**Interfaces**

**------------------------------**

Quality of a design?

* **User Experience**

Basics of user experience

1. **Usefulness**
   * Solves existing problem
   * Provide better solution to a problem
2. **Usability**
   * Easy for user to become familiar with
   * Easy to achieve objective
   * Easily accessible
3. **Desirability**
   * Emotional design
   * Invites user to interact with it, encourages user to own it or use it
4. **Accessibility**
   * Easily accessible
   * Consider different groups of users
5. **Valuable**
   * Meets end users needs
   * Improve end user experience and satisfaction

Good UI characterized by

* **Satisfied** when used
* Complete **task error free**
* Learn how to u**se system quickly**
* **Well designed**

What is design?

* Creating or shaping tools or artifacts for direct human use
* **Process**
  + Processes, methods, mistakes
* **Creating**
  + Creative endeavors, improving
* **Tools**
  + Output
* **Human**
  + End user
  + Different groups, experience, skills, needs

**Design vs Engineering**

* Design
  + Envision new possibilities, outcomes, what outcomes should result among infinite possibilities
  + Process over formulae
  + Humans are central actors “in the loop”
* Engineering
  + Mostly known outcome possible
  + Construct based on specification
  + What can be done – well established formulae

**Design vs Art**

* Design
  + Creation of something useful and usable
* Art
  + Does not bother with restriction
  + Creator vision
  + Meaningful, not necessarily usable

Why design is hard

* Judging/predicting which designs will be successful is difficult
* Design is making tradeoffs, good designs are non-obvious
* People are different; age, skills, ability
* People appropriate technology unexpectedly
* Context of use may differ than what we expect

The user is not like me

**User centered design**

**----------------------------**

Astronauts in rocket – design has many switches and buttons, some need to be reached with stick

* Design can be intended to be very serious

**UCD**

* Iterative design process that makes use of knowledge through investigation of a domain of work/play to create ideas and prototypes

**Prototypes**

* Used for evaluation, further stimulate investigation, idea and prototype generation
* Prototypes and evaluations are used to aid in production

**UCD Process**

* **Investigation**
* **Ideate**
* **Prototype**
* **Evaluate**
* **Produce**

UCD Process

* Is a process because
  + Get started in a proven track
  + Prevents designer block
  + Directs us towards final product
  + Helps us stay on schedule and on cost
  + Communicate with others
  + More reliable than intuition
  + Forces us to iterate
  + Helps to keep the user first

**UCD relies on end user information**

* End user not always right, humans make errors
  + Need to study what kind of errors end users make

**Slips**

* Error in carrying out action – motor actions or typos
* Lapse in attention, or due to change in typical circumstance

**Mistakes**

* Error choosing objective or action
  + Cognitive goals
* When a person misunderstands something in the system

Good designers help users avoid mistakes, and possibly, can avoid slips as well

User is not like me

* Familiarity with the concept
* Confidence in the system
* Designer vs user setting
* Designers have different sets of skills from the regular user
  + Regular user vs computer programmer

**Core design skills**

1. Synthesize solution from relevant constraints
2. Frame, or reframe the problem and objective
3. Create and envision alternatives
4. Select from those alternatives – intuitively how to choose the best approach
5. Visualize and prototype the intended solution
6. Think like the user when required

**User Centered Design**

**---------------------------------**

Why do some companies intentionally refuse to provide certain experiences if user demands it

UCD process

* Why do we need to investigate, ideate, prototype, evaluate, produce

**Why investigate?**

* Learn about the stakeholders, the goals, the needs
* Figure out how it’s done, what is wanted, what else has been tried before
* Need to learn as much as possible about user before we start designing

**Methods**

* Interview
* Focus groups
* Surveys
* Questionnaire
* Try it yourself
* Etc…

**Why do we ideate?**

* Generate lots of ideas
* Grasp issues and potential solutions
* Increase chance of success by considering huge volume of ideas in systematic way

**Methods**

* Structured brainstorming
* Affinity diagramming
* Personas
* Sketching
* Card sorting
* Etc.

**Why prototype?**

* Produce something tangible
* Identify challenges
* Uncover subtleties
  + Explore ideas and show intention behind a feature or overall design concept to users before investing time and money into development
* Cheap and fast
* Easy for users to react on concrete rather than abstract
* Brings subtleties and nuances to light
  + Draft version of product to explore ideas and show intention behind feature or design concept to users
* Working against some technical restraint is good

**Methods**

* **Low fidelity**
  + Paper
  + Wireframe
  + Powerpoint slides
  + Drawings
* **High fidelity**
  + HQ drawings
  + Functional prototype
  + Clickable UI
  + Simulated functionality
  + Pre-set responses

**Prototyping fundamentals**

* **Build it fast**, concentrate on unknowns
* **Don’t be attached** to prototypes
* Build them concurrently

**Why evaluate**

* Discover problems
* Assess progress
* Determine next steps
* Automated processes can find bugs but not usability issues
* Evaluation gives a way to move forward
* What needs to be fixed, added, removed
* Answers two questions
  + Did we **build the right thing?**
  + Did we **build the thing right?**

**Methods**

* Heuristic evaluation
* Usability testing
* Real-world deployment

What happens when you **discover an issue in evaluation**

* User performance
  + Return to prototyping
* Usefulness / appropriateness
  + Return to investigation
* Users don’t understand how to use it
  + Return to ideation

**Produce**

* Transfer a system high fidelity prototype to a commercial product
  + Steps required to go from functional prototype to release candidate
    - Software architecture
    - Programming, building
    - Manufacturing
    - Manuals
    - Training
    - …

**Investigation (1)**

------------------------------

* **Why do we need to investigate?**
  + Understate state of the world, state of users, what problem is
  + What user needs are
  + How can we go about doing a certain matter?
* Sometimes we can rely on users about what they want in a design
  + People really good only at framing their needs within language of their current design
  + Cant see past immediate problems to the possibilities
* Lost leg, phantom pain story
* Lessons to learn from people
  + Hours slept
    - Varies
  + Where do we walk on campus
    - People make own paths, follow paths
  + Italian vs American sauce
    - People say something, want something else
    - They care about their image
      * Present themselves in a certain way
    - Brain tries to justify actions to protect self
  + Where to put hand sanitizer
    - Not under a light switch
  + Smart home system or mobile app
    - Asking questions in the right way

What we learn

1. People have bad memories
2. People create their own way of using design when they need to
3. People bad at predicting what they prefer if its only hypothetical
4. People can be scared of changes and new things

**People are good at telling us a few things**

1. What they are **doing right now**
2. What they are f**eeling right now**
3. **What their goal is**

Interviews

* Combine interview with observation
* Watch people in their own environment
  + Watch them do everyday tasks
* Opportunities for new designs
* Workarounds
* Breakdowns
* Unexpected uses of existing tools

Two Messages

* Mismatches between what people say and do lead to wrong design
* Assumption likely to be wrong, to discover more, assume assumptions are always wrong

**Investigation (2) – IDEO Methods**

**---------------------------------------------**

IDEO Cards

* Asking users questions
  + If same thing is reoccurring, then we know of a problem

**Design methods cards**

* IBM Cards
* 18F method cards
* IDEO Method Cards

How does it work?

* **Look at what people really do**
  + Observe people to discover what they do rather than what they say they do
* **Ask users for help**
  + Enlist participation to elicit information relevant to project
* **Learn from facts you gather**
  + Analyze information collected to identify patterns and insights
* **Try it yourself**
  + Create simulations to help empathize with people and to evaluate proposed designs

**Investigation (3) – Look, Learn**

**---------------------------------------**

IDEO Methods cards -> For design

* **Look at what people really do**
  + Rather than what people say they do
* **Ask users for help**
  + Card sorting
  + Gather something specific couldn’t learn from observing
  + Elicit info relevant to the project
* **Learn from facts you gather**
  + Analyze info collected to identify patterns and insights

**Look at what people really do**

* May change mind on what you want to ask
  + Change the way you ask the question

**Methods**

* Fly on the wall
* Guided Tours
* Shadowing
* Social Network Mapping
* Time lapse videos

**Ask users for help**

* **Methods**
  + Surveys and questionnaires
  + Unfocus groups
  + Card Sort
  + Cognitive maps
  + Extreme user interviews

More Investigation

Learn from the facts you gathered

* Analyze information you’ve collected to identify patterns and insights

**Investigation (4)**

**------------------------------**

**Learn from the facts you gathered**

* Moving from investigations to ideation

**Character profiles**

* + **Personas**
    - Define who the story is about
    - Create a scenario
    - Reach a goal

**Properties of Personas**

1. Profiles
2. Personality
3. Referents & Influencers
4. Archetype & Quotes
5. Technology expertise
6. User-experience goals
7. Used devices and platforms
8. Domain details
9. Must do – never do’s
10. Brand & product relationships

* **Secondary research**
  + Can reason what has been done before – don’t make the same mistakes
* Error analysis
* Flow analysis
  + What is repetitive can be shortened
  + Enhanced before prototyping
    - Cheap to analyze user steps
* Cognitive task analysis
  + Sorting priorities in order from end users
  + Sort input from end user to learn what you want
    - Organize thoughts
* Affinity diagramming
  + Visualize thoughts and relationships

**Investigation (5)**

**---------------------------**

**Try it yourself**

* Roleplaying
  + Simple & cheap to implement
    - Observe team members roleplay
* Try-It-Yourself
  + Keeps team innovating
    - Mostly brainstorming ideas, prototype can be motivating
* Empathy tools
  + Age suit, laneway house project
* Scenarios
  + Difference between scenarios and personas
    - Scenarios focus on story or many people, persona are single character profile
    - To have scenario, user needs to have a need to do a task, but has an issue
* Scenario testing
  + Serves the scenario to end user to see feasibility of the scenario
* Paper prototyping
  + Evaluate ideas before proceeding
  + Pursue idea or not, prototype later tests the design

**Task Centered User Design**

-------------------------------------

* Technique that helps developers **design and evaluate interfaces based on users’ real-world tasks**
  + **Identification**
    - Identify users and tasks
  + **Requirements**
    - Decide which tasks and users design will support
      * System not going to support every task
  + **Design**
    - Base design representation & dialog sequence on tasks
  + **Evaluation**
    - * Cognitive walkthrough evaluation

**Phase 1 – Identification**

User

* Get in touch with potential users of system
  + Prototypical categories
  + Extremes learn about their real tasks

Tasks

* Articulate concrete, detailed examples of tasks they perform or what to perform
  + System should support
    - Routine
    - Infrequent but important
    - Infrequent
    - Incidental

What if there are no real users or tasks

* Describe expected set of users
* Describe expected set of tasks
* These become ‘assumed users and tasks’
* Verify later as information comes in
* Modify as needed
* Yogurt Example

**Phase 2 - Requirements**

What kind of requirements are we looking for?

User

* Which user types will be addressed by the interface
  + Designs can rarely handle everyone
  + Includes why particular users are included / excluded
    - Typical ‘expected’ user
    - Occasional but important user
    - Unusual user

Tasks

* Which tasks will be addressed by the interface
  + Designs can rarely handle all tasks
  + Requirements listed in terms of how they address tasks
    - Must include
    - Should include
    - Could include
      * Exclude
    - Discussions includes why these items are in those categories

**Phase 3 - Design**

* Design through scenarios
  + **Create scenarios** (Stories) with (thin) plots that capture tasks that a person would engage in
    - Develop designs to fit users and specific tasks
      * Ground interfaces in reality
    - User tasks to
      * Get specific about possible designs
        + Generate ideas
      * Consider real world context of real users
      * Consider design features work together
      * Think what user would do
        + See step-by-step when performing this task

**Phase 4 – Evaluation**

* Cognitive walkthrough design

Process

1. Select one of task scenarios
2. For each step ask the following
   * Can you build a believable story that motivates the user’s actions
   * Can you reply on user’s expected knowledge and training about the system
   * If you cannot
     + Located problem in the interface
     + Note the problem, include comments
     + Assume it has been repaired
     + Go to next step in the task

How to develop **good user tasks**

* 6 Features

1. Say **what the user wants to do** but does not say how they would do it
   * No assumptions made about the interface
   * Can be used to compare design alternatives in a fair way
2. Describe a **complete job**
   * Forces designer to consider how interface features work together
   * Contrasts how information input / output flows through dialog
   * Where does information come from, go, what has to happen next
3. Tasks are **evaluated**
   * Circulate descriptions to users, rewrite if needed
   * Ask users for omissions, corrections, clarifications, suggestions
   * Shows what want to be done, not how
4. Are **very specific**
   * Says exactly what user wants to do
   * Specifies actual items user would somehow want to input
5. Says **who the users are**
   * Names if possible
   * Says what they know
6. As a set, identifies a **broad coverage of user** and task types
   * Typical ‘expected’ user
   * Occasional but important user
   * Unusual user
   * Typical routine tasks
   * Infrequent but important background tasks are unexpected or odd tasks

Cheap shop example

**Ideation (1)**

----------------------

* How do we come up with, refine, organize ideas?

Design funnel

* Initial is biggest number of ideas gathered
  + Analyzing based on ideas that are left, narrowing down scope towards a concept.

Existing systems

* **Idea of what is currently happening**
* Sets the bar for the design
* **Good Examples**
* About ‘me’
* Ask self questions about why you like it
  + Materiable, MIT media lab
* **Bad examples**
* What is good and bad about it
* **Inspiring examples**
* What problem did they solve
* What is nice to look at, inspiring to talk about the most?
  + i.e Ecosia
* **Futuristic examples**
  + Don’t want to create product or design that will fail in a year

Explain ideas to others

* **Outsider perspective**
  + Take outsider perspective on something you know well
  + Helps simplify language/understanding
* **Insider’s Perspective**
  + Take insider perspective you don’t know well
  + Helps identify your assumptions

**Kids and Creativity**

* Kids seem creative
  + Don’t know rules, feel free to break them
  + Don’t know consequences, so take risk
  + Honest

**Ideation (2) – Sketching**

--------------------------------

Sketching to get the right design vs get the design right

**Getting the right design**

* Generate many ideas and variations
* Choose the ones that look the most promising
* Develop them in parallel
* Add new ideas as they come up
* Iterate the final choice
  + Doing sketches, working on a few ideas
    - The ones that evolve are the ones you want to go for

**Getting the design right**

* Getting the design right is investigatory phase
* It is a process, its evolutionary, about increasing the peak height

**Sketching**

* Process enables you to think through ideas and variations, convey design ideas to others early in the design phase
  + Express, develop, communicate design ideas
  + Sketching not about drawing, about designing
  + Part of ideas generating process, design elaboration design choices, engineering
* Zaha Hadid

Why do we sketch

* Early ideating that allows risk-taking and **explorations of variations**
* **Think through ideas**
* Active form of **brainstorming**
* Visualizes how things come together
* **Communicates ideas** to others to inspire new designs
* Taking the risk
  + If working with a client, sketching helps finds good ideas that will make a good design
* Helps visualizing the relationship between each sketch

Sketching Dialogue

Create

=>

Mind Sketch

(new knowledge (representation)

Read

<=

Design funnel

* Elaborate
  + Generate solutions, these are all opportunities
* Reduce
  + Decide on the ones worth pursuing
* Repeat
  + Elaborate and reduce again on those solutions

Buxton’s properties of sketches

1. **Quick**
2. **Timely**
3. **Inexpensive**
4. **Disposable**
5. **Plentiful**
6. **Clear vocabulary**
7. Distinct Gesture
8. Minimal Detail
9. Ambiguity

Forms of sketching

* **Pencil/pen drawing on paper**
* Magazine cut-offs
* Something scraped together on Photoshop
* Modifications to existing objects

**Prototyping (1)**

--------------------------------

What is prototyping

* Developed model built for purpose of **communicating, testing, evaluating design**
* Goes through multiple evaluations until acceptable outcomes are achieved. Final prototype is used for product development

**Prototype qualities**

* + Fast
  + Disposable
  + Focused

**Roles of prototypes**

* + Test -> Get feedback
  + Communicate
  + Persuade

Evaluation and user feedback is **central** in good design

* Stakeholders can see, hold, interact with prototype
* Aids communication if idea, provide focus
* Test out ideas
* Encourages reflection

Types of prototypes

* Storyboards
  + Quick, low res prototype
  + Visualize concept from start to finish
  + Developed from scenarios
* Paper prototype
  + Allows the creation and testing of UI quickly using paper only
* Powerpoint prototype
  + Cheap, quick, easy method to create clickable UI
* Video prototype
  + Show new and speculative designs, ideas, scenarios, futures or products
  + Clear idea of how users might perceive a concept that might be difficult to prototype
    - Meme long iphone
* Physical Model
  + Making physical representation of an idea
  + Early prototypes made from any material
  + Allow designer and users to interact with idea
* Software with limited functionality
  + Create model that covers only the UI of the product
  + Fully clickable and good visuals and aesthetics
  + No backend work, mainly front end skeleton

What should be prototyped?

* **Task design and user flow**
  + Based on expected tasks, what users see, what they will do
* **Screen Layouts and information display**
  + How should information be laid out to provide information as users need it?
  + How can this be optimized
  + Visual design
    - Focuses user attention, don’t want to distract them with too many elements
* **Graphic design and look & feel**
  + What should it look like
* **Technical aspects**
  + Can we actually make this work?
  + May look good, but if there are technological limitations, prototyping will discover it

Choosing prototyping methods

User flow -> Use storyboard

Screen layouts / page flow -> paper prototypes

Overall experience -> video prototype

Look and feel -> Powerpoint or PSD

Functionality -> software

Prototyping fidelity

* Low fidelity
  + Low-tech concepts
  + Turn ideas into testable artifacts
  + Collect and analyze feedback in early stages
    - Collect data and feedback quickly, cheaply
* High fidelity
  + Highly functional and interactive.
  + Very close to the final product
  + Most of the necessary design assets and components developed and integrated
  + Test usability and identify issues in workflow
    - For different users typically

Prototype fidelity

* Refers to how closely it matches the look and feel of the final system
* Vary in the areas of
  + Interactivity
  + Visuals
  + Contents and commands
* Lorem Ipsum

Comparison

Lofi

Advantages

* + Fast, Cheap, easy
  + Simulate actual product, get feedback from stakeholders

Disadvantages

* + Slow response time
  + Cant get feedback about aesthetics, user may question design quality

HiFi

Advantages

* + Better sense of finished product, judge aesthetic appeal
  + More realistic experience, evaluate the experience
  + More convincing to stakeholders

Disadvantages

* + User may focus on unnecessary details
  + Takes a lot of time to make
  + Lose track of the big picture

**Prototyping (2)**

------------------------------

* Fair and Equal
  + Equal not accurate
    - Difficult to measure workload

How do you create tasks that fit everyones skills, and is fair?

* Lead & manage
  + People don’t like to be managed
    - Leaders need to lead you to the goal
    - Their job is to make sure project delivered on time

Benefits of high-fidelity prototypes

* Realistic system response during the test
  + Done very quickly with the right tools
* Test workflow, specific UI components
* Look like “live” software to users
* Interactivity frees the designer to focus on observing the test instead of thinking what should come next
* Can be used to gather usability metrics

User interface looks exactly like the end product

* Can test the interactivity and components
* However, we understand it’s just a prototype, end user does not understand this.

Low and High fidelity

* Create a full horizontal first
* Create a few vertical implementations

Prototype functionality

* PICTIVE
  + User interface components

Difficult to prototype some piece of functionality

* Fake it with a person controlling the interface, make interaction as authentic as possible
* User has no idea interaction is being faked
  + Can prototype things that aren’t feasible (money, time), can use these simulations to get funding potentially
  + Wizard of oz technique

Video Prototyping

* Allows you to prototype without needing to be physically present
* Frees the designer from being around to simulate functionality

**Prototype Integration**

* **Evolutionary approach** to prototype integration
  + Iteratively change prototype to incorporate changes
  + Reworked prototype becomes final system
* **Modular (incremental)** approach to prototype integration
  + Build system as separate modules / components
  + Each module is designed, prototyped and built separately before being combined into final system
* **Throw-away** approach to prototype -integration
  + Used to get rapid feedback
  + Built, tested, deployed or discarded

Integrating prototypes in organizations

* **Evolutionary**
  + Altered to incorporate design changes
  + Becomes the final product
* **Modular (Incremental)**
  + Built as separate components (modules)
  + Module is prototyped and tested, then added to final system
* **Throwaway**
  + Reveal user reach, discarded
  + Creating should be rapid, otherwise its expensive