# Interação Pessoa Computador

## 1. Introduction

### Wireframe

An image or set of images which displays the functional elements of a website or page, typically used for planning a site's structure and functionality.

Human-computer interaction (HCI) is a multidisciplinary field of study focusing on the design of computer technology and, in particular, the interaction between humans and computers.

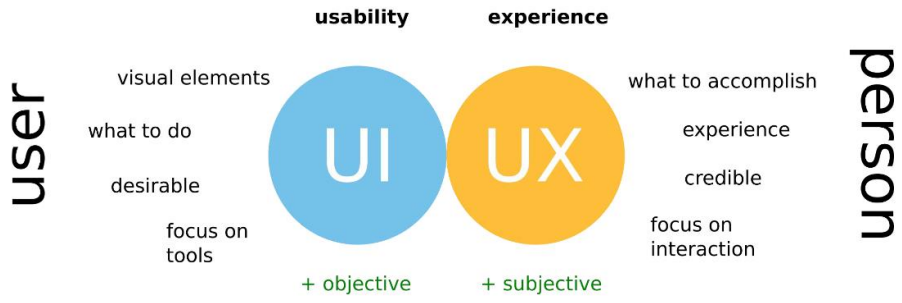
### User Interface

- "Visible" (human-stimulating) part of the system
- Enables the users to
  - Interact with the system
  - Perform their tasks
  - Get feedback/information from the system
- The User operates/interacts over/through the Interface

### User Experience

- The whole experience with a system, technology, device
- Not only the direct interaction with the artifact, but the overall context
- Involves affective component
- One does not design the user experience, but DESIGNS FOR an user experience

### UI vs UX

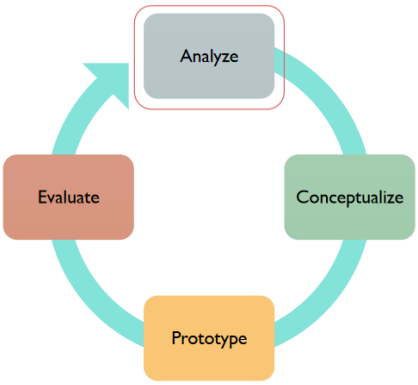


## Some Design Myths

- “Good design means good graphics”
- “Marketing dept. knows the users”
- “Good design is common sense”
- “The interface can be designed in the end”

2. **User Research**

## The iterative design cycle




### Users

- Those who will ultimately use the product
- Have needs and expectations (often not consciously)
- Their experience with the product will dictate the actual product's success
- Therefore, should be in the center of the process

### Stakeholders

- Define the initial motivations and main goals
- Point to high-level requirements
- May be wrong in their assumptions of the user's needs and expectations

### Designers

- Research the users, to know their needs and expectations (and more)
- Understand the stakeholders and the product being proposed
- Bridge the gap between both, by designing and interacting over interfaces taking all into consideration

### What to know about them?

- How they are defined
- How they differ individually
- How their expertise evolves during product usage

### How are they defined?

- What they know about the tasks
    - Previously known, recently introduced, new
    - How they have learned them
    - Their level of expertise

- What they know about the tools
    - they currently use
    - How they use them
- What mental models they have
    - How they interpret internally the system
    - What vocabulary they associate with elements and tasks -> conceptual model

### How do they differ individually?

Within a user population, there are individual differences that may lead to sub-groups of users

- Personal traits
- Physical differences
- Cultural differences
- Motivation and attitude differences

### How their expertise evolves during product usage?

- Beginner
- Advanced Beginner
- Competent
- Expert

### Task Analysis approaches

- Formal
    - Structured list of tasks, flow diagram
    - Adequate for existing systems, where there are processes already defined
- Informal
    - Follow a more exploratory approach to uncover the users needs
    - Adequate for creating new designs (or redesigning processes) - our case

### User Research Methods:

- Attitude vs behavior
- Qualitative vs Quantitative
- Implicit
    - Eye/user tracking
    - Observation studies
    - Task and reaction measurements, ...
- Explicit
    - Questionnaires/Surveys
    - Self report
    - Think aloud protocol, ...

### Personas:

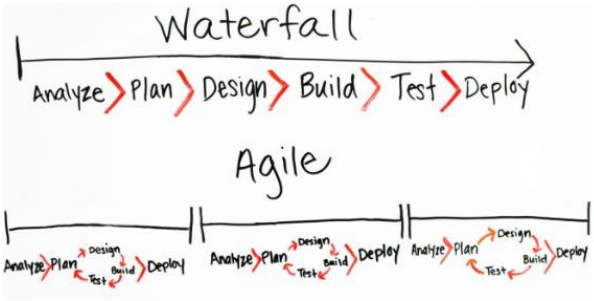
“A Persona is a user archetype you can use to help guide decisions about product features, navigation, interactions, and even visual design”

- Rich description of one of a set of typical users
- not a real person, but a realistic one, that can be related to

Theses groups are defined by:

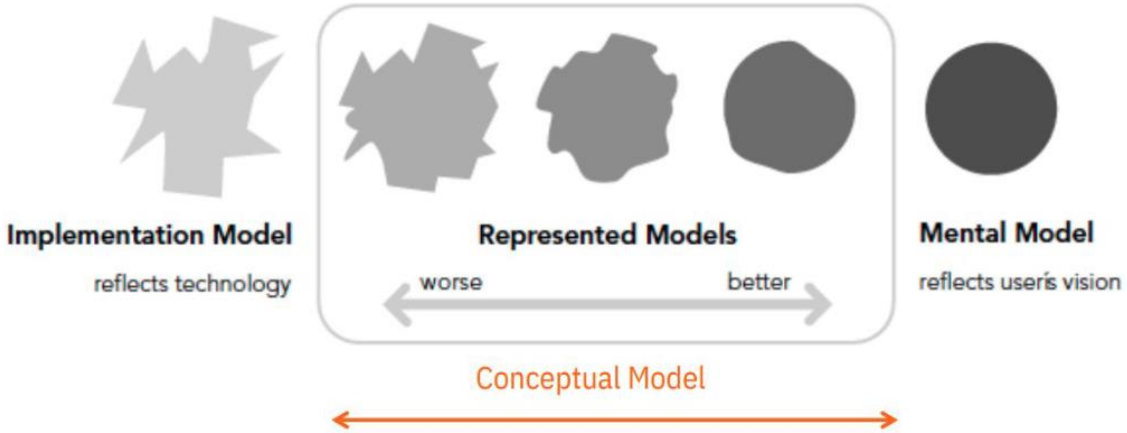
- Motivations
- Needs
- Frustration

3. **Ideation**




## Conceptual Model

- What people can do with the system
- What concepts the team needs to understand to use the system
- It refers to objects, attributes and actions in the task domain
- What are the mappings between them
- Allows the team to arrange ideas before designing the U.I.
- If possible explore metaphors well-know to the users




## Conceptual Model:

- Objects (attributes)
  - Photo (date, caption).
- Actions
  - Insert, remove, select photo.
- Relations
  - Archive has photos.

## Requirements and task selection:

- Define a set of functional requirements
- A list of specific tasks

## Usability requirements:

- Used to evaluate the usability
- Efficacy
    - All users completed task, 90% made no more than 2 errors
- Efficiency
    - Average time under 3 minutes, 90% made less than 10 clicks
- Satisfaction
    - Less than 10% unsatisfied, 90% preferred our solution

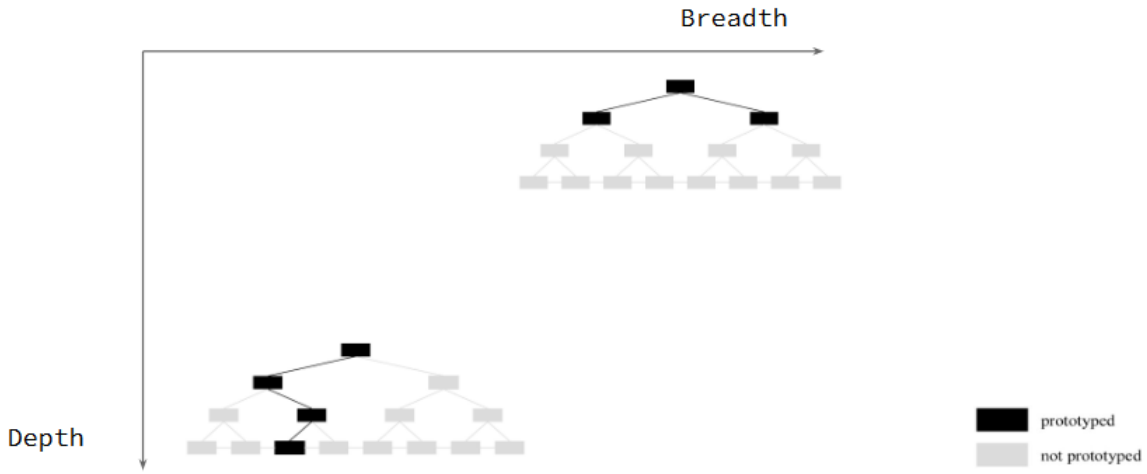
## Idea vs Ideation

An idea is an opinion, concept or notion. An idea is not the same thing as ideation. An idea is a mental thought, ideation is the process of forming that mental thought. An idea is the output of the innovation process, not the starting point.

## Some important principles

- Ideate individually first
- Bring ideas to the group
- Share in an open-minded way
- Generate more ideas
- Generate in parallel, not in sequence
- Avoid closing in on an idea too early
- Don’t interrupt other people’s ideas
- Prototype ideas to validate feasibility and interest

4. **Prototyping**




- Users don’t know what they want, but they can tell you what they don't want
- Prototype concrete, yet partial representation of the system
- Goal:
    - Try out ideas
    - Reduce time and cost needed to test a design
    - Dump bad ideas early
- Can focus on:
    - UI -> e.g. hand-drawn screens, Mock-up video
    - Device -> e.g. Cardboard model
    - Functionality -> Partially-functioning software

### Experiential techniques

- Storyboards
- Body-storming
- Wizard of Oz

### Storyboards are about communicating the concep

- A series of drawings that tell a story
- Depicts settings, context and sequences of actions/events
- A mix between comics and movie script
- Communicate context, sequence and goal

### Body-storming

- Experience physically a situation
- Imagine systems in context
- Generate ideas on tasks/functionalities

- Do role play
- Important steps
    - Choose the place
    - Prepare the space and materials
    - Recruit, test and observe
    - Debriefing with users and observers

### Wizard of Oz

- Functionalities of the system are simulated by a human (hidden or not)
- Useful for early concept testing/simulation of complex or demanding systems (e.g. speech or gesture recognition, AI...)

### Prototypes

- Fidelity focuses on appearance
- Functionality focuses on system response
- Horizontal prototypes have a broad but shallow coverage of tasks
- Vertical prototypes focus on specific parts in depth (front-end down to back-end)

### Low-Fidelity Prototypes

- Easy and fast to create
- Low-cost (hand-made or with simple tools, rough)
- Deliberately focus on the idea, not on the looks
- Quick user feedback, allows to easily test ideas
- Disposable: not feeling sorry for throwing away a bad idea
- Cons:
    - Very high-level
    - Not very scalable

### Paper Prototypes

- Can be built with simple materials and tools

### Mock-ups

- Representations of the interface in a more detailed way
- Usually created with software
- Can be lo-fi (e.g. wireframe) or hi-fi (e.g. actual hi-res widgets)
- Can be printed out to test as paper prototypes, or used digitally (desktop, mobile)
- Good for UI consistency, more scalable than sketching, and still faster and cheaper than coding
- Cons:
    - Can be slower than sketching for small/initial prototypes
    - Flow has to be induced by the tester
    - If hi-fi too early, can bias the user

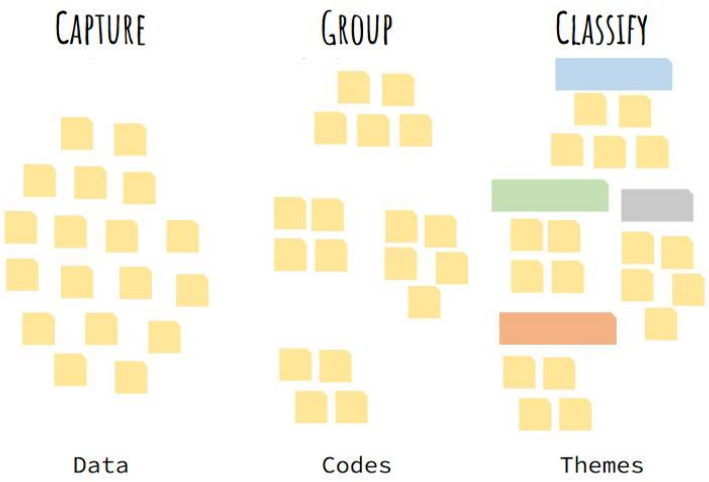
## Interactive Mock-ups

- Use tools to add interactivity to mock-ups
- Can be simple “hyperlinking” between screens through clickable areas
- Can have some more complex interactions, but not actually programmed

## Functional Prototypes

- Prototypes that already implement part of the logic of the functionalities
- Can be in the final platform, or in some rapid prototyping platform
- Usually vertical, but can be horizontal
- May be lo-fi or hi-fi (although usually at this point there are already hi-fi mock-ups available)

5. **Evaluation**




## Design process

- Interdisciplinary team
- Competence-based leadership
- No hierarchy
- Start with secondary research
- Observation
- Listening to experts who know the context
- Iterative design
- Brainstorm
- Parallel design
- Choose/vote ideas/solutions
- Mix and assign ideas
- Produce prototypes
- Test prototypes

## The 5W 1H approach to evaluation:

- Why: to check if users can use the product and if they like it.
- What: a conceptual model, early prototypes of a new system and later, more complete prototypes; finished products
- Where: in natural or laboratory settings.
- When: during design; or on finished products to assess if the requirements are met and to collect information to develop new products.
- Who: who must be involved in the evaluation
- How: which evaluation methods to use

## Types of assessment:

In a nutshell, formative assessments are quizzes and tests that evaluate how someone is learning material throughout a course. Summative assessments are quizzes and tests that evaluate how much someone has learned throughout a course.

## Usability testing

Evaluation approach that involves measuring typical user's performance on typical tasks

## Analytical evaluation

- An approach to evaluation that does not involve end-users
- Two main methods are used:
    - Heuristic evaluation:
        - a method guided by heuristics (guidelines and standards) and walkthroughs to identify usability problems;
        - walkthroughs involve experts in walking through scenarios with prototypes of the application
- Predictive evaluation:
    - methods based on theoretical models to predict user performance; Users do not need to be present

## Heuristic Evaluation

- Analytical method
- Made with experts/evaluators
- Good for evaluating initial designs and prototypes
- Advantages
    - Fast
    - Cheap
    - Easy to use
- Usability heuristics
    - Designers use to guide, evaluators to assess
- There are multiple sets
    - Nielsen; Tognazzini, Norman, Schneiderman; etc.
    - Additional list of specific heuristics

## Data types

- Quantitative (quantity, specific and measurable)
- Qualitative (quality, "open")

- Objective
- Subjective

### Methods

- Formative Assessment (usability problems)
- Summative Assessment (performance measures)

### Tasks

- Real and representative
- What and not how
- Specific
- Mixes complexities
- Summative (comparative) evaluation

### Usability Measures

- Time to complete task
- No. of mistakes made
- No. of completed tasks
- No. of clicks
- No. of help queries
- User satisfaction

### Data Collection

- Direct observation
    - Observed users performing tasks
    - In person or video
- Indirect observation
    - Diaries – manual
    - Interaction logs – automatic

### Variables

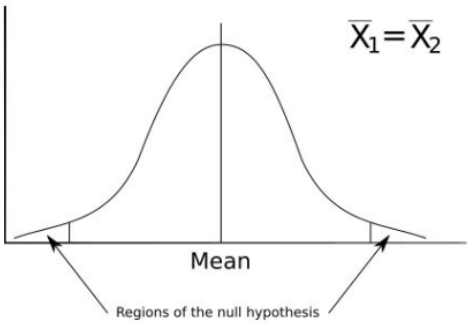
In analytical health research there are generally two types of variables. Independent variables are what we expect will influence dependent variables. A Dependent variable is what happens as a result of the independent variable.

Measures

- Nominal: Mode
  - Color, brand, name
- Ordinal: Median, Mode, Quartiles
  - [light, medium, heavy] [unhappy, neutral, happy]
- Continuous: Mean, StDev, Median, Quartiles
  - Age, height, weight, time, no. of errors

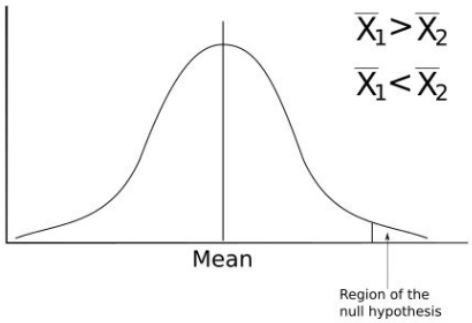
# Single-tailed vs Two-tailed

Two-tailed




Test if coin flips are biased to heads or tails, considering number of heads

Single-tailed




Test if coin flips are biased to heads considering number of heads

# Mean and Standard Deviation

$$\bar{x} = \frac{\sum x_i}{n}$$ Mean

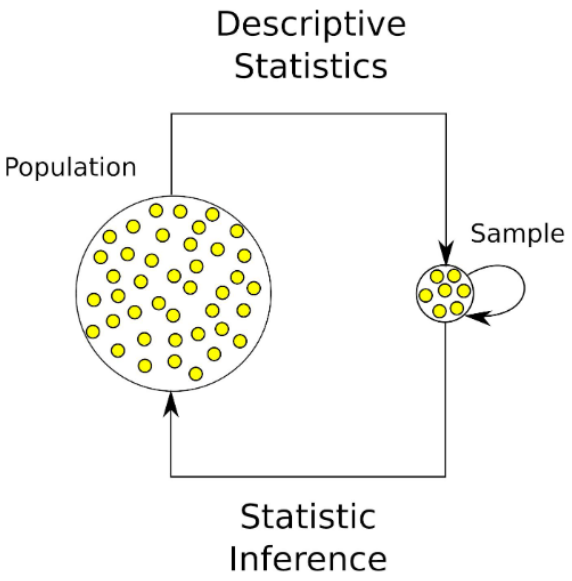
$$SQ = \sum (x_i - \bar{x})^2$$ Sum of Squared Differences (SSD)

$$s^2 = \frac{SQ}{n-1}$$ Variance

$$s = \sqrt{s^2}$$ Standard Deviation (St.Dev)

## Descriptive Statistics

- Typically first step to examine data
- Characteristics that summarize the sample
- Typical measures




## Null Hypothesis

- H0 - Null Hypothesis
  - Changes in experimental conditions do not reflect in the sample
  - E.g. “The new icons do not affect user performance”
- H1 - Experimental Hypothesis
  - Assume what we want to verify
  - e.g. “The new icons improve user performance”

## Confidence Level

- The probability of assuming H1 is true, but being wrong is designated by $\alpha$ (alpha)
- The confidence level (“grau de confiança”) is (1-$\alpha$)
- Typical values of $\alpha$ used are 0.1, 0.05 or 0.01, with the correspondent confidence levels of 90%, 95% or 99%

## T-Student for mean values

- Used for continuous variables
    - Time, errors, height, SUS, etc.
- Compare two mean values
    - e.g. the mean selection time of an icon with a trackpad vs a mouse

## 6. Nielsen's Usability Heuristics

What is the purpose of heuristic evaluation? Heuristic evaluation helps optimize usability by minimizing design deficiencies. It thus guarantees the effectiveness of the interface of a digital device, such as a website, a mobile application or SaaS software.

1. Visibility of the system status

- The system should always keep users informed about what is going on through appropriate feedback within reasonable time.
- Report what is going on clearly and objectively
- In proper time
    - < 0.1s: no indicators needed
    - 0.1s - 1s: cursor change
    - 1s - 10s: time remaining
    - > 10s: progress indicators

2. Match between the system and the real world

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

3. User control and freedom

- Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. Consistency and standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. Error prevention

- Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit the action.

6. Recognition rather than recall

- Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. Flexibility and efficiency of use

- Accelerators - unseen for the novice user - may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. Aesthetic and minimalist design

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

9. Help users recognize, diagnose, and recover from errors

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

10. Help and documentation

- It is better if the system can be used without documentation, but it may be necessary to provide help and documentation
- It should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

https://quizizz.com/admin/quiz/5d649d8ab4d2be001a88c944/jakob-nielsens-10-usability-heuristics?tags=63928e1240c8bd001ef271df&sortKey=createdAt&order=desc&queryId=63928d630dac92001db9cad4-1670549012057&searchLocale=

## 7. Help and Documentation

### User strategies for getting information

- Browsing
- Search
- Query
- Structured search
- Guided search

### Browsing

- Like walking around a store and seeing what catches the attention
- Challenges in mapping that to a digital device/screen

### Search

- Users provide search criteria
- Users knows the target, need help in finding it
- Different from querying

### Query

- Users have an idea of what they are looking for, but don't know the exact items
- Usually resort to multiple terms or criteria

### Structured search

- Helps the users converging their search focus
- Allows selecting from small sets of options
- Can be adaptive/dynamic, as each new set of options can be adjusted according to previous choices

### Guided search

- Used when there are a limited set of options and an expected sequence
- The users are guided through the process

## Documentation Types

- Manuals
- Interactive Help
- Tutorials

## Some hints

- Documentation should involve technical writers, developers and designers
- Style guides should be used
- It is not likely to have good documentation for a bad interface
- The documentation should also be tested/evaluated

## User Manual

- A resource that helps the user to learn how to use a product
- It should facilitate learning in a manageable, incremental manner.
- It should be easy to navigate
- It should be oriented to the user's tasks and needs, not the system's functionality

## Quick Start Guide

- To-the-point basic instructions for initial operation
- Initial setup and most basic/main features
- Goal: to be small enough for eager users to accept and look at it, and avoid frustration and trial-and-error

## Reference Manual

- An exhaustive reference of functionalities of the product
- To be used when the users know the functionality they want, and need details on how to use it
- Usually oriented to power or expert users
- Focus on completion and functional details

## Quick Reference Guide

- A very compact version of the reference manual
- Lists the main/more commonly used functions for easy recall of their usage
- Can be formatted as a "cheat sheet"
- Can be organized in functional blocks, alphabetically, or other
- Oriented to expert users

## Interactive Help Important Features

- Availability
  - Available anywhere in the system
- Precision and coverage
  - According to version, and covering all functionality
- Consistency
  - Within interactive help, and between it and other docs
- Robustness
  - Available even in cases of application failure
- Flexibility
  - Allow different navigation modes, be context-sensitive
- Discretion
  - Should not interfere with user's tasks or system functioning