What is HCI?

Human Computer Interaction

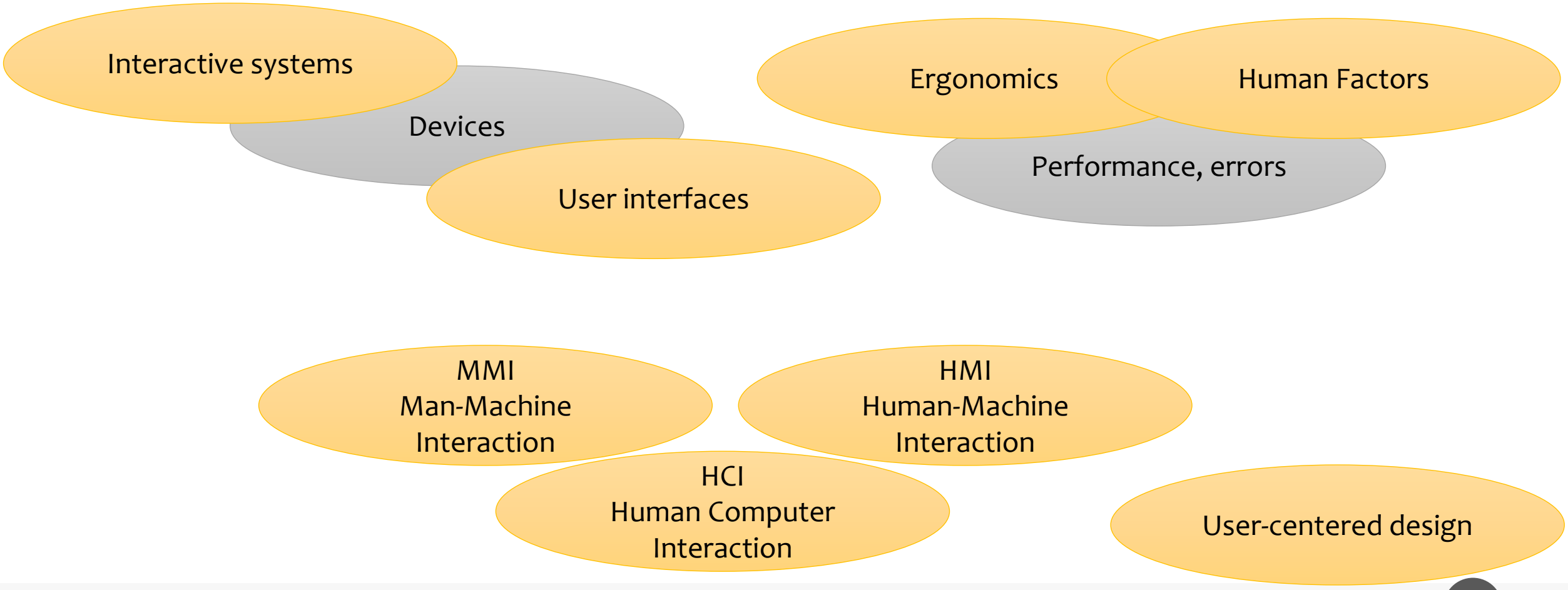
Fulvio Corno, Luigi De Russis

Academic Year 2020/2021

Goals

- What is HCI?
- What is usability?
- What is the Interaction Design Process, and how does it relate with Software Engineering processes?
- What is meant by User Centered Design?

Interconnected concepts, and evolution



The goal of HCI

Ingredients

- The **User**(s)
- The **Computer**(s)
- The **Task**(s) to be accomplished

Goal

- The system must support the user's **task,** with a focus on its **usability**
  - Useful
  - Usable
  - Used

The ingredients

The human

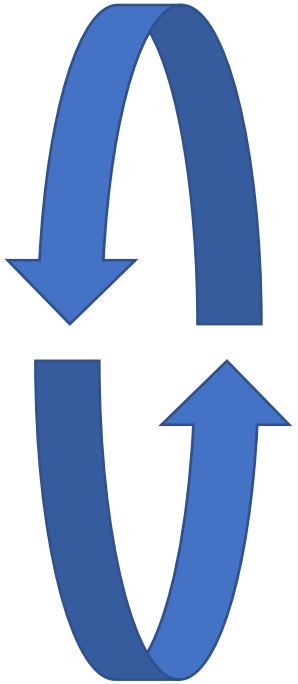
- Sensory systems
  - Visual
  - Auditory
  - Haptic
  - Spatial
- Acting systems
  - Hands
  - Voice
  - Head, Body, …
- Cognitive processes
  - Perception
  - Memory

The computer

- Input peripherals
  - Keyboard, mouse
  - Trackpad, trackball
  - Touch surfaces or screens
  - Microphone
  - Sensors
  - Card readers
  - …
- Output peripherals
  - Screen
  - Audio (voice, sounds)
  - Haptics
  - VR/AR headsets
  - …

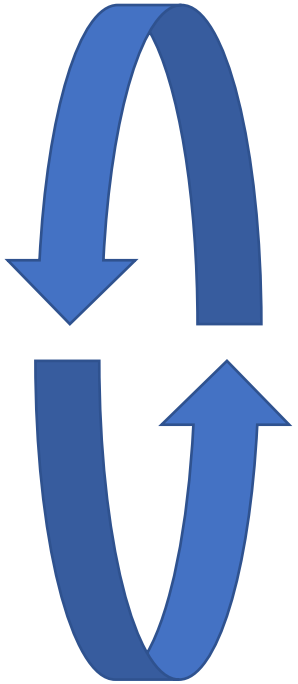
HCI is multidisciplinary

- Psychology and cognitive science
  - User perceptual, cognitive and problem-solving skills
- Ergonomics
  - User's physical capabilities
- Sociology
  - Understanding the wider context of the interaction
- **Computer Science and Computer Engineering**
  - **Building the necessary artifacts (HW, SW)**
- Business
  - Satisfying market needs
- Graphic design
  - Produce an effective interface presentation
- Technical writing
  - Documentation, manuals, on-screen content
- …



HCI is multidisciplinary

- Psychology and cognitive science
  - User perceptual, cognitive and problem-solving skills
- Ergonomics
  - User's physical capabilities
- Sociology
  - Understanding the wider context of the interaction
- **Computer Science and Computer Engineering**
  - **Building the necessary artifacts (HW, SW)**
- Business
  - Satisfying market needs
- Graphic design
  - Produce an effective interface presentation
- Technical writing
  - Documentation, manuals, on-screen content
- …



To help us in applying expertise from many different fields:

- Design methods and processes
- Models
- Heuristics
- Best practices
- Conventions
- Experiments and user studies

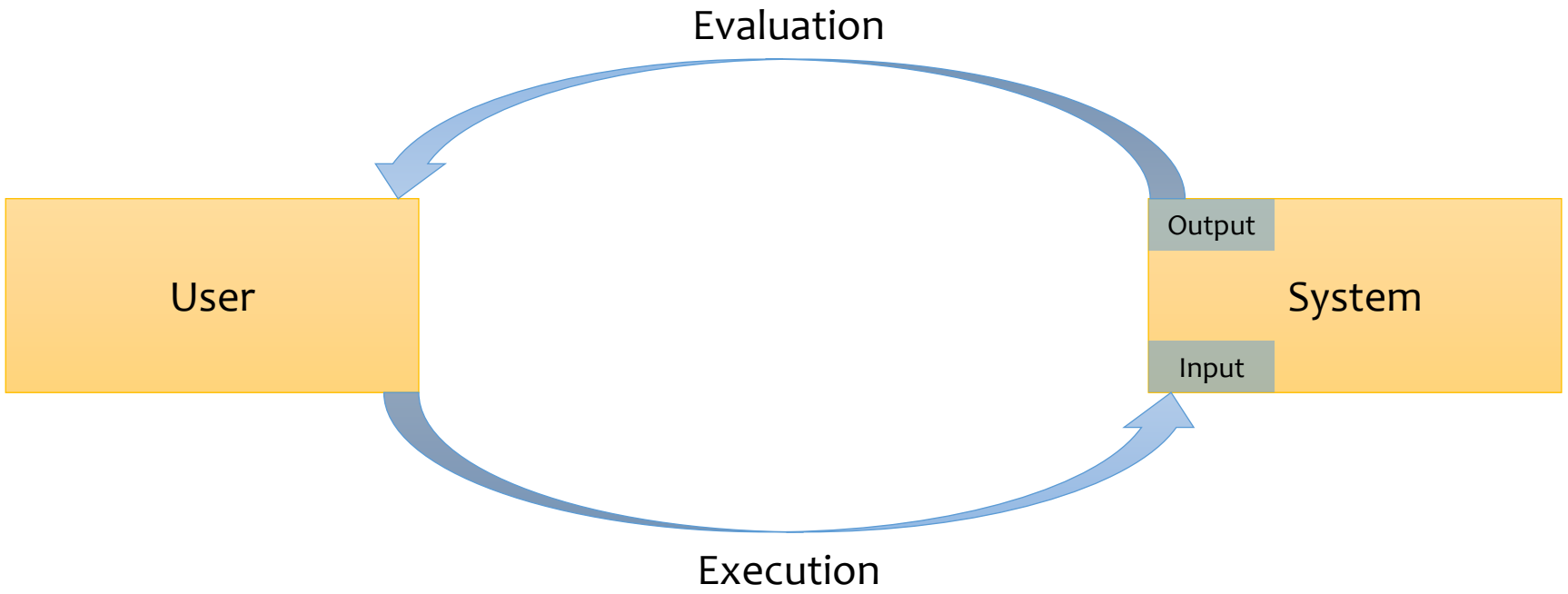
Models of interaction

A general framework to understand how User and System interact

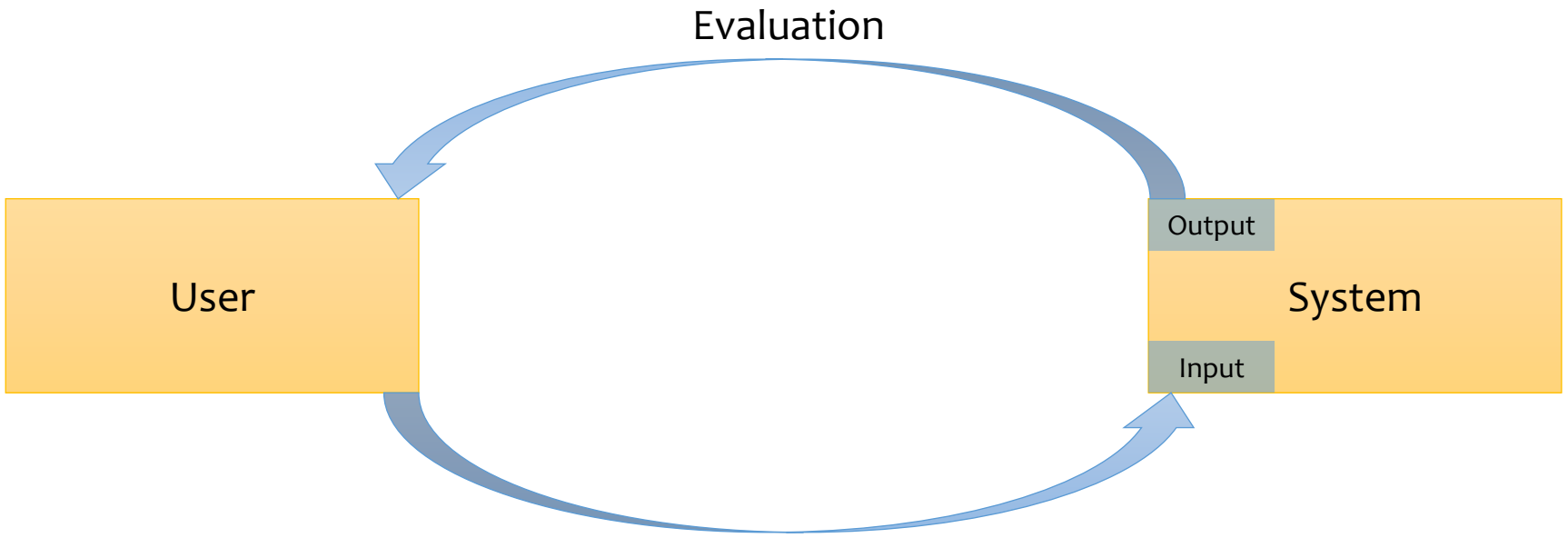
Assumptions

- The **user** wants to accomplish some **goals**, in a specific application **domain**
  - Each domain has a specific jargon, set of possible processes and goals, artifacts and building blocks, …
- **Tasks** are operations to manipulate the concepts of a domain
  - The goal is attained by performing one or more tasks
- Interaction studies the relation between User and System
  - The system possesses a **state** and "speaks" a **core language**
  - The user possesses a **state**, that includes an **understanding** of the system's state, some **intention** to perform a task, and "speaks" the **task language**

Norman's model of interaction

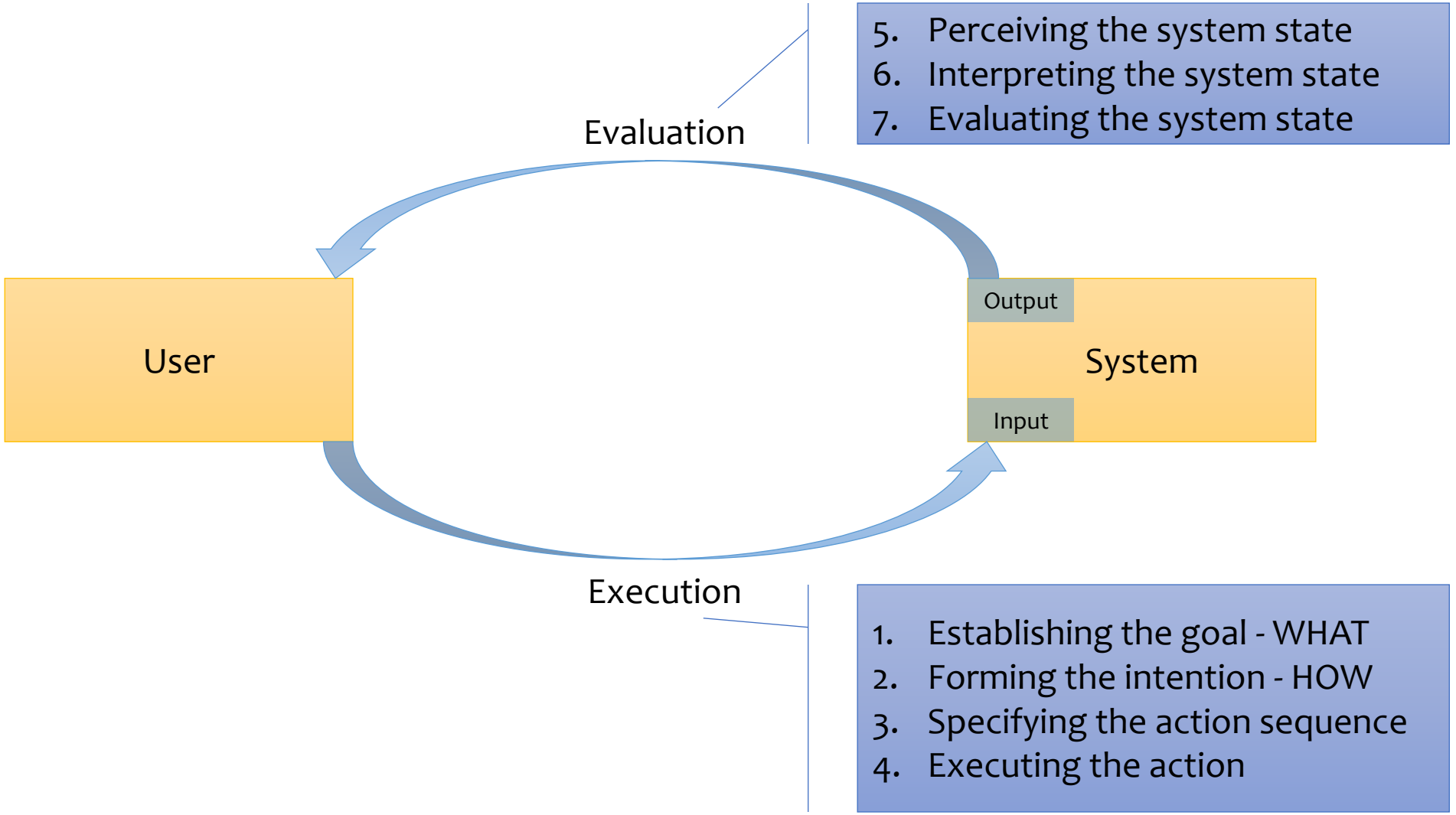


Norman's model of interaction

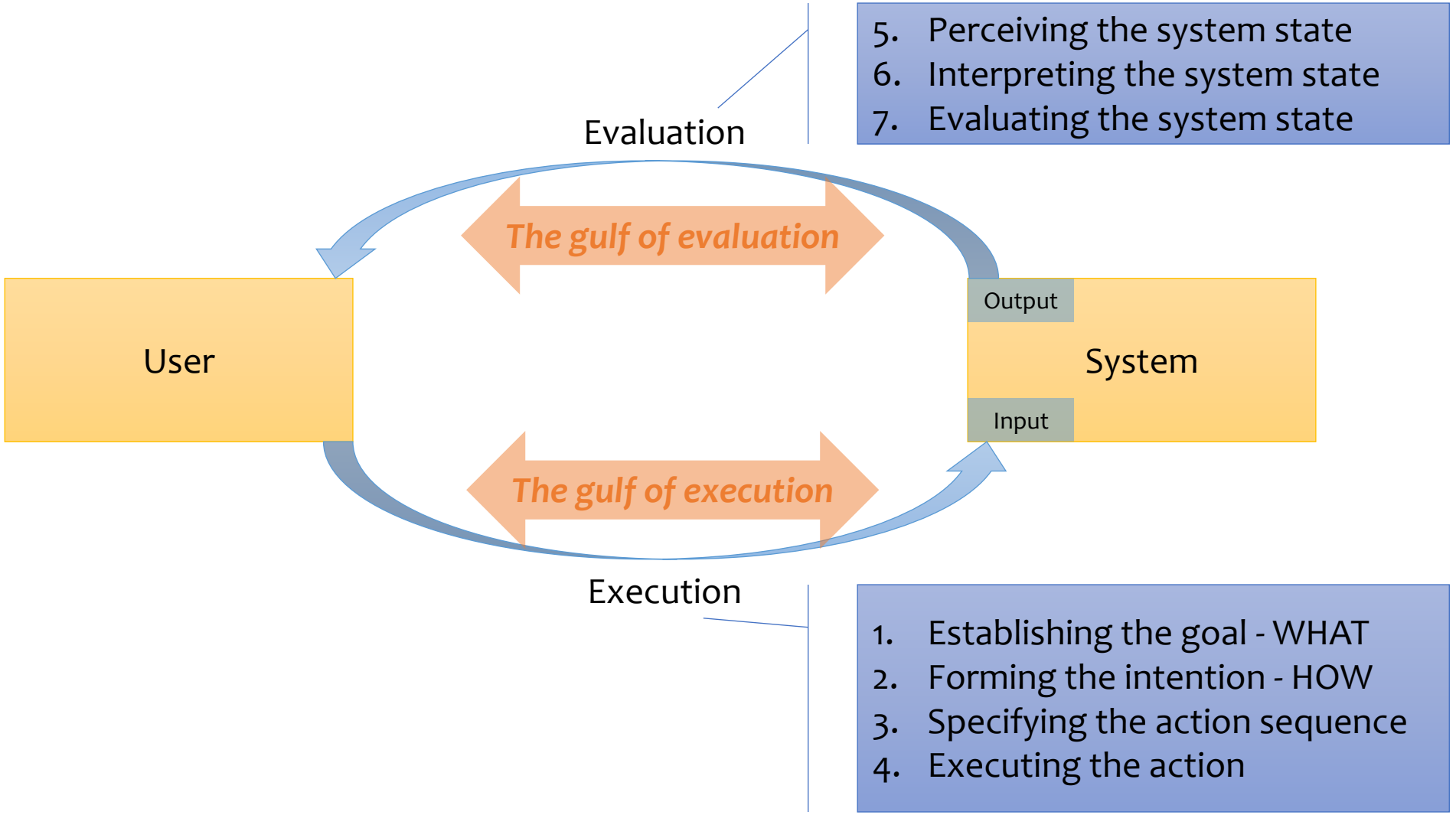


1. Establishing the goal - WHAT
2. Forming the intention - HOW
3. Specifying the action sequence
4. Executing the action

Norman's model of interaction

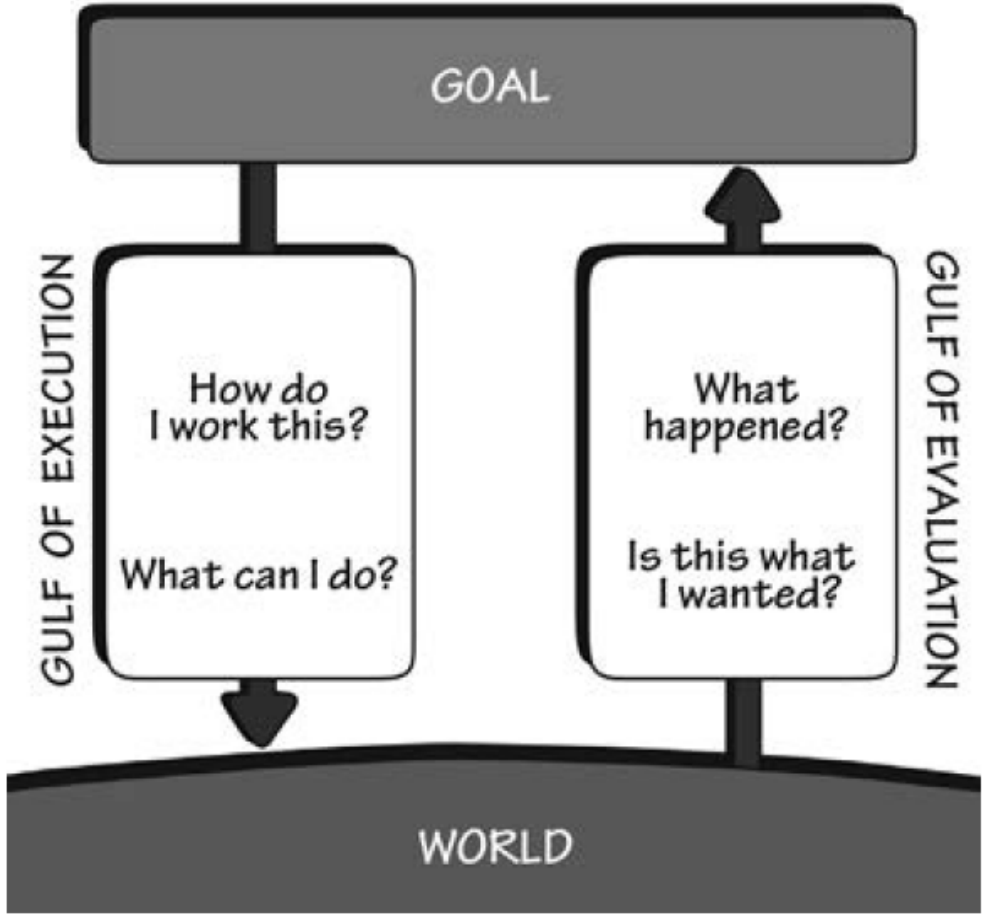


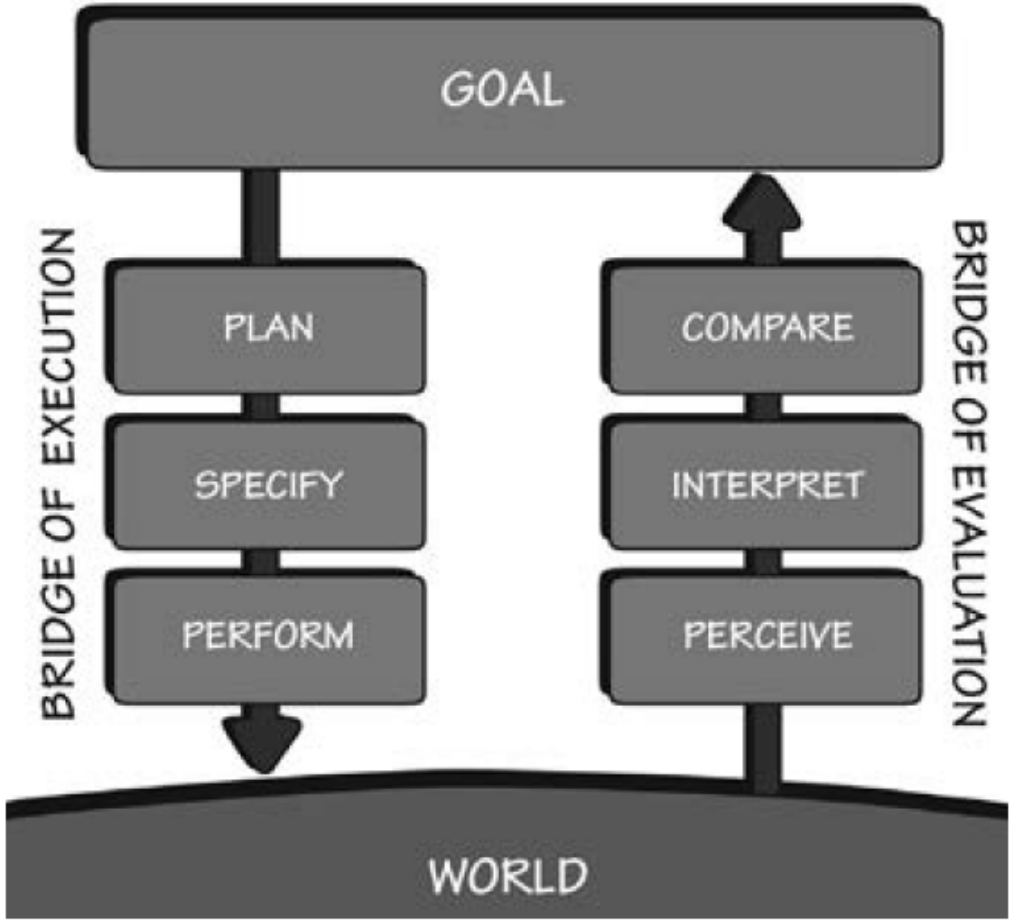
Norman's model of interaction



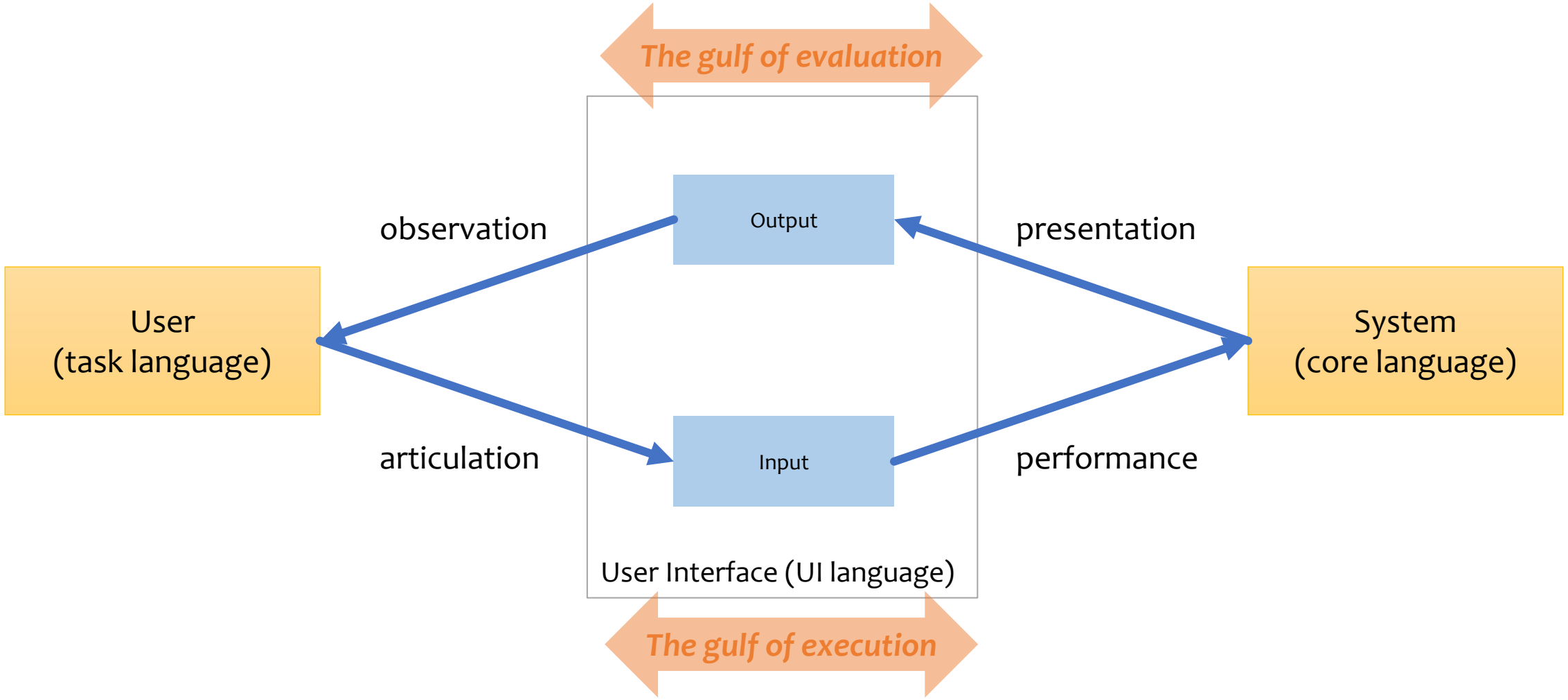
Norman's diagrams

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)





Abowd and Beale model, with explicit UI



Human errors*
in the gulf of execution

Slip

- You have formulated the right action, but fail to execute that action correctly
  - E.g., click the wrong icon, or double-click too slow, …
- May be corrected by a better interface (spacing, layout, highlights, …)

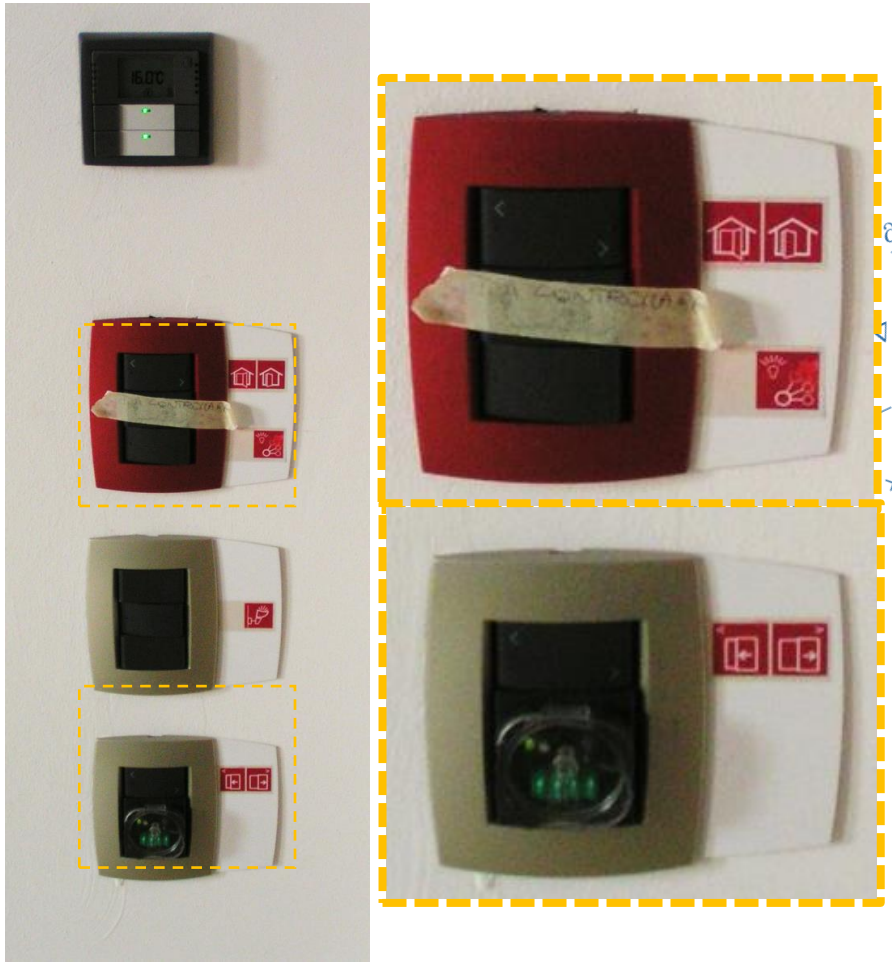
Mistake

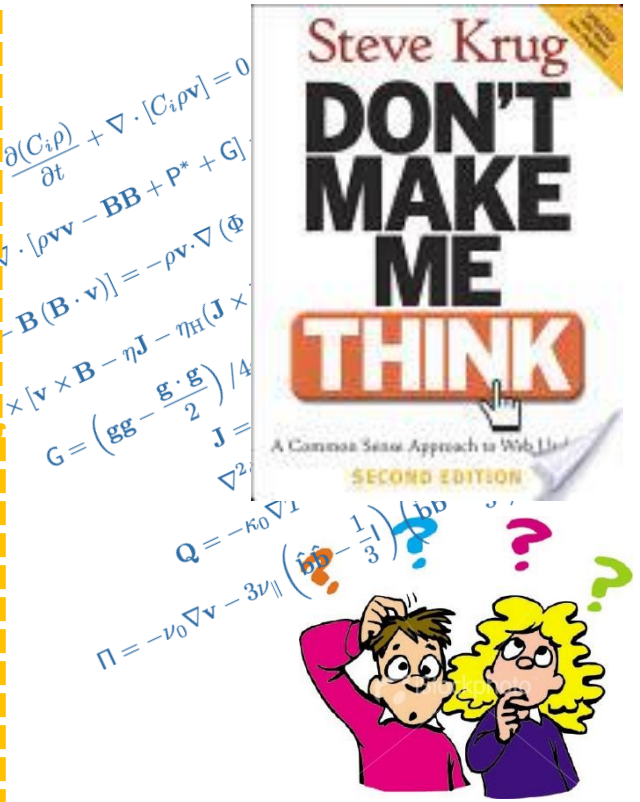
- You don’t know the system well and you may not formulate the right goal
  - E.g., click  for Zoom, but it means Search
- The user’s mental model of the system’s state is not correct
- Requires more radical redesign, or additional training

* About Human errors

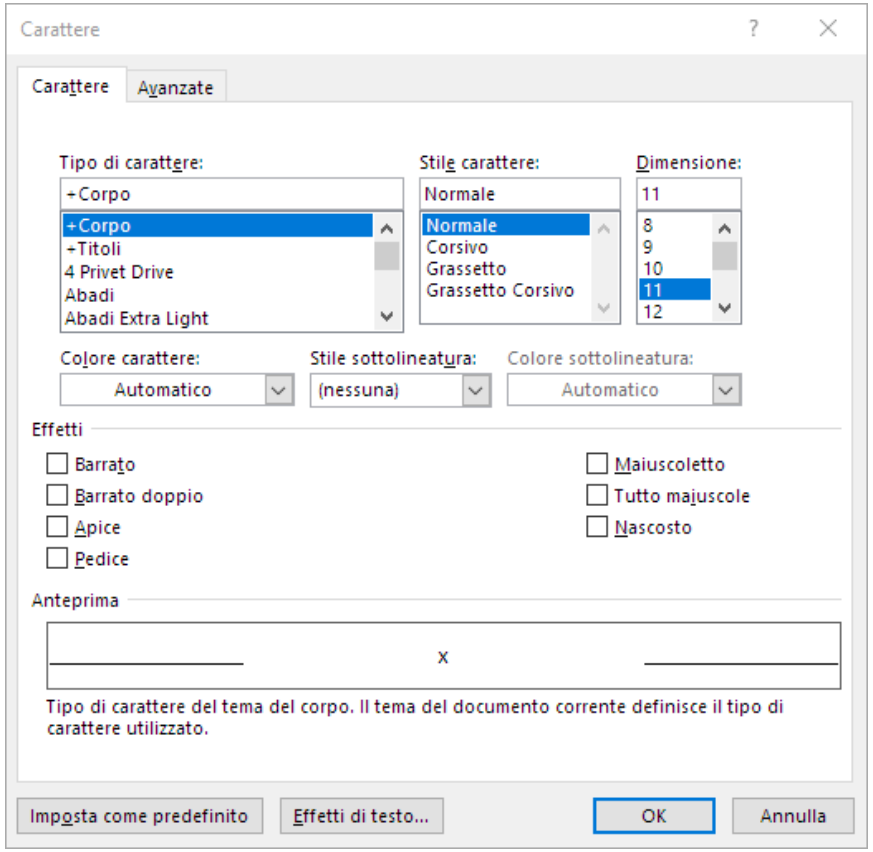
- Human errors should **never** be considered as faults of the user
- Rather, «they are usually a result of bad design» (Norman)
- Humans tend to be imprecise, distracted, not-omniscient
  - System design should anticipate this human behavior
  - Minimize the chance of inappropriate actions (evaluation)
  - Maximize the possibility of discovering and repairing an inappropriate action (execution)
  - Enable users to understand the state of the system and build an appropriate model

Example (articulation): find the right switch

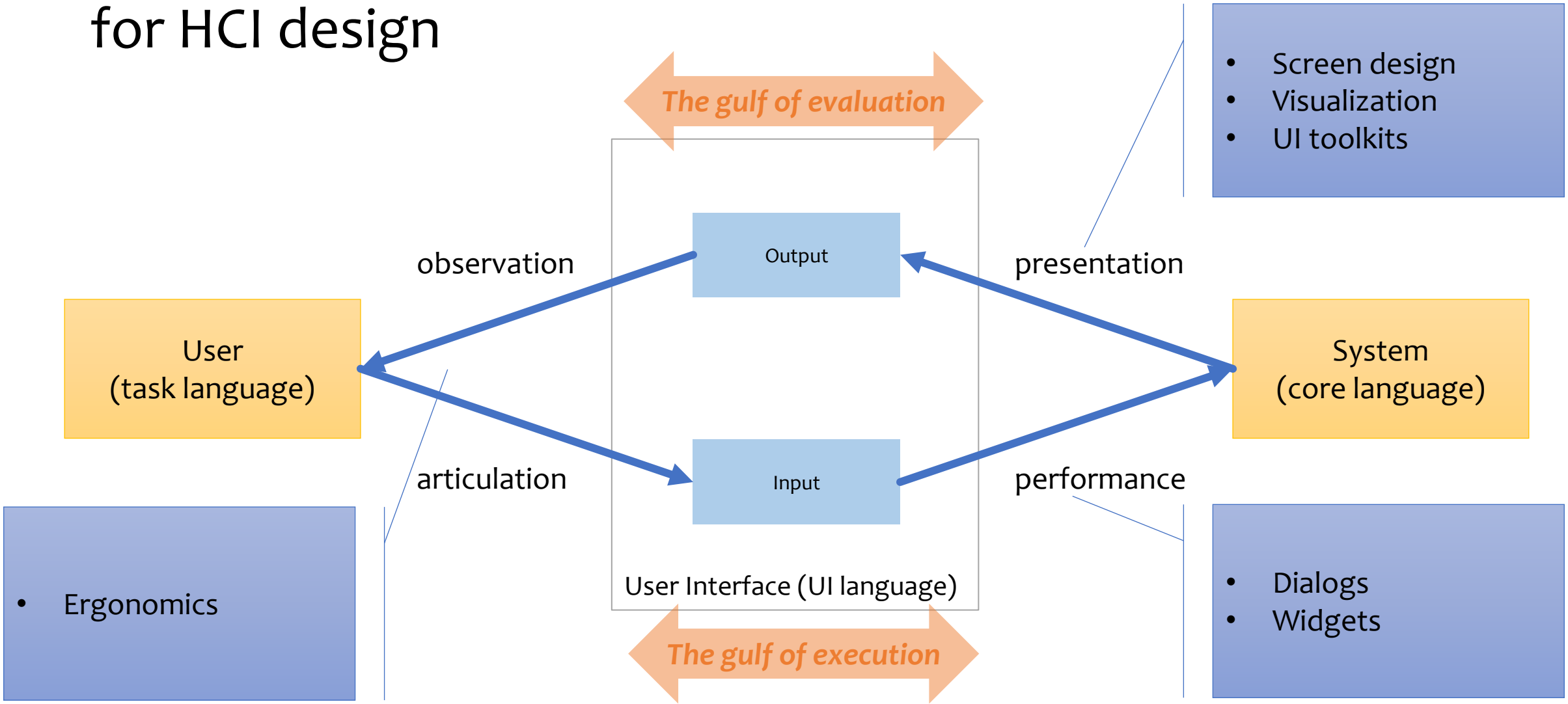




Example (presentation): what are the allowed combinations?



Tools, Techniques and Environments
for HCI design



Frameworks: major UI styles

- Command line interface
- Menus
- Natural language
- Question/answer and query dialog
- Form-fills and spreadsheets
- WIMP
- Mobile
- Point and click
- Three-dimensional interfaces

Design frameworks

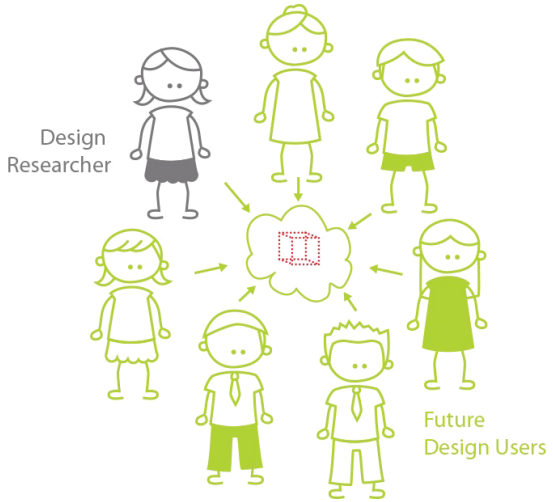
Approaches for shaping the design process

User-Centered Design (UCD)

- Avoid the risk of software project failure
  - Estimated 50% are affected by bad developer<->user/client communication
- UCD takes the needs, wants, and limitations of the **actual end users** into account **during each phase** of the design process
  - User-centered design issues are discovered during the early stages
- Benefits: systems easier to learn, with faster performance, with less human errors, encourage users to discover advanced features, and avoids "building the wrong system"
- Issues: how to find users? How many? How motivated? How to speak their language? How to extract user needs, business needs, organizational implications?

Participatory Design

- One step further than UCD, users are directly involved in the collaborative design of the things and applications they use
- Engage a group of users
  - Discussions
  - Creating scenarios, sketches, dramatizations
  - Creating and testing lo-fi prototypes
  - Continuous meetings, flexible management
  - Highly reliant on the skills of the group moderators/leaders (keep involved, filter ideas, reward participation, work around resistances, …)
  - More effective with more mature and prepared user populations (less with kids, elderly, disabled, …)



Agile Interaction Design

- Borrows ideas from agile development in software engineering
- Key: evolutionary development
  - System is built incrementally in rapid release cycles
  - Rapid prototyping techniques (for hardware, software and physical objects)
- Focus on low-cost many-iterations prototypes
- Requires fast usability inspection (extreme usability, XU)
- Makers' culture (only if it involves users!)

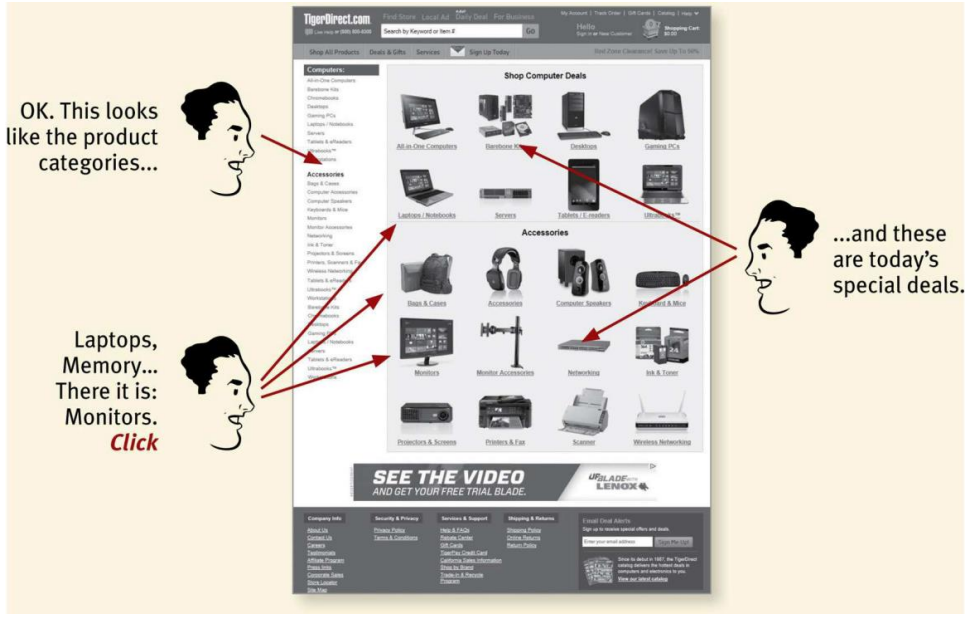
Human-centered design processes

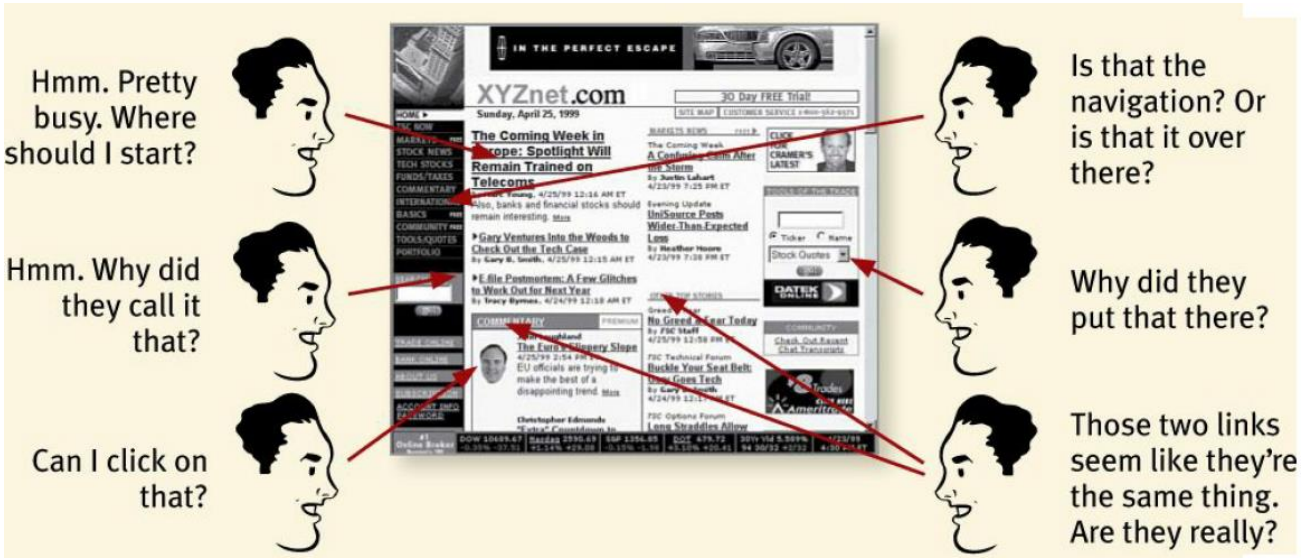
A process for designing interactive systems with a focus on usability

Usability

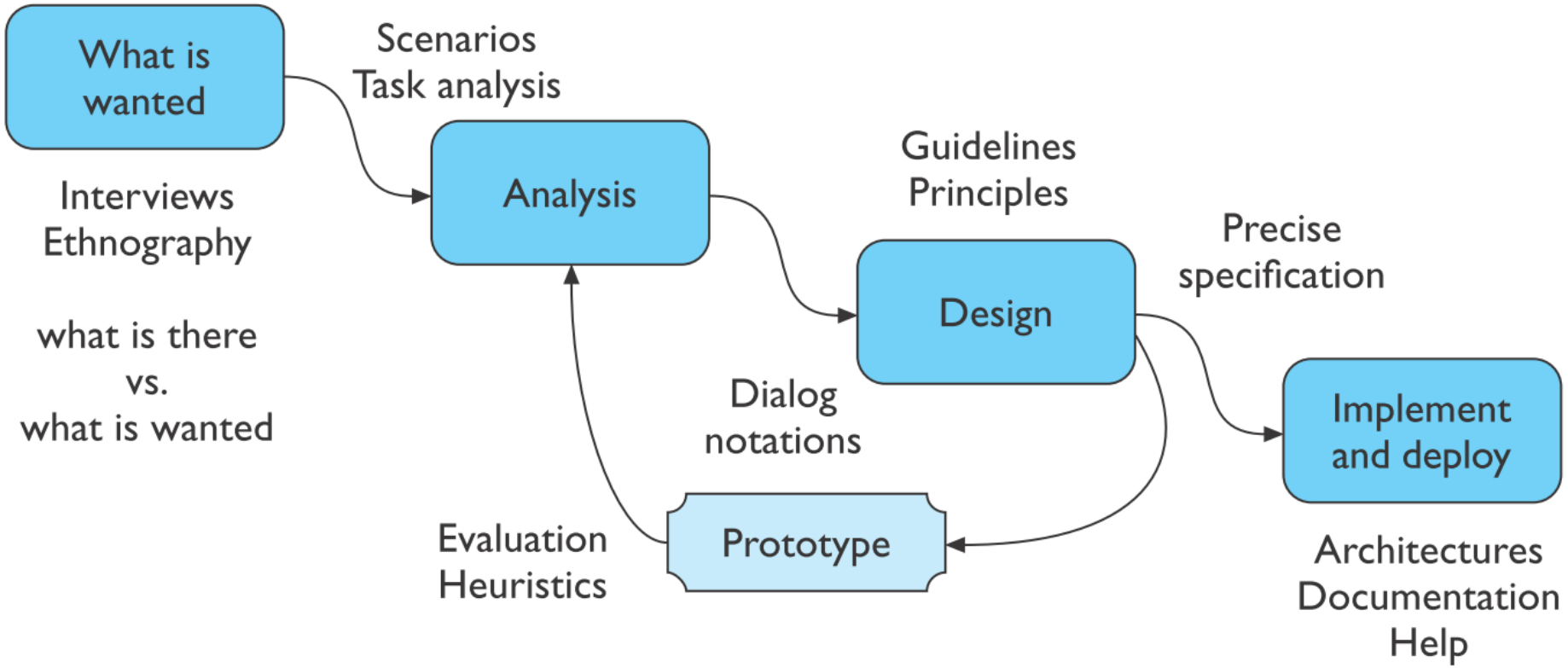
- **Usability**: how well users can use the system's functionality
- Dimensions of usability:
  - **Usefulness**: does it do something people want?
  - **Learnability**: is it easy to learn?
  - **Memorability**: one learned, is it easy to remember?
  - **Effectiveness**: does it allow reaching the goal?
  - **Efficiency**: once learned, is it fast to use?
  - **Visibility**: is the state of the system visible?
  - **Errors**: are errors few and recoverable?
  - **Satisfaction**: is it enjoyable to use?

Usability: Don’t Make Me Think



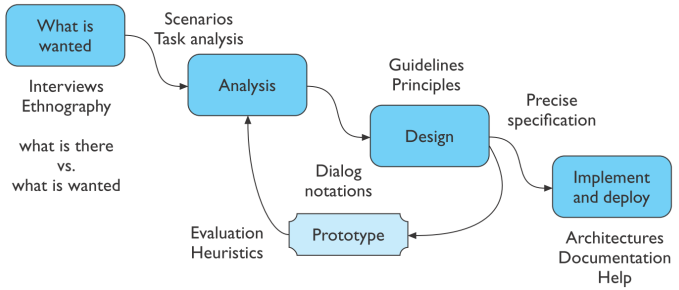


Human-centered design process
(simplified and generic)



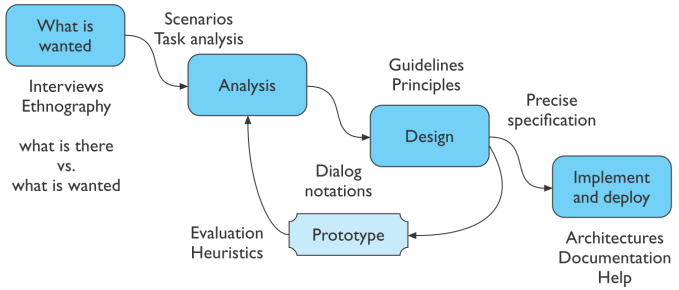
Human-centered design process – the main steps (1)

- **Requirements – what is wanted**
  - What exactly is needed? How are people currently accomplishing the goal?
  - User observation, interviews
- **Analysis**
  - Formalize and structure the requirements
  - Create interaction scenarios, stories, tasks
  - Compare current situation with expected new situation

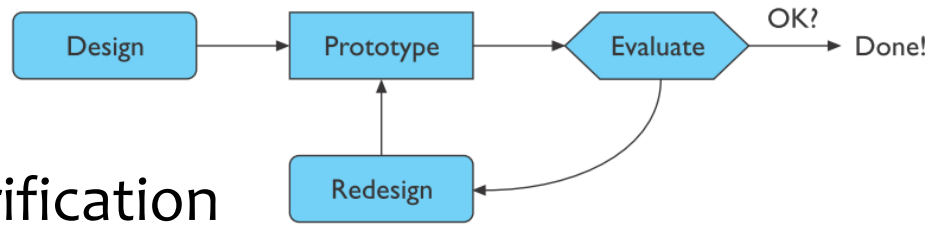


Human-centered design process – the main steps (2)

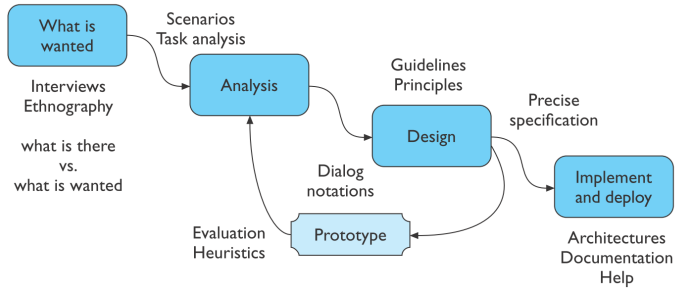
- **Design**
  - The main choices to shape the system
  - Rules, guidelines, design principles
  - Considering different types of users
  - Modeling and describing interaction
  - Visual layout
  - Consider all inputs from cognitive models, communications theories, organization issues



Human-centered design process – the main steps (3)



- **Iteration and prototyping**
  - Design must be supported by intermediate verification
  - Evaluate the design in its partial forms:
    - Prototypes
    - Evaluation metrics
  - Involving users
- **Implementation and deployment**
  - Hardware and software implementation
  - Documentation

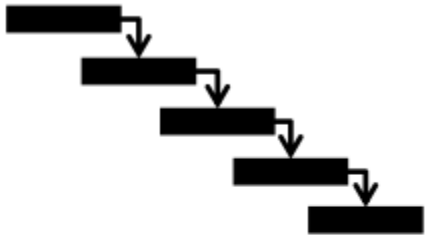


HCI in the Software Process

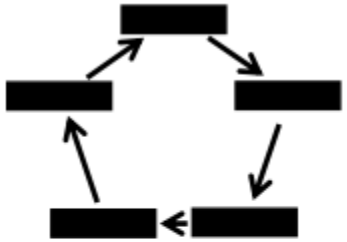
How to include Human-centered processes in Software Engineering

Software Engineering Processes

Where / how does HCI fit in?

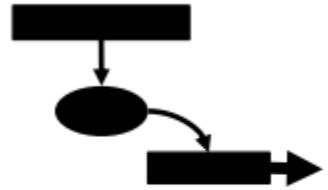


Waterfall



Iterative waterfall

Agile (scrum)

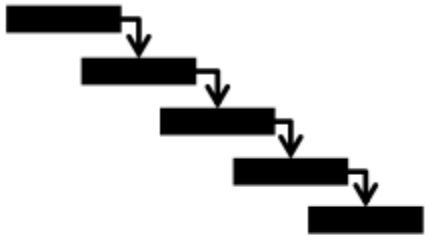


Lean

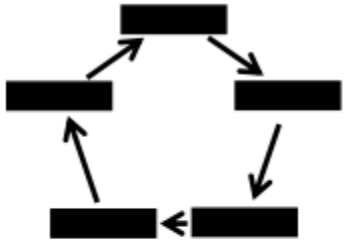
Software Engineering Processes

Where / how does HCI fit in?

Always a step ahead!

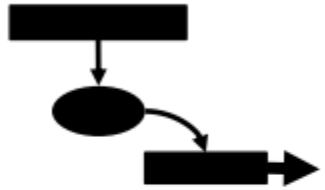


Waterfall



Iterative waterfall

Agile (scrum)



Lean

Always a step ahead

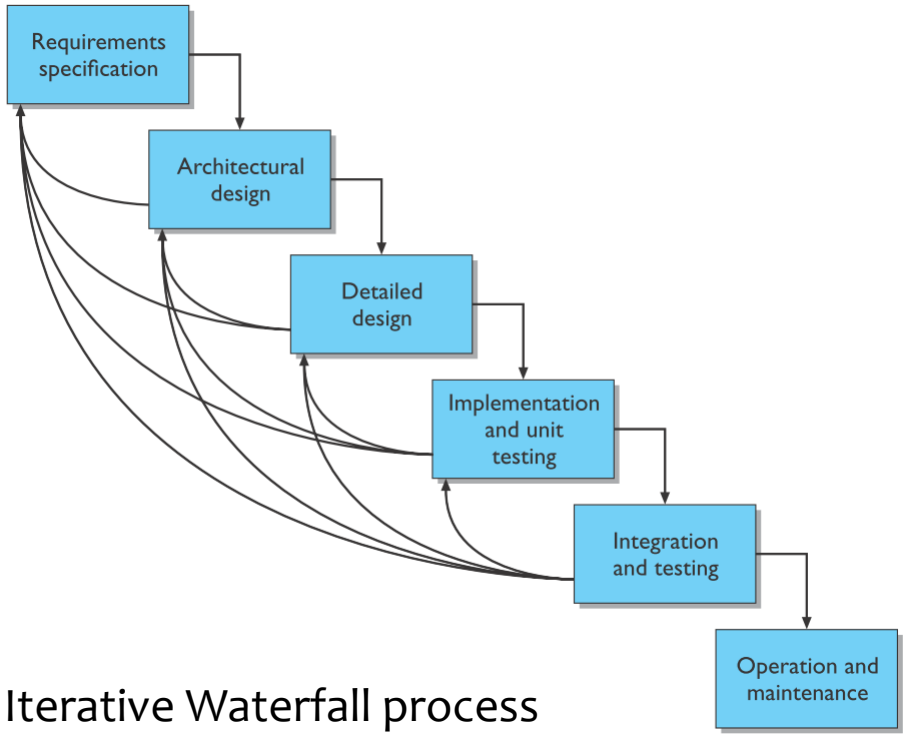
- Before
  - Every design step
  - Every implementation step
  - Any product iteration (or sprint)
  - …
- You need a user-centered step
  - Evaluate usability
  - Experiment with users
  - Evaluate alternative flows
  - Evaluate alternative layouts
  - …

- User-centered steps are cheaper than development
  - User research about users' needs to decide what to design
  - Heuristic evaluations before testing with users
  - Evaluating prototypes instead of full-fledged products
- Anticipate critical decision points later in the project

Always a step ahead

- Usability, Safety, Performance, are part of Non-Functional Requirements
- User-centered steps are cheaper than development
  - User research about users' needs to decide what to design
  - Heuristic evaluations before testing with users
  - Evaluating prototypes instead of full-fledged products
- Anticipate critical decision points later in the project

Example



Iterative Waterfall process

- Each step must be
  - Preceded with user evaluation of the design choices and formalized requirements
  - Followed with user evaluation of the result
- Must produce additional artifacts to allow this kind of iteration
  - Prototypes

References

- Robert Miller, MIT Course "6.813/6.831: User Interface Design & Implementation"
  - Spring 2018 - http://web.mit.edu/6.813/www/sp18/
  - Spring 2011 - https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-831-user-interface-design-and-implementation-spring-2011/index.htm
- Dix *et al*: Human Computer Interaction
  - Chapters 3, 5
- Norman: The Design of Everyday Things
  - Chapter 2
- Krug: Don't make me think
  - Introduction
- Shneiderman: Designing the User Interface
  - Chapter 4

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:**
  - **Share** — copy and redistribute the material in any medium or format
  - **Adapt** — remix, transform, and build upon the material
  - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
  - **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.
  - **NonCommercial** — You may not use the material for commercial purposes.
  - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **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/