2.1

Usefulness and Usability
The ultimate goal of design in HCI is to create interfaces that are both useful and useable.

> Useful - the interface allows the user to achieve a task
> Ex. map is useful but not usable

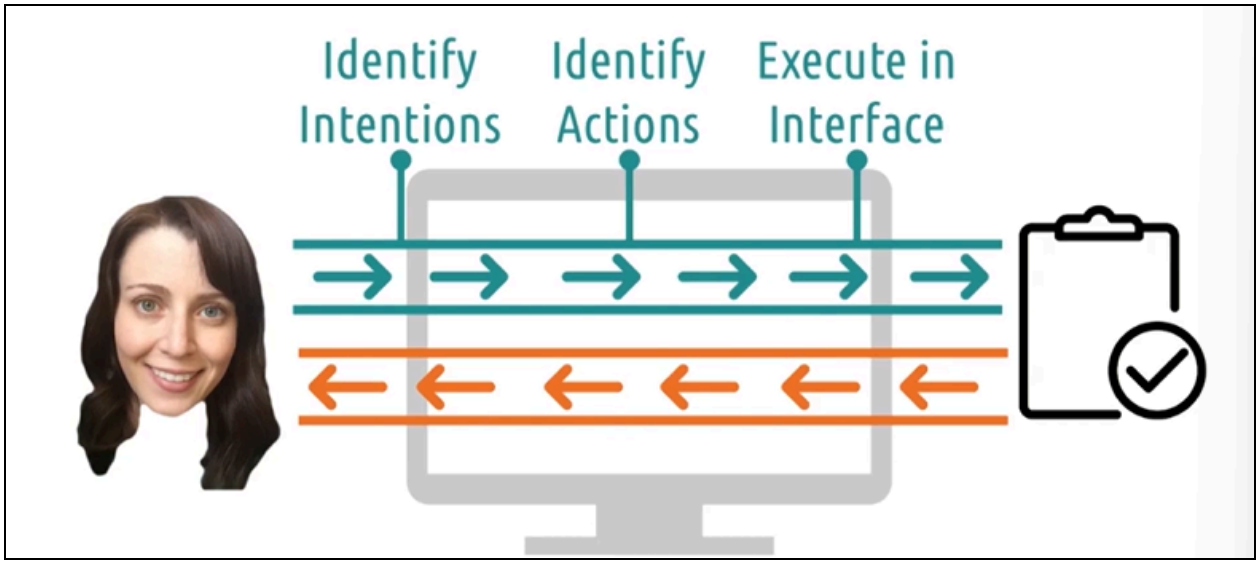
Three views of the user:
Processor: treats users as input-output machines
Predictor: the user should be able to predict what will happen as a result of their actions
Participant: extends beyond the individual, considering the user's environment, social context, and competing attention demands.

2.2 Intro to feedback Cycles - Week 2

Gulf of execution: the distance between a users goals and the actions required to realize those goals.

User: How do I know what I can do? Make the state of the system match the goal state
What's the difference between what the user thinks they have to do vs. what they actually have to do.

Steps:
1. Identify intentions
2. Identify actions
3. Execute in Interface



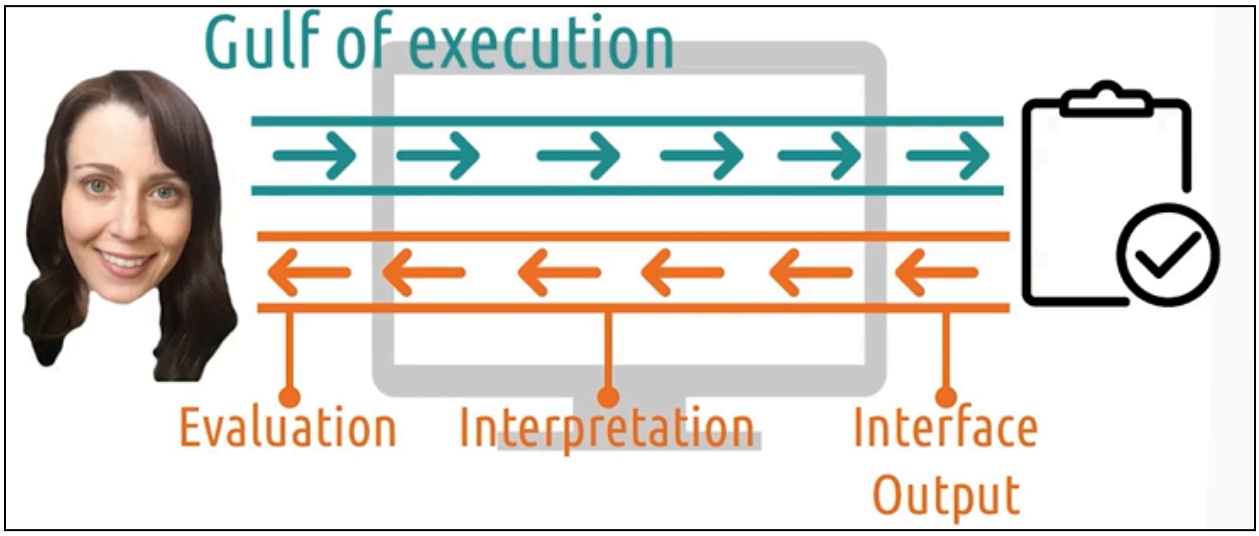
5 tips

1. Make functions discoverable
2. Let the user mess around
3. Be consistent with other tools
4. Know your user
5. Feedforward - help users predict what the next action will be ex. Loading icon

Gulf of evaluation: the distance between the effects of those actions and the user's understanding of the results

Steps:

1. Interface Output
2. Interpretation
3. Evaluation



Buffering icon helps user interpret the output and leads to a correct evaluation.

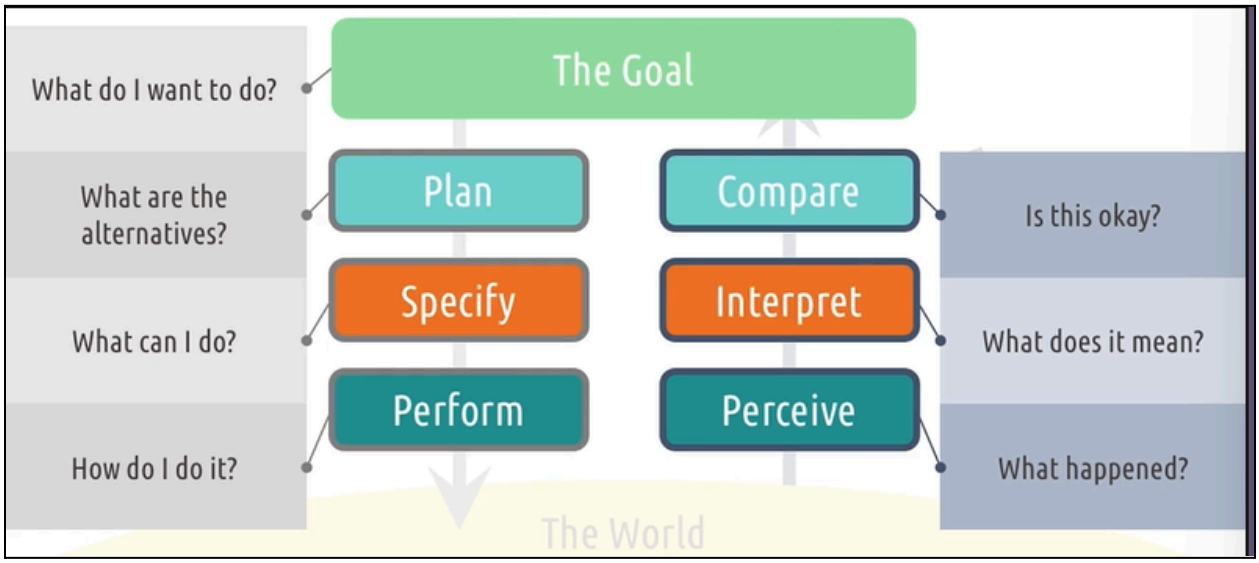
5 tips:

1. Give feedback constantly
2. Give feedback immediately
3. Match the feedback to the action
4. Vary your feedback ex. Vibration
5. Leverage direct manipulation

Feedback Cycles are Fundamental

- Do something see the result and adjust what you do the next time

Don Norman calls it a bridge of execution and bridge of evaluation



3.1 Introduction to Research Methods -Week 2

4 stage Design Lifecycle
User centered design
Qualitative vs Quantitative Data

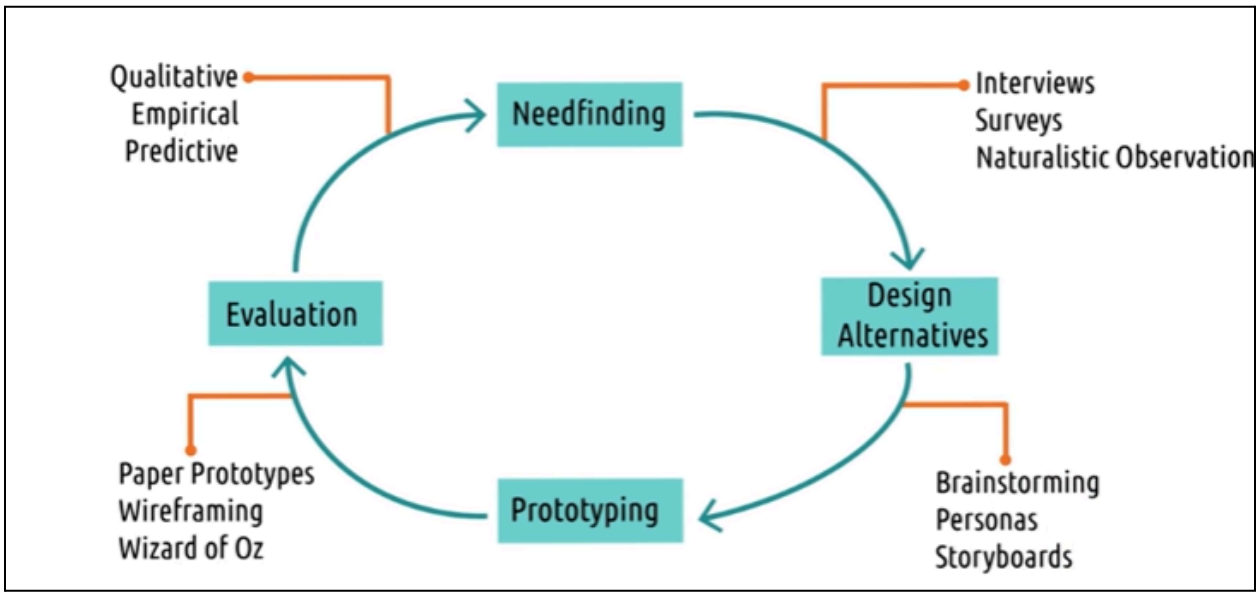
User-centered Design: Design that considers the needs of the user throughout the entire design process.

User - primary stakeholder use tool directly
Secondary don't use tool directly but interact with the output

Tertiary - affected by the existence of the tool , don interact with the input or output

The Design Life Cycle

Needfinding -> Design Alternatives -> Prototyping -> User Evaluation -> (back to needfinding)



Quantitative Data: anything numeric
Strong for a small class of things
The “what”
More expensive to analyze, its better to convert to quantitative data
General Types: transcripts, field notes, artifacts(existing UI), and anything else

Qualitative Data: natural language ex. Survey results
More prone to bias, flexible and broad, harder to generate formal conclusions
The “how or why”

4 General Types: Nominal (categories), Ordinal (scale), Interval(ex. time), Ratio

3.2 Ethics and Human Research - Week 2

IRB - institutional review board

National Research Act of 1974 lead to the creation of IRB for university research
The law dictated that the benefits to society must outweigh the the risks to the subject
Rigorous and informed consent

IRB's main task is to make sure the potential benefits of the study are worth the potential risks

So for the irb is sensitive about the perception of coercion, when participants feel coerced to participate in experiments they might be thrown off by the negative perception so that can negatively impact the results.

Protocol
A protocol is a description of a particular research project
It outlines the **procedures** that the IRB has approved regarding **consent, recruitment, experimentation**, and more.

2.3 Direct Manipulation - Week 3

Direct Manipulation: the user should interact directly with their task
Should feel natural
Ex. pinch fingers to zoom, drag and dropping files
Used to narrow the gulf of execution and evaluation

Invisible Interface: an interface that disappears between the user and their task
Ex. writing with a electronic pen vs. mouse
The interface should feel invisible from the moment the user starts using it

5 tips:

1. Use affordances
2. Know your user
3. Differentiate your user
4. Let your interface teach
5. Talk to your user

The greater cognitive loader required to use a system, the less direct the interaction with the system will feel

2.4 Human Abilities - Week 3

Psychology
Input - How stimuli are sent from the world and perceived inside the mind
Processing(cognition) - how the brain stores and reasons over the info that it has received
Output - how the brain controls the individuals actions out in the world

Types of Memory
Perceptual Store - lasts less than one second
Short-term memory - big concern when designing, people can save 4-5 chunks at a time
Long-term memory - unlimited load, for something to be in long term it needs to show up in short term many times

Procedural Learning - How to do something ex. Paste the clipboard
Declarative Learning - Knowledge about something ex. Knowing crtl+v

We want to reduce the cognitive load posed by the interface so that the user can focus on the task also understand the context of whats going on while the user is using the interface

5 tips to reduce cognitive load

1. Use multiple modalities (visual and verbal presentation) -
2. Let the modalites complement each other (ex. Text describes image)
3. Give the user the control of the pace
4. Emphasize essential content and minimize clutter
5. Offload tasks (ex, show them what they entered on the previous screen)

2.5 Design Principles and Heuristics - Week 3

Don Normans Six Principles of Design
Constantine's and Lockwood Six Principles
Jakob Nielsen's Ten Heuristics
Seven Principles of Universal Design

All of these are merged into 15 principles

Discoverability - functions should be visible to the user so that they can discover them rather than relying on them learning it somewhere else.

Simplicity - the user should only be given as much as information as they need

Affordances - Design interfaces that by their very design tell you how to use them.
The design of the thing affords or hints at the way its supposed to be used.
Look at it and understand how to use it. Ex. door handle
Defined by who the user is

Mapping - the relationship between the elements of two sets of things
Ex. the screen showing num of displays in windows settings
Shows the affect of an action

> Light switch - affordance but no mapping
> Stove burner dials - affordance and mapping

Perceptibility - the users ability to perceive the state of the system
The system to keep users informed through appropriate feedback

Consistency - refers to using controls,visualizations, layouts or anything we use in interface design consistently.
Have purpose and reuse ex. Using the word copy vs duplicate

Minimize the amount of learning the user needs to do to learn our interface, create affordances
Create expectations and conventions ex links are blue and turn purple once clicked on

Flexibility - system can cater to inexperience and experience users. Ex.ctrl+c or right click copy

Equity - provide the same means of use for all user, avoid segregating any users ex. Password requirements instructions

Ease and Comfort - design can be use with min amount of fatigue - appropriate size and space is provided regardless of users body size. Ex. size of buttons on mobile interface

Structure - design should organize user interface purposefully

Constraints - constrain the user to only performing certain actions to avoid user error

Normans Four Types of Constraints:

Physical: ex. 3 prong plug
Cultural: waiting in line,
Semantic: rear view mirror, must reflect behind
Logical: self evident based on situation. Ex. One screw left with one hole left

Become more visible since they are in the users way

Tolerance - users should not be at risk of causing too much trouble accidentally. Users need an escape like undo and redo. Minimize hazards, let user poke around

Feedback - system should give feedback so user can understand what happened, poor feedback can be worse than no feedback

System should tell you how to recover from an error without an code, plain language

Documentation - one goal is to avoid having documentation in general
The user should be able to use the interface without documentation

3.3 Needfinding and RequirementsGathering - Week 3

a

3.4 Design Alternatives - Week 3

2nd phase of design lifecycle
When we start to brainstorm how to accomplish the task we have been investigating

The biggest mistake a designer can make is jumping straight to designing an interface without understanding the users or understanding the task.

The second biggest mistake a designer can make is settling on single design idea too early.

Design space: the area in which we design our interfaces

Individual brainstorming 1:Goal is to generate lots of ideas

Brainstorm individually before brainstorming as a group

Write down everything, even a few words or sentences

Think about different types interfaces, interactions, the audience

Get silly

5 Tips: Individual Brainstorming

1.Write down the core problem
2. Constrain yourself - have at least one idea in different categories
3. Aim for 20 ideas - prevents tunnel vision
4. Take a break - minutes or days later
5. Divide and conquer - brainstorm for each smaller problem

Challenges in group brainstorming:

4 behaviors that can black progress:
Social loafing: The tendency to exert less effort working in groups than working alone
Conformity: The tendency to agree with or follow the groups reasoning and ideas
Production Blocking: The tendency of some individuals in discussion to block other individuals' participation.
Performance matching: the tendency to match one level of performance to other collaborators
Power dynamics/biases: the tendency to defer to more senior individuals or to overpower less senior individuals

Rules for Group Brainstorming

1. Expressiveness - share all ideas
2. Nonevaluation - no criticism yet
3. Quantity - as many ideas as possible
4. Building - build off of others ideas

1. Stay focused - keep the goal in mind at all times
2. No explaining ideas - say the idea and move on
3. Revisit the problem often
4. Encourage others to speak up

5 Tips: Group Brainstorming

1. Go through every individual idea - from earlier sessions
2. Find the optimal size - not more than 5
3. Set clear rules for communication - set timer to speak
4. Set clear expectation - set session time
5. End with ideas, not decision - take time to think about ideas

Personas:

Create characters to represent users

Name, face, and details

Who what when where why

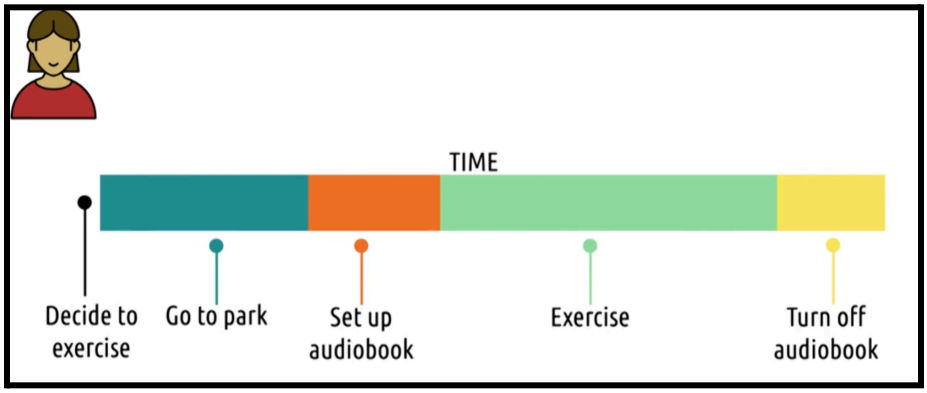
Should be different types of people

User profiles:

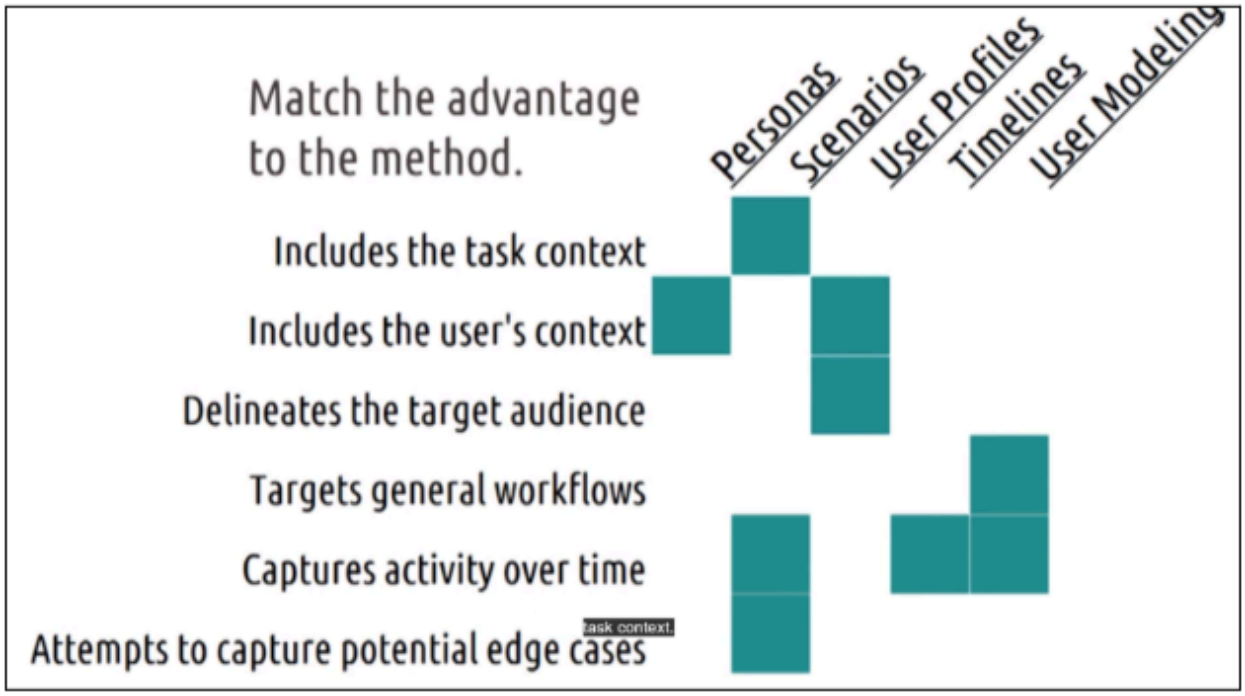
Exercise Expertise	Novice	⟷	Expert
Reading Level	Casual	⟷	Serious
Motivation	Low	⟷	High
Tech Literacy	Low	⟷	High
Usage Frequency	Low	⟷	High

Timelines:

Stretch out a persona over time



Scenarios and storyboards
Specific kinda of interactions that we want to be able to handle
Timelines are very general, show routine interactions
Scenarios are more specific



Exploring HCI Design Alternatives
a

2.6 Mental Models and Representations - Week 4

Mental model - an internal, simulatable understand of external reality

Representations: Internal symbols for an external reality
Good rep show the users how the systems work
A good rep for our problem will make the the solution self-evident

A mental model is a person's understanding of the way something in the real world works. Using mental models we generate expectations or predictions about the world then we check whether the actual outcomes match our mental model.

As a designer we want to make sure that the users mental model in our systems matches the way our systems actually work.
We can do this in two ways:

Designing systems that act the way users expects them to act
Designing system that teach the user how they react

Characteristics of a good representation/ four criteria

1. Good representations make relationships explicit
2. Good representations bring objects and relationships together
3. Good representations excludes extraneous details
4. Good representations expose natural constraints

5 Tips: mental models for learnable interfaces

1. Predictability - can a user predict what will happen ex. Grey out button
2. Synthesizability - the user should be able to predict the effects of an action before they perform it, they should also be able to see the sequence of action that led to their current state. Ex. undo menu log, CLI is good at this
3. Familiarity - similar to affordances ex. Green for good, red for bad
4. Generalizability - similar to consistency(Norman) knowledge of one user interface should generalize to others
5. Consistency - similar tasks and operations within a single interface should behave the same way. Ex. ctrl-x behavior should be consistent across the interface

User error is a failure of the interface of properly guiding the user

Slips: the user has the right mental model, but does the wrong thing anyway
Ex. popup window to save their work
User knows the right thing but does the wrong thing

Norman: action based: doing the wrong thing
Memory lapse: forgetting to do the right thing

Mistakes: the user has the wrong mental model and does the wrong thing as a result. User doesn't know the right thing to do

Norman: Rule based mistakes: user correctly asses state of the world but makes wrong decision based on it
Knowledge based: user incorrectly asses the state of the world in the first place
Memory: similar to mem slips, forgetting to fully execute plan

Learned Helplessness: A users sense that they are helpless to accomplish their goal in an interface.

The user has learned that no matter what they do they can not succeed

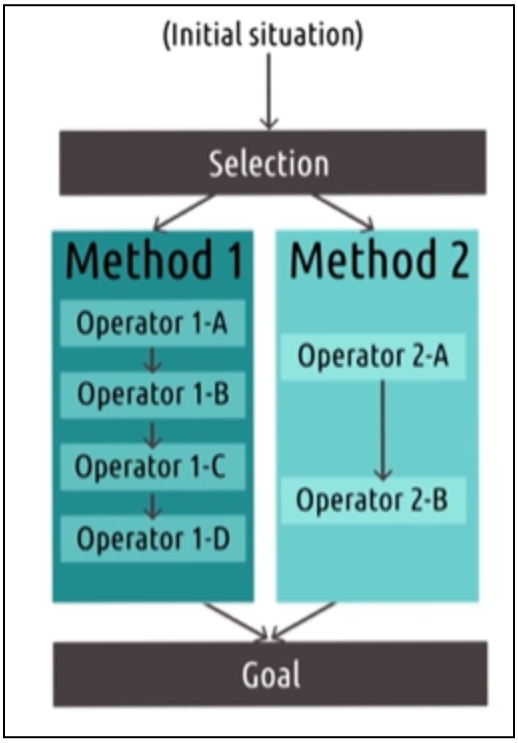
Expert blindspot, you overlook certain steps since you are an expert.
I am not my user - the motto of HCI

2.7 Task Analysis - Week 4

The task is at the heart of HCI.

Two methods for Task Analysis

Human information processor models - GOMS model
Cognitive Task analysis



GOMS model
Goals Operators Methods Selection rules

Weakness - does not address complexity, assumes the user is an expert

Strength - helps formalize user interactions, you can measure how long each method takes

5 tips for developing GOMS models

1. Focus on small goals
2. Nest goals, not operators
3. Differentiate descriptive and prescriptive
4. Assign costs to operators
5. Use GOMS to trim waste

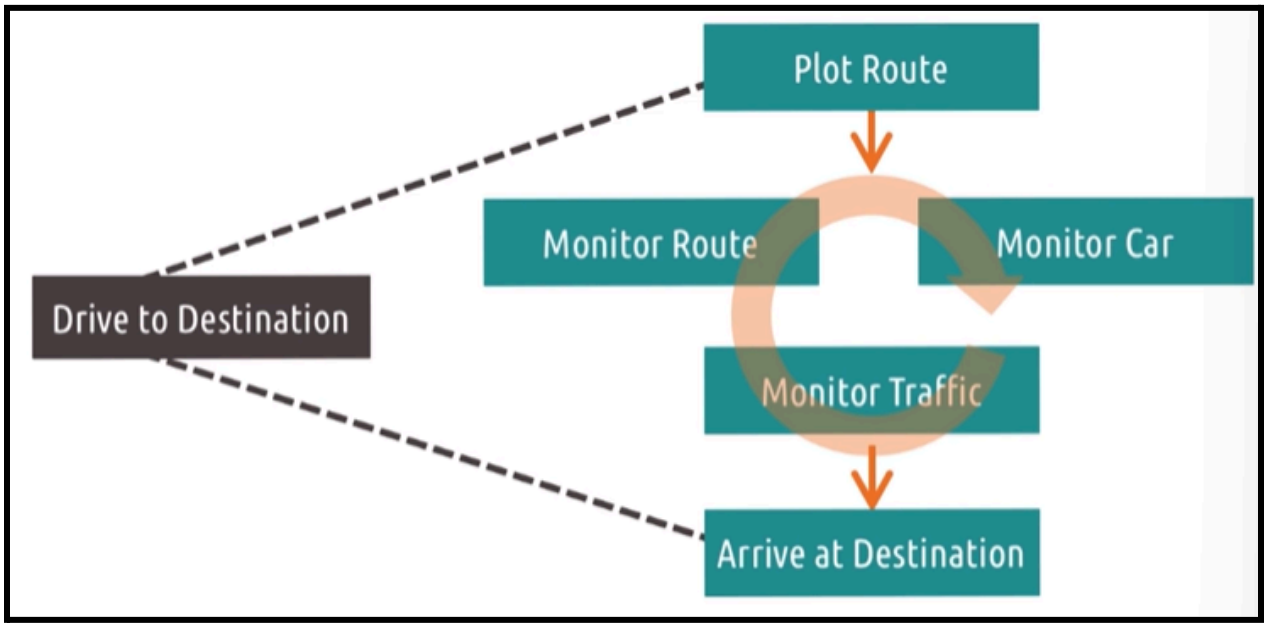
Behaviorism: an approach to psychology that emphasizes behavior as a product of stimuli and the environment

Cognitivism: an approach to psychology that emphasizes internal thought processes

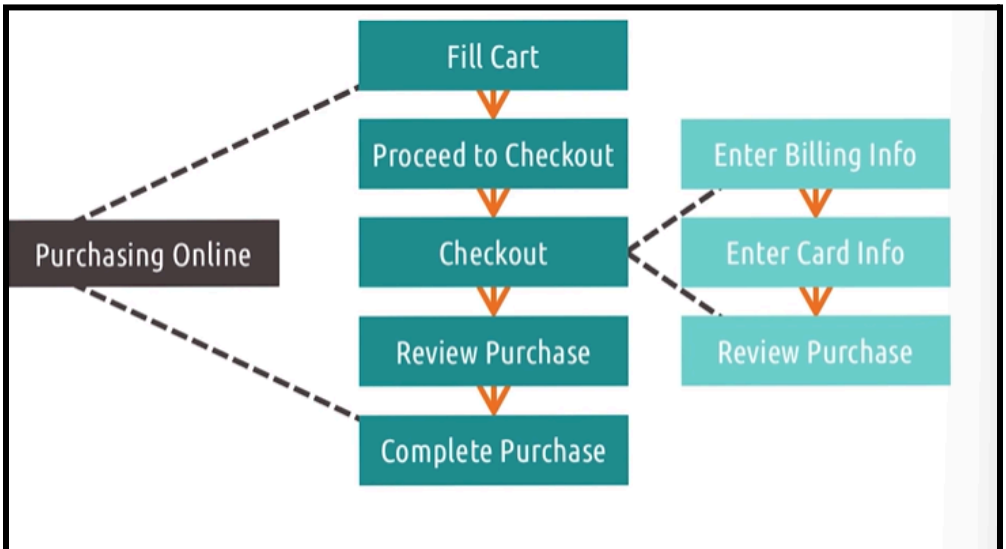
Cognitive task analysis is a type of method for approaching the evaluation of how people complete tasks

Not observable , the user would have to narrate what they are doing
Steps

1. Collecting preliminary knowledge ex. Watch the user do a task
2. Identify knowledge representations ex. Are the tasks ordered
3. Apply focused knowledge elicitation methods ex. What the user actually knows, get the user to tell you
4. Analyze and verify data acquired
5. Formatt results for the intended application



Hierarchical Tasks Analysis

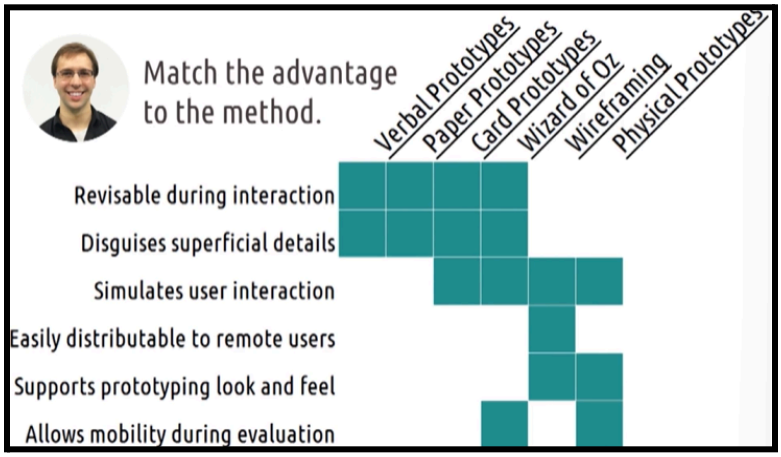


3.5 Prototyping - Week 4

Prototyping - build things to put in front of users and get feedback
Low-fidelity prototypes - things that can be assembled and revises quickly for rapid feedback form users - far from complete
    Good for evaluating the function
High Fidelity prototypes: wire frames or working versions of interface
    Good for measuring performance
4 main concepts
Representation - what is the prototype - verbal, paper, wireframe, functional
Fidelity - completeness of prototype
Evaluation - function, interface, performance
Scope - horizontal prototypes at a whole but less details
    Vertical prototype - small section in detail

5 tips: Prototyping
1. Keep prototypes easy to change
2. Make it clear that its a prototype
3. Be creative, get feedback
4. Evaluate risks
5. Prototype for feedback

Verbal prototypes: a description of the design or analogy
    Advantages: revisable during interaction and disguises superficial details
Paper prototype: draw on a paper, can quickly revise -
    Advantages: revisable during interaction and disguises superficial details
card base prototyping:
    Advantages: revisable during interaction and disguises superficial details and simulates user interaction
Wizard of Oz prototyping: a human performs the actions with the user.
    Advantages: revisable during interaction and disguises superficial details and simulates user interaction, allow mobility during evaluation
Wireframing: drawing using a software tool
    Advantages: simulates user interaction, easily distributable to remote users, supports prototyping look and feel
Physical prototypes:
it doesn't have to actually work
    Advantages: simulates user interaction,, supports prototyping look and feel, allow for mobility during evaluation



Our goal is to get feedback rapidly

3.6 Evaluation - Week 4

Evaluation: take what we designed and put it in front of users to get feedback

Qualitative evaluation: is it readable/understandable
Empirical evaluation: quantitative, has it actually improved metrics ex sales

Predictive evaluation: predict what the result of the user evaluation might be

Three types of evaluation:
- Qualitative: what do they like/dislike, whats easy/hard
- Empirical: evaluate quantitatively
- Predictive: evaluation without users

Terminology:
Reliability: Whether a measure consistently returns the same results for the same phenomenon.
Ex. asking someone the time and they give the correct answer
If we were to conduct the same procedure again how likely is it that we would get the same results? - thats reliability

Validity: Whether a measure's results actually reflect the underlying phenomenon.
Refers to how accurately an assessment measures reality
Ex. can be reliable but inaccurate, giving the wrong time

Generalizability: Whether a measure's results can be used to predict phenomena beyond what is measured.
The extent to which we can apply lessons we learned in our evaluations to people who were not in the evaluation aka broader audiences.

Precision: The level of detail a measure supplies.
The measurement of how specific some assessment is.
How specific are our conclusions and observations.

5 Tips: What to evaluate
1. Efficiency - how long does it take a user to accomplish a task
2. Accuracy - how many errors do users commit while accomplishing a task
3. Learnability - how long does it take the user to hit a level of expertise
4. Memorability - the users ability to remember how to use an interface over time

5. Satisfaction - the users enjoyment of the system or the cognitive load they experience while using the system.

Evaluation Timeline:

Overtime the methods will change
Start as formative then move to summative
The purposes

Early - Formative - evaluation with the intention of improving the interface going forward
Late - Summative - with the intention of conclusively saying at the end what the difference was, in reality hopefully we never do this.

How to fulfill the purposes
Early - Qualitative - more interpretative/informal - their goal is to help us improve or understand the task
Late - Empirical - more controlled/formal - their goal is to demonstrate or assess change

Predictive - inform how we revise and improve our interfaces over time

Lab Testing - when testing low fidelity models
Field Testing - working prototype

Evaluation Design - aka put the prototype in front of users

1. Define the task
2. Define Performance Measures - How are we going to evaluate the user's performance
3. Develop the Experiment
4. Recruit Participants
5. Do the Experiment
6. Analyze the data
7. Summarize the Data

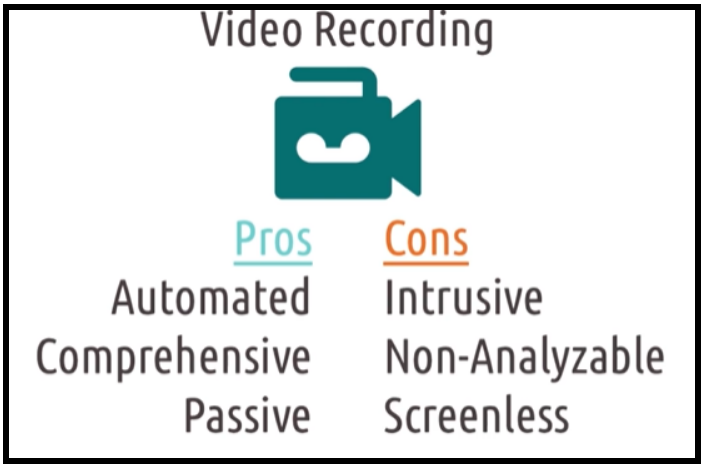
Qualitative Evaluation:

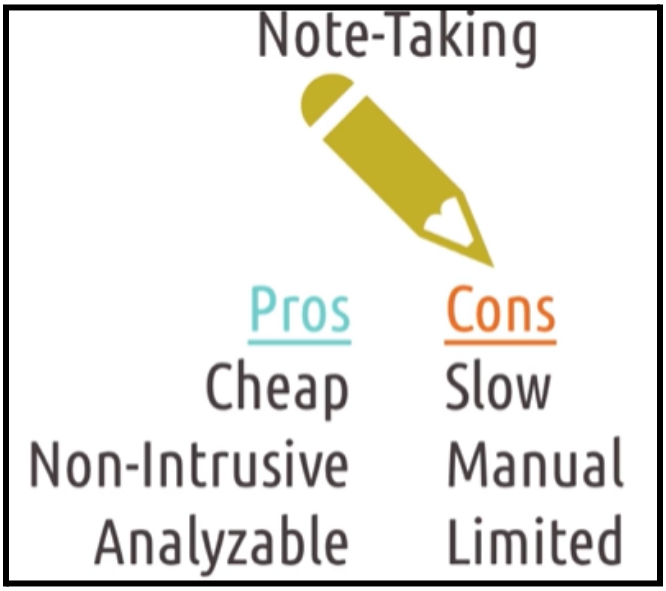
What did you like/dislike?
What were you thinking while using the interface?
What was your goal when you took that particular action?

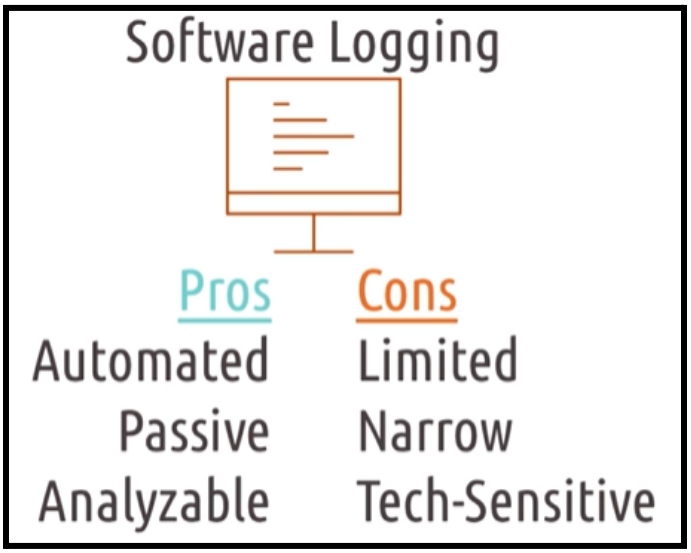
Designing a Qualitative Evaluation

Prior experience or live demonstration
Synchronous (watching the user live) vs Asynchronous
One interface vs. multiple prototypes
Think aloud protocol vs. post-event protocol (feedback at the end)
Individuals or Groups

Capturing Qualitative Evaluation







5 Tips: Qualitative Evaluation

1. Run pilot studies - friends family coworkers
2. Focus on feedback
3. Use questions when users get stuck
4. Instruct users what to do, but not how to do it
5. Capture Satisfaction

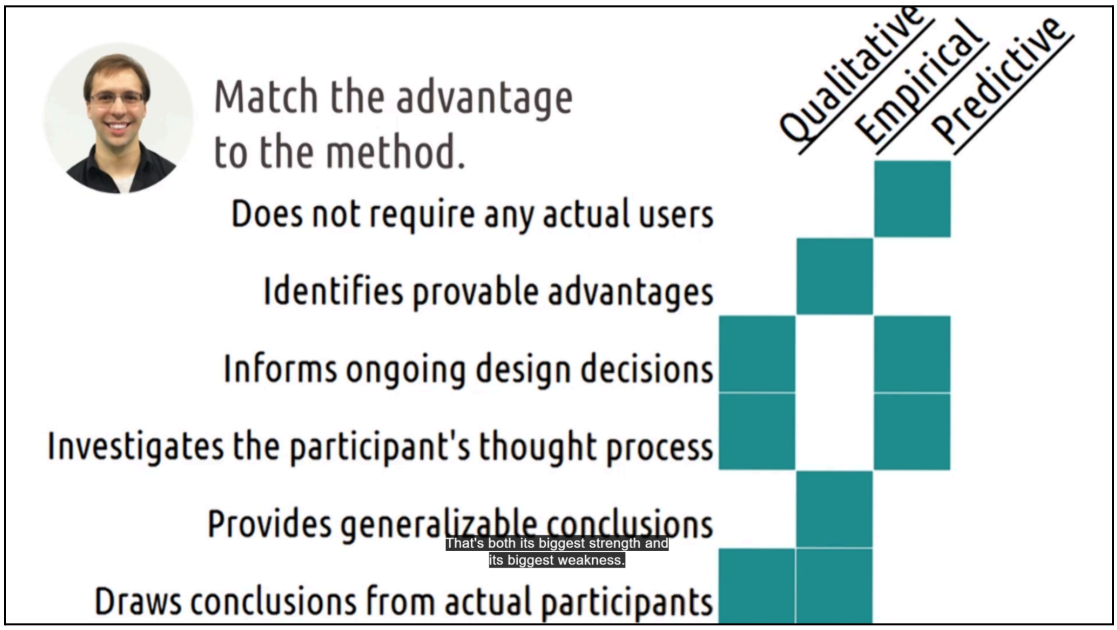
Empirical Evaluation

The goal is strong conclusions
How can we show that there is a difference between these designs

Designing Empirical Evaluation

Treatment: What a participant does in an experiment
Between Subjects Design: Comparison between two groups of subjects receiving different treatments
Within Subjects Design: Comparison within one group experiencing multiple treatments
Random Assignment: Using random chance to decide what treatment each participant receives.

Hypothesis Testing: Testing whether or not the data allows us to conclude a difference exists.



2.8 Distributed Cognition - Week 5

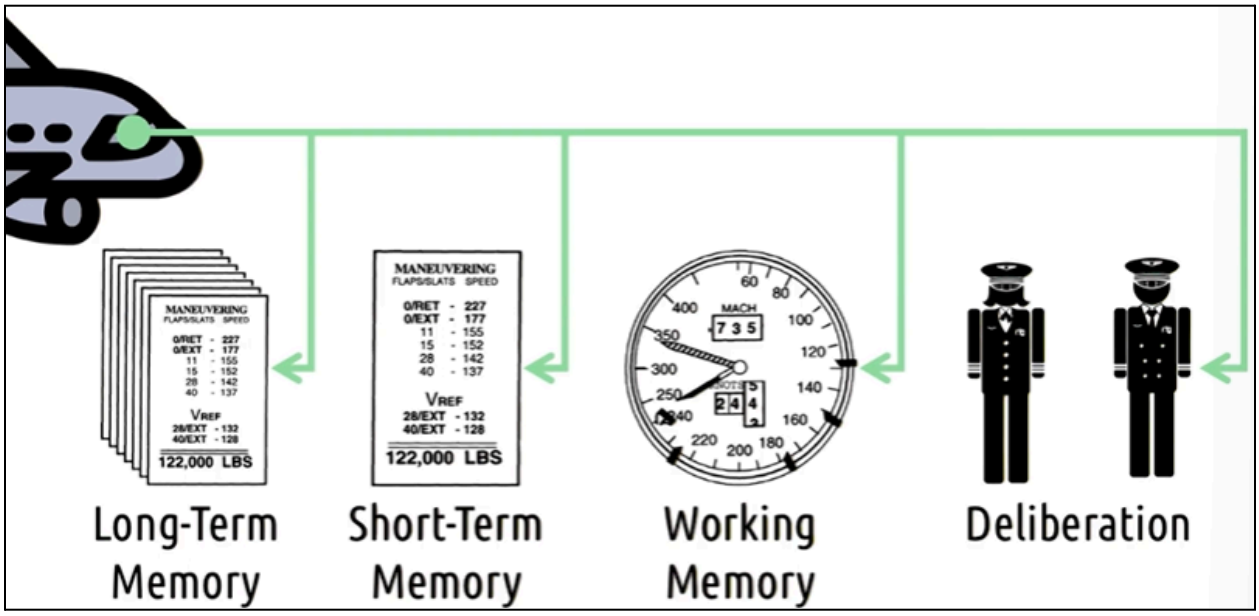
Four Models of Context

Distributed cognition - dominant theory on interplay between multiple agents, artifacts, and context
Social cognition
Situated action
Activity theory

Distributed cognition suggests models of cognition should be extended outside the mind.
Ex. doing mental math vs. using pen/paper (outside the mind tools)

How a Cockpit Remembers its speeds

The system as a whole remembers not just the human
The cognition involved in landing the plane is distributed across the components of the system.
Long term mem, short term mem, working mem, deliberation.



The different interfaces serve cognitive roles.

Distributed cognition is deeply related to the idea of cognitive load.
Cognitive load refers to your minds ability to deal with only a certain amount of information at a time.
Distributed cognition suggests that artifacts add additional cognitive resources.
Ex, using a GPS while driving distributes the cognitive load to another system
Every task offloaded to artifacts decreases the users cognitive load

Social cognition
Ex. you and your friend in the car, you are driving and your friend is giving directions so you are both capable of navigating to the new destination without a GPS.

We have to understand how social interactions work IRL in order to design for them

Situated Action

1. We must examine the interfaces we design within the context in which they're used.
2. We must understand the task the user performs grows out of interaction with the interface. We don't define it
3. The task doesn't exist until the user gets started, and once they start they define the task.

Activity Theory
Predates HCI - brought the idea of interacting through an interface
Generalizes our unit of analysis from the task to the activity. We are interested in why they are doing it.
Three main contributions:

1. Why the user is doing a task
2. Puts an emphasis on the idea that we can create low level operations from higher level actions.
3. Actions by the user can move up and down the Activity -> action -> operator hierarchy.
    a. Ex first time driving a car was a very conscious action, then after a few times it became an operator.

Activity theory and distributed cognition are driven by goals, situated action de-emphasizes goals for a focus on improvisation.

Situated actions - goals are constructed retroactively to interpret our past actions

The main difference between DC and AT is their evaluation of the symmetry between people and artifacts. AT regards them as fundamentally different, given that humans have consciousness. DC believes that artifacts can serve cognitive roles and should be conceptually equivalent to humans.

2.9 Interfaces and Politics - Week 5

Do technical devices have political qualities? Yes
Politics means whether artifacts/tech devices can personify specific forms or authority or power, whether for good or bad.

Since artifacts are interfaces that we design that change the world around us, just the way that politicians or business interests do.

Designing for change
Anticipating change from our designs

Most commonly in HCI we are interested in designing for usability. We want to make tasks easier through technology.

Third motivation is designing for change in the users behavior

Three goals of HCI:

1. Help a user do a task
2. Understand how a user does a task
3. Change the way a use does a task due to some value that we hold like safety or privacy

Two ways that artifacts can be political:

Inherently political technologies - technologies that due to their design are only compatible with certain political structures. Ex, nuclear power is connected to authoritarian power structures.

Technical arrangements as forms of order - technologies can be used to achieve changes to social order when used in the correct way. The tech itself has no political leanings but the use in a particular context for a particular purpose can accomplish some political goals. Used to achieve political motives. Replacing humans with machines in factory to disrupt union, even though machines created inferior goods.

Negative change by design
Instances where people create seemingly normal designs with underlying political motivations.

Making a bridge too low so buses can pass only rich people can go to the park

2.10 Conclusion to Principles - Week 5

3.7 HCI and Agile Development- Week 5
3.8 Conclusion to Methods - Week 5

Quiz 1

Covers:
2.1 Introduction to Principles
2.2 Feedback Cycles
3.1 Introduction to Methods
MacKenzie, I.S. (2013). Chapter 4: Scientific Foundations. Human-Computer Interaction: An Empirical Research Perspective. (pp. 121-152). Waltham, MA: Elsevier.
3.2 Ethics and Human Research

3.3 Needfinding and Requirements Gathering

2 hours
5 open-ended short answer questions
- 4 from lessons
- 1 from readings