Intro

Human-Computer Interaction (HCI): A discipline concerning the design, evaluation and implementation & study of interactive computing systems for human use (Preece, Rogers, and Sharp 2002)

Interaction Design (ID): The design of spaces for human-computer interaction and communication (Winograd 1997)

User-**centered design** (UCD) is a process (not restricted to interfaces or technologies) in which the needs, wants, and limitations of end **users** of a product, service or process are given extensive attention at each stage of the **design** process.

How is it the same/different from other development models? Waterfall? Current agile methods? RUP Model?

Usability Goals:

- Effective to user
- Efficient to user
- Safe to use
- Have good utility
- Easy to learn
- Easy to remember how to use

User experience goals:

- Satisfying, Rewarding, Fun, Support creativity, Enjoyable, Emotionally fulfilling, Entertaining, Helpful, Motivating, Aesthetically Pleasing, etc...

Interaction design/ user centered design

How is it the same/different from other development models? Waterfall? Current agile methods? RUP model?

Not very similar to waterfall.

Short cyclical process similar to an agile method.

Has time constraints like the RUP model but does not follow the same logical flow.

Affordances:

Perceived and actual properties of an object that determine how the object could be used.

Principles of affordances:

- good conceptual model
- make things visible
  - visible structure impacted by: affordances, constraints, mappings
- principle of mapping
- principle of feedback

Visibility:

Mapping:

Feedback:

Conceptual Model:

A person's understanding of how a system works and depends on the person:

User Model:

How a user would conceive the system's function

Design Model:

How a designer views the model's implementation.

Interface Metaphor:

Interface designed to be similar to some object (usually physical)

Exploit the user's existing knowledge

{Bad example: Skype webcam waterbottle}

What makes a good interface metaphor?

Not too literal: takes ideas from the physical device that transfer easily to GUIs

Doesn't clutter or complicate the interface unnecessarily

What impacts metaphor selection?

Culture, society, age demographic

Requirements

What kinds of information should we gather?

Domain knowledge (who users are)

user's physical characteristics, user's behaviour

Tasks and task characteristics

Physical environment, social environment, organizational environment

Functional needs, usability goals, values and constraints

Data Gathering Techniques:

Observation:

Pros Good for identifying constraints or needs through user mistakes or frustrations
Cons: Time intensive and intrusive

Logging:

Automated gathering of user input information

Interview (structured):

Pros: questions are predefined, can have multiple people taking notes
Cons: reach fewer people, require participant consent

Questionnaire:

Pros: can reach a large number of people, good for answer comparison
Cons: less detailed answers, can’t have person specific follow up questions

Longitudinal studies(Diaries/experience sampling):

extensive studies that involve user data collection

Ethnography:

a sociological method that explores how people live and make sense of their lives with one another in place and can include participant observation, field notes, interviews, and surveys

Sociology:

Sociology is the scientific study of social behaviour, including its origins, development, organization, and institutions. It is a social science that uses various methods of empirical investigation and critical analysis to develop a body of knowledge about social order, social disorder and social change.

Psychology:

Psychology is the study of mind and behavior.

Personas:

Turn the users into identifiable human beings with a background. Create several for each major type of user that are fictional and cover all possible users of the application.
Part of the design process when defining the users of the application.

User Characteristics:

- Age
- sex
- culture
- physical abilities/disabilities
- educational background(reading ability, comprehension)
- computer/IT experience
- motivation
- attitude

Task Analysis

Task: a set of related activities performed in some sequence
Goal: an end result to be achieved

Task Characteristics:

- Variability from one time to the next
- regularity
- knowledge and skills required
- physical environments
- time critical
- safety hazards
- alone vs. group
- other simultaneous taks

Task analysis techniques

Classes of task analysis:
describe actions to complete task (what steps the user takes to complete the task)
capture knowledge user has or needs to complete the task (how they do it)
Examples:

- Task Scenarios
- Concrete user cases
- Essential user cases
- Hierarchical Task Analysis

Task Scenarios:
A personal narrative story identifying a task with specific details.
Usually does not say anything about the system interface and is used in requirements gathering.
helps in:

- requirements gathering, design, and evaluation
- avoid making assumptions about how interface will operate/order of actions
- evaluate whether the interface will work for a particular user in a particular instance

Use Cases:
Similar to a scenario but steps through a task for a given interface and has no specific order to the subtasks. There are no personalized details and it lists all the alternate possibilities that can occur (alternate paths). It is used more in design that requirements gathering.
helps in:

- requirements gather, design, and evaluation
- evaluates whether a design will work for all possible cases

Hierarchical Task Analysis:
Breaks a task into smaller and smaller subtasks until the smallest unit of work is reached.
Helps you understand the task better.

Conceptual Design

What is it?
It is the process of establishing the underlying organization to make sure the interface is organized in a logical manner from the user's perspective and not the system perspective. The end goal is the user interface, but no graphics are considered at this stage. Bigger project takes more time for conceptual design.

When?
After some task scenarios and use cases are established.
Before interface sketching begins.

Object Based UI
Developers tend to think of objects that are available, at their disposal, and have a purpose.

Task Based UI
Users tend to think of tasks that need to be completed and it is based around those tasks.

Affinity Diagrams/Content Diagrams:
Listing each function on a sticky note then arrange the sticky notes until organization is logical, this results in a content/affinity diagram.

Prototyping

Why?
To elicit, clarify, validate, and negotiate requirements.
Choose between competing designs.
Allow users to contribute to designs.
Allow designers to focus on the task before investing time and effort into the UI
Check feasibility of ideas with users
Check usefulness of application
Allow users to test ideas

Fidelity:
The higher the fidelity, the more it looks and feels like the real system.
Low: paper prototypes, storyboards
Advantages:

- Quick and Cheap
- can be used to brainstorm competing designs
- users can create and modify them

Disadvantages:

- tends to focus on high level concepts
- hard to envision a dialog’s progression
- less compelling
- hard to detect errors

Medium: interface builders, media tools, painting/drawing package software

Advantages:

- more engaging for the end user, provides a limited scenario for the user to test
- can test more subtle design issues

Disadvantages:

- user’s reactions are usually “in the small”; blinds people to major design flaws
- Management may think it is real
- Users may be reluctant to challenge the design (as it appears to be so “final”)

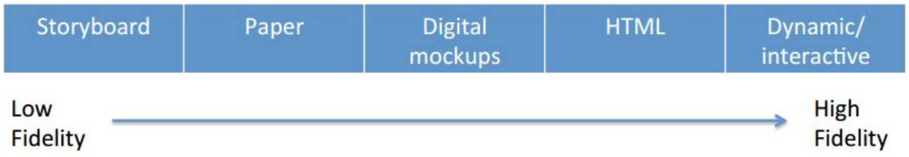
High: interactive software prototypes

Advantages:

- Shows UI layout, navigation, look, and feel
- compelling

Disadvantages:

- Finished looking product stifles feedback
- can’t be used early in design process



Approaches to limiting prototype functionality:

Full prototype:

contains complete functionality but with lower performance

Horizontal Prototype:

displays all or most of the application without going into any depth in any one part (useful for testing overall design concept)

Vertical Prototype:

carries the user deep into the behaviours of specific parts of the system (useful for testing new design concepts and technology)

Prototype Evaluation

Wizard of Oz prototyping:
Used to simulate machine behaviour by using a human operator where the user believes they are interacting with a computer but it actually a man behind the curtain. Good for complex and futuristic functionality that is not yet developed or perfected.

Advantages:

- gets a feel for whether it is worth building

Disadvantages:

Video Prototype
<<<lecture 7>>

Evaluation

x

Distributing Cognition

x

Visual Design

x

Evaluation - Data Collection

x

Experiment Design

x

Experiment Analysis

x

History

x

I/O Design

x