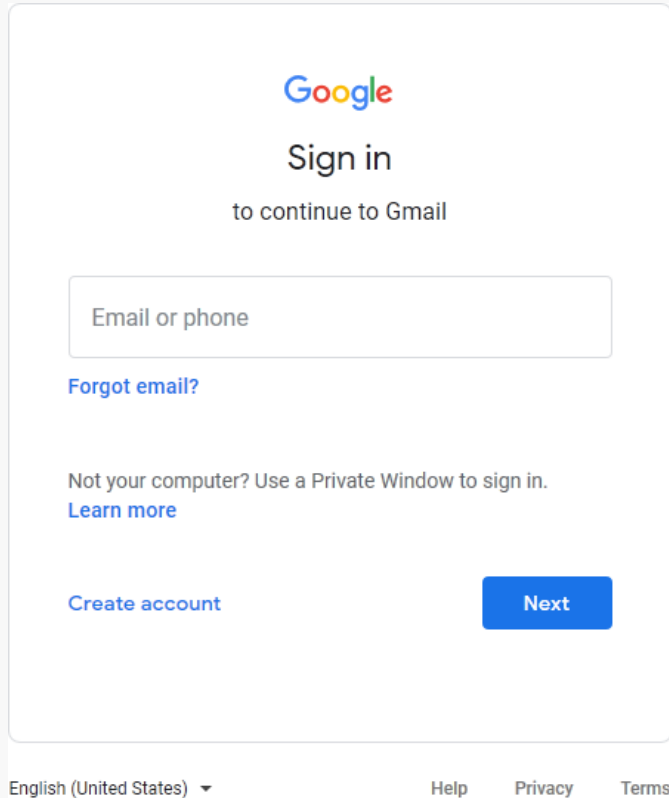
The 8 Golden Rules of Interface Design

- Strive for consistency
- Cater to universal usability
- Offer informative feedback
- **Design dialogs to yield closure**
- Prevent errors
- Permit easy reversal of actions
- Keep users in control
- Reduce short-term memory load

- Every sequence of actions should have
  - Beginning
  - Development
  - End
- Provide clear feedback at end
  - Satisfy users
  - 'Delete' current task from their working memory, prepare for the next

Clear Dialog Sequence

The 8 Golden Rules of Interface Design

- Strive for consistency
- Cater to universal usability
- Offer informative feedback
- Design dialogs to yield closure
- **Prevent errors**
- Permit easy reversal of actions
- Keep users in control
- Reduce short-term memory load

- Avoid the possibility of making errors
- Disable menu items, buttons, links, … that are not applicable
- Prevent entering illegal characters
- Offer simple, constructive and specific instructions for recovery
- Repair only the faulty part
- Errors should not alter application state (or make it easy to restore)

Error Prevention

The 8 Golden Rules of Interface Design

- Strive for consistency
- Cater to universal usability
- Offer informative feedback
- Design dialogs to yield closure
- Prevent errors
- **Permit easy reversal of actions**
- Keep users in control
- Reduce short-term memory load

- Actions should be reversible (at the cost of extra development effort)
  - Relieves anxiety
  - Encourages exploration
- Different levels of reversibility
  - A single action
  - A data-entry task
  - A complete group of actions

The 8 Golden Rules of Interface Design

- Strive for consistency
- Cater to universal usability
- Offer informative feedback
- Design dialogs to yield closure
- Prevent errors
- Permit easy reversal of actions
- **Keep users in control**
- Reduce short-term memory load

- The interface should *always* respond to user actions
- Minimize the tedious and lengthy tasks
- Avoid surprises or changes in familiar behavior
- Provide undo/redo, cancel/confirm

The 8 Golden Rules of Interface Design

- Strive for consistency
- Cater to universal usability
- Offer informative feedback
- Design dialogs to yield closure
- Prevent errors
- Permit easy reversal of actions
- Keep users in control
- **Reduce short-term memory load**

- Rule of thumb:
  - People can remember 7±2 chunks of information
- Information on a screen should not be needed (remembered) in the next screen
- No entry of phone numbers (collect from addressbook), show website location, fit long forms in a single page, …

Discussion – An Exception?
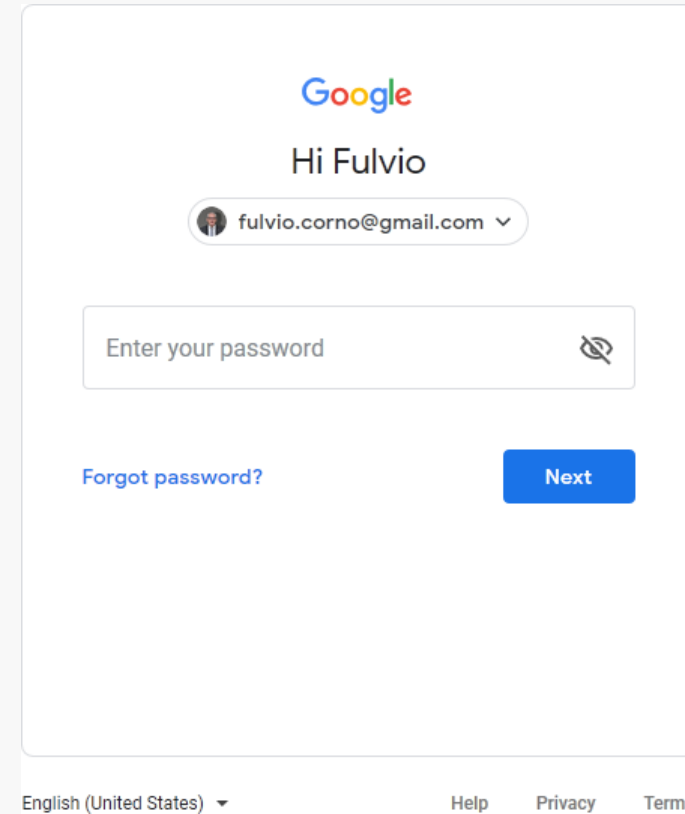
Exceptions...
sometimes entering is better than selecting
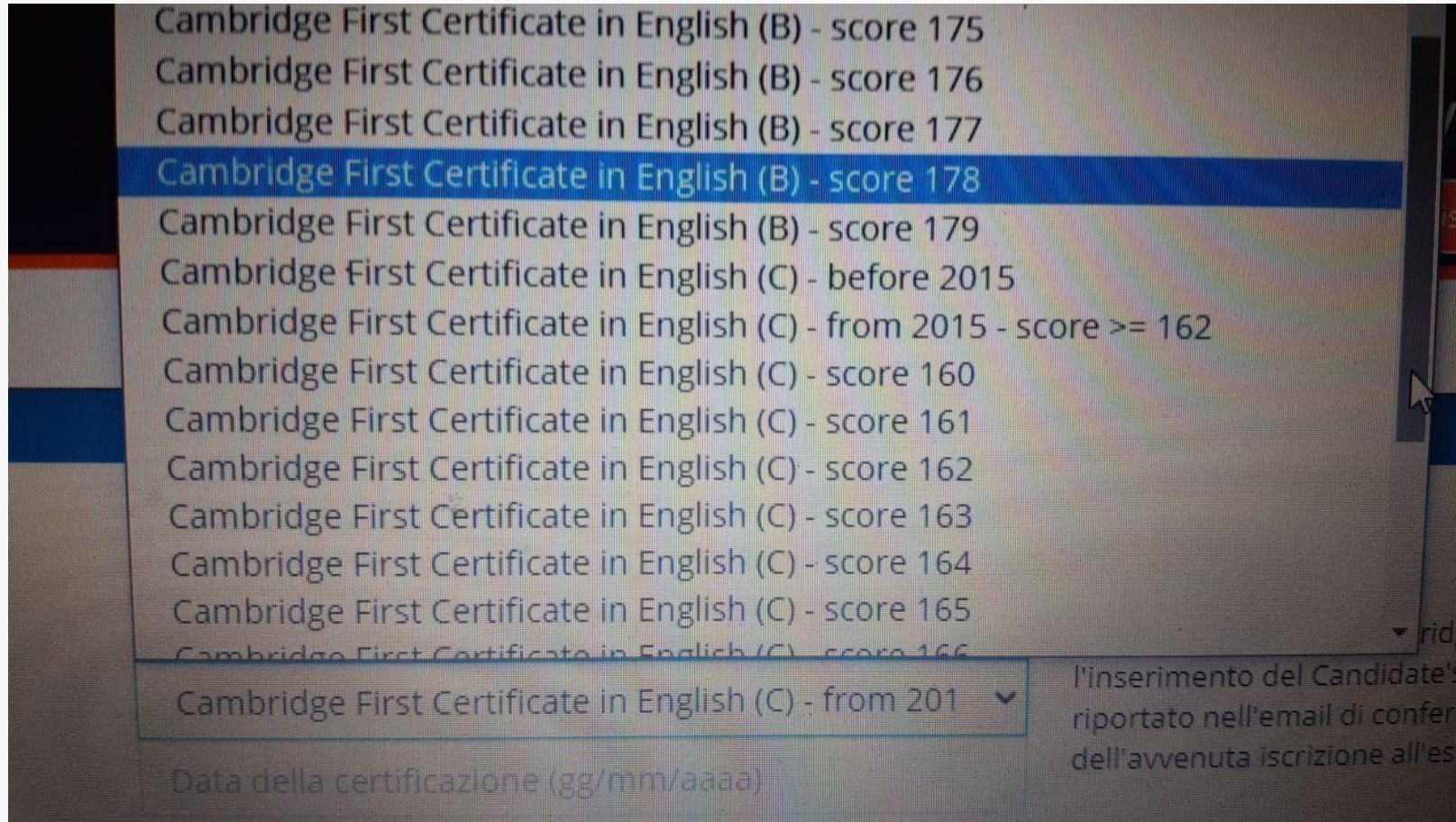
Design Principles by Benyon (I) (adapted from Norman, Nielsen and others)

Learnability – helping people access, learn and remember the system

- *Visibility* – ensure that things are visible, so users can see what functions are available and what the system is currently doing

- *Consistency* (→above)

- *Familiarity* – use language and symbols that the intended audience will be familiar with

- *Affordance* – design things so it is clear what they are for (e.g., buttons should be pushed). Maps the (perceived) properties of the objects with how they can be used

Design Principles by Benyon (II) (adapted from Norman, Nielsen and others)

Effectiveness – giving users the sense of being in control, knowing what to do and how to do it

- *Navigation* – support people in moving around the different sections: maps, directional signs, information signs

- *Control* – who is in control for the next interaction? Clear and logical mapping between controls and their effect. Relationships with the "side effects" in the real world

- *Feedback* (→feedback above)

Design Principles by Benyon (III) (adapted from Norman, Nielsen and others)

- **Safety and Security**
  - *Recovery* (→error recovery)
  - *Constraints* (→prevent errors)
- **Accommodation** – offer an interaction way that suits the users
  - *Flexibility* (→universal usability)
  - *Style* – stylish, attractive, nice-looking
  - *Conviviality* – polite, friendly, pleasant. No abrupt interruptions

Norman's Seven Principles for Transforming Difficult Tasks into Simple Ones

- Use both knowledge in the world and knowledge in the head
- Simplify the structure of tasks
- Make things visible
- Get the mappings right
- Exploit the power of constraints, both natural and artificial
- Design for error
- When all else fails, standardize

D. Norman, The Design of Everyday Things

First Principles of Interaction Design (Bruce Tognazzini, 2014)
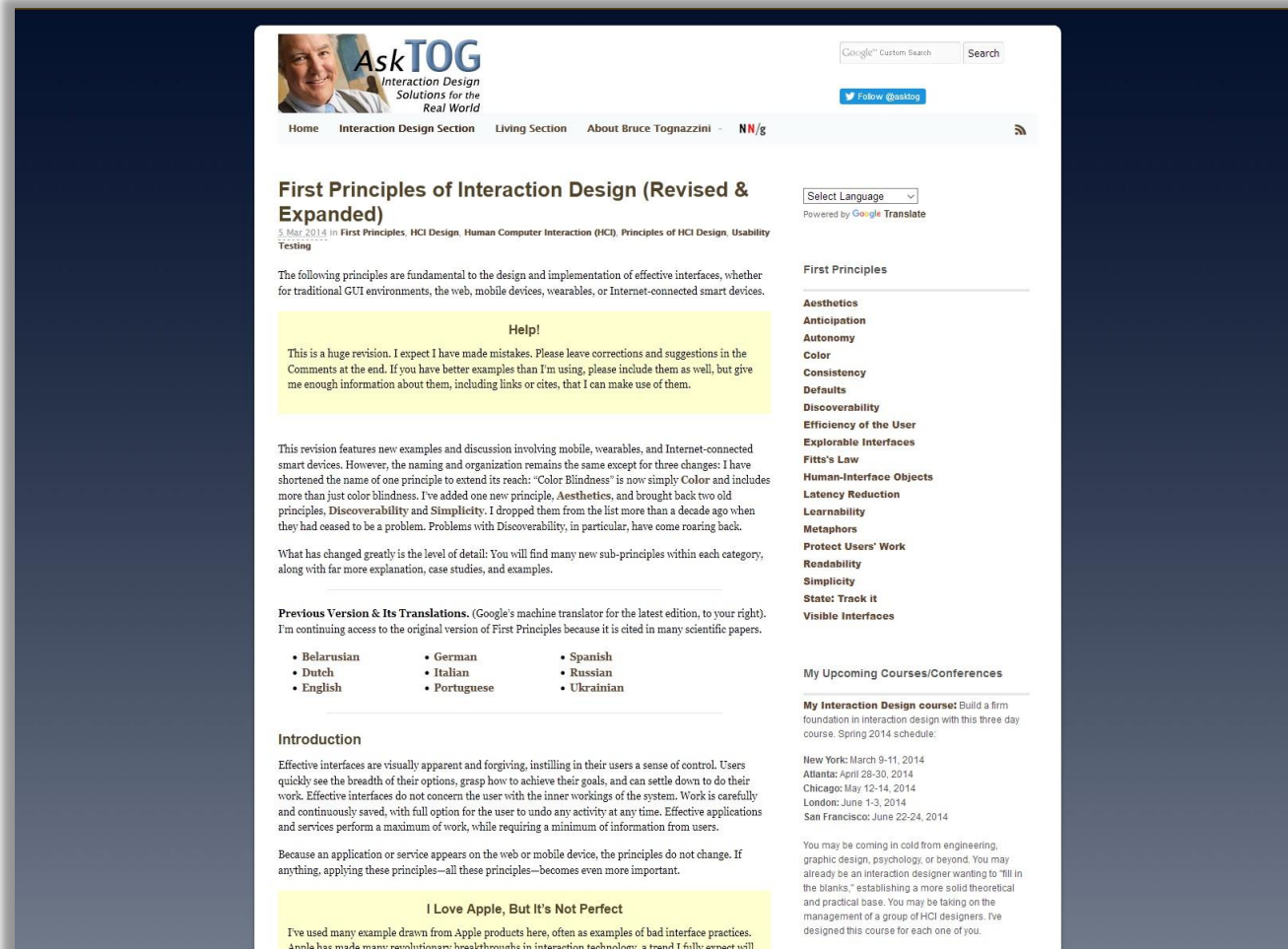
Aesthetics
Anticipation
Autonomy Color
Consistency
Defaults
Discoverability
Efficiency of the User
Explorable Interfaces
Fitts's Law
Human-Interface Objects
Latency Reduction
Learnability
Metaphors
Protect Users' Work
Readability
Simplicity State:
Track it
Visible Interfaces

Design Guidelines

Shared language to promote consistency among multiple designers in terminology usage, appearance, and action sequences

The "How"

Design Guidelines

- Concrete suggestions about "How" the Principles may be satisfied
- Often rule-based
- Based on best practices
- Encapsulate experience of expert designers
- Sometimes blessed as «standards»
- But:
  - May be too specific and hard to apply to your situation
  - Difficult to develop a general-purpose guideline

Web Style Guide

Web Style Guide by Patrick J. Lynch and Sarah Horton

Contents

About the authors

Patrick J. Lynch and Sarah Horton have been working together on award-winning interface and graphic design projects since 1991. They began collaborating on *Web Style Guide* in 1997, moving from a web-only version to print and web in 1999. The book is in its 4th edition and has been translated into more than eight languages.

- Learn more about Pat and Sarah
- *Web Style Guide, 4th Edition: Foundations of User Experience Design* on Amazon

Praise for the 4th Edition of Web Style Guide

Contents   Search   Front Matter

Web Content Accessibility Guidelines (WCAG)

https://www.w3.org/WAI/standards-guidelines/wcag/

U.S. Government Mobile User Experience Guidelines

U.K. Government Design System

https://design-system.service.gov.uk

Italian Government Guidelines and Design System

Apple HIG

Microsoft «Fluent» Design
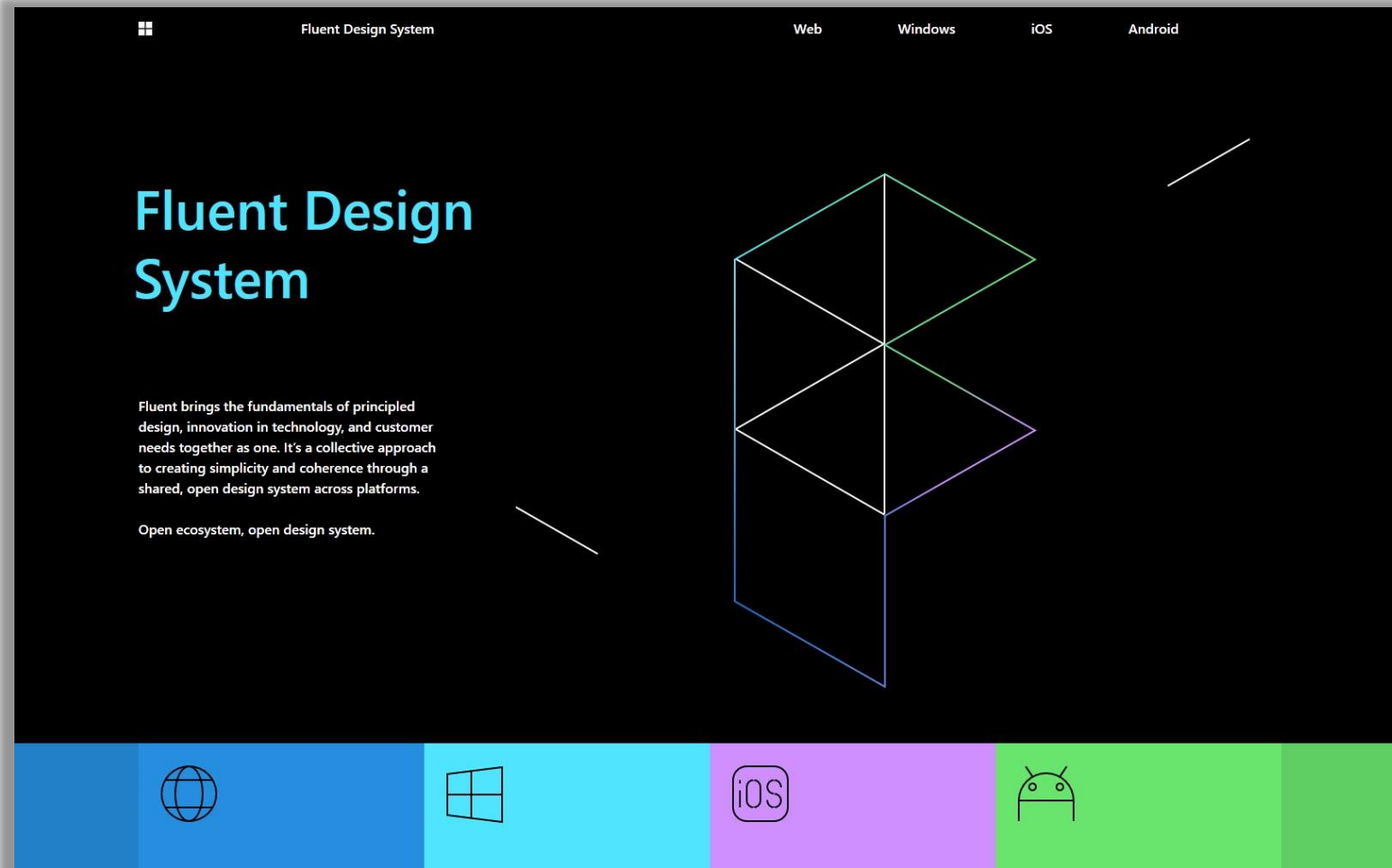
Google Material Design
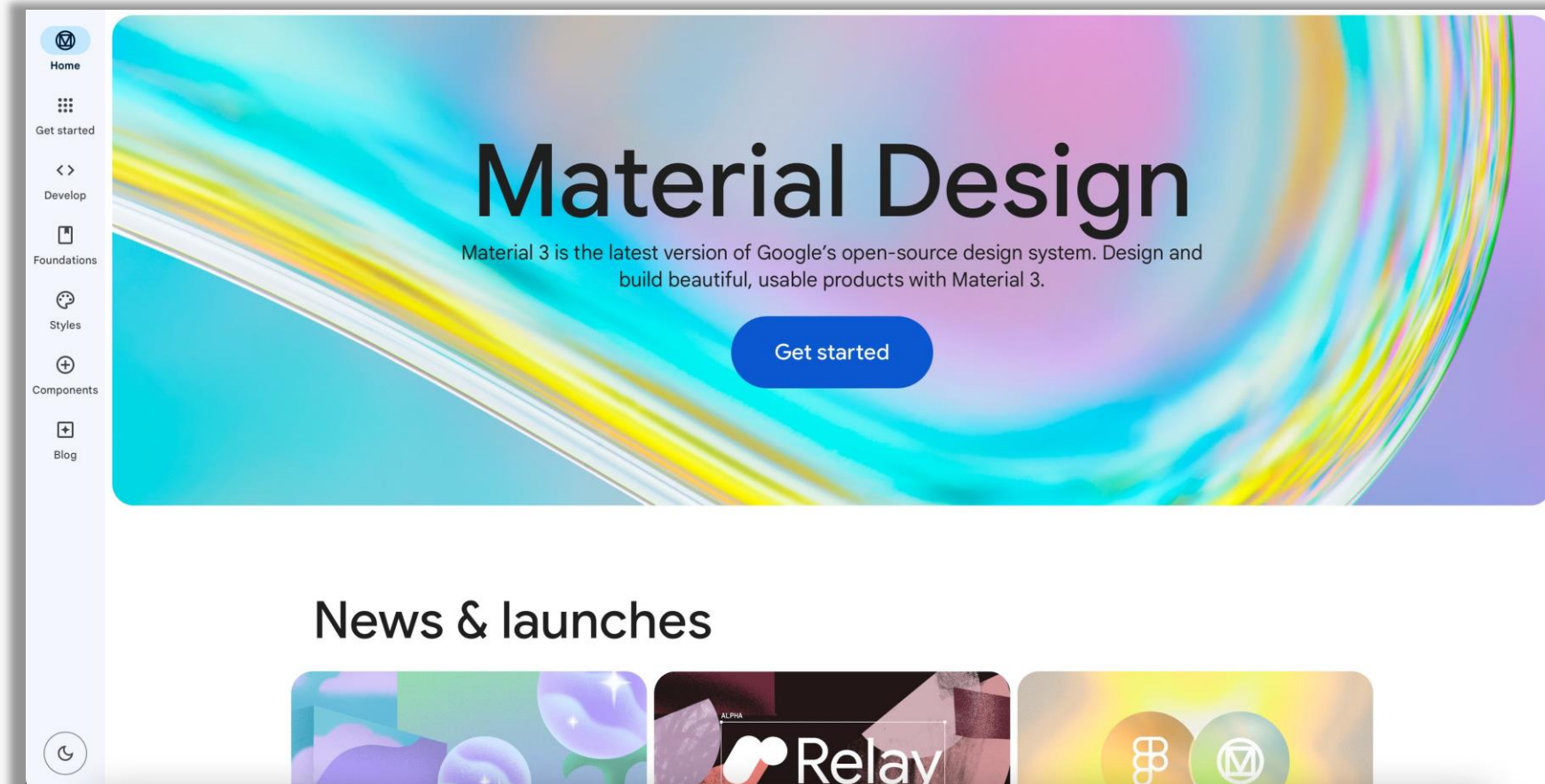
Guidelines for Human-AI Interaction

- By Microsoft Research:
  - https://www.microsoft.com/en- us/research/project/guidelines-for-human-ai-interaction/
  - https://www.Microsoft.com/en- us/haxtoolkit/ai-guidelines/
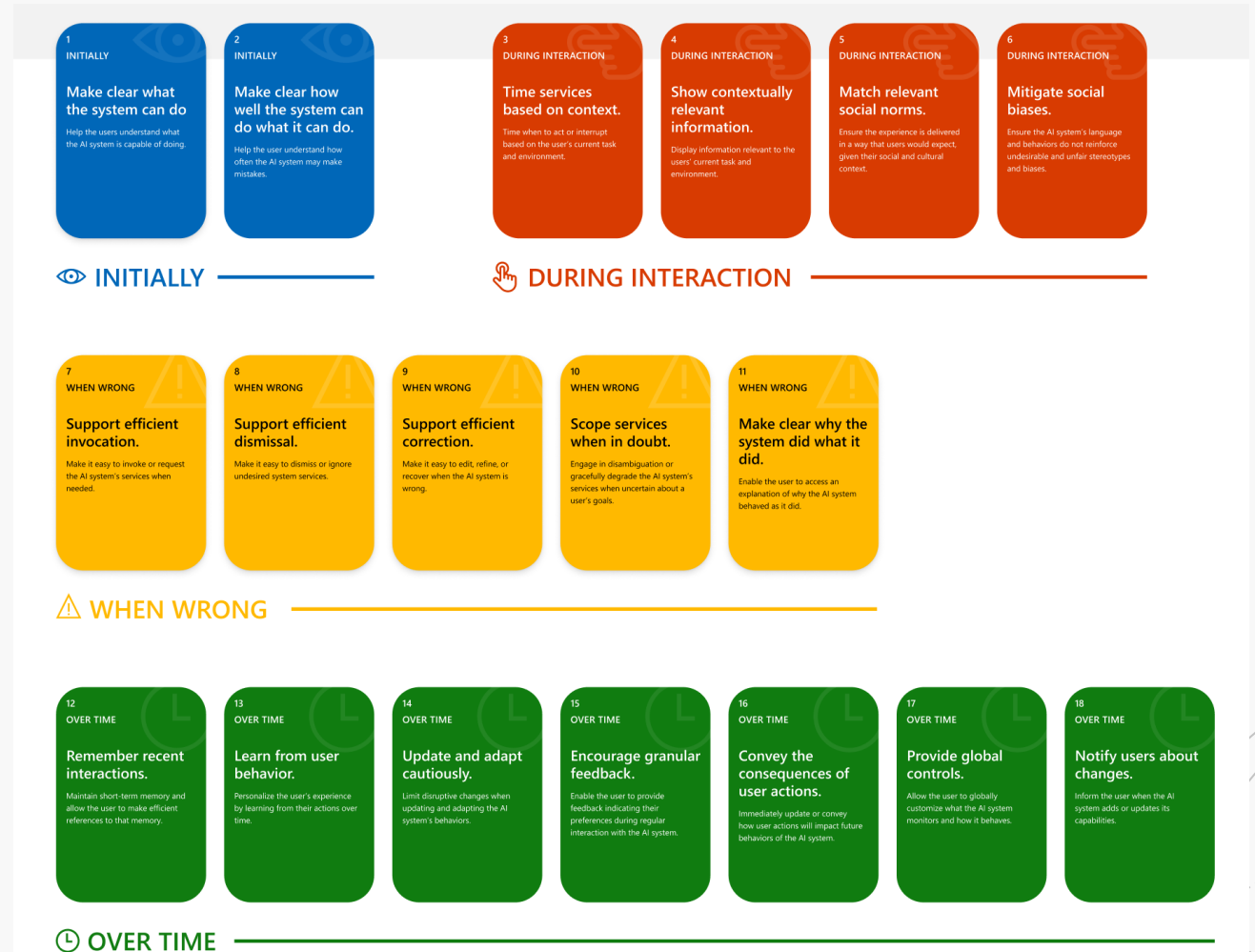
Guidelines for Human-AI Interaction: Examples

2
INITIALLY

Make clear how well the system can do what it can do.

Help the user understand how often the AI system may make mistakes.

EXAMPLE IN PRACTICE

Discover new music from artists we think you'll like. Refreshed every Friday.

▶ Play   ⇄ Shuffle
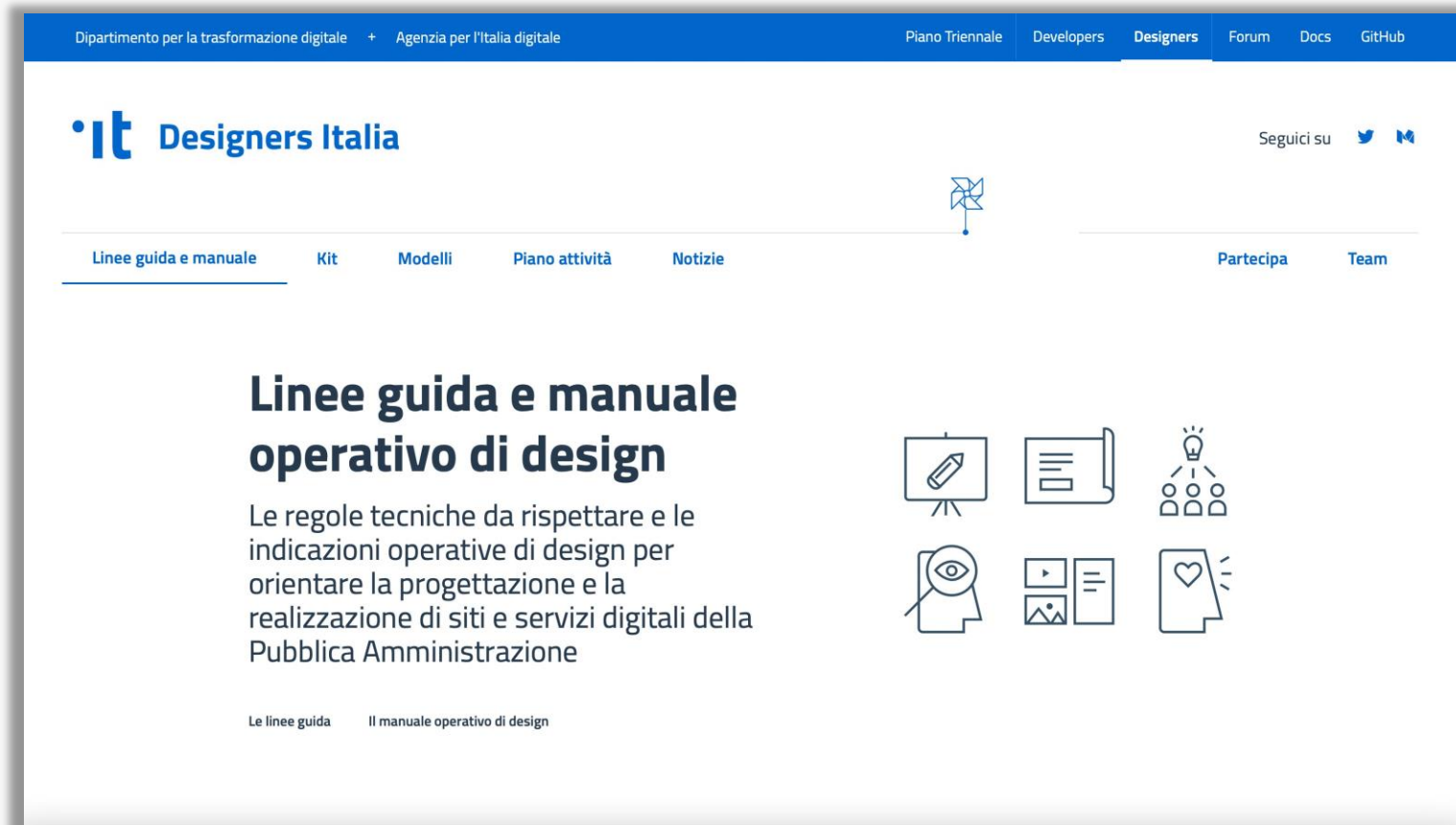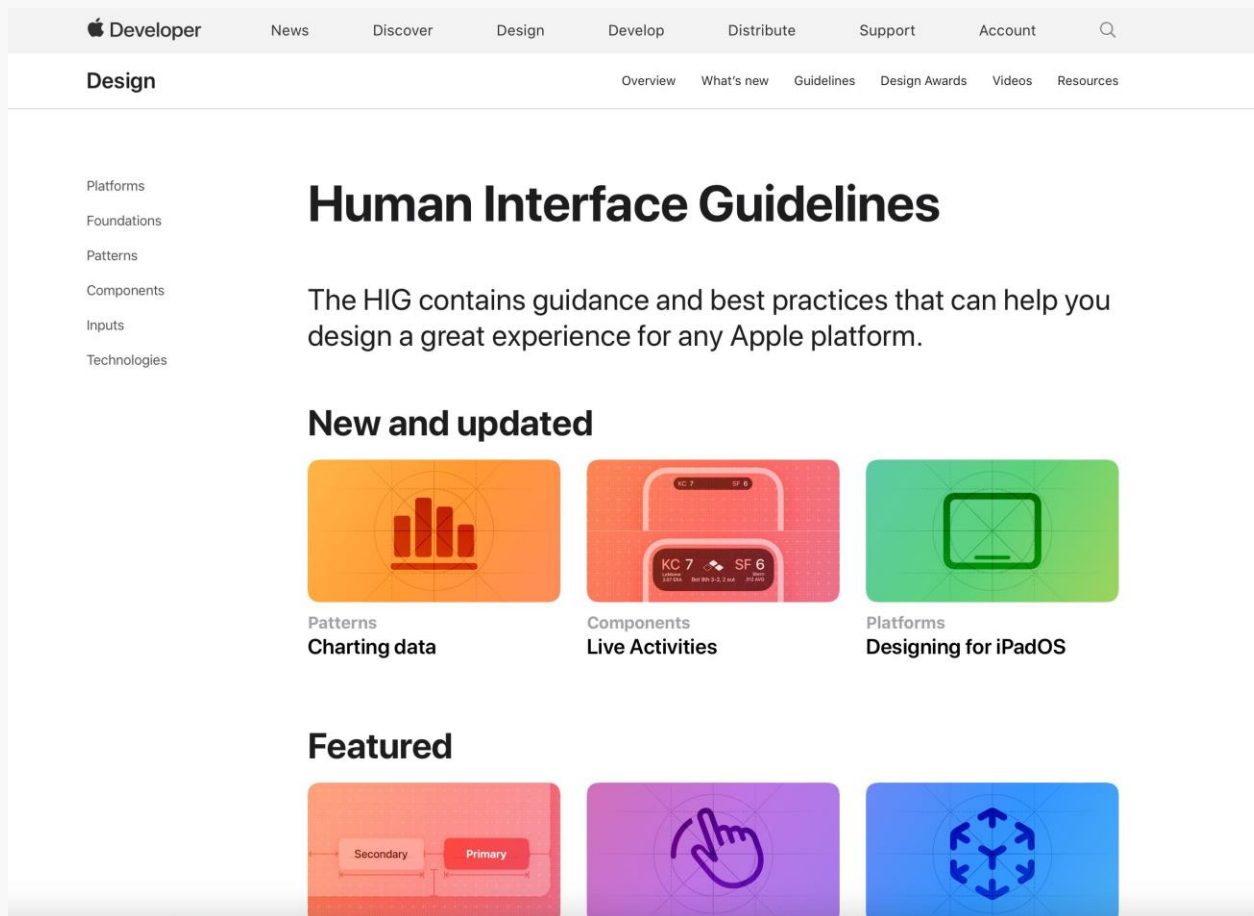
Never Not
Lauv   +

Forget to Forget   +
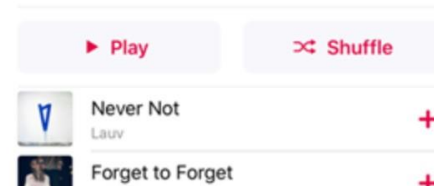
The recommender in **Apple Music** uses language such as "we think you'll like" to communicate uncertainty.

Make clear how well the system can do what it can do.   2

9
WHEN WRONG

Support efficient correction.

Make it easy to edit, refine, or recover when the AI system is wrong.

EXAMPLE IN PRACTICE

All   Images   Videos   Maps

757,000 Results   Any time ▾

Including results for keanu **reeves**.
Do you want results only for keanu reaves?

When **Bing** automatically corrects spelling errors in search queries, it provides the option to revert to the query as originally typed with one click.

Support efficient correction.   9

Guidelines for Augmented Reality

- By Apple Design: [https://developer.apple.com/design/human- interface-guidelines/technologies/augmented-reality/](https://developer.apple.com/design/human-interface-guidelines/technologies/augmented-reality/)

Creating an engaging, comfortable experience

Let people use the entire display. Devote as much of the screen as possible to displaying the physical world and your app's virtual objects. Avoid cluttering the screen with controls and information that diminish the immersive experience.

Strive for convincing illusions when placing realistic objects. Design detailed 3D assets with lifelike textures to create objects that appear to inhabit the physical environment in which you place them. Using information from ARKit, you can scale objects properly and position them on detected real-world surfaces, reflect environmental lighting conditions and simulate camera grain, cast top-down diffuse object shadows on real-world surfaces, and update visuals as the camera's position changes. To help avoid breaking the illusion you create, make sure your app updates scenes 60 times per second so objects don't appear to jump or flicker.

Consider how virtual objects with reflective surfaces show the environment. Reflections in ARKit are approximations based on the environment captured by the camera. To help maintain the illusion that an AR experience is real, prefer small or coarse reflective surfaces that downplay the effect of these approximations.

Use audio and haptics to enhance the immersive experience. A sound effect or bump sensation is a great way to confirm that a virtual object has made contact with a physical surface or other virtual object. Background music can also help envelop people in the virtual world. For guidance, see Playing audio and Playing haptics.

Minimize text in the environment. Display only the information that people need for your app experience.

References and Acknowledgments

- Ben Shneiderman, Catherine Plaisant, Maxine S. Cohen, Steven M. Jacobs, and Niklas Elmqvist, Designing the User Interface: Strategies for Effective Human-

- Computer Interaction

- Chapter 3: Guidelines, Principles, and Theories

- David Benyon: Designing Interactive Systems, Pearson, 2014

- Section 4.5: Design Principles

- COGS120/CSE170: Human-Computer Interaction Design, videos by Scott Klemmer, https://www.youtube.com/playlist?list=PLLssT5z_DsK_nusHL_Mjt87THSTIgrsyJ

License

■ These slides are distributed under a Creative Commons license "**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**"

■ **You are free to:**
  o **Share** – copy and redistribute the material in any medium or format
  o **Adapt** – remix, transform, and build upon the material
  o The licensor cannot revoke these freedoms as long as you follow the license terms.

■ **Under the following terms:**
  o **Attribution** – You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  o **NonCommercial** – You may not use the material for commercial purposes.
  o **ShareAlike** – If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  o **No additional restrictions** – You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

■ https://creativecommons.org/licenses/by-nc-sa/4.0/

Vietnam National University of HCMC

International University

School of Computer Science and Engineering

THANK YOU

Dr Vi Chi Thanh - vcthanh@hcmiu.edu.vn

https://vichithanh.github.io

SCAN ME