

UI Design Fundamentals

- UI is the visual, interactive layer users perceive and operate
- UX covers the entire end-to-end experience and task flow
- UI quality directly affects user trust, task completion, and business outcomes
- UI designers deliver components, screens, states, and responsive specifications
- By the end: apply core principles, build accessible components, evaluate interfaces



Core Principles of Effective Interface Design



- **Clarity:** every element's purpose is obvious and unambiguous



- **Consistency:** same patterns, language, and behavior throughout



- **Visual hierarchy:** size, color, and spacing guide attention



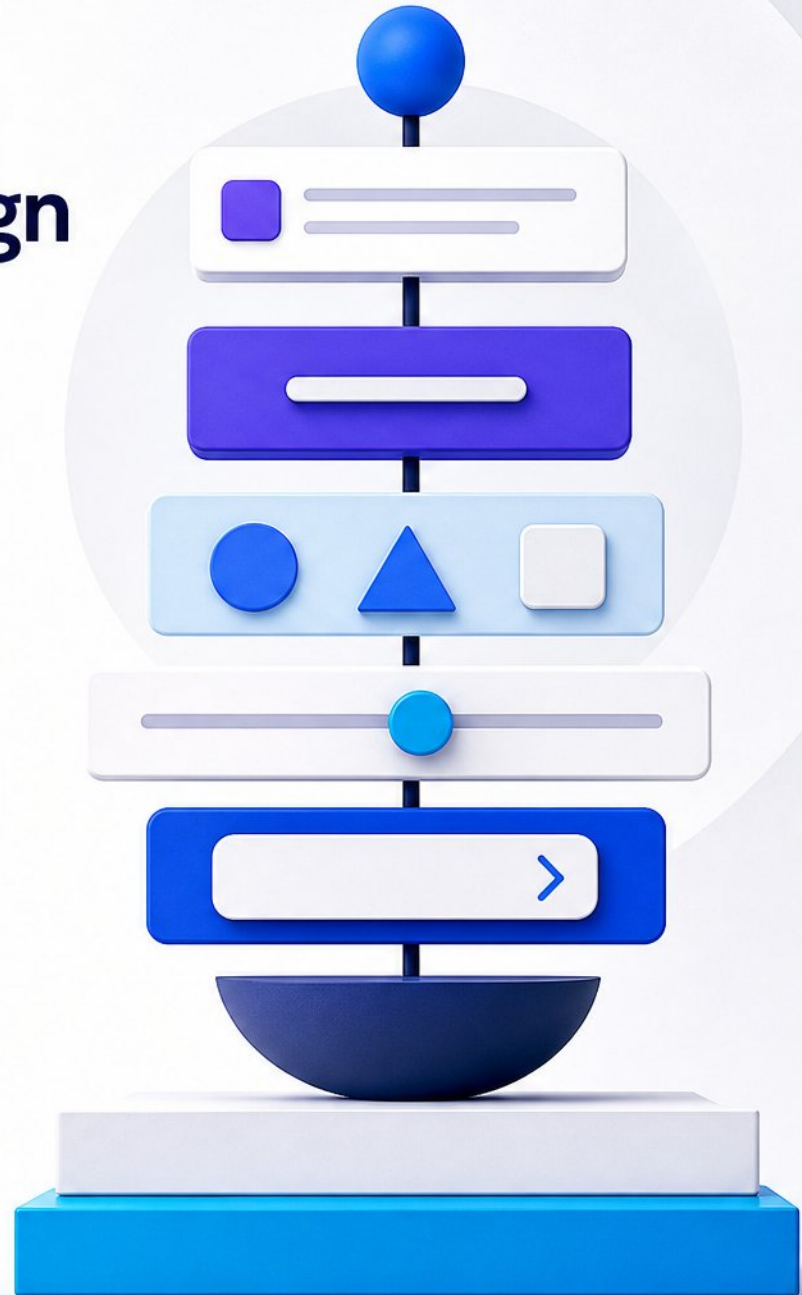
- **Feedback:** every action gets a visible, timely response



- **Affordance:** appearance suggests how to interact

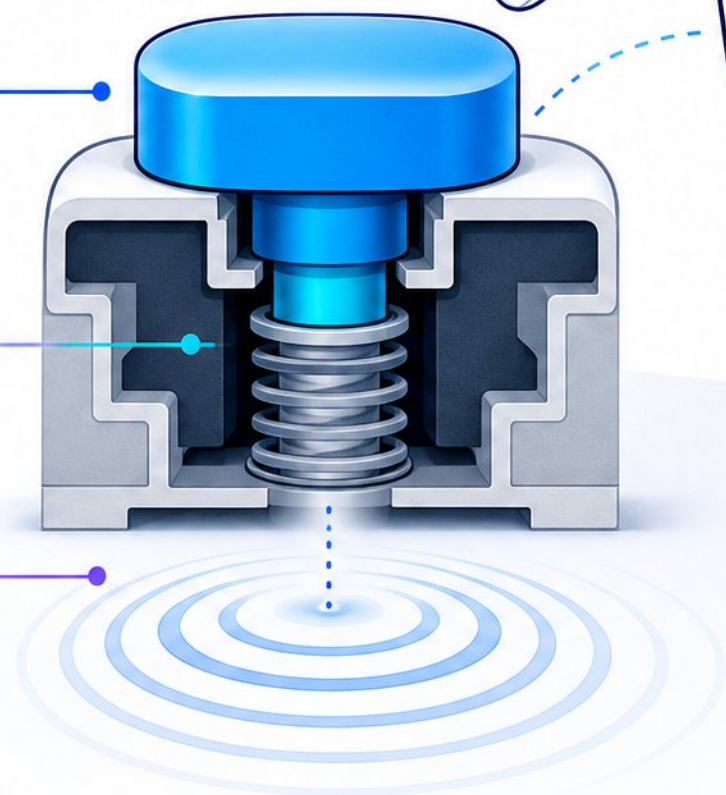


- These map to Nielsen's usability heuristics



Affordances, Signifiers, and Feedback

- - Affordance defines possible actions; signifier communicates that cue
- Types include explicit, hidden, pattern, metaphorical, negative, and false
- Feedback has three moments: preemptive, immediate, and outcome
- Under 400ms feels instant; over 2 seconds needs a progress indicator



PREEMPTIVE
Sets expectation
before action.



IMMEDIATE
Occurs right as
action happens.



OUTCOME
Confirms the
final result.

Visual Hierarchy and Layout



- Use size, color, contrast, spacing, and position to guide attention



- Apply grid systems, cards, lists, and panels for clear structure



- Follow F-pattern and Z-pattern scanning for natural content placement



- Sequence information with progressive disclosure to reduce cognitive load

F-PATTERN



Z-PATTERN



Color and Contrast for Accessible Interfaces



- Hue, saturation, value, and color scales guide UI color choices

T_T

- Normal text needs 4.5:1; large text needs 3:1



- Non-text components (borders, focus rings) need 3:1



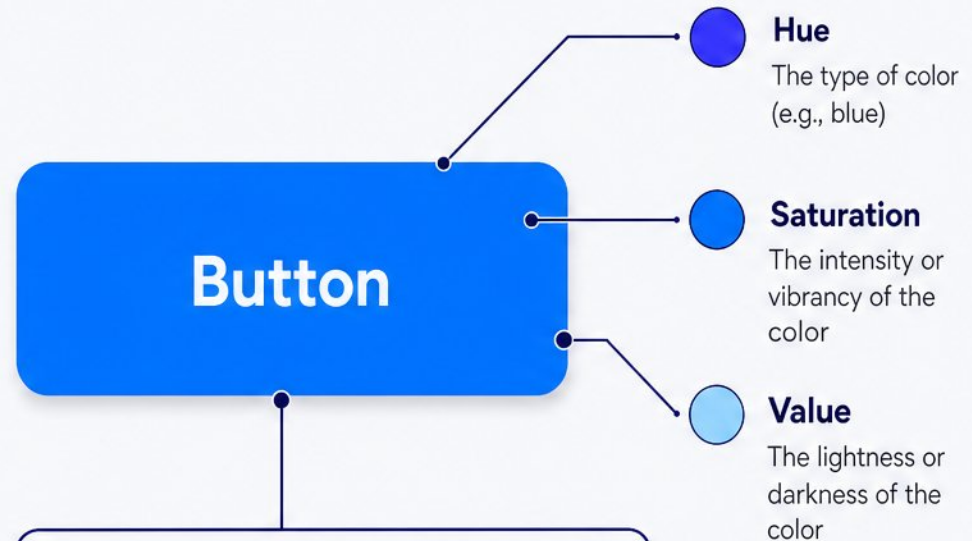
- Disabled controls and logos are exempt from contrast rules



- Never rely on color alone; add labels, icons, or patterns

UI Button Specimen

Example button analyzed for color and contrast



WCAG Contrast Ratio (Text)

Ensures sufficient contrast between button text and background.



Good contrast makes interfaces readable and usable for everyone.

Typography and Readability



- Choose typefaces for legibility on screens, not print aesthetics



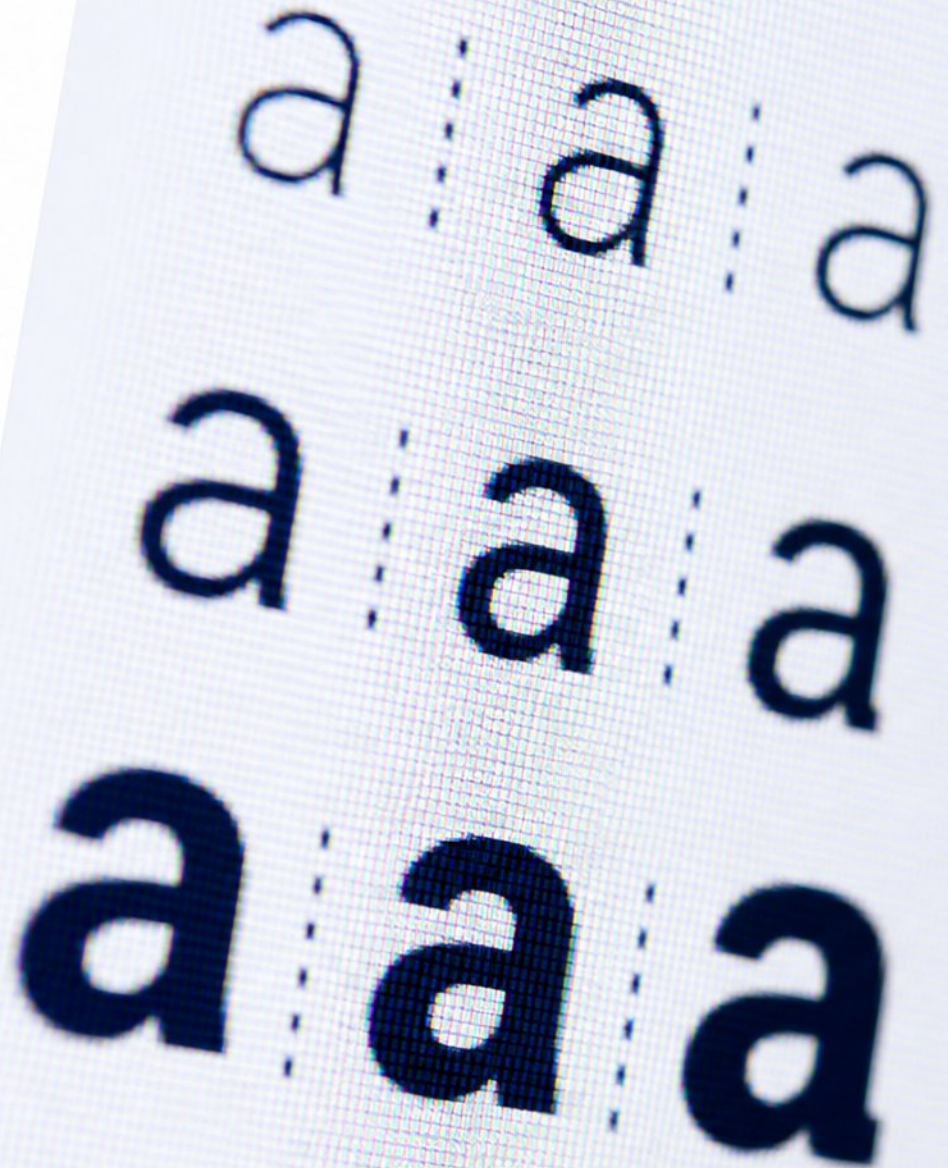
- Set practical sizes, line heights, and weights per text role



- Build a limited, purposeful type scale



- Use interface type for UI; reserve display type for moments



Components, Patterns, and Interaction States



Components are state machines: default, hover, focus, active, disabled



Prop-driven states persist; interaction states stay transient



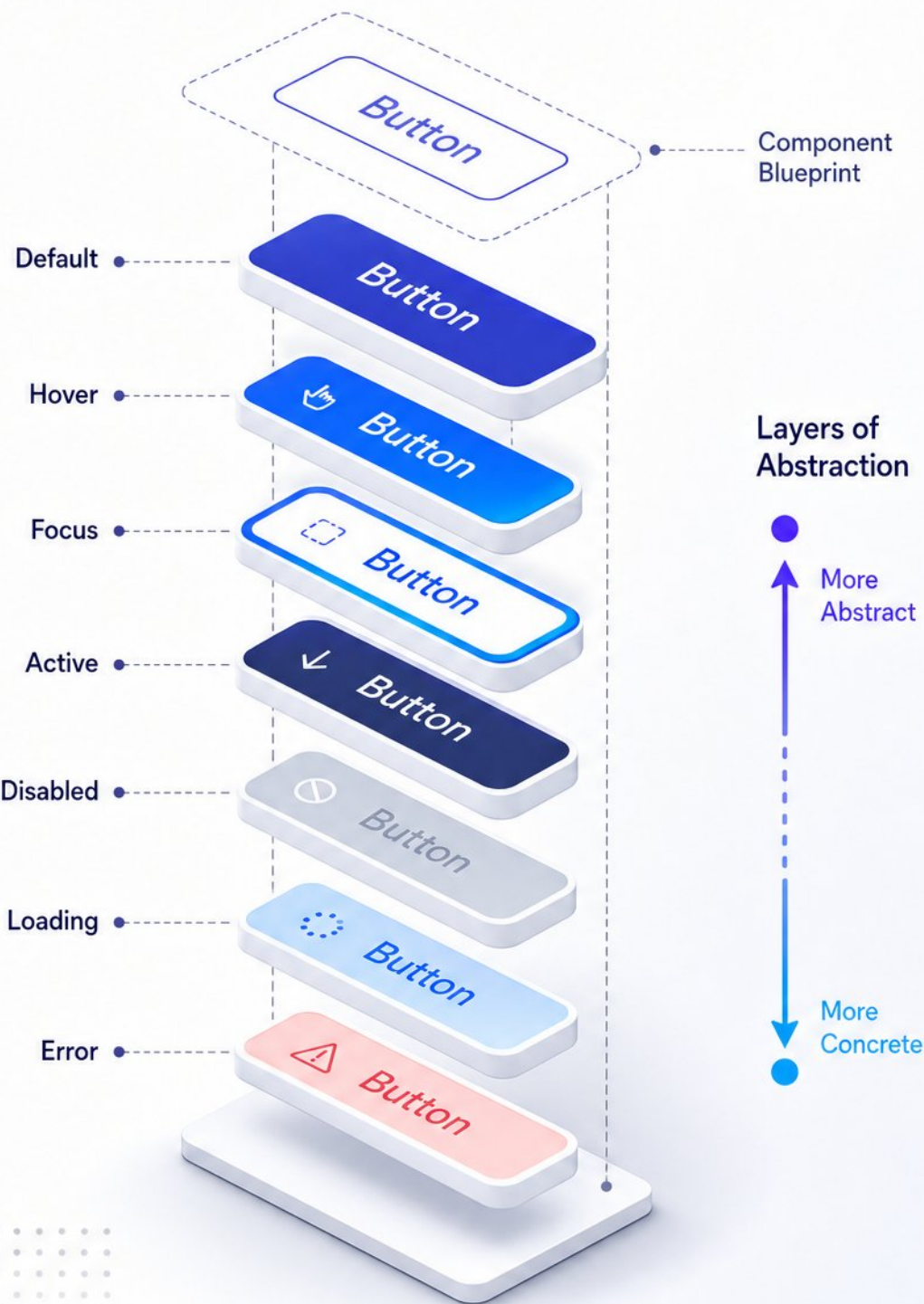
Precedence decides conflicts; layered states compose on top



Focus-visible is mandatory for keyboard accessibility



Prioritize known patterns for consistency and speed



Designing the Unhappy Paths



- Loading, error, empty, and disabled states are first-class design work



- Write specific, recoverable error messages that explain what failed and how to fix it



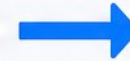
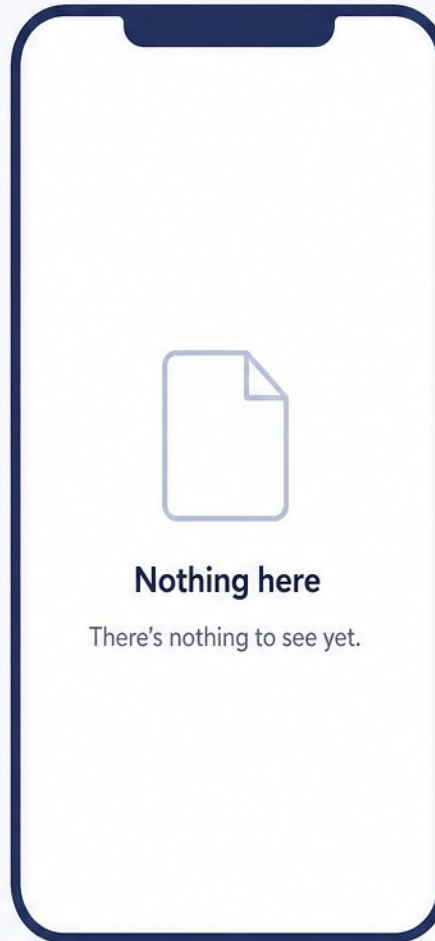
- Design empty states that orient users and invite the next action



- Use loading indicators to prevent duplicate actions and show system status

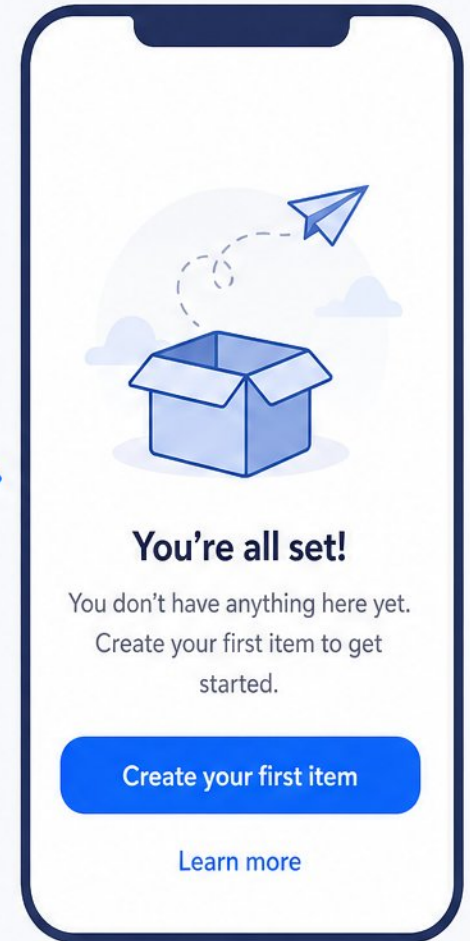
BEFORE

Dead-end empty state



AFTER

Delightful, instructive empty state

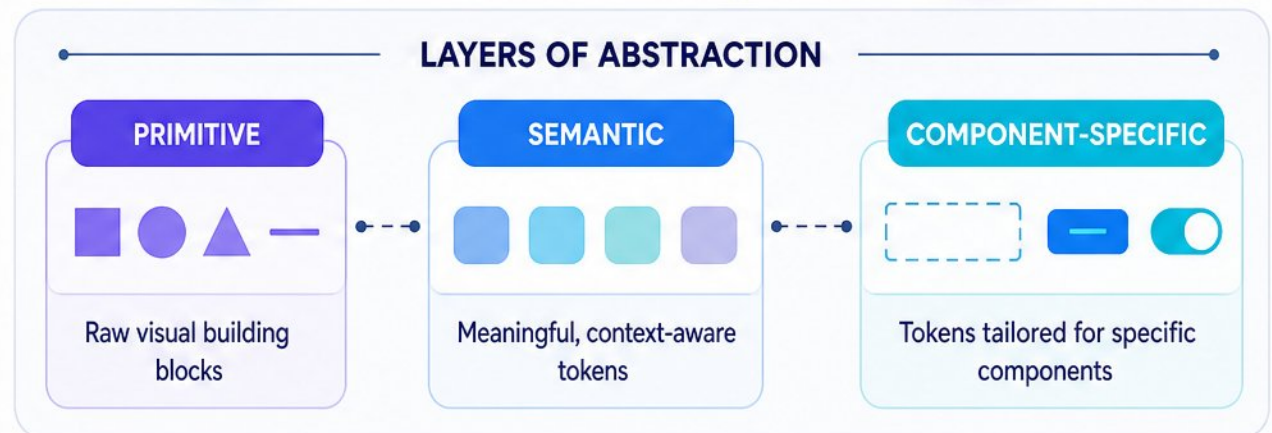
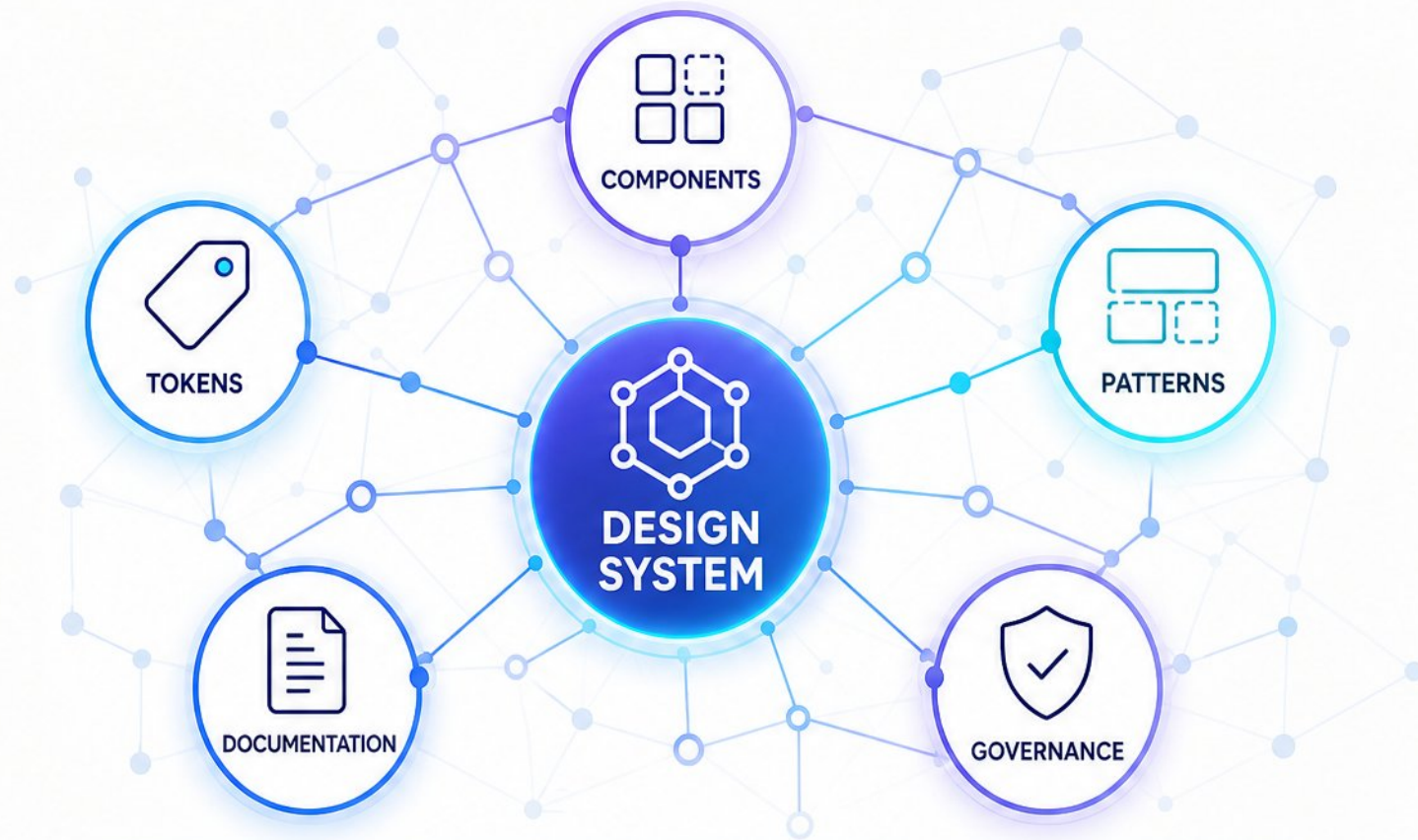


Design for the moments things don't go right.

Clear, helpful unhappy paths build trust and keep users moving forward.

Design Systems and Reusable UI

- Design system: a governed whole with tokens, components, patterns, and docs
- Design tokens are named visual decisions bridging design and code
- Tokens stack in layers: primitive, semantic, and component-specific
- Component and pattern libraries support consistency and scalability
- Governance, contribution paths, and versioning keep the system alive



Responsive and Mobile-First UI Design



- Start with smallest screens and core tasks, then enhance for larger sizes



- Use flexible grids, relative units, and content-driven breakpoints



- Provide at least 44x44px touch targets with adequate surrounding space



- Position primary actions in thumb-friendly lower reach zones



- Avoid placing interactive elements near screen edges and system gestures



Prototyping at the Right Fidelity



- Start with low-fidelity wireframes to validate structure and flow



- Increase fidelity as design confidence grows



- High-fidelity prototypes polish visual details and interactions



- Test touch, hover, disabled states, and conditional logic



- Include empty, loading, success, and error states early



Effective Developer Handoff



- Handoff is continuous collaboration, not a file transfer



- Provide states, behavior notes, responsive rules, final copy, acceptance criteria



- Map design tokens and component names to code



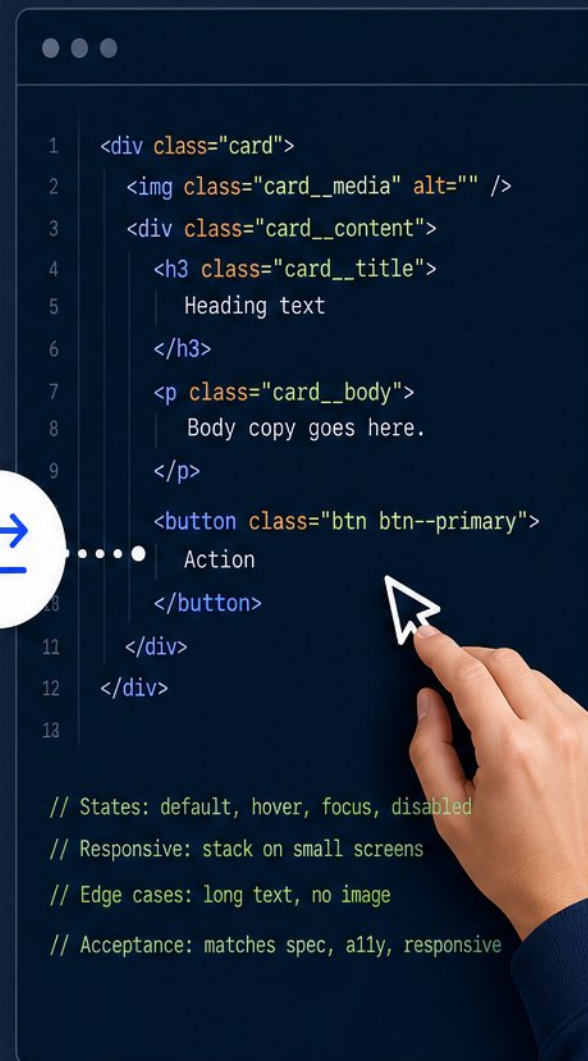
- Document logic and edge cases, not just pixels



Designer



Developer



Evaluating Interface Quality



- **Heuristic evaluation:** experts inspect UI using Nielsen's 10 heuristics



- Use 3-5 independent evaluators for better coverage



- **Usability tests** reveal barriers real users hit during tasks

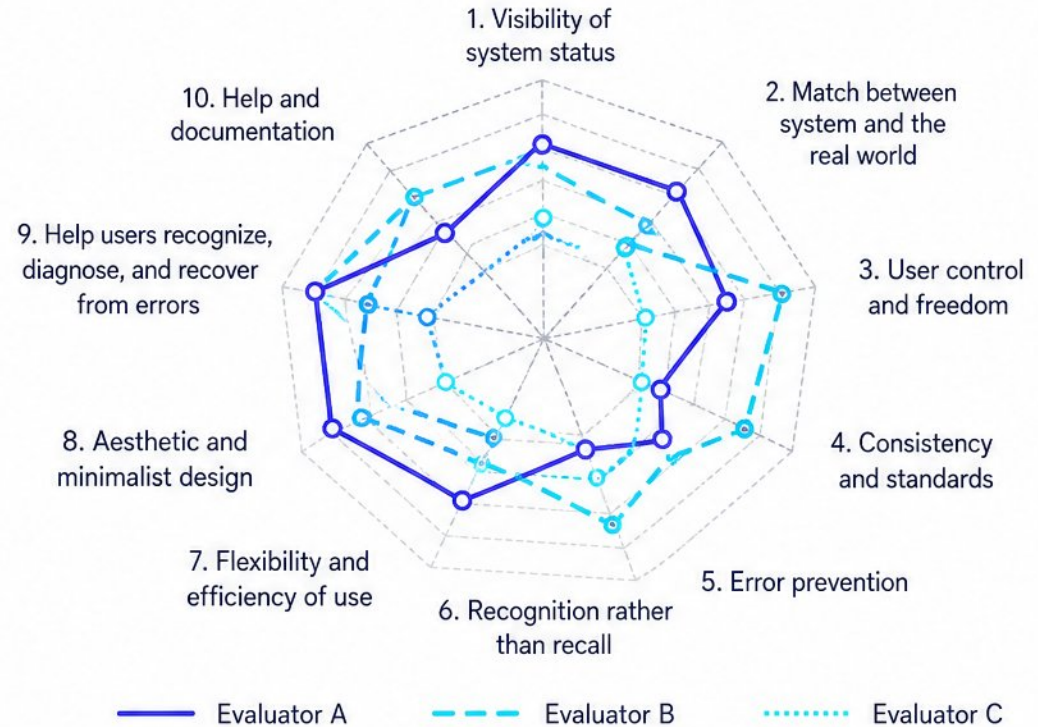


- **Design critiques** gather feedback on clarity, hierarchy, and accessibility



- **Combine expert review** with user testing for actionable findings

Heuristic Evaluation (Nielsen's 10 Heuristics)



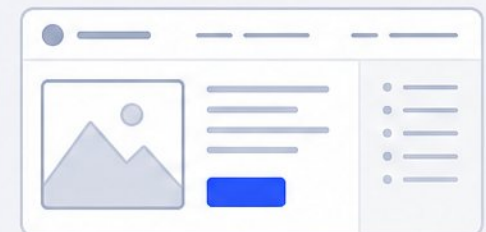
BEFORE



Unclear, inconsistent, hard to use



AFTER



Clear, consistent, easy to use

Measuring and Improving UI Design



- Track task completion, errors, and rage-tap patterns



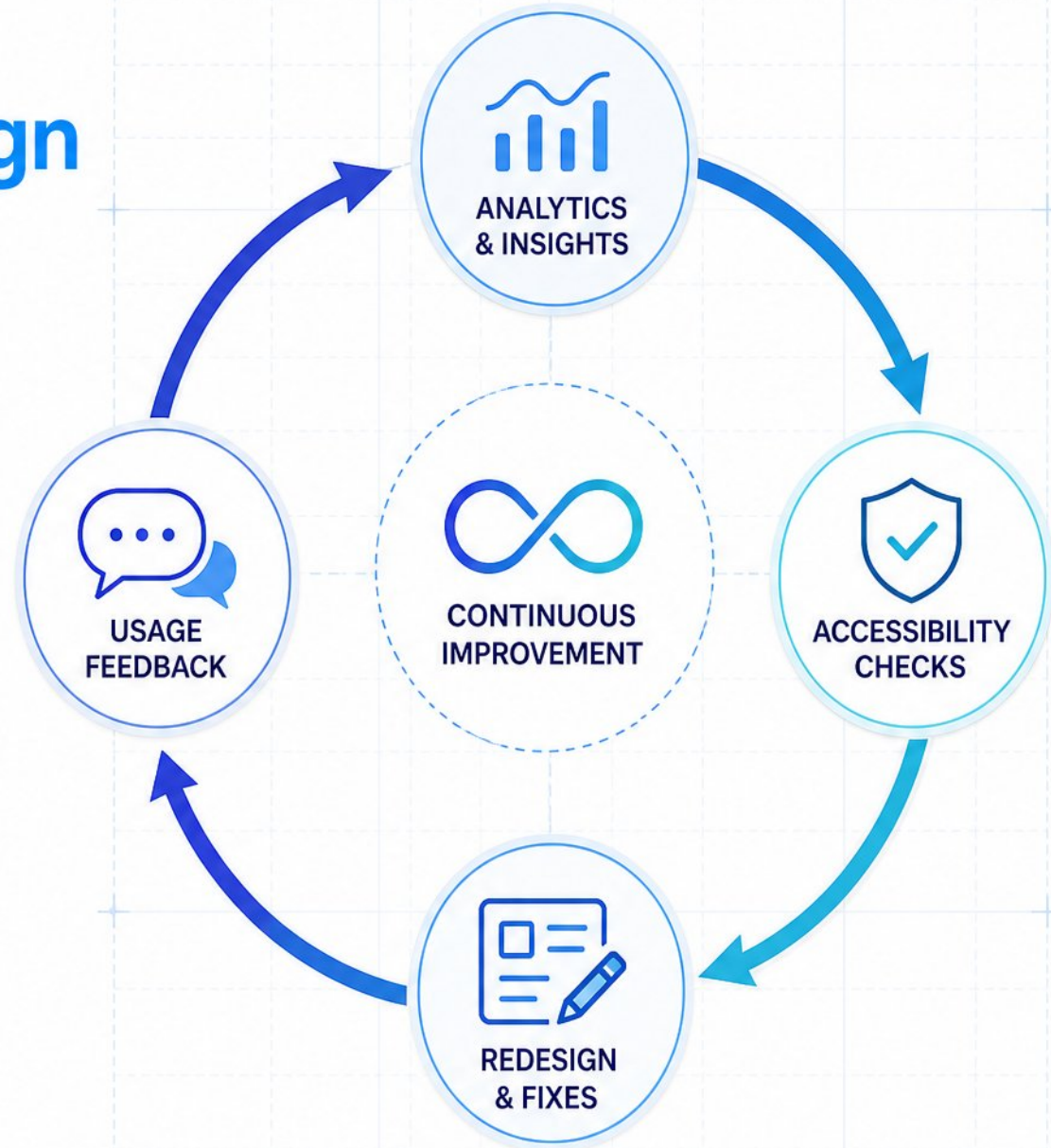
- Check contrast, focus visibility, touch targets, and keyboard navigation



- Run iterative cycles: evaluate, prioritize, redesign, retest



- Build continuous improvement through analytics, feedback, and design system care



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/69790b54/public