

# Industry 4.0 Design: Introduction to Human Computer Interaction

Prof. Andrew D. Bagdanov  
andrew.bagdanov AT unifi.it



Dipartimento di Ingegneria dell'Informazione  
Università degli Studi di Firenze

7 February 2020

# Outline

Introduction

Human Centered Design

Needfinding

Usability Analysis

Reflections

# Introduction

## Lecture presentations

- ▶ You can will find all **class lecture presentations** at this site:

<http://micc.unifi.it/bagdanov/i4.0/>



- ▶ Published here will also be links to **Colaboratory Notebooks** and any supplementary material.

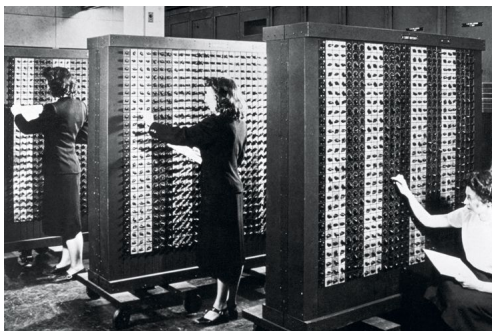


# What is HCI?

- ▶ Human-computer interaction (HCI) began to **emerge** as a discipline *per se* in the 1980s.
- ▶ Initially it was specialization of computer science that embraced **cognitive science** and **human factors** engineering.
- ▶ It has expanded rapidly for three decades, and now attracts professionals from **many other disciplines**.
- ▶ We can think of HCI as **human-centered informatics**, if we wish.
- ▶ HCI is concerned with how **humans interact** with computers, and how **computers interact** with humans.

## Some history

From the beginning there was a clear need for HCI. . .



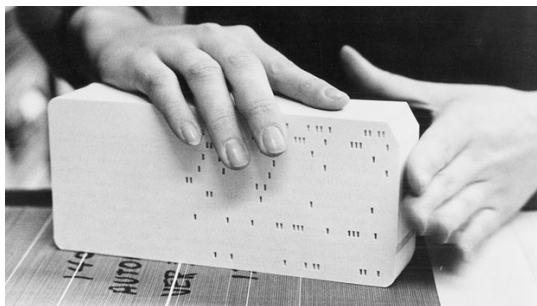
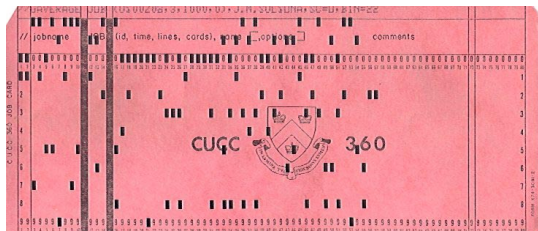
*“ENIAC’s six primary programmers, Kay McNulty, Betty Jennings, Betty Snyder, Marlyn Wescoff, Fran Bilas and Ruth Lichterman, not only determined how to input ENIAC programs, but also developed an understanding of ENIAC’s inner workings. The programmers debugged problems by crawling inside the massive structure to find bad joints and bad tubes.”*

## Punched-card I/O

What to do when data is orders of magnitude larger than memory?



# Useful, and oh so inconvenient



## If only there were a way to TELE-type

The teletype had already been in use, but we quickly learned how to use it with computers for input/output.



## When it rains, it pours

Why use paper, when you can use television? The **dumb terminal** revolutionized HCI and gradually metamorphosed into the modern PC.



## Not just about better hardware

Does anyone recognize this person?



## The compiler radically changed HCI

- ▶ In 1951, Grace Hopper invented the first compiler of a **high-level programming language**.
- ▶ This paved the way for hardware-independent programming languages (e.g. **FORTRAN** and **Lisp**).
- ▶ It is difficult to **overstate the significance of this contribution to HCI**.





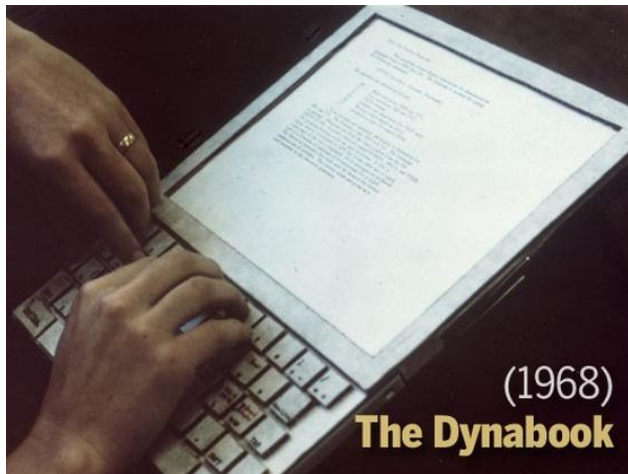
## Breaking the physical barrier

Use a **physical proxy** to represent (and control) virtual elements of the interface: the birth of the **Graphical User Interface**.



## The birth of HCI as design

**Alan Kay:** the Dynabook was at least a decade ahead of its time.



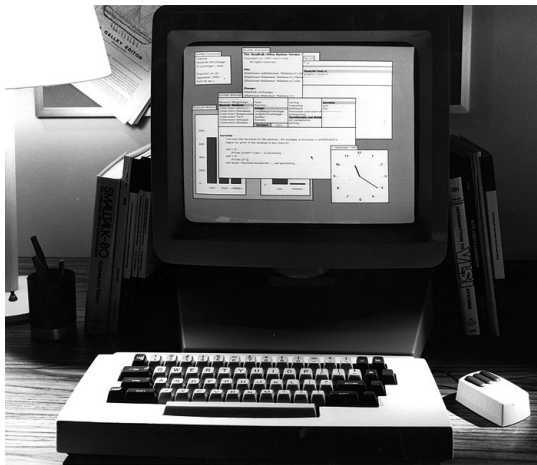
## This was as close as they could get

The Xerox Alto was the first graphical workstation – it didn't live up to the vision of the Dynabook, but the **vision** was important.



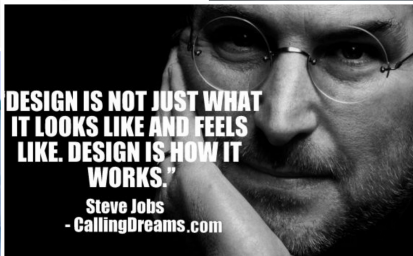
## A 40-year-old desktop

We see the first iterations of elements we take for granted in modern GUIs.



## The inheritor of the Dynabook

Steve Jobs with Apple took **design** to its limits over the last forty years.



## A timeline emerges

- ▶ **1950s-1960s: Getting data in, getting data out.** Low-level HCI.
- ▶ **1970s: The rise of the Personal Computer.** The broad projects of cognitive science, artificial intelligence, linguistics, cognitive anthropology, and the philosophy of mind, had formed at the end of the 1970s.
- ▶ **1980s: Graphical User Interface (GUI).** Interfaces designed for easier understanding of computers. GUIs were is easy to use, understand, visualize, and it improved the working environment.
- ▶ **1990s The Internet and Collaborative works.** Communication among people and computers became easier, and computers and their interfaces became highly decentralized.
- ▶ **2000s: Mobile Computing.** Mobile and Smart Phones blur the divisions between computers and between people and groups.
- ▶ **2010s: The Social Computer.** Social networks and social applications radically distribute interactions and puts computers in role as *mediators* of interaction.
- ▶ **Today: accelerating change.** Wearable computing devices, ubiquitous computing, the Internet of Things, natural interaction, surface-based computing, learning.

# Human Centered Design

# HCI as design discipline

## Three "use" words

- ▶ **Useful:** the artefacts and interventions should be functional and “do things”.
- ▶ **Usable:** they should be easy to “do things”, they should do the right thing and be enjoyable (user experience).
- ▶ **Used:** they should be attractive, available and accessible, and **acceptable** to organization/community.

## Design context (recent HCI history)

- ▶ **1980s:** first conferences and journals appeared; importance due to rise of **personal computer**; designed for **work**: *one user, one machine*.
- ▶ **1990s:** many people, geographically remote; still one user; local networks; still designed for work: one user, one machine (and one machine, one user).
- ▶ **Now:** domestic use; global networks; ubiquitous devices.



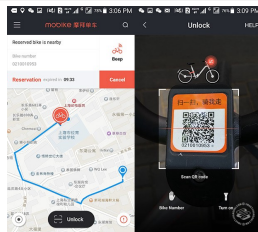
# Ubiquity of technology

- ▶ How many **computers** do you have?
- ▶ How many do you have *with you right now?*

# Ubiquity and the importance of HCI

## Daily life

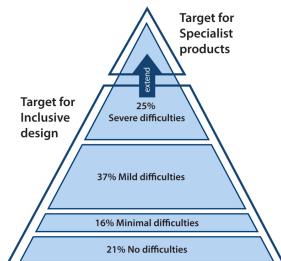
- ▶ Computers permeate **every aspect** of our daily lives.
- ▶ Even when not “using” a computer, our life is **affected** in some way by computing.
- ▶ ATMs, ticket vending machines, drink dispensing machines, etc.
- ▶ HCI is an important factor when designing any (and all) of these because it **affects our daily quality of life**.



# Ubiquity and the importance of HCI

## Accessibility

- ▶ HCI is key to building systems **accessible** to persons with disabilities.
- ▶ A core philosophy of HCI is to provide safe, usable, and efficient systems to everyone
- ▶ Any system properly designed with **user-centered** techniques will also be maximally accessible to those with disabilities.

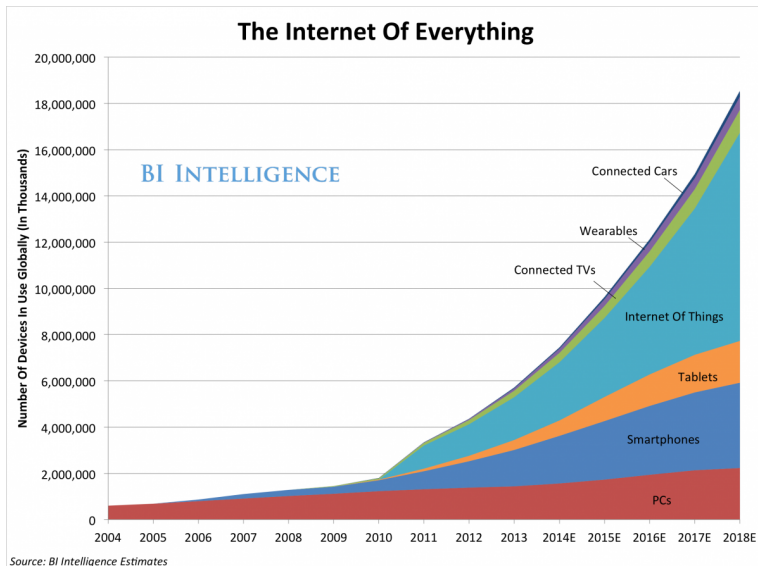


# Ubiquity and the importance of HCI

## Untrained users

- ▶ Today, the vast (**vast**) majority of users are decidedly non-expert computer users.
- ▶ Contrast this with the early days of computers (even 20 years ago).
- ▶ Users **expect** to understand the main functionality of an average program within **a few minutes**.
- ▶ Interfaces *must* be effective, obvious, easy to use, and most importantly they **must not require training**.
- ▶ We will see that this is why developing **scenarios** and **personas** for our HCIs is essential.

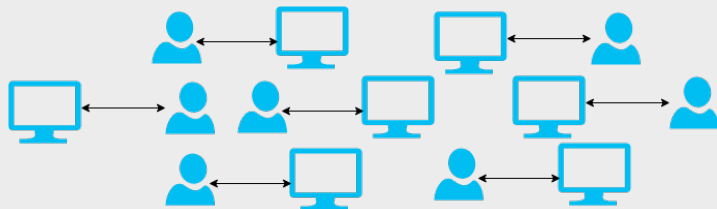
# Ubiquity and the real cost of neglecting HCI



# So what?

## 20 years ago

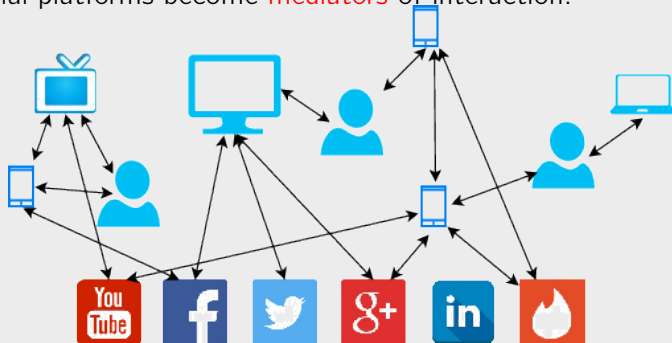
- ▶ Dialog: single user → single computer
- ▶ Very much **work and task oriented**.
- ▶ Traditional HCI: tasks, goals, work work work.
- ▶ Users **trained** to use a handful of programs.
- ▶ Elements of these programs optimized for efficiency.



# So what?

## Today

- ▶ Dialog: with the **world**.
- ▶ Much less computer/artefact/user divide.
- ▶ Wearable: **no** computer/user divide.
- ▶ Social platforms become **mediators** of interaction.



## So what?

- ▶ We can (and should) ask ourselves *why we should care about HCI (much less study it)*.
- ▶ The previous discussion should have shed some light on this.
- ▶ Technology is **ubiquitous** and we need to interface constantly with computers of all sorts.
- ▶ Studying and improving HCI thus has a **pervasive affect on nearly every aspect of our lives**.
- ▶ But, there is another reason – one perhaps even more important – related to how **we take design for granted**.
- ▶ More specifically, how **bad design** has come to pervade our lives.
- ▶ And how we have come to accept bad design as a **norm**.



## Discoverability and the “Norman Door”

- ▶ **Don Norman** (a legendary figure in the HCI and design fields) popularized the concept of the *Norman Door*.
- ▶ These types of objects can be used to explore and understand design in everyday life.
- ▶ **Question:** How do you know how to use a door?



## Five key principals

- ▶ **Affordances:** we understand what actions or properties a door (in the sense of Platonic ideal) **affords** the user. It allows passing through, it separates one area from another, etc.
- ▶ **Signifiers:** there are elements of the object's design that **signify** how to use it. A door has a handle indicating it should be **pulled**, a flat plate indicating it should be **pushed**, etc.
- ▶ **Constraints:** there may be **physical** constraints that indicate how a door **must** be used. There can also be **cultural**, **logical**, or **semantic** constraints.
- ▶ **Mappings and feedback:** there must be a direct mapping between an object's controls and their **affect** on the world.
- ▶ **Conceptual model:** perhaps the most important, we all of a **mental model** of what a door is and how to operate one.

## So, again...

- ▶ How can should we *discover* the operating principles of this door?



# The psychology of everyday actions

- ▶ We use the metaphor of **everyday things** because computers and technology have become *everyday things*.
- ▶ We will see in subsequent lectures that these questions touch on some deep themes in psychology and cognitive science.
- ▶ Understanding the intention of users, and supplying appropriate feedback when things go wrong (or go right) is essential.
- ▶ To do this we need to **understand** and model the **user**.
- ▶ But, one might ask, isn't modeling humans a **dehumanizing** act that we should avoid?

# Human error

- ▶ Well, let's think about that. . .
- ▶ How many times have you read an article about an investigation into an industrial or airplane crash that concluded the problem was “human error”?
- ▶ Isn't it kind of strange that problems get written off like this? As *human error*?
- ▶ I mean, when a bridge collapses due to harmonic oscillations, do we write it off as “harmonic error”, or “concrete error”?
- ▶ No, we say it was a **design error**.

# Design error

- ▶ Tacoma Narrows Bridge

## Human error, you say?

- ▶ The most infamous nuclear accident in the United States occurred at Three-mile Island in 1979.
- ▶ It was a classic “meltdown” in which the reactor core was destroyed and radioactive gas and water were vented into the environment.
- ▶ Cause? **Human error.**



# Human error, you say?

## Analyzing human error

- ▶ Don Norman was on the commission Three-mile Island commission.
- ▶ There were a host of human errors leading up to the accident.
- ▶ However, most were due to **confusing controls**, **confusing alarms**, and **confusing feedback** from the controls of the reactor plant.

## Towards an engineering discipline

- ▶ **Concrete fails**, so we design with that in mind.
- ▶ **Humans make errors**, why can't we design with that in mind?
- ▶ It is essential to **model** humans and their **errors** and how to prevent them (through proper **signifiers**, **feedback**, and **mapping**).
- ▶ When human error is a think of the past, **humans will actually get a much better deal in the bargain**.



## What's at stake?

- ▶ **Good HCIs** are a pleasure to use.
- ▶ I don't think that's an exaggeration: good user interfaces actually **give pleasure** to the user through their use.
- ▶ We all (I hope) have had the pleasure of using something designed extremely well, that feeling when everything just works right and **even boring tasks give pleasure**.



## What's at stake

- ▶ **Bad HCI**, however, create frustration and anger in the user.
- ▶ It conflicts with our **conceptual model** of what we are using, and they provide inadequate or confusing **feedback**.
- ▶ We **take them for granted**.
- ▶ We should **not** take this for granted, and we should **not** put up with bad HCI design.

The screenshot displays a complex software interface for order management. At the top, there are several tabs: 'Order# 9904234', 'New', 'Repeat Selection', 'OCB', 'SSF View', 'Dupes Load', 'View Invo', 'Routing Sheet', 'Print Bill', 'Call Log', and 'Cancel'. Below these are various input fields and buttons for order details, including 'Make From SC To SC', 'Ship Method', 'Service', 'From', 'To', 'Deliver By', and 'Clock Stop'. There are also buttons for 'Print', 'Add', 'Close', and 'Post'. A table at the bottom shows order items with columns for 'Units', 'Type', 'H Description', 'Price', 'Qty', 'Discounts', 'Chrgs', 'Rate', and 'Charge'. The table contains three rows of data for 'CHATE' items. At the bottom right, there is a summary bar with fields for 'Accs', 'Dy', 'S', 'M', 'T', 'R', 'F', 'S', 'D', 'Total', and 'Sub Total'.

Units	Type	H Description	Price	Qty	Discounts	Chrgs	Rate	Charge
1	CHATE	CHATE	91	94	97	50.00	40.50	
1	MAN	MAN PLO				40.00	40.00	
2	CHATE	CHATE	500	1.426	32484	3.426	500.00	793.00

## What's at stake

- ▶ Of course, it's not just **frustration** and mild annoyance at stake.
- ▶ Think about all of the HCIs used every day **in whose hands lives are being placed**.
- ▶ We cannot afford to take design of HCI lightly, we must relegate "human error" to the past with a **modern theory of design**.



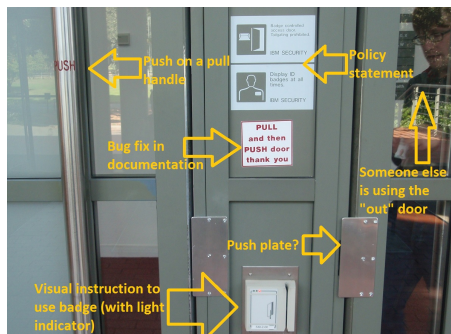
## A motivating example

- ▶ Let's say that I were to be transported instantly from this lecture hall into the cockpit of a commercial airliner.
- ▶ Let's also imagine that the plane is (for some reason) also in flight at the time.
- ▶ I feel safe saying that I would have absolutely no idea how to do **anything** useful with **any** of the controls.



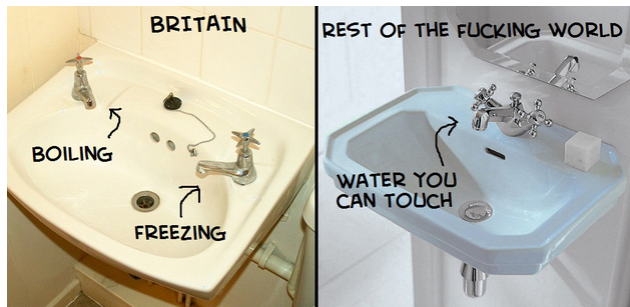
## Back to reality

- ▶ OK, the 747 cockpit is an **exaggerated example**.
- ▶ However, how many times can you remember being in a situation where you just couldn't figure out how to work something **simple**?
- ▶ Our canonical example is, of course, the **Norman Door**.
- ▶ Sometimes even with **experimentation** you can't figure out how to make the damn door do what you want.



## More objects designed to frustrate

- ▶ This design is unhappily common in the UK.



## More objects designed to frustrate

- ▶ Nothing makes one feel more **enfeebled and in need of adult supervision** that this type of failure.



## More objects designed to frustrate

- ▶ I have lived in four countries and traveled extensively: **not one of these is easy to use:**





# Getting to the point

- ▶ Why on earth does this happen?
- ▶ How is it be that objects as common as doors can **confound** us so?
- ▶ There are four **conceptual tools** that can help us analyze this:
  - ▶ **Affordances**: an innate **interaction potential**.
  - ▶ **Signifiers**: signs that let us know that an affordance **exists**.
  - ▶ **Mappings**: to leverage our ability to reason about our **3D world**.
  - ▶ **Feedback**: to help us **make sense** out of the **results** of interactions.

## Gauging potentialities

- ▶ Our world is filled with objects, many natural, the rest artificial.
- ▶ Every day we encounter thousands of objects, and though some are similar to ones we already know, many are **unique**.
- ▶ How do we manage to **figure out** how to use these new objects?



## Gauging potentialities

- ▶ The term **affordance** refers to the **relationship** between a physical object and a person.
- ▶ **Important:** an affordance is a *relationship* between the properties of an object and the capabilities of the user that determine how the object could be used.
- ▶ A **chair** *affords* support and, therefore, *affords* **sitting**.
- ▶ Most chairs can also be carried by a single person (they afford lifting), but some can only be lifted by a strong person or by a team of people.
- ▶ So, the existence of an affordance depends on **both** the object **and** the interacting user.

## Affordances are jointly determined

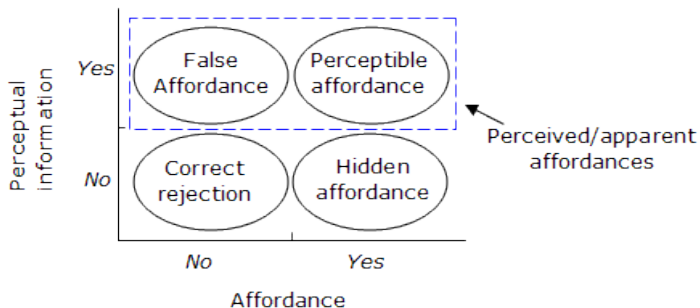
- ▶ This bears repeating and reflection: *the existence of an affordance is jointly determined by the qualities of the object and the abilities of the interacting user.*
- ▶ This relational definition of affordance can be tricky because we are used to thinking that properties are associated with objects.
- ▶ But **affordance is not a property, an affordance is a relationship.**
- ▶ Whether an affordance exists depends upon the properties of both the object and the agent.
- ▶ An affordance can be seen as a **perceived action possibility.**
- ▶ As such, its existence clearly depends on both the **object** and the **user** perceiving it.

## Perceivable affordances

- ▶ Glass affords transparency, and at the same time, its physical structure blocks the passage of most physical objects.
- ▶ So, glass affords both “seeing through” and “support”, but not the passage of physical objects (this “prevention” of interaction can be called an **anti-affordance**).
- ▶ To be effective, affordances and anti-affordances must to be **discoverable**, which is difficult for glass.
- ▶ We like glass because of its relative invisibility, but this aspect also hides its anti-affordance of blocking passage.
- ▶ As a result, birds (and people) often try to pass through windows.

## Perceivable affordances

- ▶ Affordances exist even if they are not visible, but for designers, their visibility is critical: *visible affordances provide strong clues to the operations of things.*
- ▶ If an affordance or anti-affordance cannot be perceived, a means of signaling its presence is required: this we call a **signifier**.



## From affordances to signifiers

- ▶ But the reality is that affordances exist even if they are not visible.
- ▶ For **designers**, however, their visibility is crucial because visible affordances provide strong clues about what is possible, about **how things can be manipulated**.
- ▶ **Perceived affordances** help users figure out what actions are possible – without the need for labels or instructions.
- ▶ Affordances **signal** the possibilities of interaction to the user.
- ▶ These factors we will call **signifiers**, and they may be explicit or implicit (or somewhere in between).

## Affordances versus Signifiers

- ▶ Many people find the concept of affordances difficult to grasp because they are **relationships** and **not properties**.
- ▶ As designers, we want to deal with fixed properties: we want to “put this affordance there, and that other affordance there.”
- ▶ But, as a relationship we **can't** really do this: affordances are not inherent in a single object or interface – they are **relationships**.
- ▶ After Don Norman first introduced the term **affordance** to the design world, he found designers re-appropriating it.
- ▶ **Communicating** what is possible or how to do something is **signifying** the existence of an affordance – but this is **not** the affordance itself.
- ▶ Affordances **determine** what actions are possible, while signifiers **communicate** where or how the action should take place.



# A tangible example of affordances

[Moodle and Github login examples]

# Signposts

- ▶ People need some way of understanding the objects they wish to use: some sign of what it is for, what is happening, and what alternative actions are.
- ▶ **People search for clues**, it is anything that might signify meaningful information that is important.
- ▶ **Designers need to provide these clues**: what people need, and what designers must provide, are signifiers.
- ▶ Signifiers can be deliberate and intentional, like **PUSH** on a door.
- ▶ Or they may be accidental like the **presence** or **absence** of people waiting at a train station to signifying if we have missed the train.

## Signifiers are needed

- ▶ No, signifiers are desperately needed:



## A perceived affordance

- ▶ The handle tells us to **push** or **pull**.
- ▶ Perception of affordances can have strong cultural constraints



## To summarize

- ▶ **Affordances** are the possible interactions between people and the environment – some are perceivable, others are not.
- ▶ **Perceived affordances** often act as signifiers, but they can be ambiguous.
- ▶ **Signifiers** signal things, in particular what actions are possible and how they should be done.
- ▶ **Signifiers must be perceivable**, else they fail to function.

## Physical and virtual space

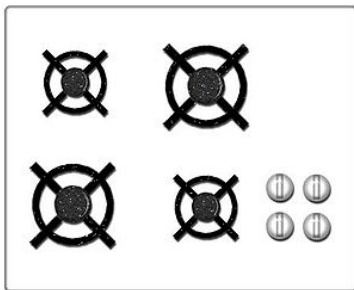
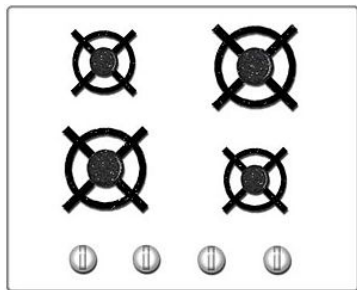
- ▶ Mapping is a term borrowed from mathematics which means the **relationship** between elements from two sets of things.
- ▶ The mapping of switches to lights specifies **which switch** controls **which light**.
- ▶ Or even better: the switch to control the **projector screen**.
- ▶ Mappings like these create implicit or explicit correspondences between **controls** (virtual or physical) and the **physical world**.

## Mapping in user interfaces

- ▶ Mapping is an important concept in the design and layout of controls and displays.
- ▶ When mapping uses spatial correspondence between the layout of the controls and the devices being controlled, it is easy to determine how to use them:
  - ▶ **Example:** to steer a car, we rotate the steering wheel clockwise to turn right – the top of the wheel moves in the same direction as the car, **this** is the **mapping**. Other choices could have been made: In early cars, steering was controlled by a variety of devices, including tillers, handlebars, and reins.
  - ▶ **Example:** bulldozers and military tanks have two **tracks** instead of wheels and use separate controls for the speed and direction of each. To turn right, the left track is increased in speed, while the right track is slowed or even reversed. This **mapping** is more complex, but still **discoverable**.

## Natural mapping

- ▶ **Natural mapping** takes advantage of spatial analogies and leads to immediate understanding.
- ▶ In such cases we say that the functioning of the device/interface is **discoverable** through use.





# Natural mapping

- ▶ Some natural mappings are **cultural** or **biological**, like how moving the hand up signifies more, moving it down signifies less.
- ▶ Other natural mappings follow from the principles of perception and allow for the **natural grouping** of controls and feedback.
- ▶ Groupings and proximity are important principles from **Gestalt psychology** that can be used to map controls to function: *related controls should be grouped together, and controls should be close to the item being controlled.*

## Natural mapping and proxies

- ▶ A device is easy to use when the set of possible actions is visible (i.e. when **affordances** are **perceived** and **real**), and when the controls and displays exploit natural mappings.

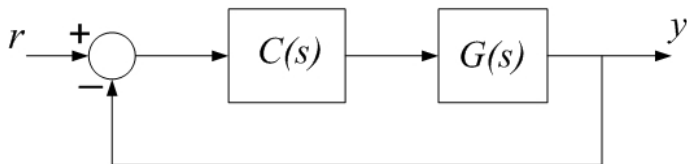


## On our way

- ▶ We are slowly moving away from **abstract** definition of interaction with everyday objects towards a **concrete** philosophy of design.
- ▶ We can already start trying to formalize how we should think about designing human computer interactions:
  1. Identify the **affordances** that should be present in the device or object or system.
  2. Ensure that affordances are **perceived** correctly, and ensure that all **perceived affordances** are not **false affordances**.
  3. When affordances are not evident, add **signifiers** to signal their presence.
  4. Use **natural mapping** between controls and outcomes to ensure **discoverability** of function.
  5. Repeat.
- ▶ We are still missing a few things: **feedback** for the user to indicate what has gone wrong (or right), and **conceptual models** in which to interpret affordances and mappings.

## Feedback and frustration

- ▶ How many times have you seen a person waiting for an elevator repeatedly press the Up button?
- ▶ How many times have **you** repeatedly pressed the pedestrian crossing button while waiting at a crosswalk?
- ▶ Have you ever wondered, while waiting at an intersection in your car, whether the sensors in the street have sensed you waiting in your car?
- ▶ What is missing in all of these cases is **feedback**: some way of letting you know that your request has been received and something is being done about it.



## Feedback and control

- ▶ Feedback is the process of **communicating the results of an action**.
- ▶ It is a well-known concept from the science of control and information theory.
- ▶ Even as simple a task as picking up a glass with the hand requires feedback to aim the hand properly, to grasp the glass, and to lift it.
- ▶ The human nervous system is equipped with numerous feedback – and we are **conditioned** to rely on feedback to accomplish **all tasks** from simple to complex.
- ▶ Given the importance of feedback, it is amazing how many **everyday objects and interfaces ignore it**.

# A running example: simple enough

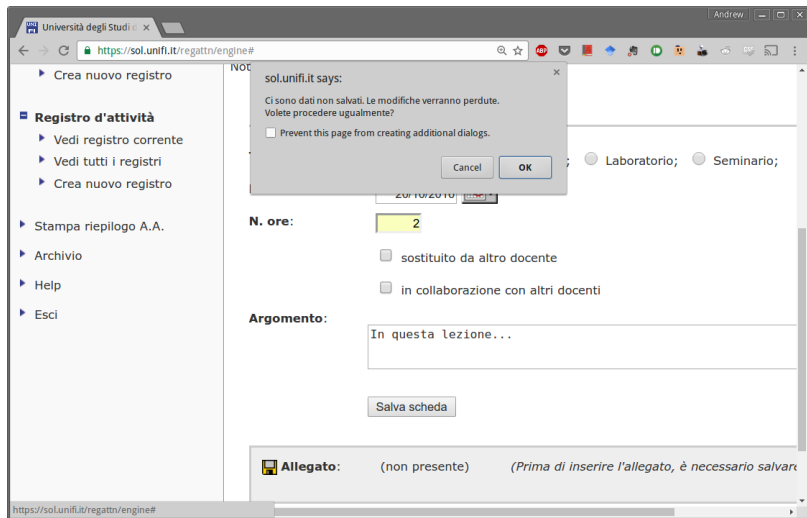
The screenshot shows a web browser window with the URL `https://sol.unifi.it/regattn/engine#`. The page title is "Università degli Studi". The main content area is titled "Note (eventuale sede del corso):" and contains the following elements:

- Navigation links: [ [Indice del registro](#) ] [ [Indice delle voci](#) ]
- Tipologia:** Radio buttons for  Lezione;  Esercitazione;  Laboratorio;  Seminario;
- Data:** Text input field containing "20/10/2016" and a calendar icon.
- N. ore:** Text input field containing "2".
- Checkboxes:  sostituito da altro docente;  in collaborazione con altri docenti.
- Argomento:** Text area containing "In questa lezione...".
- Salva scheda** button.
- Allegato:** (non presente) *(Prima di inserire l'allegato, è necessario salvarlo)*

The left sidebar contains a menu with the following items:

- ▶ Crea nuovo registro
- **Registro d'attività**
  - ▶ Vedi registro corrente
  - ▶ Vedi tutti i registri
  - ▶ Crea nuovo registro
- ▶ Stampa riepilogo A.A.
- ▶ Archivio
- ▶ Help
- ▶ Esci

# A running example: dammit!



# A running example: is it saved?

Università degli Studi x

Andrew

https://sol.unifi.it/regattn/engine#

Note (eventuale sede del corso):

[ Indice del registro ] [ Indice delle voci ]

**Tipologia:**  Lezione;  Esercitazione;  Laboratorio;  Seminario;

**Data:** 20/10/2016

**N. ore:** 2

sostituito da altro docente

in collaborazione con altri docenti

**Argomento:**

In questa lezione...

Salva scheda

**Allegato:** (non presente) *(Prima di inserire l'allegato, è necessario salvarlo)*



## Immediate, informative feedback

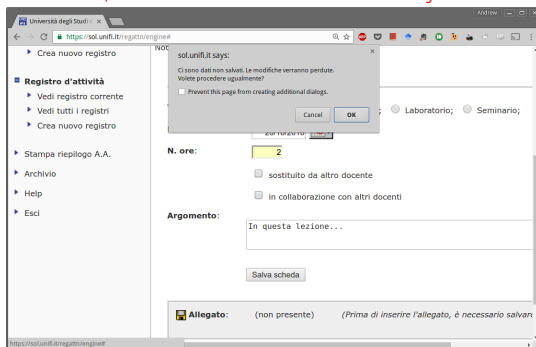
- ▶ **Feedback must be immediate:** even a delay of a tenth of a second can be disconcerting.
- ▶ If the delay is too long, people can give up and go do other activities – and **this is annoying to the people.**
- ▶ **Feedback must also be informative:** simple visual flashes or beeps are usually more annoying than useful – they tell us something has happened, nothing about **what** has happened.
- ▶ **Visual and auditory feedback is risky:** in many cases we are uncertain about which device has created the sound; if the signal is visible, we may miss it unless our eyes are on the correct spot at the correct time.
- ▶ **Poor feedback can be worse than no feedback at all:** it is distracting, uninformative, and in many cases irritating and anxiety-provoking.

## Just the right amount

- ▶ **Too much feedback** can be even more annoying than too little.
- ▶ Examples are **inappropriate, uninterpretable feedback**.
- ▶ Are we all familiar with the concept of the **backseat driver**? They are often correct, but their remarks and comments can be so continuous that instead of helping, they become an irritating distraction.
- ▶ **Machines that give too much feedback are like backseat drivers**: it is distracting to be subjected to continual flashing lights, text messages, or beeps and boops.
- ▶ Too many announcements **cause people to ignore all of them**, or wherever possible, disable all of them – which means that **critical and important ones might be missed**.

## Final words on feedback

- ▶ **Feedback has to be planned:** all actions need to be confirmed, but in a manner that is unobtrusive.
- ▶ **Feedback must also be prioritized:** so that unimportant information is presented in an unobtrusive fashion, but important signals are presented in a way that does capture attention.
- ▶ **Feedback is essential, but it has to be done correctly.**



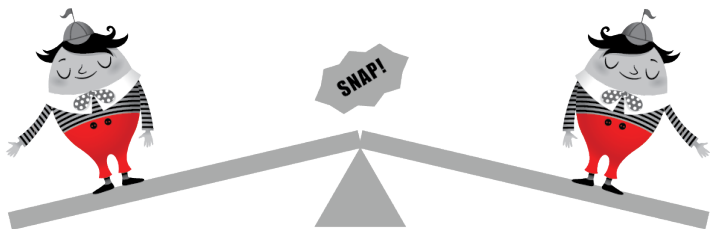
# Needfinding

# Needfinding

- ▶ **Needfinding** is the process of observing people to discover their needs, goals, and values.
- ▶ It is often associated with the process of developing new products or even new businesses.
- ▶ The main element is the **investment of significant time, effort or money** in the development of **something new**.
- ▶ Whether a new product or a new HCI system, it always makes sense to understand whether a genuine **need** exists.
- ▶ A good starting point is to clearly identify an **existing problem or need**, because finding a big problem and need often yields important untapped opportunities.
- ▶ Observing people also helps build empathy and think from their point of view.
- ▶ So, **how do we observe people and identify their needs?**

## Needfinding (why?)

### The twin anti-poles of design failure



**Doing precisely  
what the user asks**

**Assuming you know what's  
best and ignoring the user**

## Needfinding: participant observation

- ▶ It is essential to **observe the users and their behavior in context** (performing the activity).
- ▶ This is vital to learning and understanding their experience.
- ▶ While observing, we seek answers to these questions:
  - ▶ What do people do now?
  - ▶ What values and goals do people have?
  - ▶ How are these particular activities embedded in a larger context?
  - ▶ What similarities and differences are there across people?
  - ▶ Are there any hacks or workarounds used?



Observations



Interviews



Extreme users, lead users

## Needfinding: What do people do now?

- ▶ Getting a full understanding of the culture, practices, and rituals of your target audience provides an understanding of your audience that is extremely helpful.
- ▶ Digging deep into the motivations, emotions, and aspirations of your audience allows you to better understand where to begin.
- ▶ It is essential to understand how **real people** manage this process **today** – this implies understanding how **all** types of users.
- ▶ Too many software systems are implemented and deployed without a complete understanding of **how things are currently managed**.
- ▶ Without understanding this, it is hard to expect improvement over the *status quo*.



## Needfinding: What values and goals?

- ▶ Most often, we want to build technologies that align with what people care about and what they hope to accomplish.
- ▶ This doesn't mean literally building what people have asked for: people often (usually) don't know how to achieve their goals – especially for disruptive technologies.
- ▶ Instead, we must design technologies that will weave themselves into the fabric of everyday life, even if they introduce new concepts and functionality.
- ▶ **Main point:** people cannot be relied upon to tell us **how** they should accomplish their goals, but through observation and interview we **can** uncover what those goals are, and what **values** should be preserved (e.g. saving time).

## Needfinding: What is the context?

- ▶ For a public transportation user, a bus or subway segment is a **part of a larger activity** like getting to a friend's house, commuting to work, or going to the grocery store.
- ▶ By understanding the constraints and goals of the large activity, we can **derive ideas that are otherwise missed** if we think narrowly about the bus ride.
- ▶ By figuring out why someone would choose to take the bus or not take the bus, we as designers might end up with something more broad, like creating a mobile application that helps people figure out when a bus is coming.
- ▶ Taking this broader view can help us be more effective as designers by helping to **design for the larger activity that people are engaged in**.

## Needfinding: Hacks?

- ▶ Uncovering **hacks** that people have discovered for accomplishing tasks is a **gold mine for designers**.
- ▶ This is because hacks represent methods and techniques that **accomplish actual goals** and **respect the values** of people in the system.
- ▶ Often, these can be translated directly into features in our systems.



## Needfinding: observing the experienced

- ▶ A good strategy for finding needs is to be an apprentice under someone who has experience with the area.
- ▶ One illustration of the power of being in the presence of an expert comes from Jack Whalen from Xerox PARC:
  - ▶ While studying a call center for photocopier repair, he found that diagnosing photocopiers over the phone is really hard.
  - ▶ Unsurprisingly, after studying the support staff they found that the most effective person was the person who had been working there the **longest**.
  - ▶ To their surprise, however, the second most effective person wasn't the person that had been working their the second longest, but the person who had been **sitting next to** the most effective person.
  - ▶ Thus, by sitting next to an expert, these repair technicians were able to pick up **informal skills** of doing repair work that **aren't written down anywhere**.

## Needfinding: observing the experienced

- ▶ If you see something that catches your eye while being an apprentice, be sure to **interact** and **validate** it so you can better understand why things are done the way that they are.
- ▶ Additionally, pay attention to all the **artifacts** that compose people's work, because the ways that people have **hacked** their equipment to make their work more effective is an indication of ways to innovate.
- ▶ For example, **Post-Its** have traditionally been used as artifacts to help users more easily navigate different routines.
- ▶ A harmful attitude about designing, although less common nowadays, is the tendency to **think people who can't figure out how to use technology are simply incompetent.**

## Needfinding: observing the experienced

- ▶ Using another example from Xerox PARC, Lucy Suchman recorded a video (that has now become legendary) of two people trying to produce a double sided copy of 50 pages of paper:
  - ▶ VIDEO
  - ▶ According to legend, when Lucy shared the video with the executives, they wrote the users off as **dumb**.
  - ▶ However, when it was revealed that the two users were **Allen Newell** and **Ron Kaplan** – two of Xerox's premier research scientists, they were no longer able to say that it was because the users did not know how to use technology.
  - ▶ Ultimately, this video shows that if you are **unfamiliar** with a particular piece of technology, it can be difficult to figure out how to use it **without an intuitive user interface**.

## Needfinding: interviewing

- ▶ Of course, apprenticing with a company is probably a luxury for most people.
- ▶ Another way of observing people to identify needs is **interviewing**.
- ▶ When it comes to interviewing, one key distinction that must be made is the difference between **what people say** and **what people do**.
- ▶ For example:
  - ▶ Walmart conducted a study asking its customers **whether they would like the aisles to be less cluttered**.
  - ▶ Unsurprisingly, the participants of the study responded: **well, yes**.
  - ▶ Walmart then proceeded to declutter their aisles, remove inventory, and **lost a billion dollars in sales**.

## Needfinding: interviewing

- ▶ What happened? In this situation, Walmart made two key mistakes:
  - ▶ First, they **listened to what people said** rather than paying attention to what people **did**.
  - ▶ Second, they asked a **leading** question.
  - ▶ Those two mistakes led them to do exactly the **opposite** of what would be most effective.

*"If I asked people what they wanted, they would have said faster horses." – Henry Ford*

### The twin anti-poles of design failure



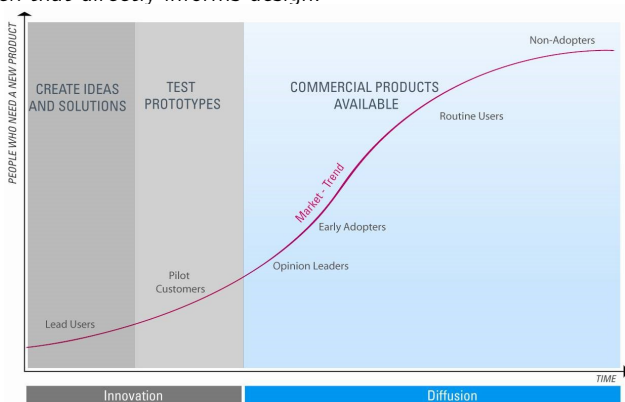
**Doing precisely  
what the user asks**

**Assuming you know what's  
best and ignoring the user**



## Needfinding: interviewing

- ▶ The key is to be **open to any insights** that users or potential users may have.
- ▶ One group of users, known as **lead users**, can be extremely helpful in the interview process as well as the development process.
- ▶ Lead users are the **knowledgeable early adopters** that can help provide valuable feedback that directly informs design.



## Needfinding: asking good questions

- ▶ Questions to avoid:
  - ▶ **Questions that ask what users would do**, like, or want in hypothetical situations. These questions often generate replies based on a person's ideal, hypothetical world, not necessarily what they would actually do.
  - ▶ **Questions that ask how often users do things**. Asking a user how often they go to the gym generates responses that reveal how often they **wish** they went to the gym, not actually how often they do.
  - ▶ **Questions that ask how much they like things on an absolute scale**. Asking a user how much they like something is often not extremely valuable or insightful.
  - ▶ **Binary questions**. Questions that only have two answer choices, like questions that ask on an absolute scale do not provide any significant insights into how you should build your product.

## Needfinding: asking good questions

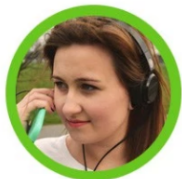
- ▶ Good questions are questions that are **open-ended**, have a specific goal, and are unbiased.
- ▶ For example, instead of asking how **often** a person goes to the gym, you should ask them to tell you about the **most recent time** that they went to the gym.
- ▶ Having a specific goal with each question allows you to uncover specific things with each question.
- ▶ If a question leaves your interviewee **stumped** for a little bit, that is a good thing.
- ▶ One common problem for new interviewers is that they often say that there's nothing to be found for the problem they're tackling because it's either **impossible** or **obvious**.

## Needfinding: personas

- ▶ The final step, after conducting as many user interviews as you possibly can, is to develop **personas**.
- ▶ Personas are fictional characters that you use to represent the **demographics** of your users.
- ▶ For example, if you are developing a messaging application, your personas can include **teenage males** who are using your messaging app to communicate with friends, as well as **middle-aged mothers** who are using the app to message their kids.

## Needfinding: personas

- ▶ Example persona:



**ZOE**

age 18-22, single female, living with friends

Zoe studies as a graphic designer at a small art school. She aspires to one day work at an agency and eventually run her own.

She is constantly using dribbble to share her work and explore the work of others on her Macbook Pro. She regularly uses Photoshop and illustrator to create different designs, often showing her close friends before uploading it to dribbble.

She regularly uses a notebook to keep track of any sketches or ideas she may have regarding something that she would like to design.

## Needfinding: personas

- ▶ The value in having personas is being able to quickly **pinpoint different use cases** among different demographics of people.
- ▶ By assigning a **concrete name** to a certain demographic, it becomes easier to keep that specific demographic in mind.
- ▶ Personas should be as detailed as possible, and be a compilation that represents **different groups** of your users.
- ▶ Beyond just demographic information, however, a persona should also capture a person's motivations, beliefs, intentions, behavior, and goals.
- ▶ In other words, **give your persona a story to tell.**

## Needfinding: Personas

- ▶ **Personas are humanized descriptions of users.**
- ▶ They are useful for anything that has to do with making stuff or selling stuff.
- ▶ In marketing classes, we learn that the way you describe a market is something like “males 25-35 with incomes over \$30,000/year”.
- ▶ But who is that guy? How would you figure out what innovative new product or feature he might like?
- ▶ Personas are used extensively in marketing, but they are an extremely useful tool for identifying **archetypal** users of your systems.
- ▶ Here I will give some general advice for building personas for your design process.

## Needfinding: Personas

- ▶ Good personas **tell a story**.
- ▶ This isn't to discount the importance of being **rigorous and quantitative**, it's to anchor that rigor in a **meaningful starting point** and identify the right tools for that starting point.
- ▶ Avoid bullet points: this "plastic persona" you see is the **opposite** of what you want.



- Women
- Age 28-45
- Have kids
- Socialize with other mom's
- Online with Facebook
- 86% said they'd like to be more organized
- 70% said they'd use an application that organizes them

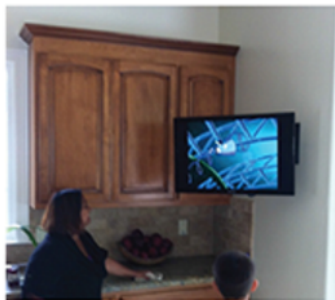


## Needfinding: Personas

- ▶ The **photo** is clipart from the Internet, the description is a set of **generic** bullet points and they don't say much of anything about who this person really is or what they **need**.
- ▶ **Get real photos**, create a collage of real stuff and if you're saying that your target persona Tweets or posts certain types of content online, get examples.
- ▶ If you're saying that they hate doing paperwork, get samples of the kind of paperwork they have to do.
- ▶ Write **full sentences** that deliver a **narrative**.

## Needfinding: Personas

- ▶ The 'organic persona' is a better start: an actual paragraph of description, and a photo was taken in the field, which is **where you should be when you're developing personas**.
- ▶ One of the top failure modes of persona creation is that they have their **humanity peeled away or just never acquire it**.
- ▶ **Giving them a name is a trick that helps humanize them**.



Mary is a mom by choice. She had a successful career in accounting, but welcomed the opportunity to be a stay at home mom. She loves it. But it's not like having kids purged her creative, social instincts. She wants to connect, she wants to learn, she wants to interact. Being a mom is a job and she wants to do it well. That means corresponding with other mom's on relevant topics and keeping the family calendar in ship shape. She posts to Facebook at least twice a week and responds to other moms' items more often than that. . . She. . .

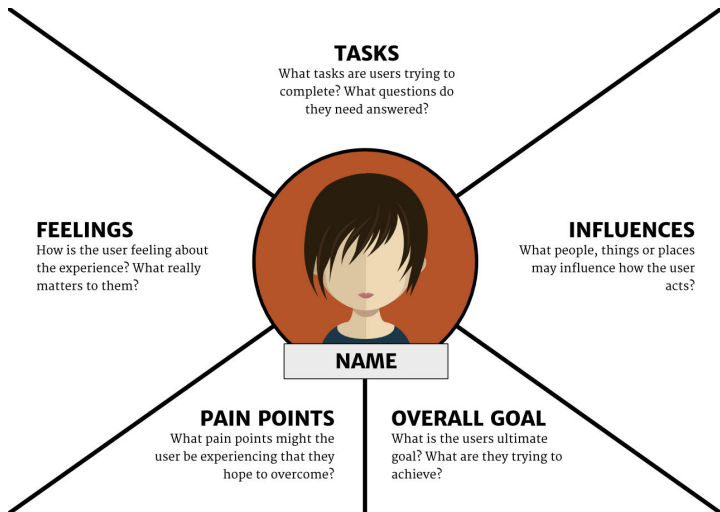
## Needfinding: Personas

- ▶ Once you've humanized your persona, you'll want to operationalize it in your particular area (online banking, dating, shopping for power tools, whatever).
- ▶ The **Think-See-Feel-Do checklist** is a good way to frame this.

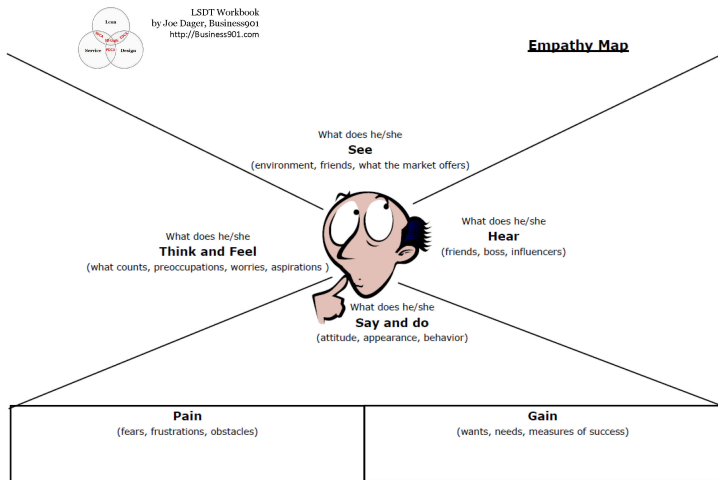
<b>THINK</b>	What is their point of view on your area of interest? What do they like, dislike about it? What's the difference in their view between how it is and how it should be?
<b>SEE</b>	What influences and informs them about your area of interest? Where do they get that? Peers? Media?
<b>FEEL</b>	What are the underlying emotional drivers in the area? How does that influence what they do and their interest in alternatives?
<b>DO</b>	When you observe them, what do they actually do?

- ▶ [This website](#) has excellent resources for working with personas.

# Needfinding: Empathy Maps



# Needfinding: Empathy Maps

Adapted from [xplane.com](http://xplane.com)

# Summary

- ▶ **Needfinding** is an essential step if you want to make *any* claim to be doing **Human Centered Design**.
- ▶ The **goal** of needfinding is to arrive at the **personas** that constitute the **model** of our users.
- ▶ Having this model is **how** we put the **human** in our design.
- ▶ And, in the end, how we can – by incorporating human factors into our design – hope to put an end to **human error** as an excuse for **design failure**.
- ▶ **Important:** none of this is **easy**.

# Usability Analysis

# Usability testing: basics

- ▶ What is **usability testing** in software and what are the benefits to the **end user**?
- ▶ Think of it as carefully designed tests to determine whether the application built is **user-friendly** or not.
- ▶ Usability Testing is a **black box** testing technique.
- ▶ It also reveals how comfortable users are in terms of the **flow**, **navigation** and **layout**, **speed** and **content**.
- ▶ Usability Testing tests the following features of the software:
  - ▶ How easy it is to **use** the software.
  - ▶ How easy it is to **learn** the software.
  - ▶ How **convenient** is the software to end user.



## Usability testing: benefits

- ▶ Usability testing includes the following five components:
  - ▶ **Learnability**: Can users accomplish basic tasks the first time they encounter the design?
  - ▶ **Efficiency**: How fast can experienced users accomplish tasks?
  - ▶ **Memorability**: When a user returns to the design does she remember enough to use it?
  - ▶ **Errors**: How many errors do users make, how severe are these errors and how easily can they recover from them?
  - ▶ **Satisfaction**: Does the user like using the system?
- ▶ Benefits of usability testing to the end user or the customer:
  - ▶ Better **quality** software.
  - ▶ Software is **easier** to use.
  - ▶ Software is **more readily accepted** by users.
  - ▶ **Shortens the learning** curve for new users.

## Usability testing: other advantages

- ▶ Advantages of usability testing:
  - ▶ Usability tests can cover **other types of testing** such as functional testing, system integration testing, etc.
  - ▶ Usability testing can be very **economical** if planned properly.
  - ▶ If proper resources (experienced and creative testers) are used, usability tests can help even **before the system is finally released** to the user.
  - ▶ Usability testing can discover **potential bugs** in the system not visible to developers.
- ▶ Usability testing is broad area and requires a **high level of understanding** of the field along with a **creative mind**.
- ▶ In this lecture I will introduce the basic concepts and try to give some concrete advice for designing your own usability tests.

## Usability testing: bottom line

- ▶ The biggest challenge designers face isn't how technology works – **it's how humans work.**
- ▶ What users **say** versus what they **do** are two completely different things, and **the only way to verify is to test.**
- ▶ Usability testing is more than a just a checkbox on a list of product requirements – **it is the most convincing support for your design decisions.**
- ▶ Just like in software development, for usability analysis is it essential to **test early and test often.**
- ▶ Every product is different, so **there is no magical usability test** that will tell you everything you need to know.
- ▶ Usability testing helps you see the bottom line of **whether your design works or doesn't.**

## Defining goals for usability testing

- ▶ The first step in usability research should always be knowing **what you want to get out of it**.
- ▶ This isn't always as easy as it sounds: you must **categorize your testing goals** and know what type of data is **most appropriate**.
- ▶ **Michael Margolis** (Google Ventures Design Studio) believes the first step to determining objectives is knowing the right questions to ask.
- ▶ It is important to focus the team on **research questions**:
  - ▶ Research question: **Why do people enter the website and not watch the demo video?**
  - ▶ Dictating methods: **We need to do focus groups now!**

## Knowing what to measure for UX testing

- ▶ Now that you roughly know your goals, you must decide **how** to apply usability testing to accomplish them.
- ▶ What **type of feedback** would be most helpful for your results?
- ▶ Does your team need a **graph** or a **rating scale**? Personal **user accounts** or **numbers**? **Written responses** or **sound bites**?

Type	Example	Results
<b>Verbal Response</b>	Describe and demonstrate what, if anything, was most frustrating about this site.	Spoken answers correlate with where a participant is at in the study. Make great clips for a highlight reel.
<b>Multiple Choice</b>	Do you trust this company? • Yes • No	Great for collecting responses that are <b>categorical</b> . These can be nominal (cats or dogs?) dichotomous (yes or no) and even ordinal ( <b>Likert</b> scale agree/disagree).
<b>Rating Scale</b>	How likely are you to return to this site again? 1    2    3    4    5 <i>Not at all likely</i> <i>Very likely</i>	Good for collecting ordinal variables (low, medium, high) and are very recognizable especially within the United States.
<b>Written Response</b>	What do you think is missing from this page, if anything?	Good for running post-study analysis. How many people used the same answers? Quick quotes for building user stories.

## Usability metrics

- ▶ These are **quantitative** data about usability.
- ▶ **Usability metrics** are statistics measuring a user's performance on a set of tasks. The website [usability.gov](http://usability.gov) lists some of the most helpful:
  - ▶ **Success Rate**: In a given scenario, was the user able to complete the assigned task?
  - ▶ **Error Rate**: Which errors tripped up users most? These can be divided into two types: **critical** and **noncritical**. Critical errors prevent a user from completing a task, while noncritical errors simply lower the efficiency with which they complete it.
  - ▶ **Time to Completion**: How much time did it take to complete the task?
  - ▶ **Subjective Measures**: Numerically rank a user's satisfaction, ease-of-use, availability of information, etc.

# Usability metrics: task summarization

- ▶ Task summaries look like this:



## Quantifying qualitative data: SEQs

- ▶ The **Single Ease Question (SEQ)** is a 7-point rating scale to assess how difficult users find a task.
- ▶ It's administered immediately **after** a user attempts a task in a usability test.
- ▶ After users attempt a task, ask them this simple question: Overall, how difficult or easy was the task to complete?

Overall, how difficult or easy did you find this task?

Very Difficult							Very Easy
1	2	3	4	5	6	7	
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

- ▶ **Labels and values:** label the end points only and provide numbers from 1 to 7. Slight changes are far outweighed by the very salient event of the just attempted task. Users have little problem expressing their **frustration** or **delight**.



## Quantifying qualitative data: SEQs

- ▶ **SEQs work well:** Despite their simplicity, the SEQ performs about as well or better than more complicated measures of task-difficulty (see [this CHI paper](#) for a comparison).
- ▶ **Ratings of difficulty correlate with other metrics:** the correlation between user responses on the SEQ and task-time and task-completion is around  $r = .5$ .
- ▶ **Users respond differently:** Some users will make everything a 6 or 7 while others will use the full range of the scale. It's very common for people to use rating scales differently, but these differences tend to average out across tasks.
- ▶ **SEQs are technology agnostic:** A feature of task-difficulty ratings is that users tend to respond to what they expect given the device, fidelity of the interface, and nature of the task.

## Quantifying qualitative data: SEQs

- ▶ **Ask Why?**: When users rate a task difficult, it's good to know why they did. When a user provides a rating of less than 5, it's a good idea ask them to briefly describe why they found the task difficult.
- ▶ **If you're going to ask something**: There is quite a bit of evidence that if you're going to ask the users **one question** about their experience with a task, **it should be the SEQ**.
- ▶ **Use it for everything**: You can use SEQs to ask about many aspects of usability: the scale can also be "1. Strongly Disagree – 7. Strongly Agree"
- ▶ **Mix it up**: don't always ask questions that demand the **same response** (1 or 7).

# UX Testing: The Pilot Test

- ▶ No matter what type of test you choose in the end, you should always start with a **pilot test**.
- ▶ Pilot testing is like a **test run** of your greater user test.
- ▶ You will conduct the test and collect the data in the same way you would a real test, but the difference is that you **don't analyze or include the data**.
- ▶ In most cases, **something will go wrong with your first test**.
- ▶ If you want the most reliable data, run a pilot test or two until you feel **you** understand the process and have **removed all the kinks**.

## UX Testing: Types of tests

- ▶ There are four general types of usability tests:
  - ▶ **Scripted use of the product:** These tests focus on specific usage aspects. The degree of scripting varies, with more scripting generating more controlled data.
  - ▶ **Decontextualized use of the product:** Tests that don't use the product – at least in the actual testing phase – are designed for broader topics and generating ideas.
  - ▶ **Natural (and near-natural) use of the product:** Seek to analyze common usage behaviors and trends with the product, doing well with data authenticity at the cost of control.
  - ▶ **Hybrid:** Hybrid tests are creative and non-traditional tests geared towards understanding user mentality.

## UX Testing: Types of tasks

- ▶ Each type of test is divided into tasks.
- ▶ The first distinction to make is whether tasks will be phrased **directly** or in terms of **scenarios**:
  - ▶ **Direct Tasks**: are instructions such as “Find a turkey recipe on the Food Network,” or “Learn about wiener dogs on the blog.” Direct tasks are more **technical** in nature, and their unnatural nature can **detract from user experience**.
  - ▶ **Scenario Tasks** phrase the instructions in a real-life example: “You’re going to a high school reunion this weekend. You want to find a nice outfit on the Macys website”. Scenario tasks are more common than direct tasks because the user can **forget they are being tested**.

## UX Testing: Types of tasks

- ▶ The next distinction to make is whether tasks will be **closed** or **open-ended**:
  - ▶ **Closed tasks**: have clearly defined success or failure. These are used for testing specific factors like **success rate** or **time**. For example (from a Yelp redesign case study): “Your friend is having a birthday this weekend. Find a venue that can seat up to 15 people.”
  - ▶ **Open-ended tasks** are ones the user can complete in several ways. These are more subjective and most useful when trying to determine how your user **behaves spontaneously**, or how they **prefer to interact with your product**. For example: “You heard your coworkers talking about Yelp. You’re interested in learning what it is and how it works.”

## UX Testing: Finding your audience

- ▶ The core component of these usability tests are **actual people**.
- ▶ To think of your participants as **merely test subjects** is a mistake – they are all individuals with their own personalities and their own way of doing things.
- ▶ When focusing in on your test group, it's important not to **obsess over demographics**.
- ▶ The biggest differentiator will likely be whether users have **prior experience** or are **knowledgeable** about their domain or industry – not gender, age, or geography.
- ▶ If you find you have more than one target group, remember to **test each group independently of each other**.

## UX Testing: Recruiting participants

- ▶ Knowing who you want for the test is only half – **you still need to get them to come.**
- ▶ These are some effective methods to find test participants:
  - ▶ **Existing Users:** By definition, these are your target users. Even if you're researching a new product, if your company has produced similar products in the past there's a chance they both target the same type of person.
  - ▶ **UserTesting.com:** A website designed specifically for this. It lets you select users by age, gender, location, and even more customizable options. The site delivers audio and video of users actually testing your site or app.
  - ▶ **Hallway Testing:** "Hallway" testing is a term that means random, as in whoever is walking by the hallway when you conduct the test. Remember that the farther you get from your target audience, the less helpful the data.



## UX Testing: Hallway testing

- ▶ Hallway testing is a usability test set-up in a high foot traffic area, utilizing **bystanders** to test your product.
- ▶ Participants will be people who **happen to be walking down the hall** and are able to afford 5-10 minutes of their day.
- ▶ Some concrete Hallway testing tips:
  - ▶ **Location is extremely important.** Choosing the optimal location for your hallway test is invaluable. Ensure your location will have heavy foot traffic. Be aware of the timing of your hallway test. Do not schedule hallway tests during inconvenient hours or major events.
  - ▶ **Plan ahead.** As impromptu as hallway testing sounds, it takes a lot of planning. Preparation for a hallway test can start as early as a month before the actual test.
  - ▶ **Set up early.** Give yourself ample time to set up and get the testing team situated. You should arrive at least 30 minutes before you are scheduled to begin testing.

## UX Testing: Hallway testing

- ▶ Concrete Hallway Testing tips (continued):
  - ▶ **Review and practice.** Run through the test script multiple times and inform team members exactly what you are looking to identify during the test. It is key to inform team members about the duties of their role.
  - ▶ **Use greeters.** Greeters play an important role during hallway tests, namely identifying and recruiting the test participants. It is important to have outgoing greeters to get people involved in the testing sessions, some personalities are better suited for this task.
  - ▶ **Be mindful of time.** The optimal time for an individual hallway test is **10 minutes**. Focus and interest tend to wane if you keep them any longer.
  - ▶ **Explain the purpose to the participant.** Let the tester know the ultimate goal of the hallway test. They have agreed to the parameters you have set for the test, do your best to keep them focused on the task at hand.

## UX Testing: Hallway testing

- ▶ Concrete Hallway Testing tips (continued):
  - ▶ **Reward your volunteer.** Participants are volunteering part of their day to aid your product, you should reward them for their kindness. Simple gestures can be effective, like handing out pens or candies.
  - ▶ **Look to improve.** Always be aware of how you can improve your testing processes. A team debriefing session should be a requirement after hallway testing sessions.
  - ▶ **Guerilla Testing Technique**

## UX Testing: Types of bias

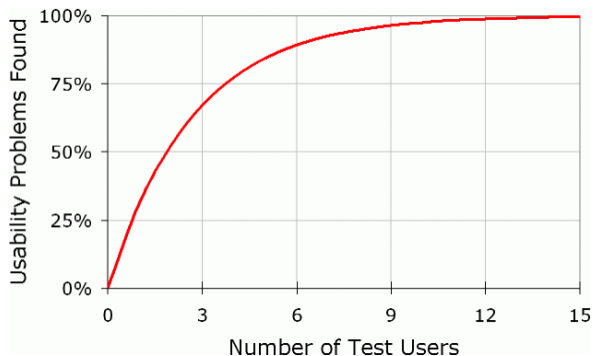
- ▶ It's important to understand the subtle biases that creep into **moderated** (and **unmoderated**) usability tests:
  - ▶ **Hawthorne Effect**: "You and all those people online and in the next room are watching my every keystroke, I'm going to be more vigilant and determined than I ever would be to complete those tasks – I'll even read the help text."
  - ▶ **Task-Selection Bias**: "If you've asked me to do it, it must be able to be done." This state of **knowing** is unnatural.
  - ▶ **Social Desirability**: Users generally tell you what they think you want to hear and are less likely to say disparaging things about people (seen and unseen) and products. Users tend to blame **themselves**.
  - ▶ **Availability**: If a user has two hours during the day to volunteer for a study, it limits the users you are testing to those who are available. (Also problematic for the jury system in the USA.)
  - ▶ **Honorariums**: If the honorarium the user receives is the sole motivator, the quality of the data can be questionable. In unmoderated studies (e.g. **Mechanical Turk**), users will **cheat** to

## UX Testing: Types of bias

- ▶ Types of bias (continued):
  - ▶ **Note Taking**: “I see you wrote something down after I did that so I must have done something wrong.” Users who are aware the moderator is taking notes may become more self-conscious about the actions they are taking.
  - ▶ **Recency and Primacy Effects**: The Recency Effect is the tendency to weigh recent events more heavily than earlier events. Conversely, weighing events that happened first more heavily is called the Primacy Effect. Users typically perform **worse** on their initial tasks.

## UX Testing: More is Not Always Better

- ▶ Testing interfaces on many users is not necessary in order to identify issues – **even 8-10 users can be enough**.
- ▶ This curve was first shown by Jakob Nielsen and Tom Landauer<sup>1</sup>:



<sup>1</sup>Nielsen, Jakob, and Landauer, Thomas K.: "A mathematical model of the finding of usability problems," Proceedings of ACM INTERCHI'93 Conference, 1993.

## UX Testing: More is Not Always Better

- ▶ Some observations:
  - ▶ **Zero** users is not enough.
  - ▶ As soon as you collect data from a single test user, **your insight skyrockets** and you have learned almost a third of all there is to find.
  - ▶ The difference between **zero and even a little bit of data** is astounding.
  - ▶ When you test the second user, you will discover overlap with the first one – but people are different and you will still learn something.
  - ▶ As you add more and more users, you learn less and less because you will **keep seeing the same things again and again**.
  - ▶ After the **fifth** user, you are wasting your time by observing the same findings repeatedly but not learning much new.
- ▶ Summary: **Elaborate usability tests are a waste of resources. The best results come from testing no more than 5 users and running as many small tests as you can afford.**

## UX Testing: Takeaway

- ▶ Design is a highly iterative process, and **all of the intuition in the world won't identify gaps in your product.**
- ▶ As designers, we are **too familiar with our own work** to be able to spot where it fails.
- ▶ The only way to truly improve a design is to test it on **real users and watch how they interact with it.**
- ▶ Testing with a live app can uncover problems and turn a **mediocre** effort into a **spectacular success** – *with only a little work.*



## The AirMouse usability test

- ▶ Let's spend a few minutes looking at a usability test **case study**.
- ▶ **AirMouse** studied how to emulate the behavior of a physical mouse using the 3D hand model generated by a **Leap Motion device**.
- ▶ **Machine learning** techniques were used to recognize the "mouse grab" pose and a small set of **mouse gestures**.
- ▶ It was evaluated with a usability test to assess user opinions and by a set of **timed tests** to collect an objective measure of the performance and an estimation of the learning curve.
- ▶ **[Demo Video]**

## AirMouse: The Testing Protocol

1. **Timed test:** 5 colored squares are presented on the screen, the user has to follow the instruction on the screen and click the right square. The application records the total time spent to complete the 16 instructions.
2. **Task oriented test:** The user is asked to complete a series of routine tasks, the supervisor is allowed to help the user by giving some tips.
3. **Timed test again.**
4. **Free Use:** The user is left free to use device to navigate Google Maps; the suggested task is to find his house on the map (by dragging the map / zooming in and out).
5. **Final timed test.**

# AirMouse: The Timed Test

- ▶ [Make webpage go now]

## AirMouse: The Tasks

- ▶ (a) Open Firefox browser
- ▶ (b) Open a new tab (plus button in the tab bar)
- ▶ (c) Open the website in the first panel
- ▶ (d) Scroll down until a red button is found
- ▶ (e) Click the red button (it will open Google News)
- ▶ (f) Click the second news
- ▶ (g) Close the tab
- ▶ (h) Open a new tab
- ▶ (i) Swap the third and the fourth panel
- ▶ (j) Close Firefox

# AirMouse: Example Questions

## 1. Task completion required too much effort \*

Il completamento dei compiti ha richiesto troppo sforzo

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

## 2. Task completion required mental concentration \*

Il completamento dei compiti richiede concentrazione mentale

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

## AirMouse: Example Questions

### 3. I was able to complete the tasks very quickly \*

Ho completato i compiti molto rapidamente  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

### 4. The device is accurate \*

Il dispositivo è accurato  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

## AirMouse: Example Questions

### 5. Actuation required a lot of force \*

L'utilizzo richiede molta forza fisica  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

### 6. The operation is very smooth \*

Il funzionamento è molto fluido  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

## AirMouse: Example Questions

**7. I felt very uncomfortable during operation \***

Sono stato scomodo durante l'utilizzo

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

**8. The system is very easy to use \***

Il sistema è molto semplice da utilizzare

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree



## AirMouse: Example Questions

**9. The gestures and hand poses are difficult to remember \***

Le "gestures" sono difficili da ricordare

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

**10. I learned to use the system better after a little while \***

Sono stato in grado di utilizzare il sistema dopo poco tempo

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

## AirMouse: Example Questions

**14. Most people will learn to use the system quickly \***

La maggior parte delle persone impareranno a usare il sistema rapidamente  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

**15. Overall, I am satisfied with the system \***

In generale sono soddisfatto del sistema  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

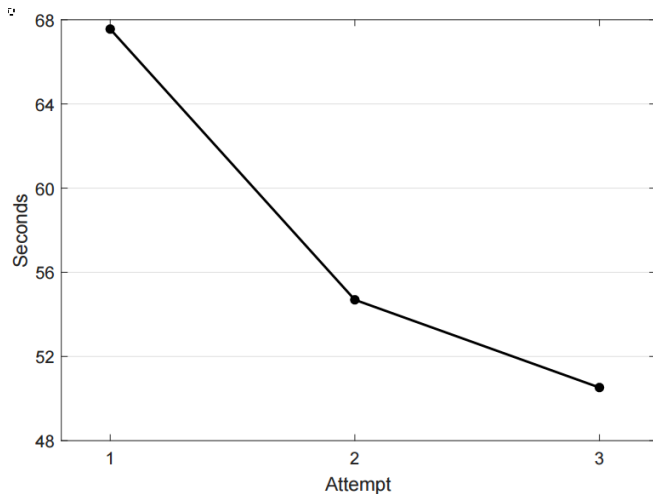
## AirMouse: The Qualitative Results

- ▶ The following table summarizes the answers to the task-oriented questions:

N	Question	Mean rating	$\sigma$	% Agree
1	Task completion required too much effort	3.2	1.6	19%
2	Task completion required mental concentration	3.6	1.3	19%
3	I was able to complete the tasks very quickly	4.2	1.5	44%
4	The device is accurate	4.9	1.4	69%
5	Actuation required a lot of force	2.1	1.2	6%
6	The operation is very smooth	4.6	1.5	56%
7	I felt very uncomfortable during operation	2.5	0.8	0%
8	The system is very easy to use	4.5	1.2	50%
9	The gestures and hand poses are difficult to remember	1.7	0.8	0%
10	I learned to use the system better after a little while	5.7	1.3	81%
11	The system is illogical or inconsistent	1.3	0.5	0%
12	Position and posture are comfortable	3.9	1.2	25%
13	Help given during the test is very important to understand how the device works	4.9	1.2	50%
14	Most people will learn to use the system quickly	5.3	1.3	75%
15	Overall, I am satisfied with the system	5.7	1.0	81%
16	I like the similarity with a physical mouse	5.1	1.8	63%

## AirMouse: The Quantitative Results

- ▶ The following plot summarizes the quantitative results:



## AirMouse: Takeaway

- ▶ The AirMouse Usability Study is an excellent example of **controlled evaluation** of an interface idea with **known problems**.
- ▶ It demonstrates both the **limits of intuition** and the **value of usability testing with real users**.
- ▶ See the full code and usability tests (including questionnaire) in the github repository:

<https://github.com/buddino/AirMouse>

# Summary

- ▶ We have seen a broad overview of the critical issues in **usability testing**.
- ▶ UX testing is an extremely dynamic topic – **necessarily so due to the fast pace of innovation**.
- ▶ For you, or for anyone with **limited resources** and **modest goals**, **hallway testing** is probably the best compromise.
- ▶ You should use **SEQs** as the basis for collecting quantitative usability data.
- ▶ There are many, **many** tools and services available to help collect and analyze usability data (see **Resources** section).

# Summary

- ▶ There is no **magic formula** for usability testing.
- ▶ It is part **science** (a small part) and part (a large part) **art**.
- ▶ It is **effective already with only 5-10 users**.
- ▶ The important things to do are:
  - ▶ Have **clear ideas** about what you want to obtain (i.e. **craft** your tasks and questions to your needs).
  - ▶ **Test often** and with relatively **few subjects** to optimize results.
  - ▶ Use **SEQs** with a 1–7 **Likert Scale**.
  - ▶ Summarize results both **qualitatively** and **quantitatively**.

# Reflections



## General references

- ▶ Two excellent books (one technical, one philosophical) on design:
  - A. Cooper, R. Reimann, D. Cronin (2007). *About Face 3: The Essentials of Interaction Design*. Indianapolis, Indiana: Wiley.
  - Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic books.
- ▶ Conferences and journals focusing on HCI:
  - ▶ The ACM Conference on Human Factors in Computing Systems (CHI)
  - ▶ ACM Transactions on Computer-Human Interaction (TOCHI)

# Usability Testing Resources

- ▶ Here I have gathered some links to tools and in-depth information.
- ▶ Tools:
  - ▶ [Optimal Workshop](#) offers a range of UX evaluation products covering most of the types of testing discussed today.
  - ▶ [UserTesting.com](#) offers remote testing support with video.
  - ▶ [Some unmoderated testing tools](#)
- ▶ An excellent [starter kit](#) for usability testing (includes **forms** and **checklists** for interviewers and interviewees).
- ▶ [Tips on moderating UX testing sessions](#)
- ▶ [A/B Testing in depth](#)
- ▶ [Guide to field studies](#)
- ▶ [Guide to diary studies](#)