Pere-Pau Vázquez – Dep. Computer Science, UPC

Interacció i Disseny d'Interfícies

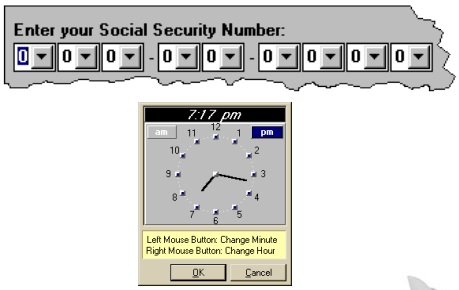
Organization

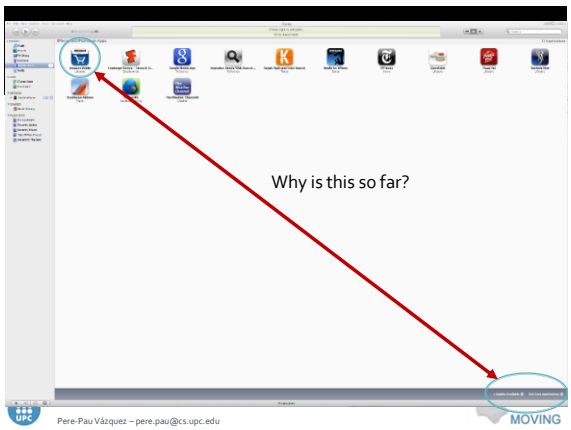
- **Contents**
  - HCI, UX, usability introduction (1 session)
  - Interaction (2 sessions)
  - Usability evaluation (1 session)
  - Visual elements in design (1 session)
  - RV, RA, GPUs (1 session)

Organization

- Useful references:
  - http://usability.gov
  - http://www.smashingmagazine.com
  - http://www.nngroup.com/articles/
  - http://www.interaction-design.org
  - http://www.usabilitycounts.com

Motivation





Motivation

Motivation

What about this game?

Is this interesting at all? How difficult?

Definitions

- What does HCI mean? Which are its objectives?
  - **Human computer interaction** is a field that deals with the study of how humans interact with machines
    - HCI is a very relevant issue when evaluating the quality of an application.
      - An application must fulfil its requirements,
      - It has to provide an easy access to its features.

Definitions

- HCI is about understanding and critically evaluating the interactive technologies people use and experience
- HCI is about understanding contemporary human practices and aspirations

Definitions

- When an application is difficult to use, it is perceived as a low-quality application.
  - **User Interfaces** may be determinant on ease of use perception of application
  - UI: tools and methods that are used to communicate between the user and the system

HCI. How did all begin?

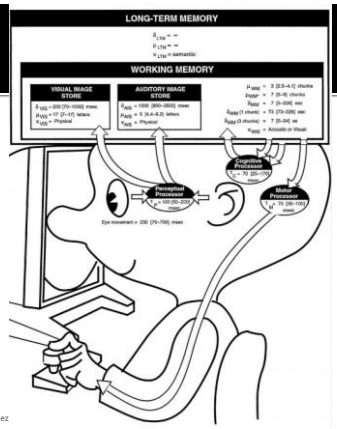
- Studies how humans interact with machines:
  - Began by combining the data gathering methods and intellectual framework of experimental psychology with tools developed from computer science
    - Spreadsheets and text editors
  - Human factors had developed empirical and task-analytic techniques for evaluating human-system interactions in domains such as aviation and manufacturing

HCI. How did all begin?

- Studies how humans interact with machines:
  - Started to incorporate theories of writing, reading, and media, with **empirical user testing**.
    - Contributions from educational and industrial psychologists, technical writers, experts in human factors and ergonomics, information architects…
  - And finally… Designers

HCI. Initial models

- The Model Human Processor was an early cognitive engineering model intended to help developers apply principles from cognitive psychology.



HCI. Initial models

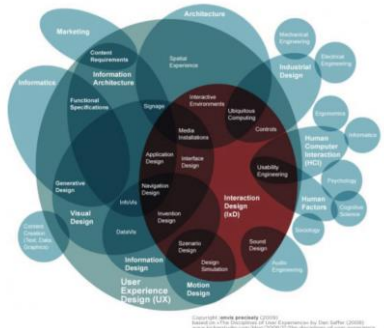
- Software crisis in the 70s lead to focus software engineering with a new view
  - Including non functional requirements such as usability and maintainability

HCI. Initial models

- One of the original focus of HCI was *usability*.
  - Originally stated as "easy to learn, easy to use"
  - More on this later today…
- Helped to influence computer science and technology development more broadly and effectively
- It grew to include other areas, not restricted to computer science

HCI and its neighborhood

HCI and its neighborhood



Definitions

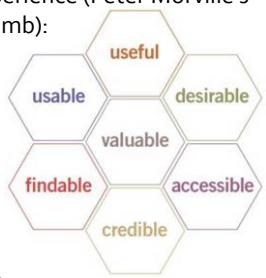
- User experience:
  - "Experience or User Experience is not about technology, industrial design, or interfaces. It is about **creating a meaningful experience through a device**."
  - "the perception left in someone's mind following a series of interactions between people, **devices**, and events"
  - What you **remember and feel** from the use of a device

Definitions

- User experience:
  - The iPhone case

Definitions

- User experience (Peter Morville's honeycomb):



Definitions

- Interaction Design:
  - "Interaction design is about shaping digital things for people's use"
    - How we interact with **devices** ("digital things")

Definitions

- Interaction Design:
  - Find/frame the problem and find solutions
  - Like all design processes, usually iterate through:
    - Sketch
    - Evaluate
    - Improve

HCI & UX. Some requirements

- Most of the work here will be carried out through Interface Design.
- Interfaces must guarantee:
  - Nice and comprehensible appearance
  - Must provide the user with a feeling of controlling the application
  - UIs must be transparent to the implementation
  - User actions should be undoable
    - Work must be often saved automatically
  - Maximum outcome with the minimum inputs

This Is Why the iPhone's Screen Will Always Be 3.5 Inches

Why does the iPhone have a 3.5-inch screen? Why do larger smartphones feel awkward on your hand? Dustin Curtis has an answer, and I think it is spot on:

> *Touching the upper right corner of the screen on the Galaxy S II using one hand, with its 4.27-inch screen, while you're walking down the street looking at Google Maps, is extremely difficult and frustrating. I pulled out my iPhone 4 to do a quick test, and it turns out that when you hold the iPhone in your left hand and articulate your thumb, you can reach almost exactly to the other side of the screen.*

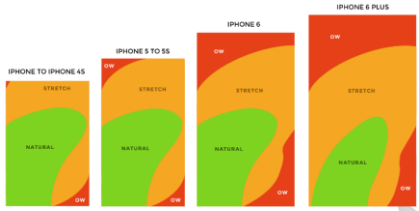
His graphic shows this clearly. It makes total sense. And that is exactly why we would never see any larger screen iPhone. That 3.5-inch screen will be the ideal size until all humans are 7-feet tall and have hands the size of frying pans. [deurtis]

MOVING

HCI & UX. Some requirements

HCI & UX. Some requirements

- For mobile take into account the thumb zones



Usability

- Usability: Defined in ISO 9241 standard as
  - The ability in which a product may be used by **specific** users in order to carry out **specific** tasks *effectively, efficiently, and with satisfaction* in a **specific** use environment.
  - **Usability is always referred to a concrete user group and a concrete user application**

Usability

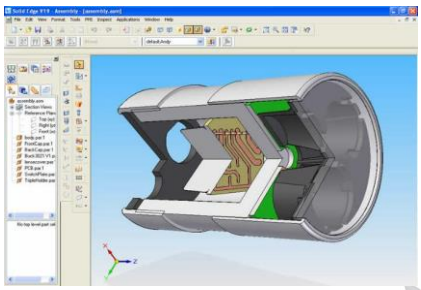
- Usability:
  - **Efficacy** is the ability of correctly and completely achieving a certain goal.
  - **Efficiency** is the relation of used resources and the completeness and correctness of achieved goals.
  - **Satisfaction** is the comfort and acceptation of a system by the users and other people that are affected by its use.

Usability. Different scenarios

Usability. Different scenarios

Usability. Different scenarios

Usability. Different scenarios



Outline

- Introduction
  - **Human-Computer Interfaces**
  - Usability

Human-Computer Interfaces

- Systems requirements/limitations:
  - Desktop devices
  - Mobile devices
- History of Human-Computer Interaction
- GUI Programming

Human-Computer Interfaces

- **Systems requirements/limitations:**
  - Desktop devices
  - Mobile devices
- History of Human-Computer Interaction
- GUI Programming

HCI. Desktop systems

- Desktop systems:
  - Large screens
    - Space for *everything*
  - Mouse pointer
  - Keyboard
    - Adequate for creating content
  - Large resolution

HCI. Desktop systems

- Desktop systems: Windows 7

HCI. Desktop systems

- Desktop systems: Mac OSX

HCI. Desktop systems

- Desktop systems: Ubuntu Linux

HCI. Mobile systems

- Mobile systems:
  - (Relatively) Small size
    - Must carefully think on what to fit
    - Notifications often not properly solved
  - Interaction with the finger/stylus
  - (Almost) No keyboard
  - Small resolution
  - Software limitations
    - No flash in some devices

HCI. Mobile systems

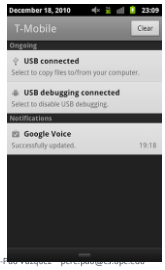
- Small screens

HCI. Mobile systems

- Small screens

HCI. Mobile systems

- Small screens. Other notification methods



HCI. Mobile systems

- Small screens. Other notification methods

HCI. Mobile systems

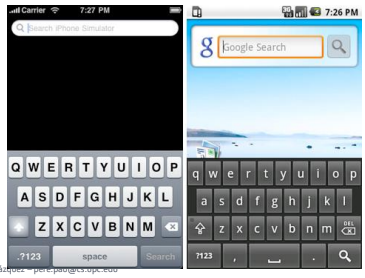
- Finger interaction
  - Reduces the effective resolution
  - Obscures part of the screen

HCI. Mobile systems

- Lack of keyboard
  - Need of a virtual keyboard onscreen:
    - Further reduces the space
    - Prone to errors
    - Slow (unless you have a fantastic corrector or Swype)
    - Not very adequate for creating content
    - **Did I say it is PRONE TO ERRORS?**

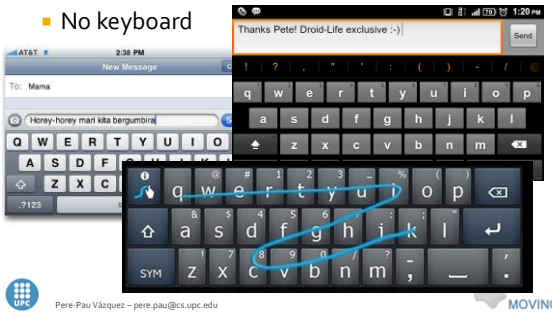
HCI. Mobile systems

- No keyboard



HCI. Mobile systems

- No keyboard



HCI. Mobile systems

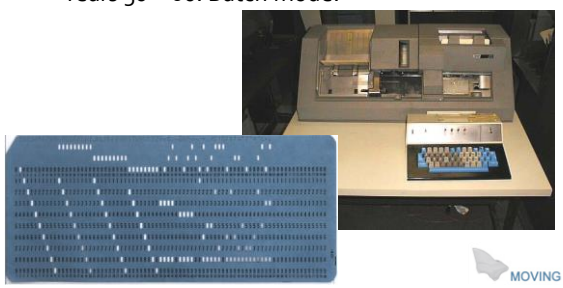
- Tablet systems:
  - (Relatively) Large size
    - May fit what we need
  - Interaction with the finger/stylus
  - (Almost) No keyboard
  - Software limitations
    - No flash in some devices

Human-Computer Interfaces

- Systems requirements/limitations:
  - Desktop devices
  - Mobile devices
- **History of Human-Computer Interaction**
- GUI Programming

HCI: History

- Years 50 – 60: Batch mode.

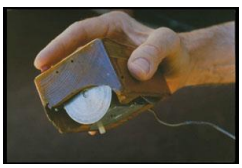


HCI: History

- Years 60 – 80: Shared-time systems. Command-line mode

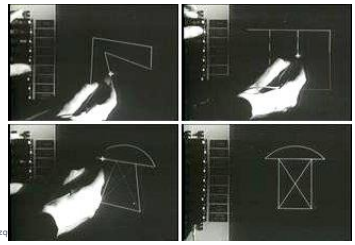
HCI: History

- Years 80 – middle 90s: Graphical interfaces.
  - Interaction metaphor: desktop
  - Interaction tool: mouse pointer.



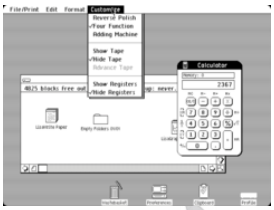
HCI: History

- Years 80 – middle 90s: Graphical interfaces
  - Sketchpad: Ivan Sutherland's PhD Thesis (1963)



HCI: History

- Years 80 – middle 90s: Graphical interfaces



HCI: History

HCI: History

- Years 80 – middle 90s: Graphical interfaces

HCI: History

- Middle 90s – mid 2000s: New interfaces that use gestures and voice recognition

HCI: History

- Mid 2000 – actuality: Tactile screens.

Human-Computer Interfaces

- Definitions
- Systems requirements/limitations:
  - Desktop devices
  - Mobile devices
- History of Human-Computer Interaction
- **GUI Programming**

HCI. GUI (& app) Programming

- Tools for Desktop Development:
  - Windows:
    - Visual C++ and Microsoft Foundation Classes (MFC)
  - Platform independent:
    - Java tools (and *xml* files)
    - GTK+
    - Qt

HCI. GUI (& app) Programming

- Tools for Mobile Development:
  - Native tools
    - Provided by the OS designers
    - Focus on the OS features
  - Cross-platform
    - Provided by third-party institutions
    - Focus on facilitating the development
  - Other third-party software
    - Focus on facilitating the development

HCI. GUI (& app) Programming

- Two main ways to develop:
  - Web apps
  - Native OS apps

HCI. GUI (& app) Programming

- Web apps. Pros:
  - Develop once & deploy everywhere
    - Almost any system has a capable browser
  - Easy updating
    - App is loaded everytime the browser connects to the page
      - Only needed to change the server code
  - Well-known tools and techniques
    - PHP, Java…

HCI. GUI (& app) Programming

- Web apps. Cons:
  - Not as rich as native apps in terms of:
    - UI
    - Communication
    - Access to local resources
  - Inefficient and insecure communication protocol
  - Difficult to write (need to know many different technologies)
  - Mainly designed for large displays with mouse

HCI. GUI (& app) Programming

- Native apps. Pros:
  - Richer UI
  - Many controls
  - Safe and fast access to local resources
    - GPS, camera, files…
    - Efficient communication
    - Any protocols allowed
  - Slower variety in languages and tools
  - Designed for small screens and touch controls

HCI. GUI (& app) Programming

- Native apps. Cons:
  - No universal access
    - Each OS has a different app format and development environment
  - Difficult to manage updates
    - Require individual (user guided) updates per device
  - Less general than desktop programming
    - Though a lot of new material is on the web

HCI. GUI (& app) Programming

- Native app development:
  - In Java: Under almost any desktop OS (Linux, Windows, Apple…)
  - iPhone require Macs (and Objective-C)
- Installation is quite easy
  - Can be done through a web page, e-mail, USB connection, Dropbox…
  - Of course also with Google Play, Amazon App Store…

HCI. GUI (& app) Programming

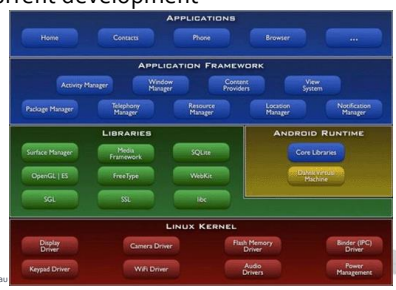
- Tools for Mobile. Native tools
  - Provide the most efficient, compliant code
    - Higher level of control over GUI
    - Steeper learning curve
  - Android SDK: eclipse + plugins, Android Studio
    - Java, widely available
  - iOS SDK: XCode + iOS development toolkit
    - ObjectiveC, only Mac
  - Windows Phone: Windows Phone SDK + XAML
    - C#, Windows

HCI. GUI (& app) Programming

- Tools for Mobile. Cross-platform:
  - PhoneGap (Cordova): Web-based development (HTML5)
    - The result is a web app
  - Xamarin: Development in C#
    - No abstraction of the GUI (different development for each platform)
  - Appcelerator Titanium: Development in JavaScript
    - Abstraction of GUI: facilitates one development, multiple deployment
  - Tools for Mobile: AppInventor
    - Block-based development

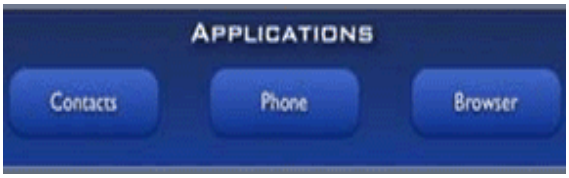
HCI. GUI (& app) Programming

- Current development



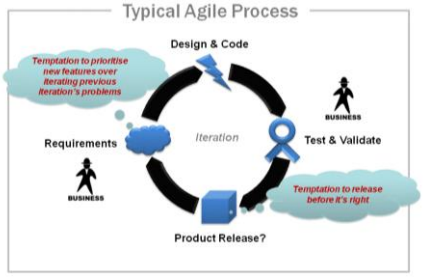
HCI. GUI (& app) Programming

- Development in the past (e. g. in the nineties)



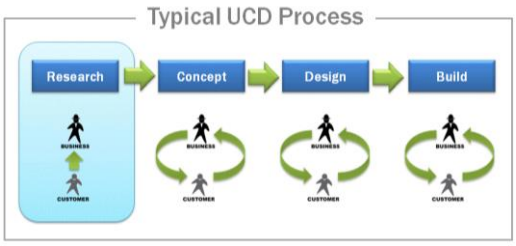
HCI. GUI (& app) Programming

- Software development. Agile



HCI. GUI (& app) Programming

- Software development. UCD



Outline

- Introduction
  - Human-Computer Interfaces
  - **Usability**

Usability

- Definitions
- Usability requirements
- Usability profiles
- Universal design
- Usability problems

Usability

- **Definitions**
- Usability requirements
- Usability profiles
- Universal design
- Usability problems

Usability. Definitions

- Usability: Defined in ISO 9241 standard as
  - The ability in which a product may be used by specific users in order to carry out specific tasks effectively, efficiently, and with satisfaction in a specific use environment.
  - Usability is always referred to a concrete user group and a concrete user application

Usability. Definitions

- Usability:
  - **Efficacy** is the ability of correctly and completely achieving a certain goal.
  - **Efficiency** is the relation of used resources and the completeness and correctness of achieved goals.
  - **Satisfaction** is the comfort and acceptation of a system by the users and other people that are affected by its use.

Usability

- Definitions
- **Usability requirements**
- Usability profiles
- Universal design
- Usability problems

Usability. Requirements

- Effective user interfaces:
  - Achieve required performance by operator, control, and maintenance personnel.
  - Minimize skill and personnel requirements and training time.
  - Achieve required reliability of personnel-equipment/software combinations.
  - Foster design standardization within and among systems.

Usability. Requirements

- Usability can be treated as another engineering task.
  - Requirements analysis:
    - Task and subtask identification
    - Reliability ensuring
    - Standardization and portability
    - Schedule fulfilling

Usability

- Definitions
- Usability requirements
- **Usability profiles**
- Universal design
- Usability problems

Usability. Profiles

- Usability profiles:
  - Life-critical systems
  - Industrial and commercial
  - Home and entertainment
  - Exploratory applications
  - Collaborative systems

Usability. Profiles

- Life-critical systems:
  - Complex
  - Training

Usability. Profiles

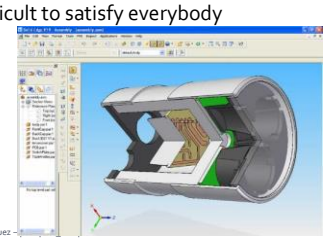
- Commercial:
  - Little training
  - Speed matters

Usability. Profiles

- Home and entertainment
  - Little training
  - Subjective satisfaction matters

Usability. Profiles

- Creative applications:
  - Large variety of (often highly motivated) users
  - Difficult to satisfy everybody

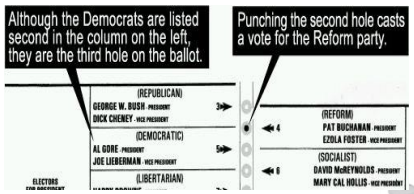


Usability. Profiles

- Collaborative interfaces:
  - Require different information types
  - Difficult to present information

Usability. Profiles

- Socio-technical systems:
  - Broad audience
  - Novice users



Usability

- Definitions
- Usability requirements
- Usability profiles
- **Universal design**
- Usability problems

Usability. Universal design

- Great amount of diversity between humans.
  - Different abilities, backgrounds, motivations, personalities, cultures, and work styles

Usability. Universal design

- Take care of:
  - Variations in physical abilities and workplaces
  - Diverse cognitive and perceptual abilities
  - Personality differences
  - Cultural and international diversity
  - Users with disabilities
  - Adult users
  - Children
  - Accommodating to software and hardware diversity

Usability

- Definitions
- Usability requirements
- Usability profiles
- Universal design
- **Usability problems**

Usability. Problems

- Organizational fragility:
  - Internet access to services
    - Services prone to malicious attacks
      - Sony PlayStation Network
    - Server failures
      - BB Messaging servers down for 1+ day…
    - Storage or access services dependent on non dependable networks (e. g. the Amazon Web Services)
      - If AWS has problems other (Dropbox, Trello…) will have problems too

Usability. Problems

- Anxiety or rage:
  - Use of computers produces anxiety on a high amount of people
    - Computer shock, web worry, network neurosis, or computer rage
  - UI guilty of many of such cases

Usability. Problems

- Alienation:
  - People spend more time using computers
    - They become less connected to other people

Usability. Problems

- Complexity of public services:
  - Interactions with public institutions can nowadays be carried out using the Internet
    - How can I get my tax paying info?
    - How can elder people?

Usability. Problems

- Invasion of privacy:
  - Internet connected devices are a hole to security
    - Famous actresses' photos
    - Facebook

Usability. Problems

- Invasion of privacy: The Facebook example
  - Remember that now Facebook owns Instagram, WhatsApp…
- It is not a particular case of Facebook:
  - Google scans your e-mails (MS Scroogled campaign)
  - iOS had your locations stored as a plain file
  - WhatsApp servers seem to be not very safe
  - Path had problems of privacy

Usability. Problems

- Invasion of privacy: Facebook

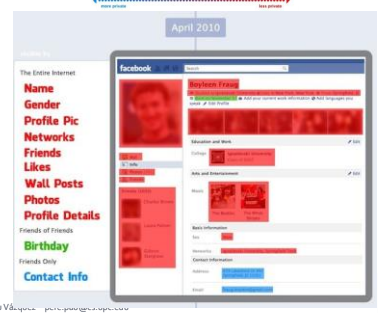
Usability. Problems

Usability. Problems

Usability. Problems

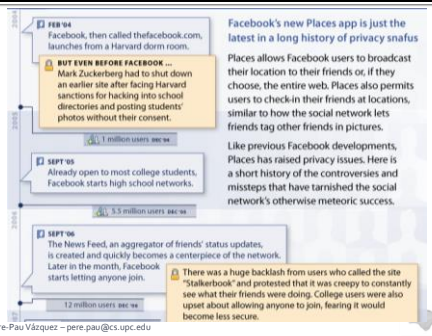
Usability. Problems

Usability. Problems



Usability. Problems

Usability. Problems. Facebook



Usability. Problems. Facebook

Usability. Problems. Facebook

This is frightening, isn't it?

Usability. Problems

- UI design. Great variety of problems:
  - **Development:** Poor countries and poor people
  - **Medical Informatics:** Remote surgery, remote diagnosis…
  - **Electronic commerce:** Commercial transactions have been continuously increasing for the last years.
  - **Government services:** 24 hour / 365 days services
  - **Terror prevention and response:** 11-S terror attacks
  - **Creativity support tools:** Artists

Professors d'IDI - UPC

Interacció i Disseny d'Interfícies