UI Hall of Fame or Shame

User Centered Design for Technology in the Developing World

6.831 Guest Lecture
Clayton Sims ('08, MEng '09)
Dimagi, Inc.

Who am I?

- Mobile Software Division Lead
  - Public health and medical software systems
  - Majority of work in developing nations
  - OpenRosa core maintainer
- My Biases
  - Focus on Mobile Technology, especially Cheap Phones
  - Most experience in Sub-Saharan Africa
  - Not Much Experience in > 10k User Deployments

Technology in Developing Nations

- 85% of the world's population lives in a developing* nation
- The ICTD Research Field is Dedicated to the Creation of Technology in these nations
  - OpenMRS
  - OLPC Project

*UN HDI Index < .9
The Wrong Question

- How is User Centered Design Fundamentally Different in the Developing World?
  - Well, it isn’t.

- Principles are the Same
- Avoid Temptation to Ditch a Disciplined Approach

A Better Question

- How is User Centered Design Practically Different in the Developing World?
  - Radically Different User Populations
  - Resource Limitations
  - Different Expectations for Adoption

Challenges

- Can’t Rely on Prior Knowledge
  - Computer exposure is almost always limited

- You are not the User
  - Seriously this time
  - Neither are your friends

Advantages

- Old Paradigms Die Hard
  - Design isn’t weighed down by convention
  - We’re slow to adopt potentially better interfaces

- Barriers to Adoption are Different
  - Less existing infrastructure to integrate with

Challenges

- User Population Traits are Unique
  - Learning about one population won’t translate to learning about another

- Limited Access to Users
  - Often can’t schedule weekly formative testing

Advantages

- Scale of Impact
  - Good systems make a difference

- It’s Just More Fun
  - To learn how wrong you are
Importance of Attitude

• Kill Your Darlings
  – Your work may become obsolete, unnecessary, or may fail. Live with it. Move on.
• Understand the Implications
  – What if you succeed?

User Centered Design for ICTD

• Examining User Populations
• Reviewing Design Techniques
• Thinking about ‘The Interface’
• Avoiding Pitfalls

User Populations

• Avoid Generalizations
  – I can’t tell you about your users in particular
• Some Common Traits
  – Varying literacy levels and languages
  – Little or no Exposure to Computers
  – Differences in Usability Priorities

Limited Literacy

• Avoid Excessively Verbose Labels
• Consistency in Visual Displays is Valuable
• Alternative Display Modalities
  – Speech Systems
  – Iconographic UI

Language Concerns

• Limited Geographies Still Have Multiple Languages
  – There are 29 Different Languages in India with more than 1 Million Native Speakers
  – Internationalize on first round (and make it easy)
  – Plan for the unexpected

Limited Exposure to Computers

• Trajectories Should be Clear and Narrow
  – It should be clear what actions can be taken
  – Actions should be grouped visually
  – Tend towards avoiding “overloading” screens
• Focus on Imperative Language
  – “View Orders” v. “Orders” as a selection
Limited Exposure to Computers

• Beware of Seemingly Natural Groupings
  – Login Screens
  – High Level “Settings” Fallbacks
  – Popup Menus

• Be Flexible on Inputs
  – Especially with newer systems
  – If a lot of your users seem to something it should work, maybe it should
  – Redundancy is preferable

Difference in Priorities

• Users are more tolerant of lessened efficiency
  – Although there’s a bias in that notion
• Discoverability and Learnability are extremely important
  – Users are more hesitant to explore and find features

• Error Recovery and Prevention
  – Users familiar to computers expect errors and bugs
  – Users with limited exposure to computers often find it extremely hard to recover from errors
  – This problem tends to alleviate with time

Responsibilities

• Avoid Overpromising in UI
  – If it looks possible, it should be
• Beware Automatic Inclusion
  – Be explicit in Opt-In actions
  – Understand cultural dangers and avoid assumptions of impact

User Centered Design for ICTD

• Examining User Populations
• Reviewing Design Techniques
• Thinking about ‘The Interface’
• Avoiding Pitfalls
Task and Domain Analysis

• Computerized Tasks are Often Based on Real World Analogs
  – This consistency is valuable
  – But don’t get caught up trying to emulate existing systems
  – Task and Domain Analysis doesn’t stop with replicating an existing process

User Inclusion

• Participatory Design and Formative Evaluation are Vital
  – Ongoing evaluation is a necessity
  – Participatory design often generates unexpected and useful information and feedback

Feedback

• Users are more often hesitant to provide negative feedback
  – Make Alternatives Clear
  – Elicit Negative Opinions Explicitly and With Groups
  – Observe and Query
• Especially for Professional Users
  – Looking to be successful with existing system, not ‘complain’ about it

Prototyping

• Avoid Computer Prototypes as Long as Possible
  – Many confounding variables exist with any
    Computer Prototype if users aren’t comfortable with computers
• Establish the Idea of Non-Permanence
  – “Once it’s on the Screen, it’s finished”

Prototyping

• Environment Can be Hard to Control
  – Lighting Conditions
  – Weather
  – Space Concerns
• Better to be Flexible than Exact
  – Don’t over prepare artifacts
Prototyping

- Maximize Team
  - Guide
  - Communicator
  - Emulator
  - Multiple Observers
- Prototype on Groups of Users
  - More willing to explore
  - More willing to explain

User Centered Design for ICTD

- Examining User Populations
- Reviewing Design Techniques
- Thinking about ‘The Interface’
- Avoiding Pitfalls

Prototyping

- Users Might need More Guidance
  - Exploring might not be intuitive
  - Try to provide as much up front as possible
- Have a Clear Goal for Prototyping
  - High Noise Level in Data
  - Formative evaluation is also useful

What is ‘The Interface’

- Everything
  - Operating System
  - Computer Power Button
  - Keyboard/Mouse
- You’re on the hook for all of them
  - Not just for learnability/understanding
  - Environmental Factors can make your systems unusable
**What is ‘The Interface’**

- Extraneous Layers to Interface are Damaging
  - Confusing for new users
  - Damaging to Efficiency and Error Prevention
- Lock it Up
  - Controlled Sandbox
  - Provide Easy Reset to a Central Location
  - The whole system should be a concern early in the design process

**What is ‘The Interface’**

- All-in-one systems are the most desirable
  - Laptops have become a much more popular choice for deploying technology than desktops
  - Custom all-in-ones (OLPC) are a far-extreme of this solution

**Touch Screens**

- Extraordinarily Intuitive
  - Viewport and Input all in one place
- Very good for larger systems
- Simplifies Input Needs
  - No mouse
  - No keyboard (if you can avoid it)

**Touch Screens**

- Concerns
  - Even Further Extracts External Components
  - Touchscreen software environments have a lot of extraneous inputs possible
  - Avoid flash and hidden input modes and features which are based on pre-existing mental models that experienced computer users have

**Voice Interfaces**

- Voice Interfaces Greatly Increase Accessibility
  - Literacy is far less important
  - Voice Interfaces are linear and imperative by default
- Scale well in Breadth, but not Depth
  - Many users isn’t much harder
  - More complicated tasks make system polynomially harder to use

**Voice Interfaces**

- Pseudo-Human Interfaces are Tricky
  - Need to filter extraneous input
  - Potentially unrealistic expectations of capability
  - Machine learning with every dialect/accents/language probably isn’t realistic
Voice Interfaces

• Set Realistic Expectations
  — System should be forthcoming about inadequacies
  — Computerized voices are less subjectively satisfying, but have less errors
  — Keypads perform better in experiments

Cell Phones

• Cellular Phones are the Most Ubiquitous Form of Technology in the Developing World
  — If users have been exposed to computing, odds are it is with a cell phone
  — Many users are familiar with high level interface
    • Keypad
    • Screen
    • Some similarity of software

Cell Phones

• Leverage Consistency
  — Emulate Phone UI whenever possible
  — Work with USSD or other build in protocols if possible
  — Don’t build a custom Phone Interface if you can avoid it

Cell Phones

• Challenges
  — Extremely difficult to sandbox
  — UI Capabilities are Extremely Limited
  — Hard to find a reasonable platform
  — Technology changes fast, your platform/app could be obsolete in 6 months

User Centered Design for ICTD

• Examining User Populations
• Reviewing Design Techniques
• Thinking about ‘The Interface’
• Avoiding Pitfalls

Avoiding Pitfalls

• Making every interaction a single decision
  — There is a place for guiding the user, it is not every screen
  — Wizards are a dangerously tempting paradigm
Avoiding Pitfalls

• Ignoring Efficiency and Expert Users
  – Lack of expertise is limited only by time and exposure
  – If your system is good, eventually it will have expert users and you still want it to be usable

• Ignoring User Roles and Gradations
  – Many features of a system don’t need to be available to every user
  – If some users are comfortable with computers and others aren’t, establish what tasks are simply out of scope

• Biting off more than you can chew
  – Iterative design and testing can show early on what features are impractical
  – Stable End to End behavior of central features should be always be the first step

• Ignoring What Real Users Need
  – An Awesome UI isn’t worth much if it isn’t solving a real problem
  – Much developing world technology is funded by grants and gifts, avoid flash and pomp

Questions?

Followups Welcome
csims@dimagi.com
tsim@mit.edu
617-863-9873