Lecture-0

State of the Art in HCI

(1) Brain-computer Interfaces (BCIs):-

- Direct communication between brain and computer
- Example ⇒ Neuralink (Elon Musk) → control using thoughts
  ⇒ CTRL-Labs (Meta) → wristband, decode neural signals
- Impact ⇒ enable interaction with physical movement,
  ⇒ beneficial for users with disabilities.

(2) Multimodal Interaction:-

- Combine multiple input modes (speech, gesture, touch, gaze, haptics) for seamless interaction.
- Example ⇒ VR (Virtual Reality) → Eye Tracking + speech command
  ⇒ Meta Quest / Leap Motion → Hand tracking
- Impact ⇒ Interfaces are intuitive
  ⇒ More natural

(3) Affective Computing (Emotion-aware Interfaces):-

- Systems that detect and respond to user emotions
- Example ⇒ Affectiva → Facial expression recognition
  ⇒ Emotion-aware chatbots
- Impact ⇒ Interface that adapt to user mood,
  ⇒ Improving UX and accessibility.

④ Augmented Reality & Mixed Reality:

Example ⇒ Apple Vision Pro
⇒ Meta Quest 3
⇒ Microsoft HoloLens.

Features ⇒ Spatial computing: Interaction with virtual ↑ in physical space
⇒ Context-aware interfaces: Enhance real-world tasks.

Impact ⇒ Immersive learning
⇒ remote collaboration
⇒ initiative spacial UIs

⑤ Conversational & Generative AI:

Example ⇒ ChatGPT ⇒ Amazon Alexa
⇒ Google Gemini ⇒ Character.AI

Features ⇒ Natural, context-aware conversations
⇒ Voice + gesture interfaces by LLMS

Impact ⇒ Human-like assistants
⇒ Language based UIs
⇒ Personalized Interactions

⑥ Wearable and Invisible Interfaces:

⇒ Smart textiles (Google Jacquard)
⇒ Skin-based interfaces (iSkin)
⇒ Smart rings, glasses, earbuds

Impact ⇒ Continuous interaction integrated into daily life

⑦ Zero UI & Ambient Computing:

- Interfaces without screens - interaction through environment and context.
- Example ⇒ Smart Homes - react to presense, voice, habits
  ⇒ IoT devices working together without explicit commands.
- Impact ⇒ Frictionless, intuative interaction mixed into everyday environments

⑧ TUIs & Shape Shifting Devices:

- Physical objects that user manipulate as digital input
- Example ⇒ MIT's inform - shape displays that changes form
  ⇒ Tern - Tangible blocks for programming
- Impact ⇒ Merges physical and digital for intuitive learning and design

⑨ Inclusive & Adaptive Accessibility Technologies:

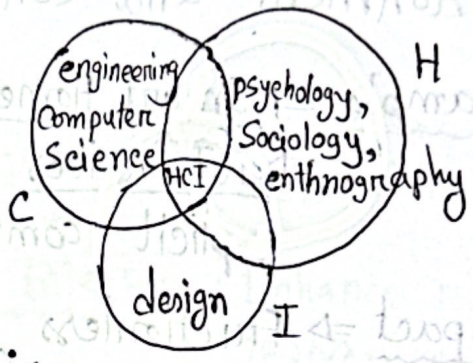
Features: ⇒ Real Time Sign Language recognition
⇒ Voice - controlled navigation for blind users
⇒ Adaptive keyboads and UIs for neurodivergent users

Impact ⇒ Accessability - first design
⇒ Enabling border participation

Lecture - 1

Introduction to HCI

- Human Computer Interaction = HCI
- HCI is about the design and use of computer technology



- **Design** – making things in such a way as to best accomplish a particular purpose
- **Interface** – Human and computer interacts using it. and that interface is what we design.

Why Interaction design is hard?

⟹ Because designers and engineers are not the user. They already know how system works inside out, which users don't know – only see interface and try to achieve goal. That's why interface design is hard to predict where users might struggle and what might confuse them.

What do you mean, when we say "User is always right?"

⟹ If users are confused, frustrated, and make mistakes, then it's design's fault, not the user's. User's reaction shows what's wrong in the design.

Don't expect users to be Designers either :-

Example — i) Telephone Handset Weight.
ii) 30 google search Results
iii) Command abbreviation

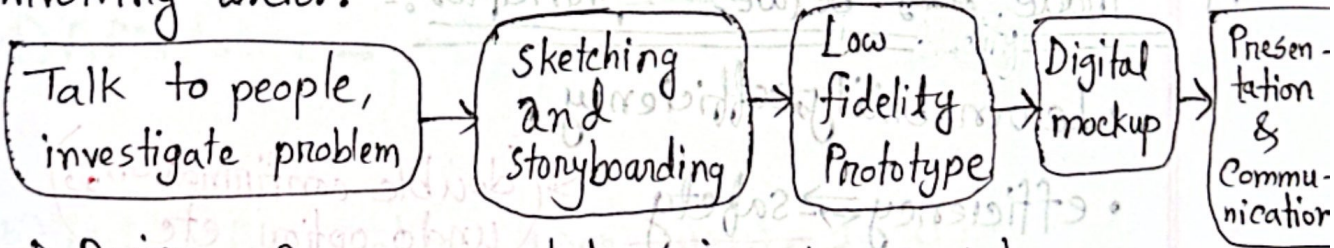
So, Users can identify problems, not effective solutions.

What to design?

Ans1 ⟹ Design as a process —

(1) Combine needs, technology, context from all relevant constraint
2) Frame or reframe the problem and objective
3) Create and envision alternatives (sketching, brainstorming)
4) Select from those alternatives (choose effective one)
5) Visualize and prototype the intended solution (digital mockup, working model, build and test prototype)

Design as a process as iterative, exporative and constantly involving users.



Ans2 ⟹ Design as open ended series of principles :-

(1) Usability → how well users use functionality
(2) Learnability → How easy to learn?
(3) Efficiency → how quickly can be used?
(4) Safety → are errors recoverable?
(5) Accessibility
(6) Aesthetics, minimalism
(7) Ergonomics
(8) Flexibility, Expressivity
(9) Control, Malleability

So, which principle to use/emphasize?

- It depends on the user, task, context.
- There are Trade-offs between principles So, we can't apply them mindlessly.
- Emphasize depends on user :-

→ Novice user → needs learnability
→ Experd user → " efficiency

All users can be novice or an expert.

- Emphasize depends on task :-

→ Critical Task → emphasize safety
(emergency system, Amber alert, medical devices)

→ Less critical Task / repitive Task → need efficiency
(unlocking phone, typing message)

Trade offs between principles :-

- Learnability ⇄ efficiency
- efficiency ⇄ safety → (double confirmation, undo option etc.)
- Safety ⇄ Flexibility
- Aesthetics, minimalism ⇄ Usability, Learnability
- Ergonomics ⇄ Aesthetics, flexibility

Design for physical comfort (large buttons, spacing etc)

More trade offs :-

→ In HCI, we will ignore these trade-offs because, these are not interface related

- Functionality ⟷ Performance, cost, maintainability
- Maintainability ⟷ Performance, cost
- Security ⟷ Performance, usability, cost
  - authentication, encryption

- High functionality reduce Usability by causing confusio

Five key Ideas of HCI :

(1) Visibility –

(2) Feedback –

(3) Goal → A state, user wants to reach (to save a file, to call someone)

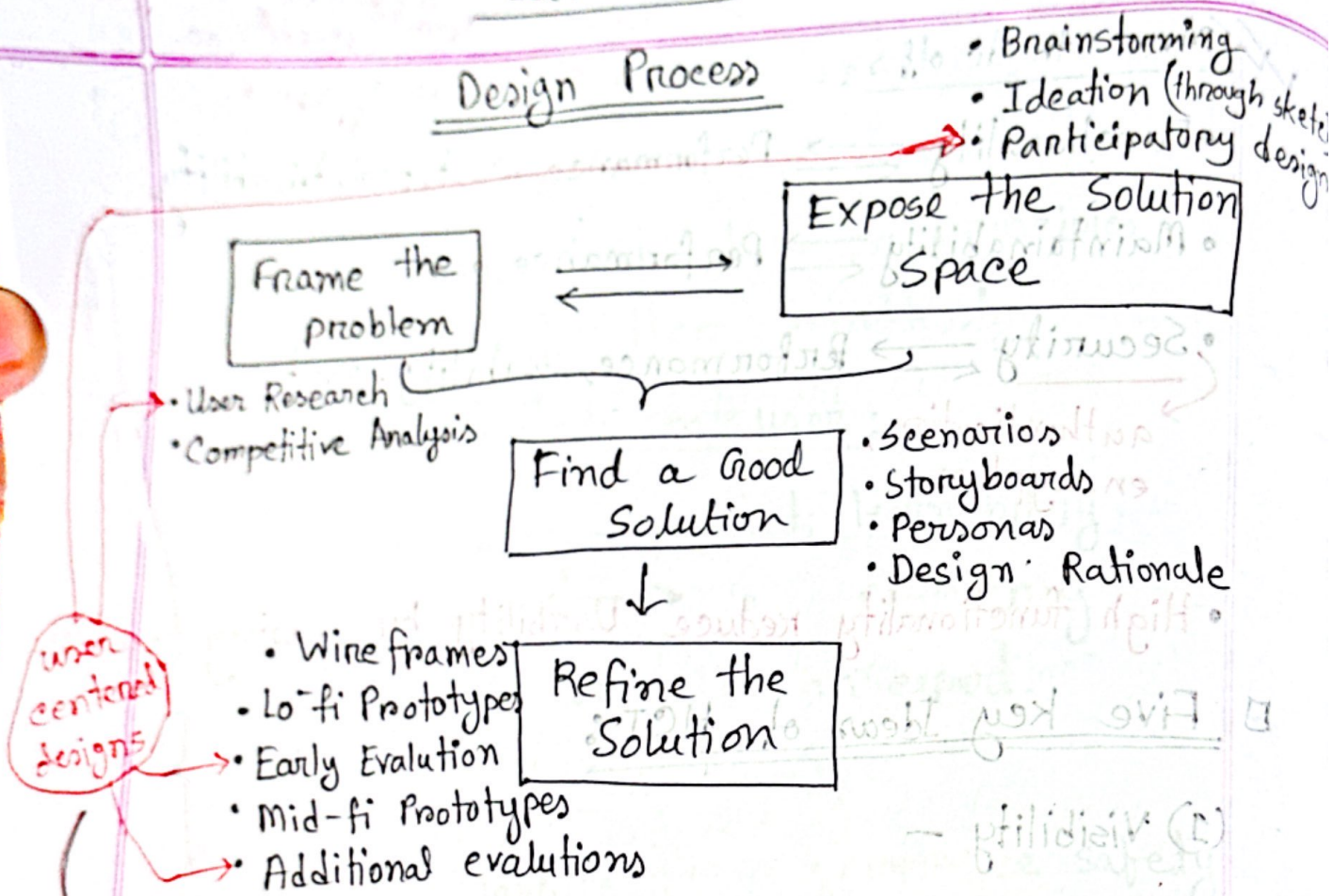
(4) Affordance →

(5) Task → An action user wants to do

"Goals beget task, task beget goals)

Lecture 2

Design Process

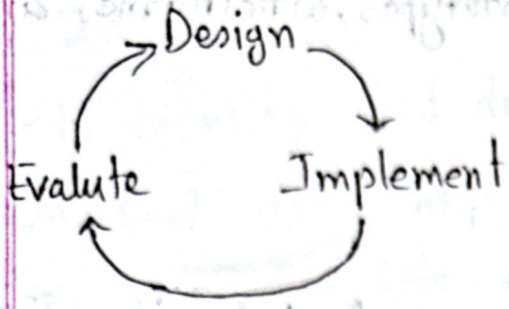


Why do we need user centered design?

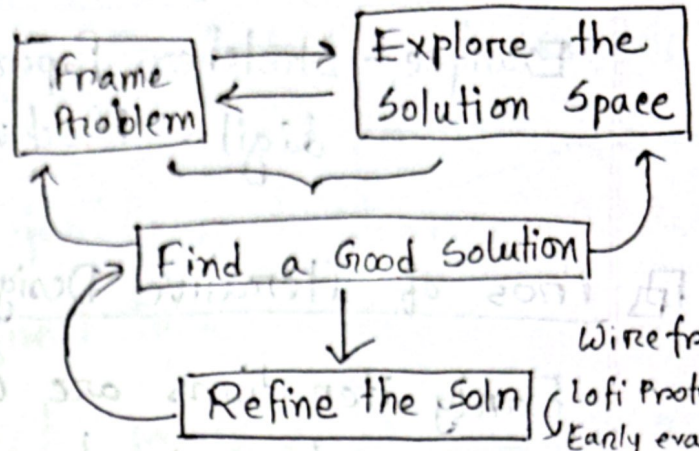
→ It ensures products that solve real problems.
→ It prevents resource waste on irrelevant f
→ improves usability, satisfaction, adoptibility, affordance
→ aligns innovation with actual human experience

example → Silicon Valley's 'Juicero' failed because doesn't solve real problem for users
→ University websites make users struggle to find what they want (campus map, faculty contact)

How do we add users feedback to the process?



Itenative Design

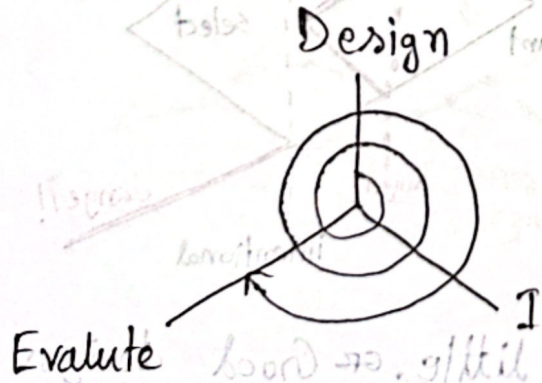


Itenative Design

Why is waterfall method bad for designing UIs?

UI design is risky. In waterfall method, users are not involved in validation until acceptance test at the end. Most of the design and development is already done — so, fixing UI issues become costly and time-confusing. It causes difficulty to redesign

We won't get it right in the first time, so we can also use Spiral Model.



Spiral Model

It increase in fidelity with each itenation

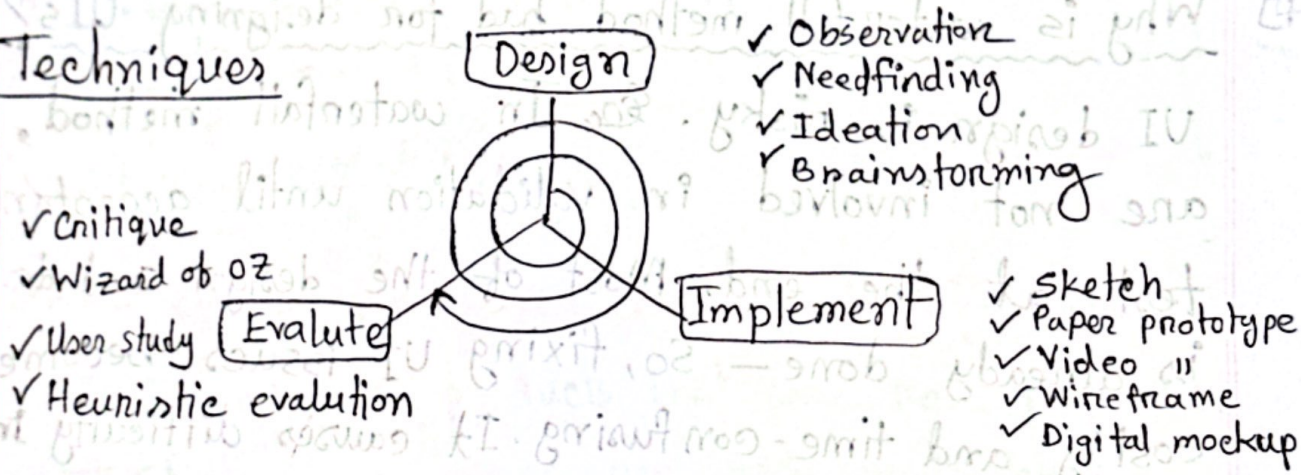
Early Prototyping:

Example - Sketches, Paper prototypes, wireframe, and digital Mockup.

Pros of Iterative Design:

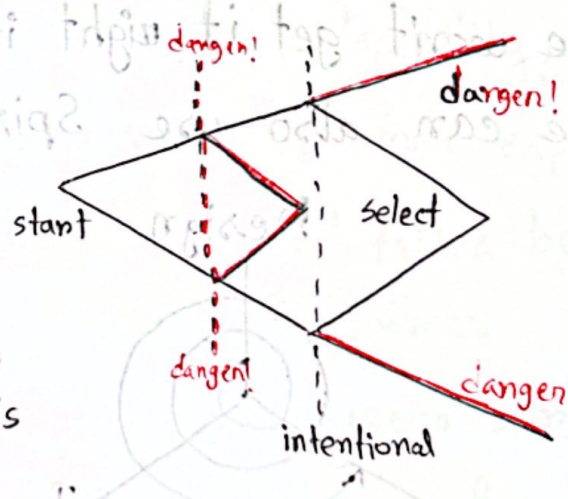
Early iterations are cheap and later iterations use richer implements after UI risk has been mitigated. More iterations ⟶ Better UI.

Techniques



Design Diamond:

Jumping to solutions too quickly or too little divergence doesn't solve real needs. Too much divergence causes delays, confusion. Main danger is losing balance between exploring too broadly or too little. Or Good design comes from maintaining balance.



Role of Critique:

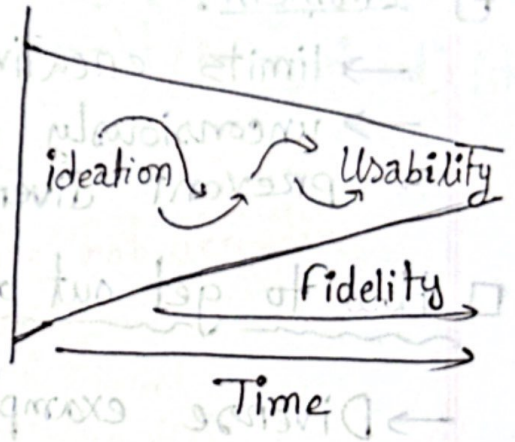
Ideas can be both good and bad. Critiques help to identify bad design, so we can avoid implementing it. Bad ideas help to justify good ideas and turn ~~make~~ them into a great idea

Critiques refine concepts, justify decisions, ensure creative, user centered final design.

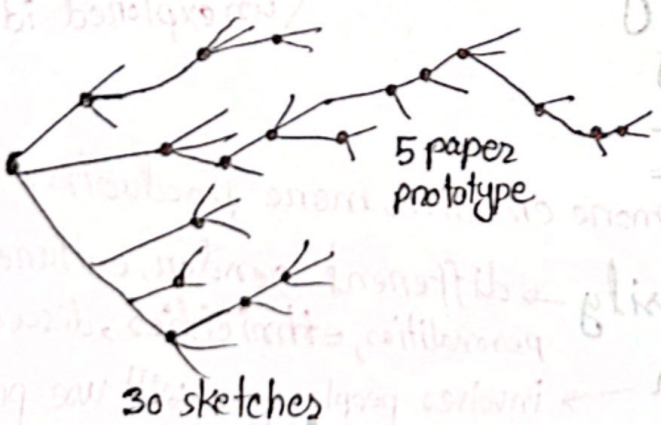
Combine Design Diamond, Iterative Design & Spiral Model –

1) start with low fidelity
2) move to higher

we can design diamond process and it gets smaller over time.



- Use parallel design early to explore broadly.
- Use serial design later to refine deeply.



Parallel design:-

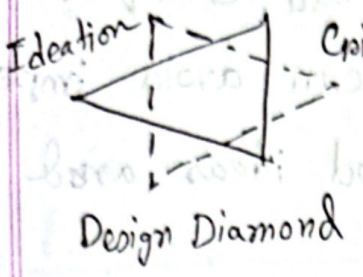
→ easy to do sketchin on lo-fi design
→ build and test multiple

Serial design:-

→ One design at a time
→ Later stage prototype

Lecture 3

Ideation and Critique



Ideation = generating diverse, creative ideas

- **Design Fixation** ⇒ Happens when designers get stuck on a single idea / familiar pattern.
- It comes from one's experience in life and habits.

Problem:
- → limits creativity and exploration.
- → unconsiously repeat old solutions
- → prevent diverse or innovative design

How to get out of fixation?
- → Diverse example
- → Quantity over Quality → (generate as many ideas as possible)
- → Mapping the Design space → (all possible direction, visualize + testing, unexplored ideas)
- → Parallel Prototyping
- → Attribute Listing

Collaborative Ideation:-
- → Diverse teams → more creative, more productive ideas
- → Designing for diversity → different gender, culture, ages, personalities, ethnicities, disabilities
- → Participatory Design → involves people who will use product

Critique :-

1) Presenter explain artifacts
2) Critics ask question about artifact and give feedback
3) Presenter give answer/clarify, note down feedbacks

Presenters Tip :-

- Take notes of feedback
- Don't take feedback personally
- You are the "Final decision Maker", you are not losing authorship

Critics Tip :

- "I wish", "I like", "What if" method
- Socratic method: ask why?
- Limit personal pronouns ("You")
- Include clarity and rationale and honest (both good, bad) feedback

- In brainstorming → we are not criticizing
- In critique → we are not defending

- get feedback early, often and cheaply
- Focus on improvement

Lecture 4

Design Principle - I

How do people learn a new UI?

✓ Learn by doing

- explores interface to achieve goal. When stuck, seek help or read instructions.

✓ Learn by watching

- UI should itself communicate how it works, how to use it. But not many explanatory text

How can we design more learnable UIs?

Ans 1 ✓ Design in Affordance (pull or push?)

- perceived affordance + actual affordance
  - perceived affordance: (what user think they can do with an obj)
  - actual affordance: (what the obj actually do)
- Technology affordances —come→ from physical world (scrolling bar, trash icon for deleting files)
- Some affordance have all on its own (without physical metaphor)

Button & link Search http://google.com

Drop down arrows w.google.com/ig

Mouse cursor ↖ ☝ ↕ I

✓ Feedback :

- actions should have immediate visible effect
  - low-level feedback (button press, cursor chang, highlight o hover
  - high-level " (new page loads)

Ans 2 Using Recognition Over Recall :

- Prioritizing making information easy to recognize rather than require users to recall from scratch memory

✓ Recognition is much easier.

Ans 3 Consistency :

- Similar things should look and act similar
- Different things should look different

(1) Internal consistency → within application (same terminology and layout)

(2) External consistency → with other applicat
  - ✓ common widget appearance
  - ✓ design patterns common accross appli

Consistency → reduce user confusion, errors
→ increase efficiency

Ans 4 **Metaphors:** a way to bring outside world in interface so that user has less to learn

Example — Desktop metaphor

Files and Folders ⟶ Store documents

Trash can ⟶ delete

Windows ⟶

Not perfect simulation of real world but it helps to recognize, increase learnability

Advantages:

- Highly learnable
- User's existing mental model

Dangers:

- can limit design
- Break down over time
- not always useful
- can die
  - ↳ Example: Floppy disk icon

✓ Use metaphors if you have good one

Ans 5 **Mapping:** Physical arrangement of controls should match arrangement of function.

example — Stove knobs (চুলার সুইচ), Can control

✓ Reduce errors

✓ Increase learnability

✓ Support natural interaction

Ans 6 **Visibility and Exposing State:**

- **Visible selection State:** When user selects an obj, highlight the selection, don't leave invisible
- **Visible Navigation State** (Breadcrumps, pagination, Tabs)
- **Self-Disclosure** (~~visi~~ commands and syntax visible and learnable, specially command based interfaces)

Example – auto suggestions,

- **Visible Modes** - Modes are state changer (caps lock, Shift key, drawing tool mode)

Use spring loaded modes or otherwise have a lot of visual cues (symbol) to clue users on its currently active mode

□ **How does learning break down?**

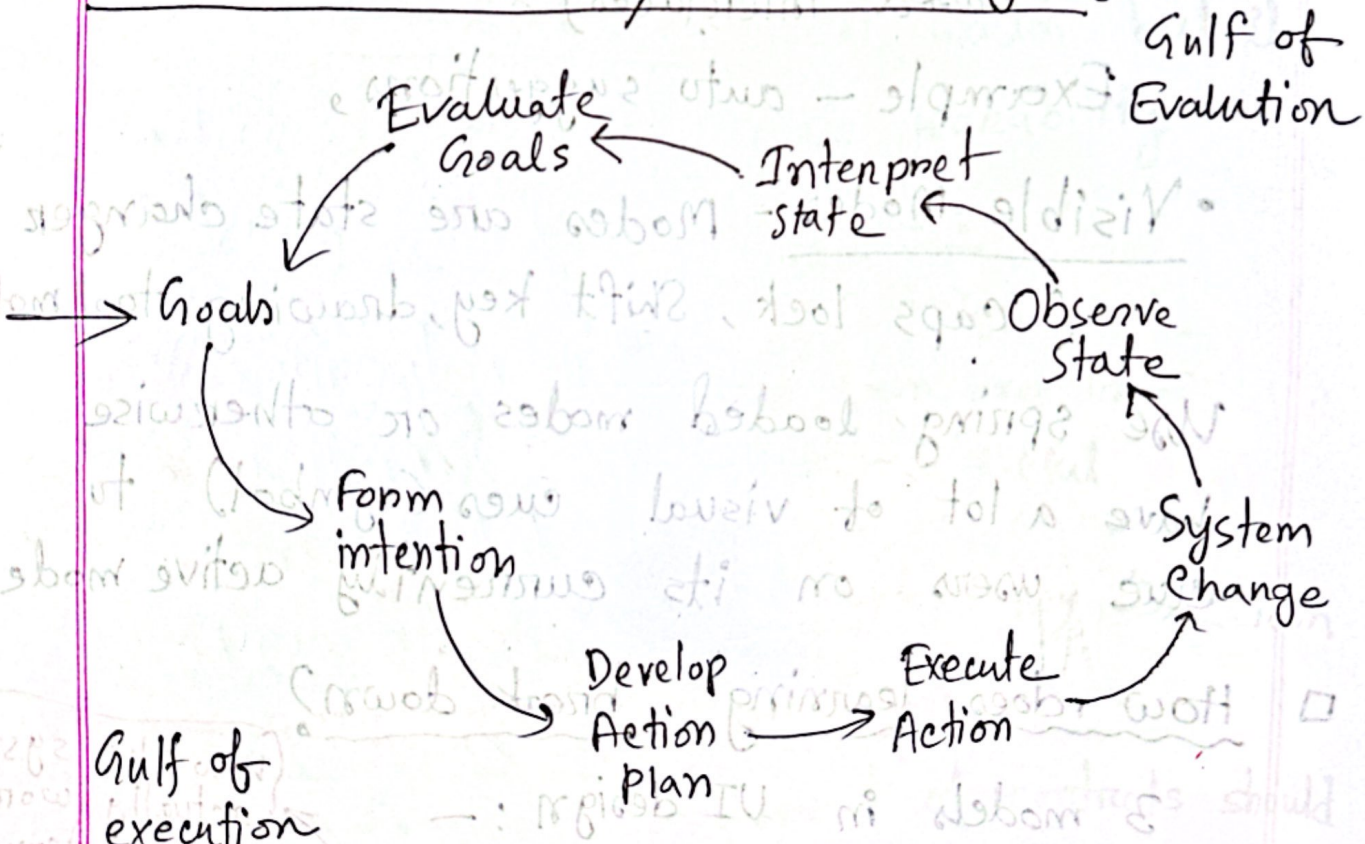
3 models in UI design :–

1) System model (implementation model) ← how the system actually works internally
2) Interface model (manifest model) ← How the system is presented to the user via interface
3) User mental model (conceptual mode) ← How user thinks the system works

System ↔ Interface
User ↔ System
} So there is always a mismatch.

Interface should help bridge the gap between user's mental model and system model.

Norman's Execution/Evalution Cycle:



Lecture - 5

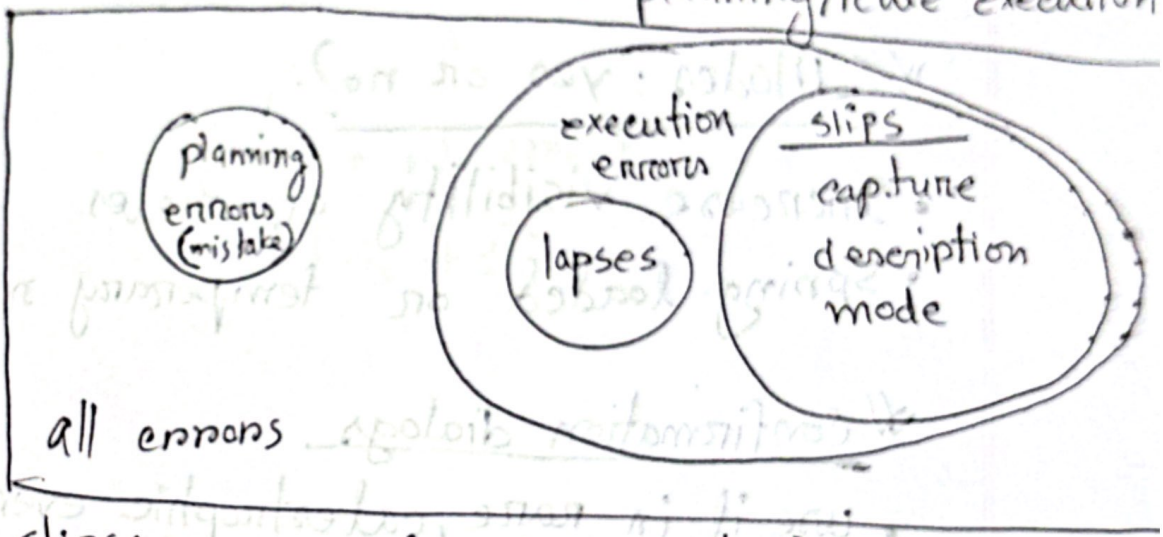
Safety → prevent errors of users

Kind of errors

Slip – Failure due to execution/control (Ctrl-V, Ctrl-C)

Laps – " " " memory (forgot to add attachment in email)

Mistakes – Error made in planning/rule execution



Types of slips:

- **Capture** – Start executing one action, but does another (starting similar) ←
  ":wq" → ":w" টাইপ করে "Enter" দিয়ে ফেলা
- **Description** – Two action very similar; User wants to do one, but accidentantly does another (Send | Send + ...)
- **Mode** – States which has same action, different meaning (doesn't notice caps lock/mode)

Preventing errors

✓ Remember consistency

- Diff things should look diff

So, keep dangerous command away from common ~~ones~~ ones.

✓ Modes : yes or no?

- Increase visibility of modes
- Spring-loaded or temporary modes

✓ Confirmation dialogs

- use it in rare, catestrophic events
- make it look different than anything else
- Draw attention to it.

example- no 'OK' button forces you to think

Efficiency:

✓ Chunking

✓ Memory - Grouping into chunck improve efficiency

✓ Fitts law

✓ Make frequentlly used targets big
✓ Puts targets used fogether near each othe
✓ Avoid steering task
✓ Use screen corners and screen edges.

fitts law

$$T = \text{Reaction time} + \text{Movement Time}$$
$$= a + b \log (D/S + 1)$$