

Follow-up: Classification & Root Cause

Facilitator:

Participant ID:

3. Classifying Problems & Errors *Slips require better UI constraints. Mistakes require better user guidance.*

Slips (Action-based Errors) User had right goal but hit the wrong button. <ul style="list-style-type: none">• Capture Slips: Habit beat intended action.• Description Slips: Target looked like 'X'.• Mode Errors: User forgot app state.		
Mistakes (Goal-based Errors) User formed wrong goal or idea of how app works. <ul style="list-style-type: none">• Rule-based: Wrong "if-then" logic.• Knowledge-based: No rule, faulty logic.• Memory-lapse: Forgot the plan mid-task.		

4. Summary & Root Cause *Distill raw observations into design requirements.*

Discoverability Failures: <i>Features requiring trial-and-error</i>	1. <input type="text"/>	
	2. <input type="text"/>	
Heuristic Violations: <i>(e.g. Lack of Undo, Poor Consistency)</i>	<input type="text"/>	
The "Five Whys" Analysis: <i>Take the biggest problem and ask "Why?" until you find the real design flaw.</i>	<div>Problem: <input type="text"/></div> <div>1. Why? <input type="text"/></div> <div>2. Why? <input type="text"/></div> <div>3. Why? <input type="text"/></div> <div>4. Why? <input type="text"/></div> <div>5. Why? <input type="text"/></div>	
Root Cause:	<input type="text"/>	
Redesign Recommendation: <i>How to prevent this error?</i>	<input type="text"/>	

Facilitator Reference Guide

1. Classification of Action-Based Slips

*Slips occur when the user has the **correct goal** but the execution of the action is flawed due to inattention.*

- **Capture Slips:** A frequent or recently performed habit “captures” the intended action because the two sequences share a similar start (e.g., a user automatically types their home address when they intended to type a new destination).
- **Description-Similarity Slips:** The user performs the correct action on the **wrong object** because the target and the incorrect object look similar (e.g., clicking a “Delete Map” icon because it looks like the “Clear Search” icon).
- **Mode Errors:** Occur when a control has different meanings in different states, and the user forgets which mode they are in (e.g., trying to pan the map while the “Drop Pin” mode is still active).

2. Classification of Mistakes

*Mistakes occur when the user’s **goal or plan is wrong**, even if they execute the actions of that faulty plan perfectly.*

- **Rule-Based Mistakes:** The user misinterprets the situation and applies the **wrong “if-then” rule** (e.g., believing that if they zoom in all the way, a specific map layer will automatically appear, which is not how the app works).
- **Knowledge-Based Mistakes:** The user has no rule to follow and tries to solve the problem with **faulty logic** in a novel situation (e.g., assuming a map uses imperial units when it actually uses metric).
- **Memory-Lapse Mistakes:** The user **forgets the goal** or the current state of the plan due to an interruption (e.g., after a phone call, the user forgets why they opened the map and starts a different, irrelevant task).

3. The 10 Usability Heuristics

Broad rules of thumb for interaction design to identify potential “holes” in the system.

1. **Visibility of System Status:** The design should keep users informed about what is happening through **appropriate feedback** within a reasonable time.
2. **Match Between System and the Real World:** Use words and phrases **familiar to the user**; follow real-world conventions (e.g., “Compass” vs “Coordinate Vector”).
3. **User Control and Freedom:** Provide a clearly marked **“emergency exit”** to leave an unwanted state (e.g., an “Undo” or “Cancel” button).
4. **Consistency and Standards:** Follow **platform conventions** so users don’t have to wonder whether different words or actions mean the same thing.
5. **Error Prevention:** Eliminate error-prone conditions or present users with a confirmation option before they commit to an action.
6. **Recognition Rather than Recall:** Minimize memory load by making objects and options **visible**. The user should not have to remember info across screens.
7. **Flexibility and Efficiency of Use:** Shortcuts—like **touch gestures**—can speed up interaction for expert users.
8. **Aesthetic and Minimalist Design:** Interfaces should not contain irrelevant info; every extra unit of info competes with relevant items for visibility.
9. **Help Users Recognize, Diagnose, and Recover from Errors:** Error messages should be in **plain language**, indicate the problem, and suggest a solution.
10. **Help and Documentation:** It is best if the system is intuitive, but provide searchable, task-focused documentation to help complete specific goals.