Humans and Computers

Course Information

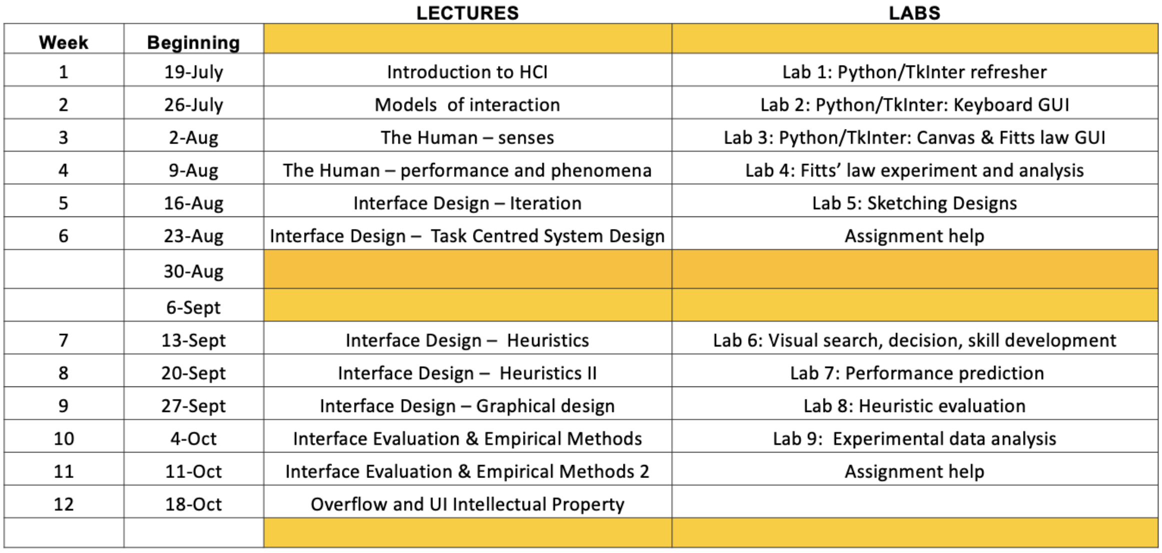
Lecturers Details

- Lecturer: Andy Cockburn
  - Email: andy.cockburn@canterbury.ac.nz
- Tutors:
  - Katia De Lu:
    * Email: katia.delu@canterbury.ac.nz
  - Stewart Dowding:
    * Email: stewart.dowding@canterbury.ac.nz
- Team alias: team368@cosc.canterbury.ac.nz

Schedule

Topics

- Introduction
- Models of interaction and interface technology
- The human
- Interface design
- Evaluation
- UI intellectual property



		LECTURES	LABS
Week	Beginning		
1	19-July	Introduction to HCI	Lab 1: Python/TkInter refresher
2	26-July	Models of interaction	Lab 2: Python/TkInter: Keyboard GUI
3	2-Aug	The Human – senses	Lab 3: Python/TkInter: Canvas & Fitts law GUI
4	9-Aug	The Human – performance and phenomena	Lab 4: Fitts' law experiment and analysis
5	16-Aug	Interface Design – Iteration	Lab 5: Sketching Designs
6	23-Aug	Interface Design – Task Centred System Design	Assignment help
	30-Aug		
	6-Sept		
7	13-Sept	Interface Design – Heuristics	Lab 6: Visual search, decision, skill development
8	20-Sept	Interface Design – Heuristics II	Lab 7: Performance prediction
9	27-Sept	Interface Design – Graphical design	Lab 8: Heuristic evaluation
10	4-Oct	Interface Evaluation & Empirical Methods	Lab 9: Experimental data analysis
11	11-Oct	Interface Evaluation & Empirical Methods 2	Assignment help
12	18-Oct	Overflow and UI Intellectual Property	

Figure 1: Course Schedule

Assessment Structure

- Labs (13.5%)
  - 1.5% per lab *changed due to COVID*
  - Binary marking scheme - go to the lab, get full marks
- Assignments have been merged due to COVID (30%)
  - Usability analysis and storyboard (25%)
    - Wed 22nd September 5:00 pm
    - Teams of ~six, forming own groups
  - Design Specification and Rationale (15%)
    - Wed 20th October 5:00 pm
- Exam (56%)
  - TBA
  - *weighting changed due to COVID*

Textbooks/Resources

- Designing with the Mind in Mind

    - Based on COSC368, Old lecturers thoughts
    - Author: Jeff Johnson, Morgan-Kaufmann

- Papers on ACM Digital Library
- Other materials on Learn

Lectures

Lecture One - Introduction

Technologies in this course

- TKinter
    - Lab one - Refresher
    - Lab two - Keyboard GUI
    - Canvas & fitts law GUI
- Python

> NOTE: Labs will be used as the basis for analysis in assessments, so we need to build them

What is HCI?

Human computer interaction (HCI) is a discipline concerned with the design evaluation and implementation of interactive computing systems for human use, and with the study of major phenomena surrounding them.

What is HCI Responsible for?

1. Learnability
2. Efficiency
3. Subjective satisfaction
4. *Memorability*

- Mostly encapsulated in Learnability

5. *Errors*

- Opposite of Efficiency

Lecture Two - Goals of HCI

Knowing the user: Preliminary Factors

- safety considerations
- need for throughput
- Frequency of use
- Physical space, lighting, noise, pollution
- Social context
- Cognitive factors: age, fatigue, stress, focus

Managing complexity

- Poorly designed interfaces amplify complexity
- Well designed UI's make interfaces as simple as possible, but no simpler
- Sometimes it may be appropriate to over-simplify *cater to an audience*

Models of Interaction

- A model is a simplification of reality
- They are useful when they help us understand a complex artifact

Don Norman's Model of Interaction

- Helps understand the designer's role in creating a system that is used by a thinking person
- Generally designers have a design model that is incomplete
- Then we get a system image, that is working, but in high incite, we would have built it differently
- The user has a model, that is weak, we need to try to map the designers model with the user model to create a mapping of the system image.

Lecture Three - The Human

Don Norman's Execute-Evaluate Cycle

- Execute:
  - Goal > Intention > Actions > Exeution
  - Gulf of Execution: Problem arrives when executing intention/action
- Evaluate:
  - Perceive > Interpret > Evaluate

    - Gulf of Evaluation: Problem assessing state, determining effect, etc. **UISO Interaction Framework**

- Emphasises translations during interaction - This is a cycle

    1. Articulation: user's task language to input language
    2. Performance: callbacks, etc.
    3. Presentation: show new state
    4. Observation: interpretation
    5. Back to step one

Mappings

- Good mappings (relationships) between User and I/O, increasing Usability.
- We can try to allocate real world relationships by using mappings to real world items

    - We can also use spacing and different design to make things more intuative

Affordance

- Objects afford particular actions to users

    - Buttons afford pushing, chairs for sittings, sliders for sliding, dials for turning, handles for pulling.
    - Poor affordance encourages incorrect actions
    - Strong affordance may stifle efficiency

Over and Under-Determined Dialogues

- Ideally dialogue is well-determined, natural translation from task to input language
- **Under-determined** - User knows what they want to do, but not how to do it
- **Over-determined** - User forced through unnecessary or unnatural steps

Direct Manipulation

- Visibility of object

- Direct, rapid, incremental, reversible actions

    - This allows users to learn the interface and experiment without loss of data or risk
    - Unix's `sudo rm -rf / --no-preserve-root` is **NOT** an example of this as it is not reversible

- Rapid feedback

- Syntactic correctness

  - Disable illegal actions

- Replace language with action
- Advantages:

  - Easy to learn
  - Low memory
  - Easy to undo
  - Immediate feedback to user actions
  - Enables user to use spatial cues

- Disadvantages:

  - Consumes screen real estate
  - High graphical system requirements
  - May trap user in *beginner mode*

Lecture Four - User Interaction and Psychology Behind Design

Psychological and physiological abilities have implications for design

- Perceptual: how we perceive things (input)
- Cognitive: how we process information
- Motor: how we perform actions (output)
- Social: how we interact with others

We are using these as a way to treat the human as an information processor, we are essentially mapping humans to a model in order to create a general solution.

The Human Information Processor

Based on a book by: card, Moran, Newell 1983

- Underlying psychology of interaction
- Predictive engineering models (GOMS/KLM)

  - GOMS, A library of models in order to map user interfaces and engineer effectively

- Extensive empirical validation
- Core computer science
- We will further break up the human brain into:

  - Long-term memory

- Working memory
    - Visual image storage
    - Auditory image store

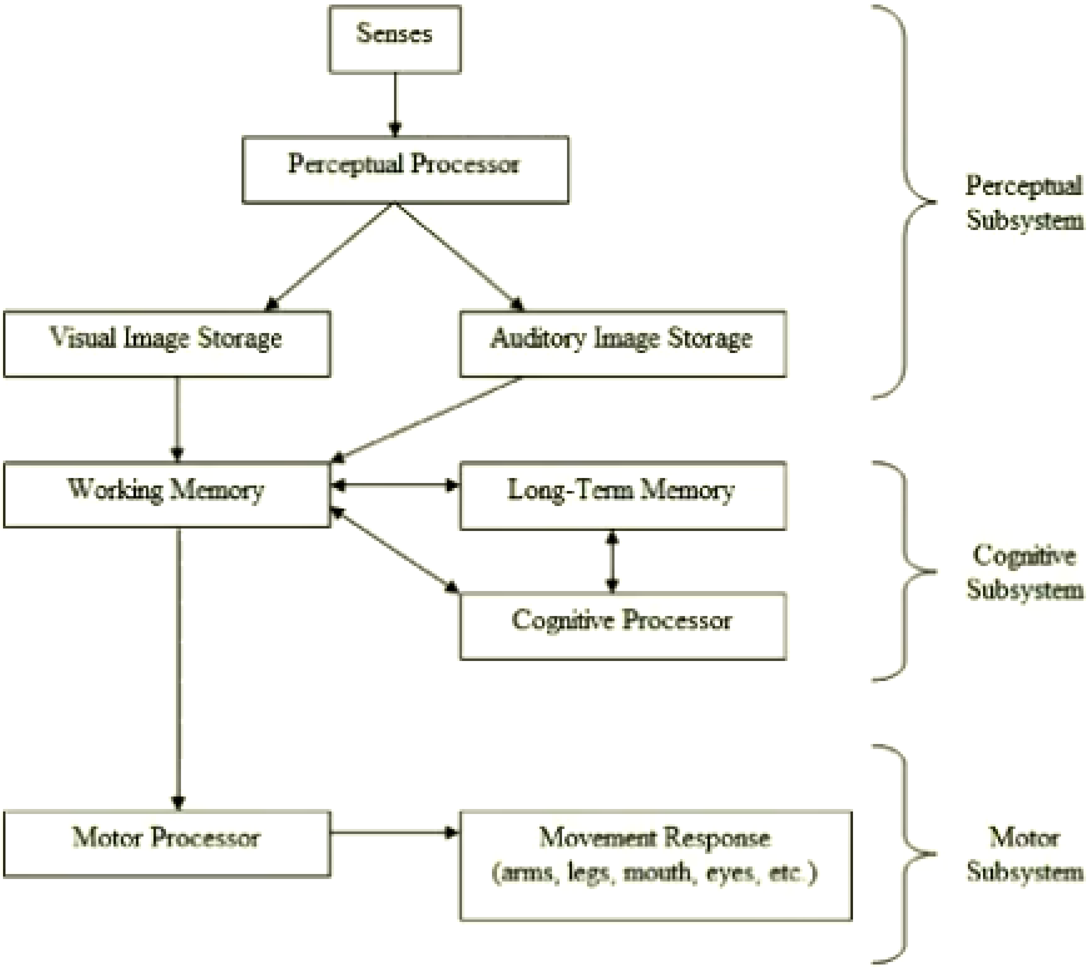


Figure 2: Sensory Model

- Human Input
    - Vision, hearing, haptics, olfaction
    - Vision:
        - Photoreceptors cells, rods: low light, monochrome, Cones: normal light, colour in fovea
        - Fovea: detailed vision of ~2 deg/sec
        - Retina: non-detailed vision of ~120 deg/sec; sensitive to movement

- Human Output
  - Pointing, steering, speech, typing, …
- Human Processing
  - Visual search, decision times, learning
- Human Memory
- Human phenomena & Collaboration
- Human Error
- And UI implications of each

Visual Acuity

- Point acuity:
  - One minute of arc
- Grating acuity:
  - 1-2 minutes of arc
- Letter acuity:
  - Five minutes of arc
- Vernier acuity:
  - 10 seconds of arc

We can do the maths to figure out the acuity in order to figure out if the text/shapes is readable in order to create user interfaces for a wider audience. For example we can calculate letter acuity in order to see if text will be visible to a whole stadium.

Eye movement

- Fixations: Visual processing occurs when the eye is stationary (nearly)
- Saccades: Rapid eye movements (900 deg/sec), blind
- Eye movement used as input via eye-tracker
  - Midas touch problem
- Smooth-pursuit: for tracking moving objects up to 100 deg/sec; cannot be induced voluntarily
  - Relevant in scrolling e.g. (SDAZ)
  - The problem with scrolling is not computers cannot scroll fast and accurate, but that we cannot keep up with the scrolling (our eyes fall into motion blur).

Lecture Five - Human Input: Depth-based UI's

Size and Depth Cues

- Familiarity
- Linear perspective
- Horizon distance
- Size constancy
- Texture gradient
- Occlusion
- Depth of focus
- Aerial perspective
- Shadows/Shading
- Stereoscopy

Depth-based UI's: 3D

- The real world is 3D
- So all interaction should be 3D, right?
    - No this is just a bad idea (lecturers opinion)
- 3D can be invaluable for interaction with 3D objects or in 3D Environments
- Terrible for navigation, to many mechanics
    - Occlusion is a huge issue (lots of stuff overlapping, cannot see)
    - Complexity

Depth-based UI's Zooming

- Overview first, zoom and filter, details on demand
- Allowing users to see everything, then allowing the user to get more specific when they want to
- Example: *zillow* website

Input: Haptics

- Proprioception: sence of limb location +
- Kinaesthesia: particularly limb movement +
- Tactition: sensations
- Potentially powerful: e.g. Braille

Human Output

- Motor response times depend on stimuli

  - faster for combined signals

- Muscle actions
  - Isotonic: contraction yields movement
  - Isometric: contraction with no movement

Fitts' Law

- A model of rapid, aimed human movement
- Predictive of tasks; descriptive of devices
- Derived from Shannon's theory of capacity of information channels
- Extremely accurate and extensively validated for many types of aimed pointing
  - Consider velocity profile

Lecture Six - Input Devices: Pointing, Scrolling and Textual

Fitt's Law

- Movement of time (MT) is linear with *ID*
  - $MT = a + bID$ or $MT = a + blog_2(A/W + 1)$
  - Reciprocal of slope *(1/b)* also called throughput or bandwidth of device, measured in bits/second

> NOTE: we are expected to be able to reproduce these formulae

Pointing and Scrolling

- Human output received as system input
- Direct vs Indirect
- Control: Position (zero-order), rate (first order), acceleration (second-order)
- Isotonic (force with movement)
- Isometric (force without movement)
- Control-Display gain and Transfer Functions

Steering Law

- A model of continuously controlled 'steering'
  - $MT = a + bID$ or $MT = a + b(A/W)$
  - A is the tunnel length; W is tunnel width
  - (A/W) is still called the `index of difficulty`

Text Input

- Alternative keyboard (Devorak)
- Chord keys
- Constrained keyboards
- Reactive/predictive systems (e.g. Dasher)
- Gestural input (unistrokes, shapeWriter/swipe)
- Hand-written recognition

Visual search time

- Extensively researched in psychology
- Linear complexity $O(n)$, unless we can create a pop out effect, then complexity is $O(1)$
- We should try to avoid this if possible, because it is slow with large values of n

Lecture Seven - Human Processing: Visual Search Time

- Extensively researched in psychology
- Visual search time is a linear function
- Pop-out effects can reduce linearity to O(1)

Hick/Hyman Law of Decision Time

- $T = a + b \times H$
- H is information entropy
- With $H_i = log(\frac{1}{p_i})$
- For all n with equal probabilities $H = log_2(n)$
- We make frequent decisions quickly
- Decision times are fast $O(log\, n)$
- Applies to name retrieval and location retrieval
- In GUI's replace visual search with decision from spacial stability

We want to design for novice users to be able to transition to expert users.

Lecture Eight - Human Memory

Zipf's Law, Pareto Principle

- Frequency of words (Zipf 1932)
  - $P_n = n^{-a}$

  - P_n is a scaling factor of frequency of n^{th} rank word
  - $a = 1$
  - Models fequency of commands, URL's, apps, windows, etc.

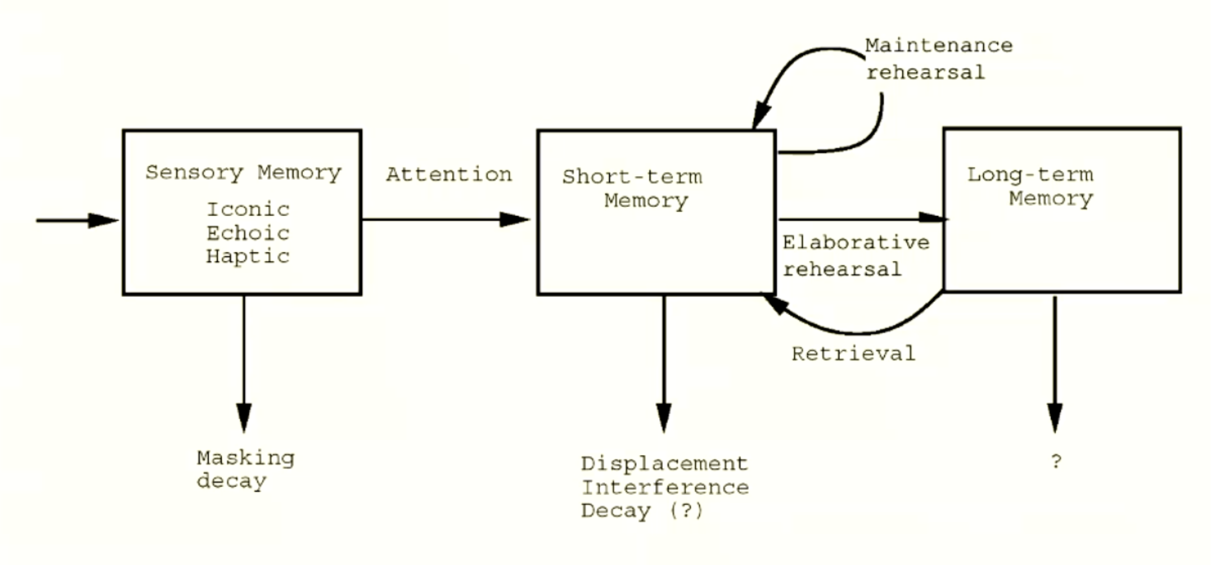


Figure 3: Human Memory: Simplified Model

Short-term memory

- Input from sensory or long-term memory
- Capacity up to 7 (+/-) 2 chunks
- Chunks used to aid storage and reconstruction
- Fast access, rapid decay
- Constant update and interference
- Maintenance rehearsal

Human Error: Mistakes

- Errors of conscious decision making
- Due to incorrect or incomplete model of system
- Only detected with feedback

Human Error: Slips

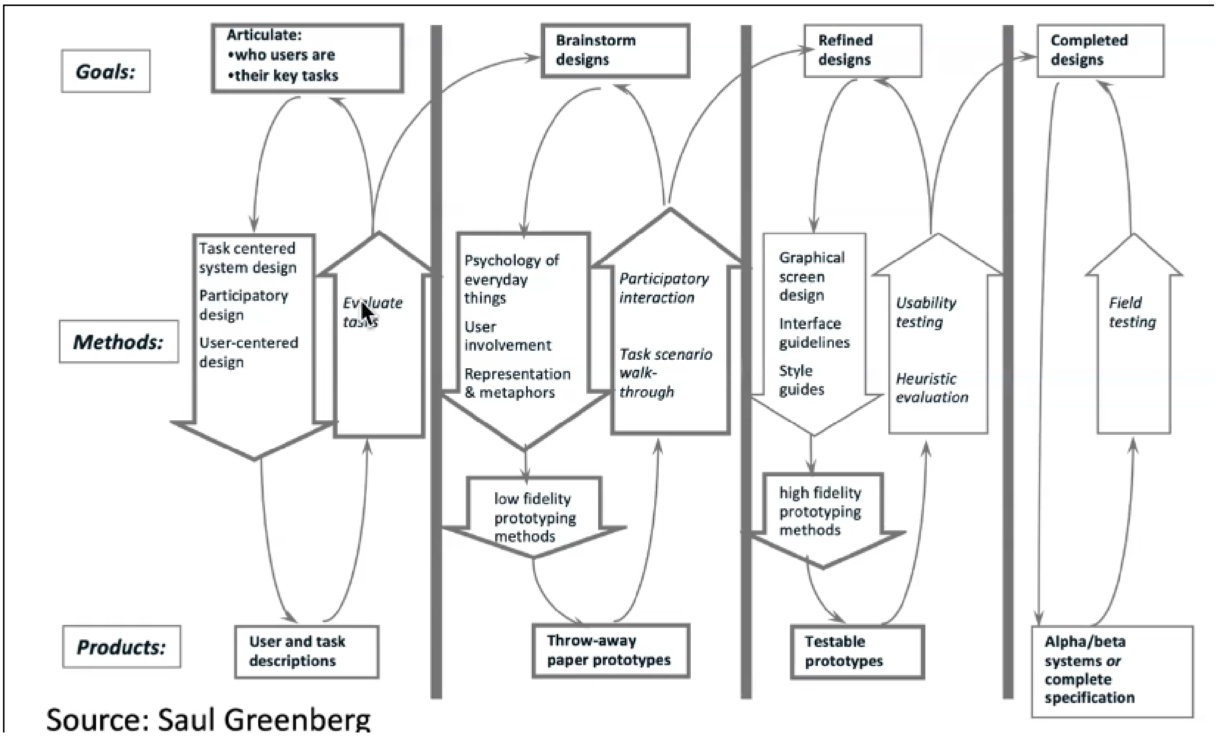
- Errors of automatic and skilled behaviour
  - Capture error
    - Two action sequences with common starting point

- Description error
    * More than one object allowing the same/similar action
- Data-driven error
    * External data interferes with STM
- Loss-of-activation error
    * Goal displaced/decayed
- Mode Error
    * Right action, wrong system state
- Mode slip
    * Pointing or steering error
- Premature closure error
    * Dangling UI actions required after perceived goal completion
- What is a mode?
    * system partition
    * Modal dialog
    * Ensure modes are visible and noticeable

Lecture Nine: Design Process

Assignment Information: What we are covering

We will cover the full design process, we will be implementing up to brainstorming for the assignment, and will be covering refining design and implementing next term. Below is the full design process outline.



Source: Saul Greenberg

Figure 4: Design Process

Iterative Design

- Elaboration: get the right design
- Reduction: get the design right

1. Starts up with design and marketing
2. More and more marketing and less design
3. Sales comes in and design and marketing both decrease

Supporting Rapid Iterations

- Fudd's first law of creativity: to get a good idea, get lots of ideas
- But lots of ideas will take lots of time to implement
- Time is precious
- Requires rapid creation and evaluation
- Rapid prototyping

Early design

- Is when we start prototyping
- We must have:

    - brainstormed different representations
    - choose a representation
    - Rough out interface style
    - Fine tune interface, screen design

Low fidelity prototypes: sketches and wireframes

- Outward appearance and structure of intended design
- Necessarily crude/scruffy
    - Focus on high level concepts
    - Fast to develop
    - Fast to change
    - Low change resistance
    - delays commitment
- Annotations/sequence to shoe UI progression

Lecture Ten: Early UI Concepts

Sequential sketches/storyboards

- Show state transitions in your wireframes, *how the user changes from page to page*
- Focus on *main* interactions (Zipf's law)

Medium fidelity prototypes

- Functionality is mimicked by a person
- The wizard must know the algorithm
- Good for complex/futuristic ideas
- Facilitates motion paths
- Links between states, etc.
- There are lots of wire-framing tools
    - moqups.com
    - balsamiq.com

Must have:

- Series of key frames
- State progression is clear
- Walk-through evaluation

First we can use wireframes, secondly we can use simulations and animations in order to have a better visual representation of how it will work when implemented.

Watch out for when doing further mock ups:

- Giving perception of `nearly completed`
- Reluctance to change
- Excessive focus on presentation rather than approach

System Centered System Design (A bad approach)

- Focusses on the system and designers needs
- Asks the question, what is easy?

Task-Centred System Design

- HCI equivalent of requirement analysis
- Exactly and specifically who are the users and what they will use the system for?
- Critical difference between:
  - "The User": a pretend person who will adapt to the system
  - "Mary": A reality-based sanity check for designers
- User Identification
  - Identify categories of end-users, with specific exemplars:
    * Casuals
    * Power users
- Record what the users wants to do, but minimize a description of how they do it (*Vim is an example of this*)
- Record the complete task: input source, output destination
- Uniquely enumerate tasks for identification
- Identify broad coverage of users and use cases

Lecture Eleven: Task-Centered System Design

What is TCSD?

- HCI equivilent of requirements analysis/use cases
- Exactly and specifically who are the users and what will they use the system for?
- Critical difference between:
  - Our percieved user

  - An actual user

- TCSD: A reality-based sanity check for system designers
- Focus is on generating the designs
- should be based around user's needs, abilities, context, work, etc

Cautions using TCSD

- Tasks and task scenarios often embody process
- Hard to record identified tasks and write task scenarios that are independent of interface or workflow prescription
- This may hinder identification of alternate ways to achieve tasks
- Can be hard to find people `responsible` for new tasks in a system

Phases of task centered system design

1. User Identification

- Identify categories of end-users with specific examples
  - Including typical users and extreme users (edge cases)

2. Task Identification

- Record what the user wants to do, but minimise description of how
  - No interface assumptions
  - Can be used to compare alternative designs
- Record the complete task, input source, output destination
- Identify Users
  - design success depends on what users know
  - Tested against specific individuals
- Uniquely enumerate tasks for identification
- Identified tasks are circulated for validation
- Identify broad coverage of users and tasks

Participatory Design - an extension of UCSD

- Problem:
  - institutions can be wrong
  - Interviews lack precision/context and can mislead

  - Designers cannot know user’s needs sufficiently well to answer all questions likely to arise during design

- Solution:

  - Designers need access to a pool of representative end users
  - Not managers, union-reps; real users
  - These users are full members of the design process

- Talk to users
- Interviews with users

  - discover users culture, requirements, expectations

- Explain designs

  - get output at all design stages
  - important to have visuals/demos: prototypes

- Walk-through with the user

  - *Let them demo an interface idea or sketch*

Lecture Twelve: Heuristics

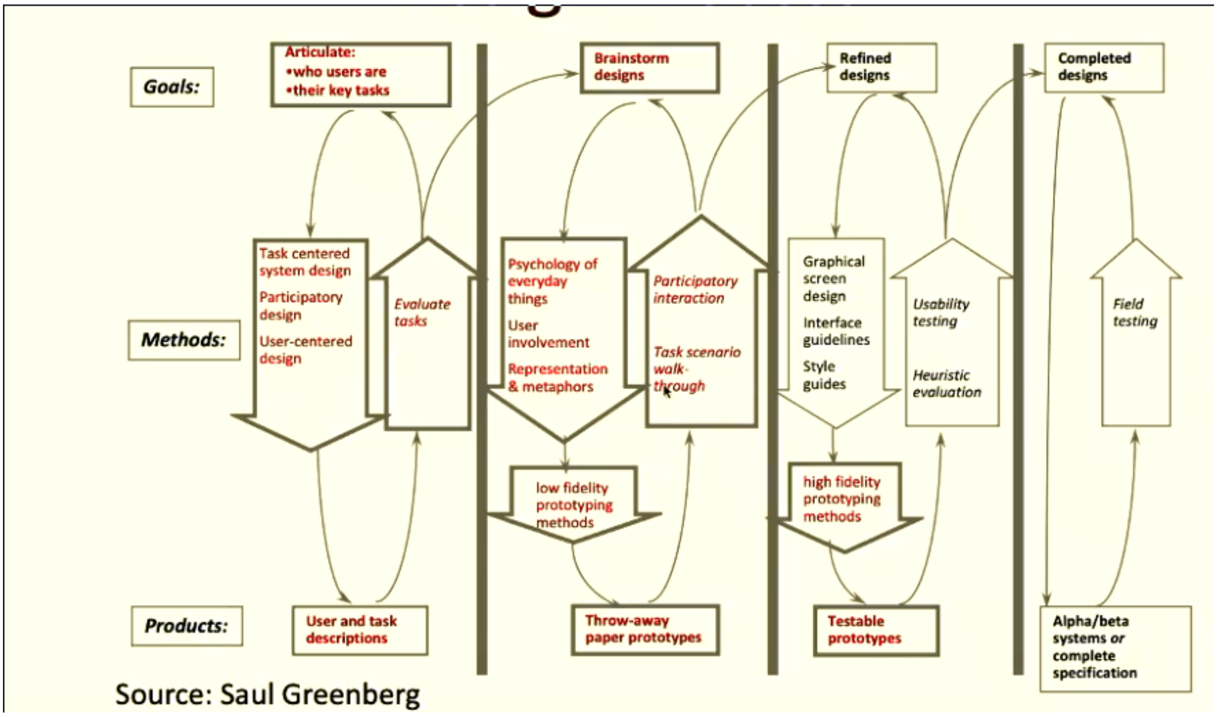


Figure 5: Design Process Lifecycle

Usability Heuristics

- Encapsulate best practice in `rules of thumb`
- Identify common pitfalls
- Are highly redundant

Types of Usability Heuristics:

- Formative: guide's design decisions
- Summative: evaluate systems

Nielsen's Heuristics

1. Simple and natural dialogue

- Managing complexiity: as simple as possible but no simpler
- Organisation of the interface:
  - Presentation: simple and natural?
  - Navigation: simple and natural?

- Graphic design
  - Organise, Economise, Communicate
  - Employ a graphic designer
- Use windows frugally
- Less is more

2. Speak the user’s language
3. Minimise user memory load
4. Consistency
5. Feedback
6. Clearly marked exits
7. Shortcuts
8. Good error messages
9. Prevent errors
10. Help and documentation