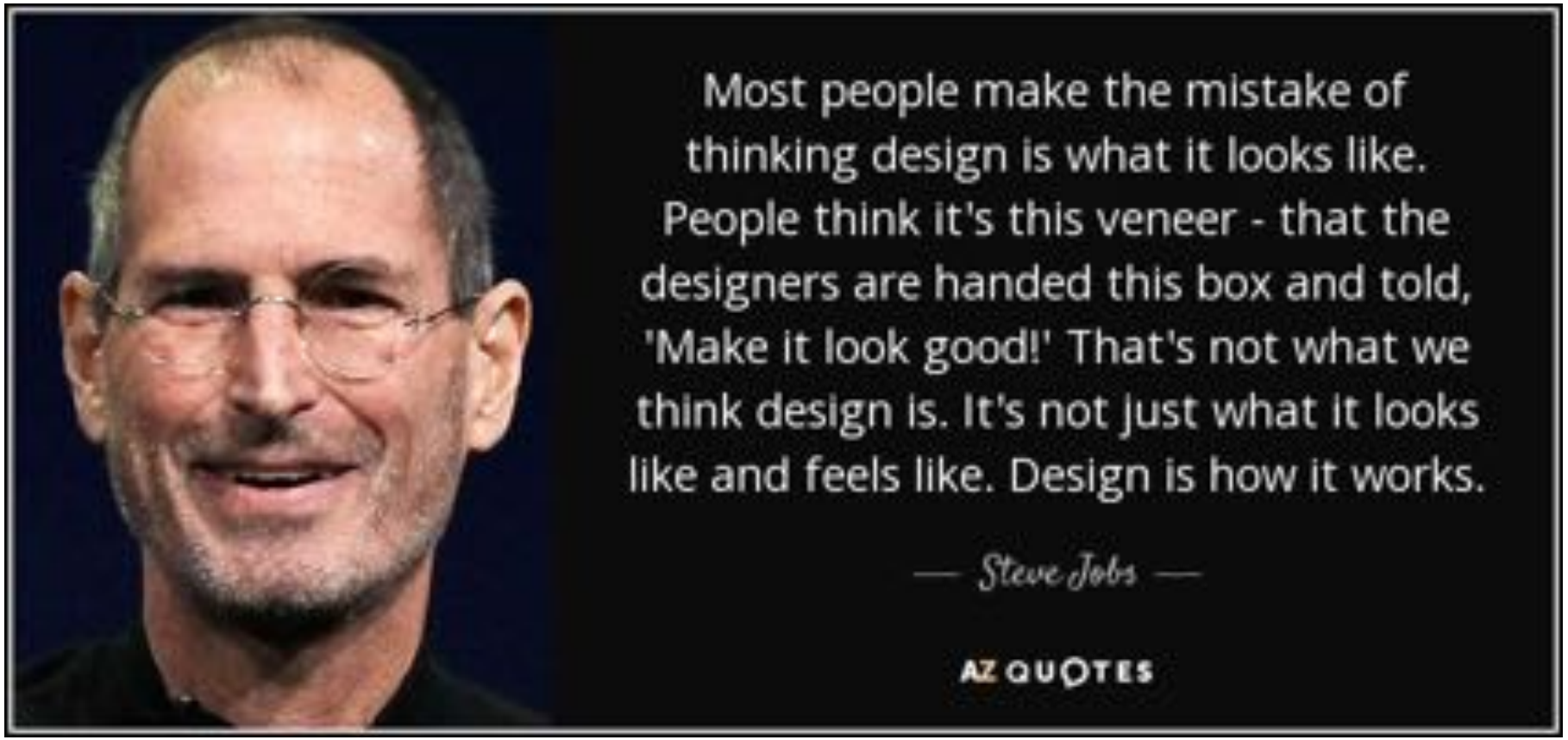
17-423/723: Designing Large-scale Software Systems

Design for Usability

April 8, 2024

Learning Goals

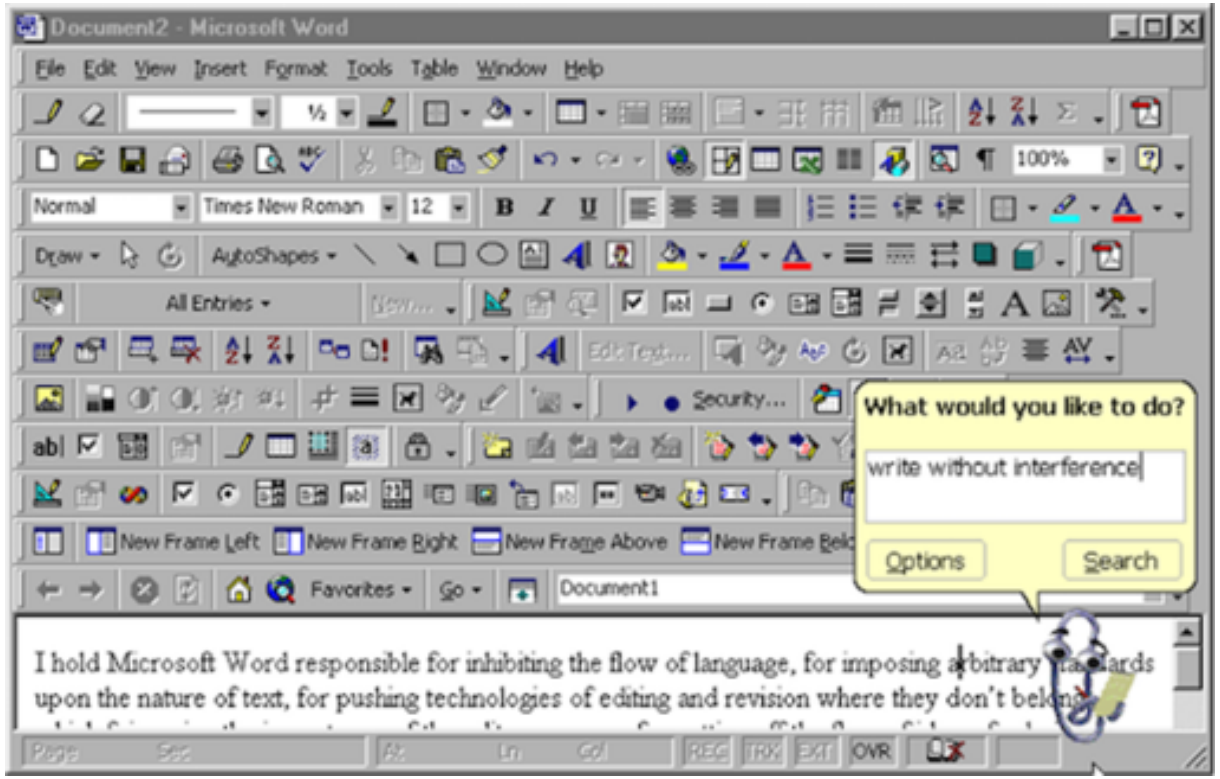
- Describe the basic concepts in usability and the goal of usable design
- Identify a user's mental model for the system being designed
- Identify potential mismatches between the mental model and the system
- Apply strategies to help adjust the user's mental model to the system


Most people make the mistake of thinking design is what it looks like. People think it's this veneer - that the designers are handed this box and told, 'Make it look good!' That's not what we think design is. It's not just what it looks like and feels like. Design is how it works.
— Steve Jobs —
AZ QUOTES

Usability Concepts

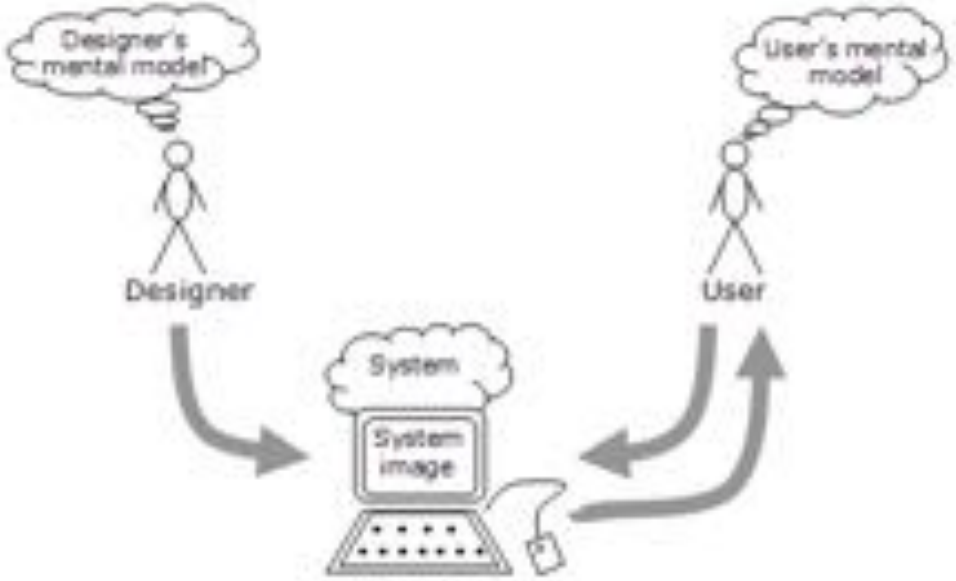
- **Learnability**: How easy is it for users to perform a task the first time?
- **Efficiency**: After learning, how efficiently can users perform the task?
- **Memorability**: Can users remember to perform the task after a period of not using the system?
- **Errors**: How often do users make errors, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction**: How pleasant is it to use the design?

Interaction Cost

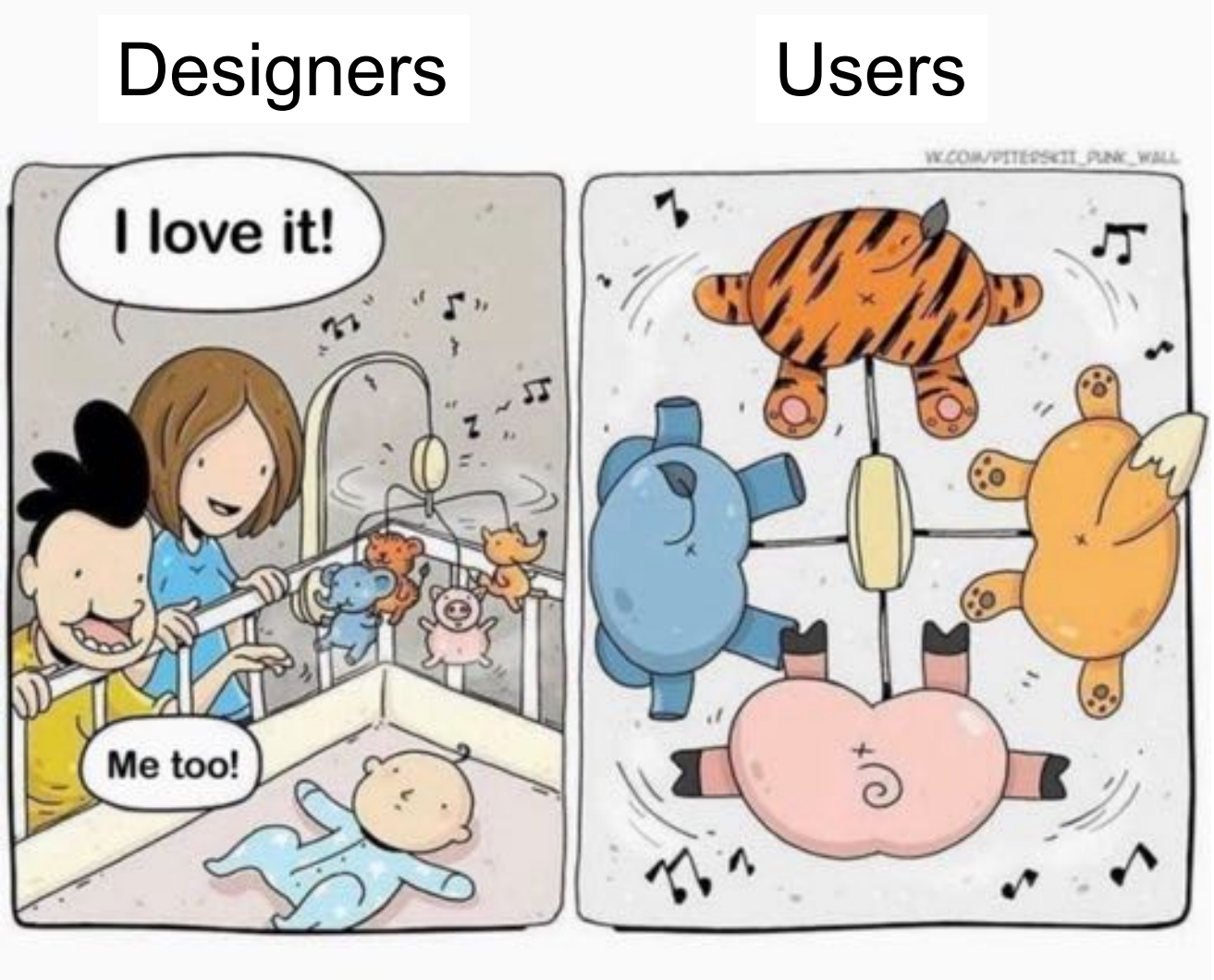


- Amount of mental & physical effort to perform a desired task
  - Reading, scrolling, clicking, typing, switching contexts, memorizing
- **Goal of usable design**: Minimize interaction cost while allowing users to achieve their goals

Mental Model



- A person's expectation and belief of how a system works
  - What is the sequence of actions that I need to do to complete my task?
  - What is the current state of the system?
  - What actions can I perform from this state? What will happen next if I perform Action X?



What the designer sees ≠ what the user sees!

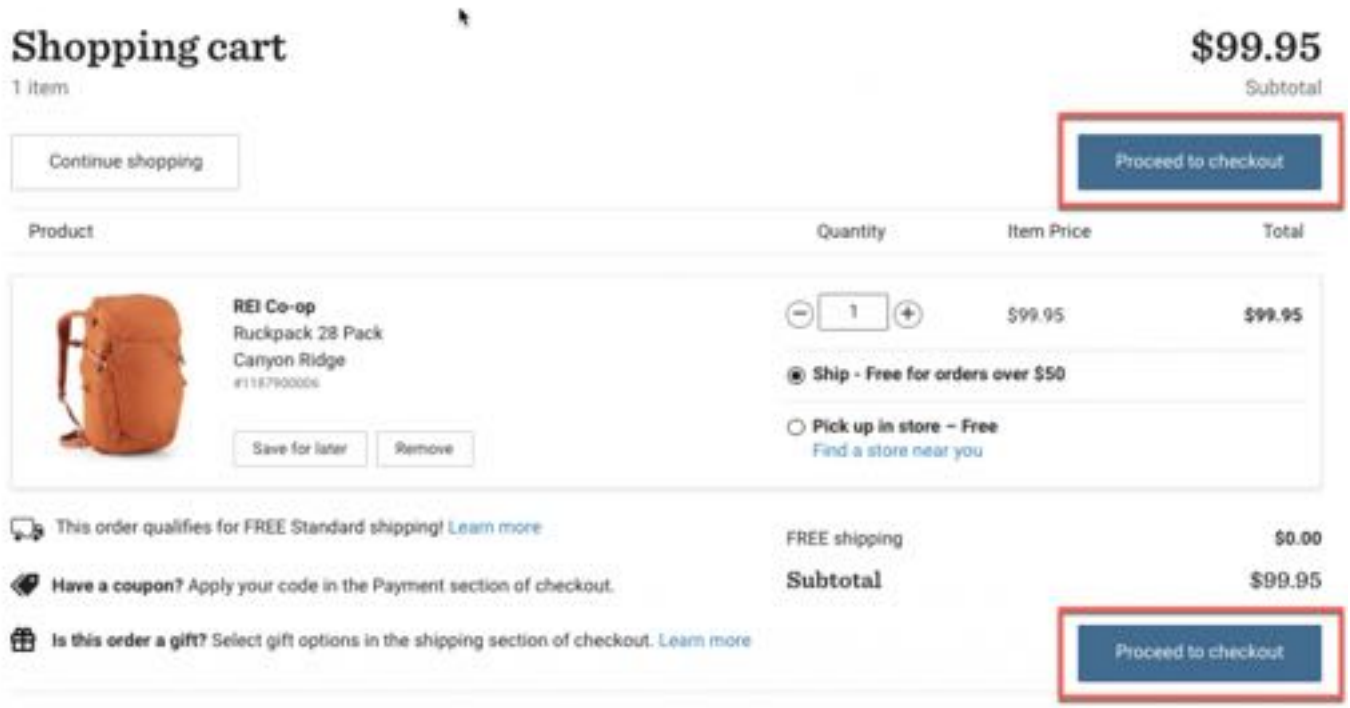
Mental Model Mismatch

- Divergence between a user's mental model & actual system behavior
- A mental model mismatch can manifest as the user:
  - Showing confusion about the current status of the system
  - Being unsure about what actions are available and/or trying out random actions
  - Performing an incorrect/erroneous action
  - Restarting or simply quitting the system

Mental Model Mismatch

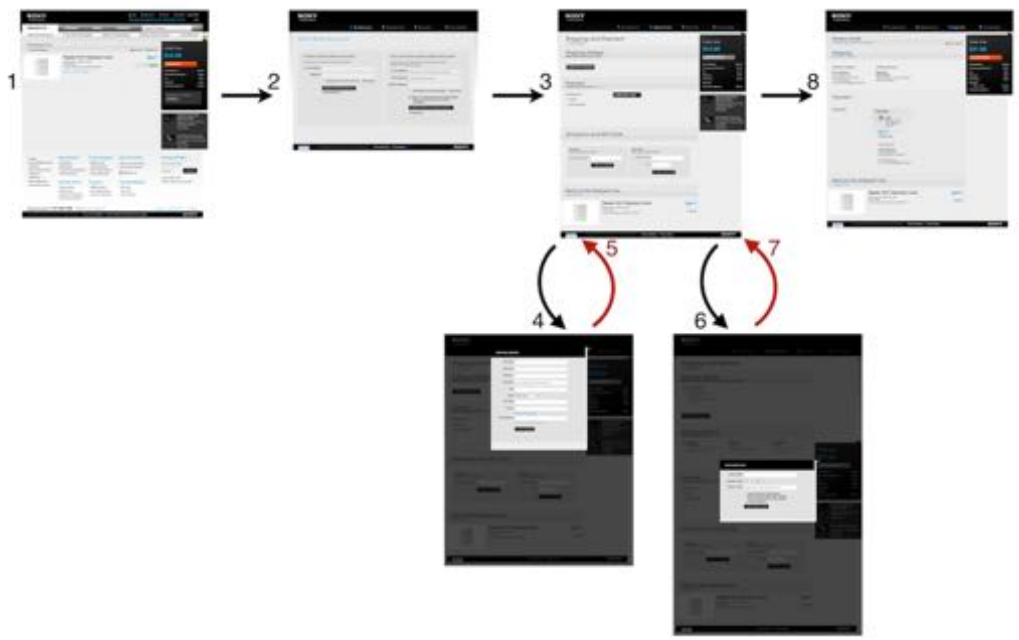
- Divergence between a user's mental model & actual system behavior
- A mental model mismatch can:
  - Cause confusion and frustration in users
  - Increase interaction costs
  - Increase chance of user errors
- Usually these lead to negative consequences for the system
  - Loss of users and revenues, complaints, low product ratings, accidents…

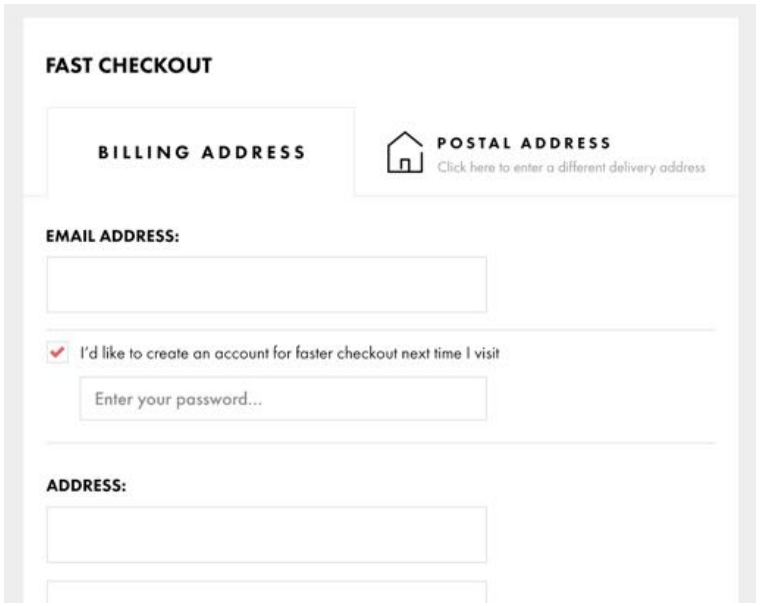
Example: Shopping Cart Checkout



- Standard mental model for online shopping:
  - Browse for items -> Add items to cart -> Choose checkout -> Enter shipping & billing data -> Press submit -> Get confirmation

Example: Shopping Cart Checkout

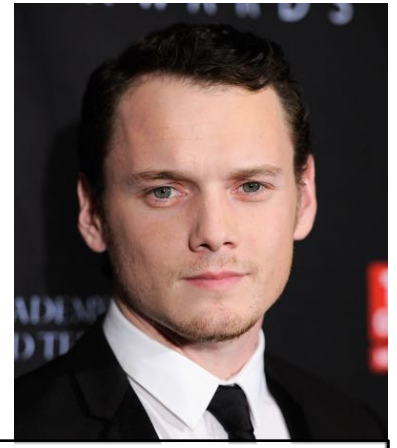




- Common anti-pattern: Non-linear interaction process
  - Interrupt the flow: Create an account, open a new dialog to enter a preferred address, suggest other items to buy…
  - Deviates from the user's mental model
  - ~60% of customers abandon their shopping cart; failure to convert into sales!

Source: Why Your Checkout Process Should Be Completely Linear

Example: Gear Shifter

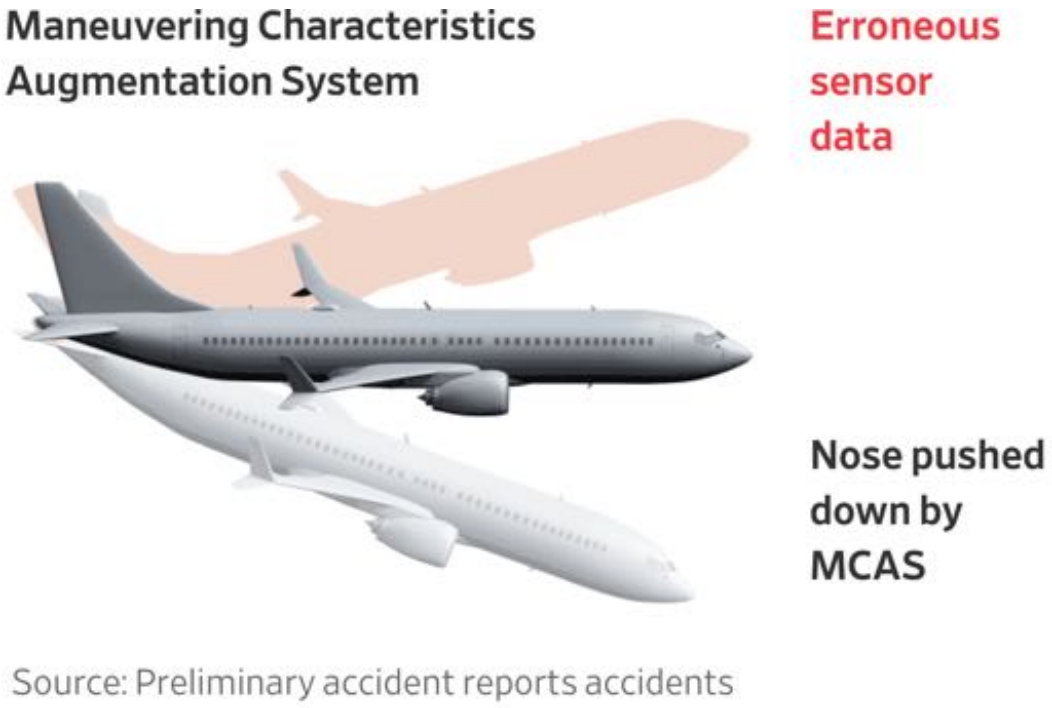


Recalled Shifter May Have Played a Part in Actor Anton Yelchin's Death

His Jeep Grand Cherokee has the type of shifter many find confusing

- Fiat Chrysler vehicles (mid 2010s)
- A new gear shifter design; radical departure from standard design
- Drivers frequently became confused between modes (e.g., Park vs. Reverse)
- 266 crashes, 68 injuries, 1.1 million vehicles recalled

Example: Boeing 737 MAX



Source: Preliminary accident reports accidents

- MCAS: Keep the plane nose down if the detected angle is too high
- A faulty sensor indicates high nose angle; MCAS is activated
- Pilot sees nose being pushed down and attempts to correct, but is unaware that MCAS is overriding the control

Boeing 737 MAX

- Boeing skipped out on proper pilot training (to save costs)
- Pilots often confused & not equipped to respond to MCAS failures
- Two major accidents involving a MCAS failure:
  - Lion Air Flight 610: 189 deaths (2018)
  - Ethiopian Airlines Flight 302: 157 deaths (2019)

Discussion: Mental Model Mismatch

- Working in groups, discuss an application or a product that you've used in the past where you experienced a mental model mismatch
- How did the mismatch manifest itself?
- What negative consequences did the mismatch have?

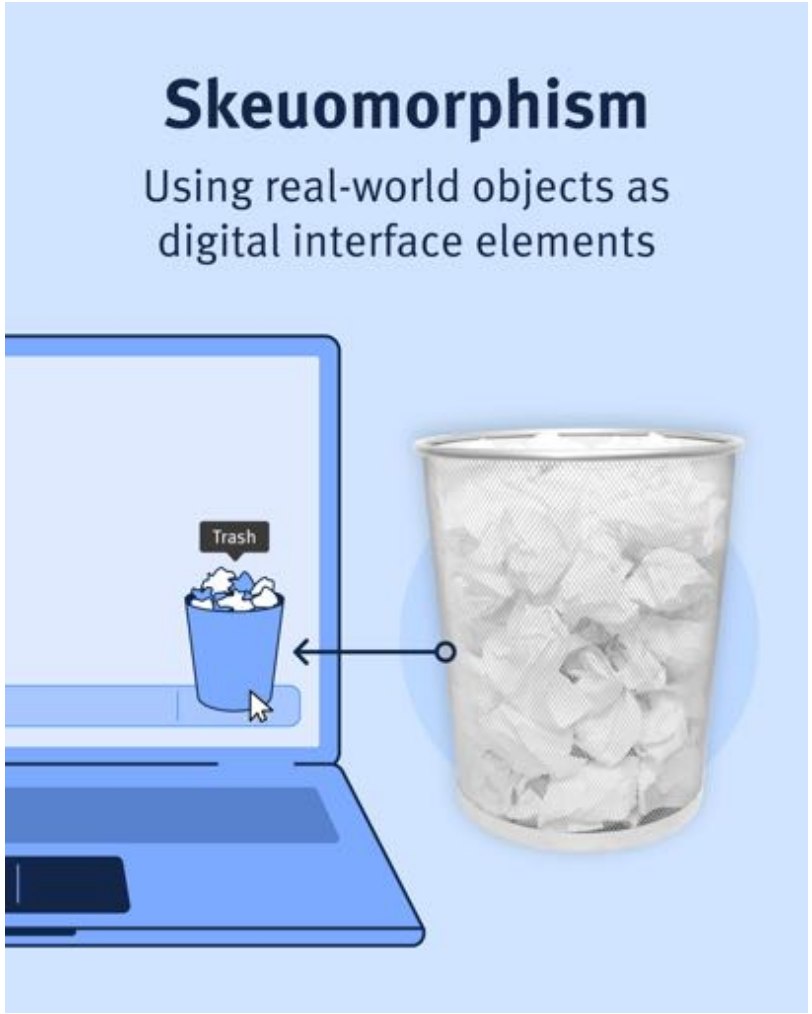
Principle: Mental Model Alignment

- **Principle**: The user’s mental model must be consistent with the observable behavior of the system
- **Strategies** for achieving alignment:
  - Identify the user's existing mental model
  - Design the system to conform to the user's mental model
  - Adjust the user's mental model to conform to the system

Identifying User's Mental Model

- Find similar systems & identify a common mental model across them
  - Mental-model inertia: Users tend to stick to an existing model and are reluctant to change
    - Users rarely read documentation or manuals
    - Users are unwilling to learn unless there are clear benefits
  - Be conservative; don't innovate in user interfaces unless necessary
  - Leverage the user's existing knowledge from other contexts (e.g., skeuomorphism)

Skeuomorphism



Source: Nielsen Norman Group

- Incorporate real-world elements that are familiar to the user
- **Goal**: Leverage familiarity to reduce learning time and interaction costs
- **Example**: Trash cans in OS desktops
  - But be aware of subtle differences vs. the real world!
  - “Deleting” things by moving into trash doesn’t free up disk space; must empty the trash first
- **Q. Other examples?**

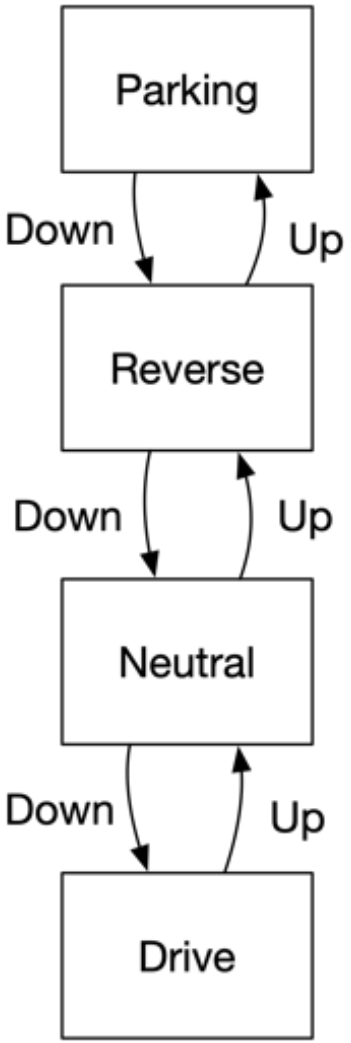
Identifying User's Mental Model

- Find similar systems & identify a common mental model across them
- Perform user testing to identify a pre-existing mental model
  - Build a mock-up or prototype
  - Ask potential users to perform common tasks & observe their interactions
  - Record user errors or unexpected behaviors
  - Perform an interview to identify user confusion

Adjust the System to the Mental Model

- During design: Document and systematically compare the user's mental model against the actual system to identify potential mismatches
- After deployment: Collect & analyze user complaints and errors to identify unforeseen mismatch

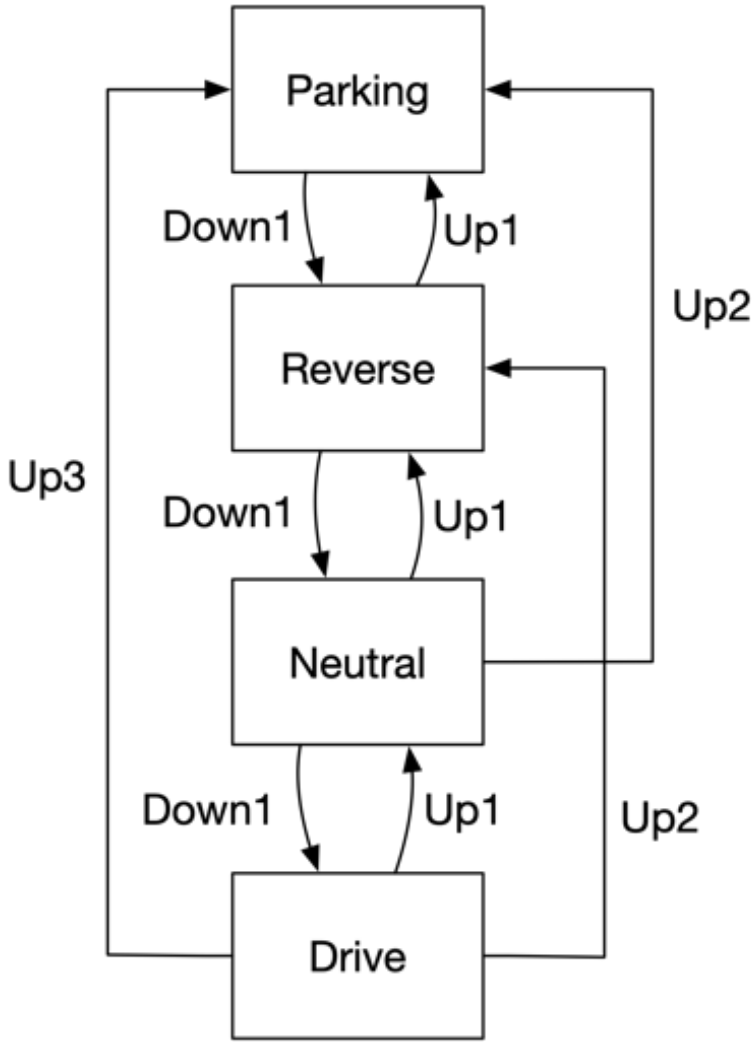
Documenting a Mental Model



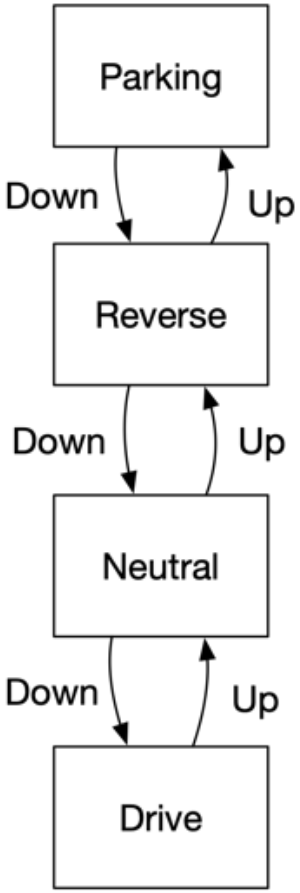
- State machine as the user's expectation of how system behaves
- Node: A possible state of the system
- Edge: An action that moves the system from one state to another
- **Example**: Stick gear shifter
  - Move up & down to change the gear one at a time
  - Common design: For many drivers, the mental model of how gear shifting works

Comparing Mental Model vs. System

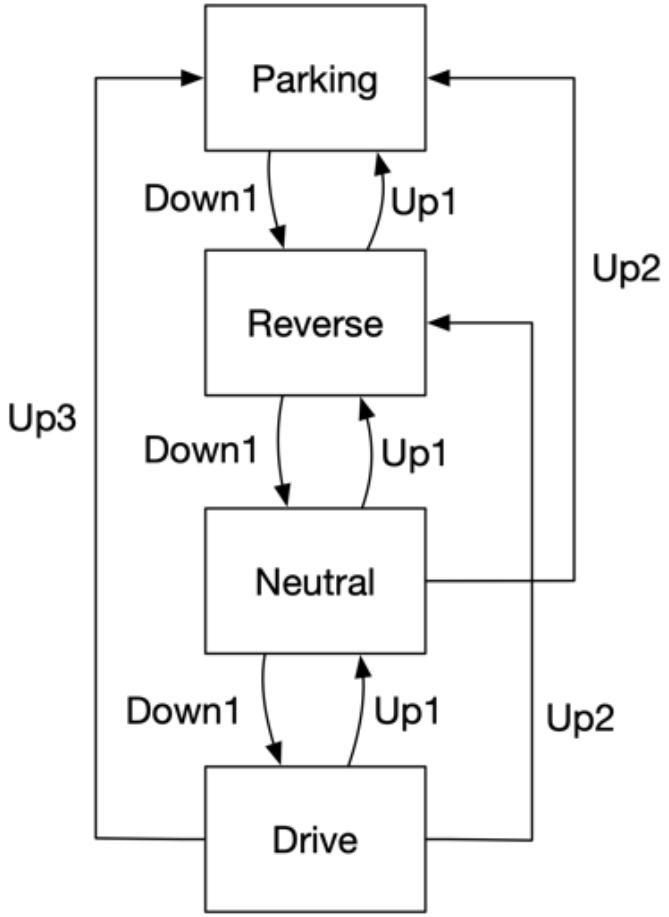
- “Monostable” gear shifter in Chrysler vehicles
- Change gear by moving up/down quickly 1-3 times
- State machine as a model of the actual system behavior



Comparing Mental Model vs. System



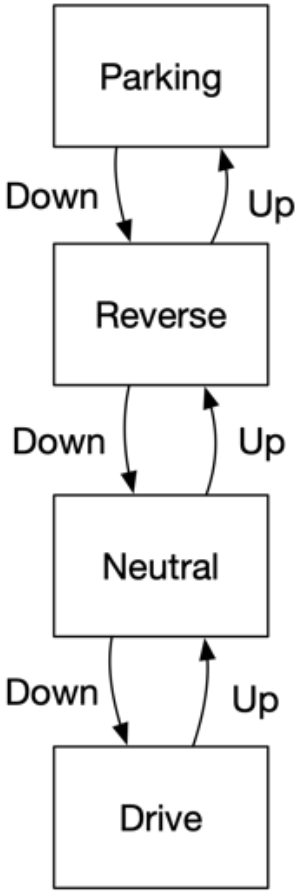
Mental model



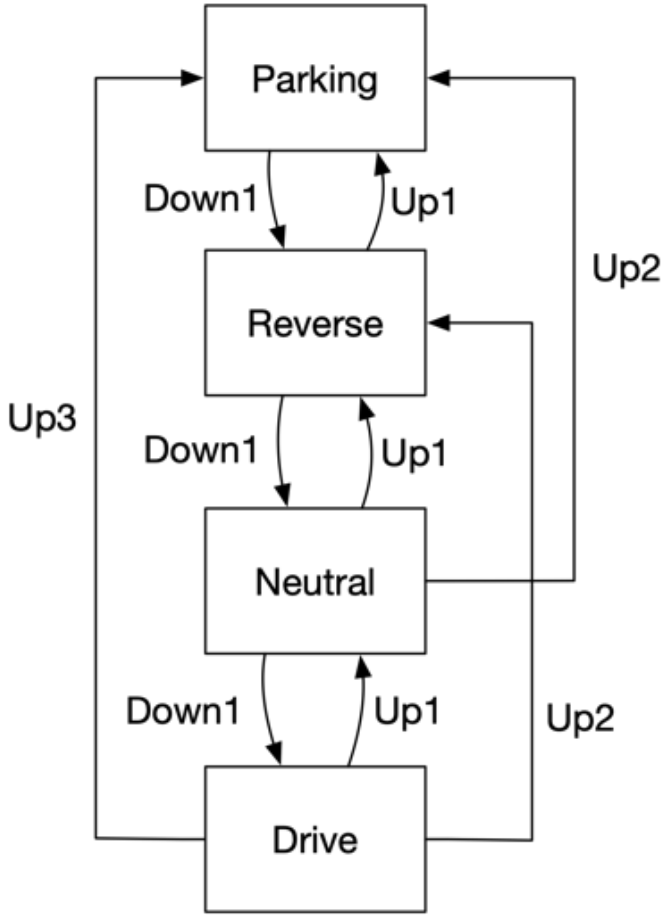
System model

- Compare the two state machines and identify possible mismatch
  - Ideally, the machines have the identical behavior (also called bisimulation)
- Can the state machines ever end up in different states?
- Can the system end up in a state where the user can't perform a desired action?
- Do actions and states have a valid correspondence across the state machines?

Comparing Mental Model vs. System



Mental model



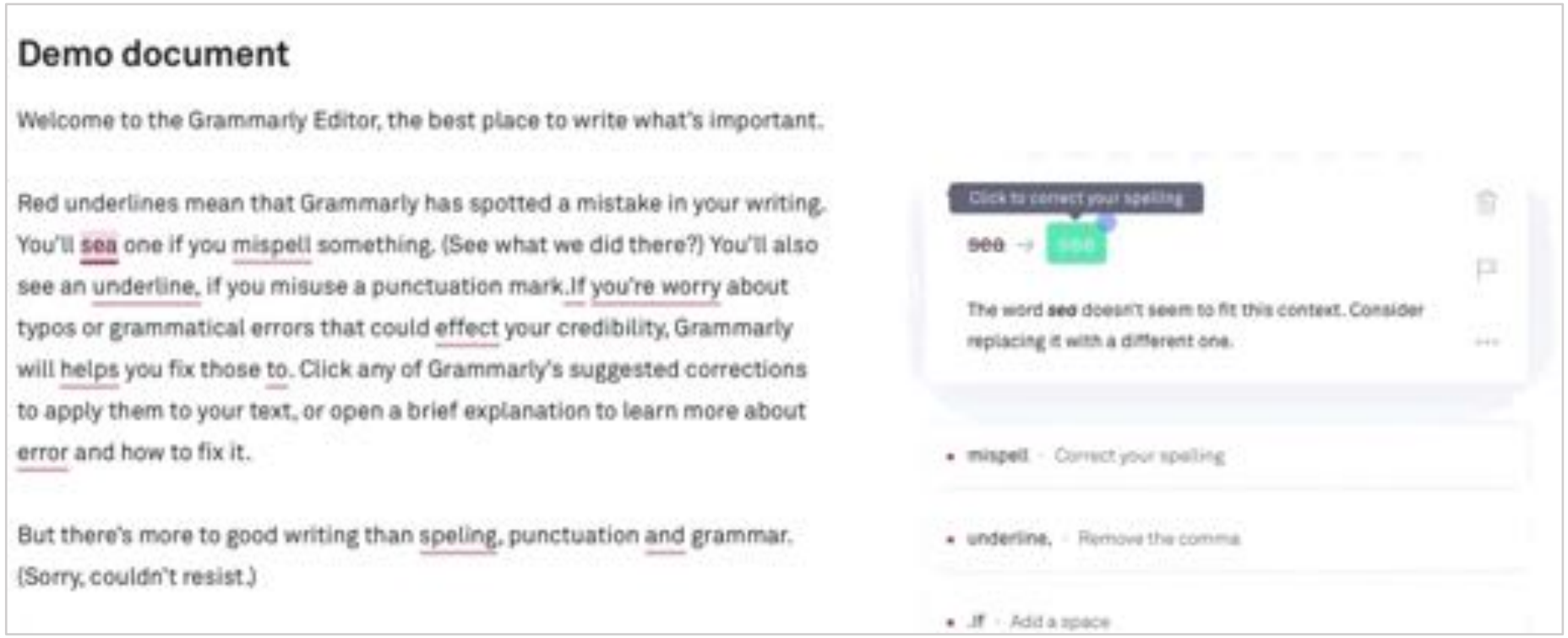
System model

- **Q. Is there a potential mismatch? What could go wrong?**

Adjusting the Mental Model to the System

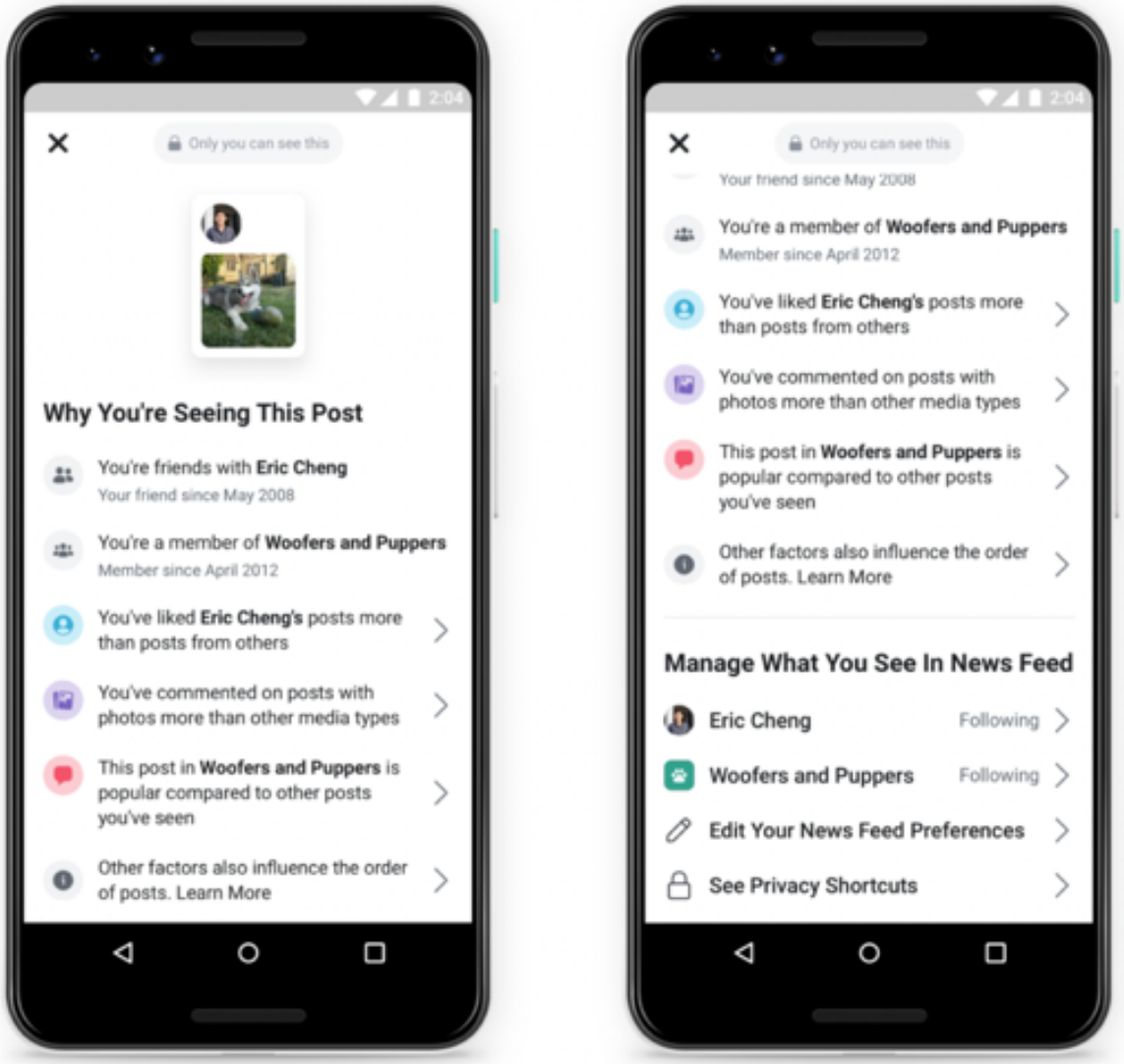
- Certain innovative applications/products have user interactions that do not fit into an existing mental model
- Provide an aid to help the user adjust or develop a proper mental model that aligns with the system
- **Strategies**
  - Set the user's expectations through onboarding
  - Increase transparency about how the system works by explaining its behavior to the user

Onboarding



- Introduce the user to the expected interactions with the system
- Provide examples of how the system works
- Be explicit about what the system can and cannot do

Explaining System Behavior



- Be transparent about how the system behaves
- Inform the user about available actions
- Aid the user in gradually building or adjusting their mental model

Designing for Usability: Tips

- Developers focus on design & code, and often do not see the software from the user's perspective
- Ultimately, it's the users who decide how the software will be used
  - They are always right, even if they seem erratic or incompetent
  - Software that is not usable will likely be misused or not used at all
- Understanding the user's mental model is the key to usable software
- Engage with users & identify their common mental model
- Be conservative! Use an interaction design that matches an existing mental model
- If the product is innovative, explicitly guide the user in building an accurate mental model through onboarding and transparency

Summary

- Exit ticket!