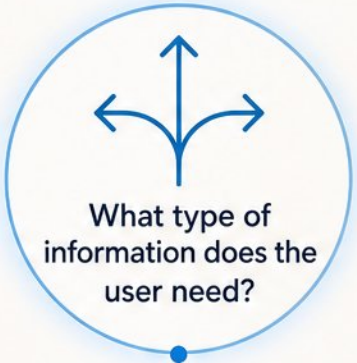
Introduction to Designing Helpful Tooltips and Contextual Help

- - Tooltips: brief, hover-or-focus overlays for just-in-time info
- - Distinct from popovers, inline help, coach marks, and errors
- - Core goal: reduce cognitive load and prevent errors
- - Training covers foundations, design, writing, accessibility, and iteration

Tooltip, Popover, or Something Else?

A Decision Framework



Tooltip

non-essential, read-only microcontent triggered by hover or focus

Popover

richer, interactive content that supports user actions

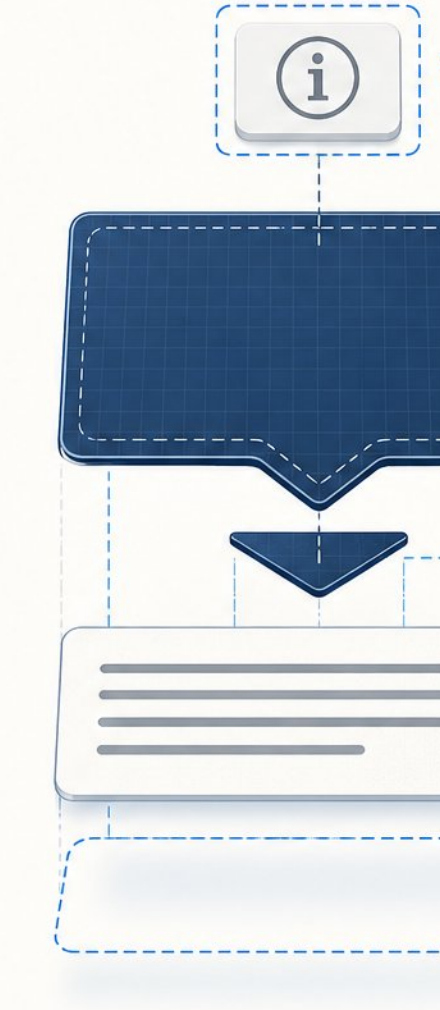
Inline text

essential instructions permanently visible in the UI

- Tooltip: non-essential, read-only microcontent triggered by hover or focus
- Popover: richer, interactive content that supports user actions
- Inline text: essential instructions permanently visible in the UI
- Use tooltips for unlabeled icons, jargon, shortcuts, and disabled states
- Start without tooltips; add only when data shows persistent confusion

Anatomy and Visual Design of Effective Tooltips

- Trigger element, container, pointer arrow, optional dismiss button
- Anchor positioning with position-try-fallbacks prevents overlap
- 4.5:1 text contrast, 3:1 boundary contrast, 20–30 chars per line
- Consistent padding, readable typography, distinct but subtle appearance
- Handle default, hover, focus, and touch states predictably



TRIGGER ELEMENT
Interactive element that invokes the tooltip

CONTAINER
Holds the content and provides visual structure

POINTER ARROW
Connects the tooltip to its anchor element

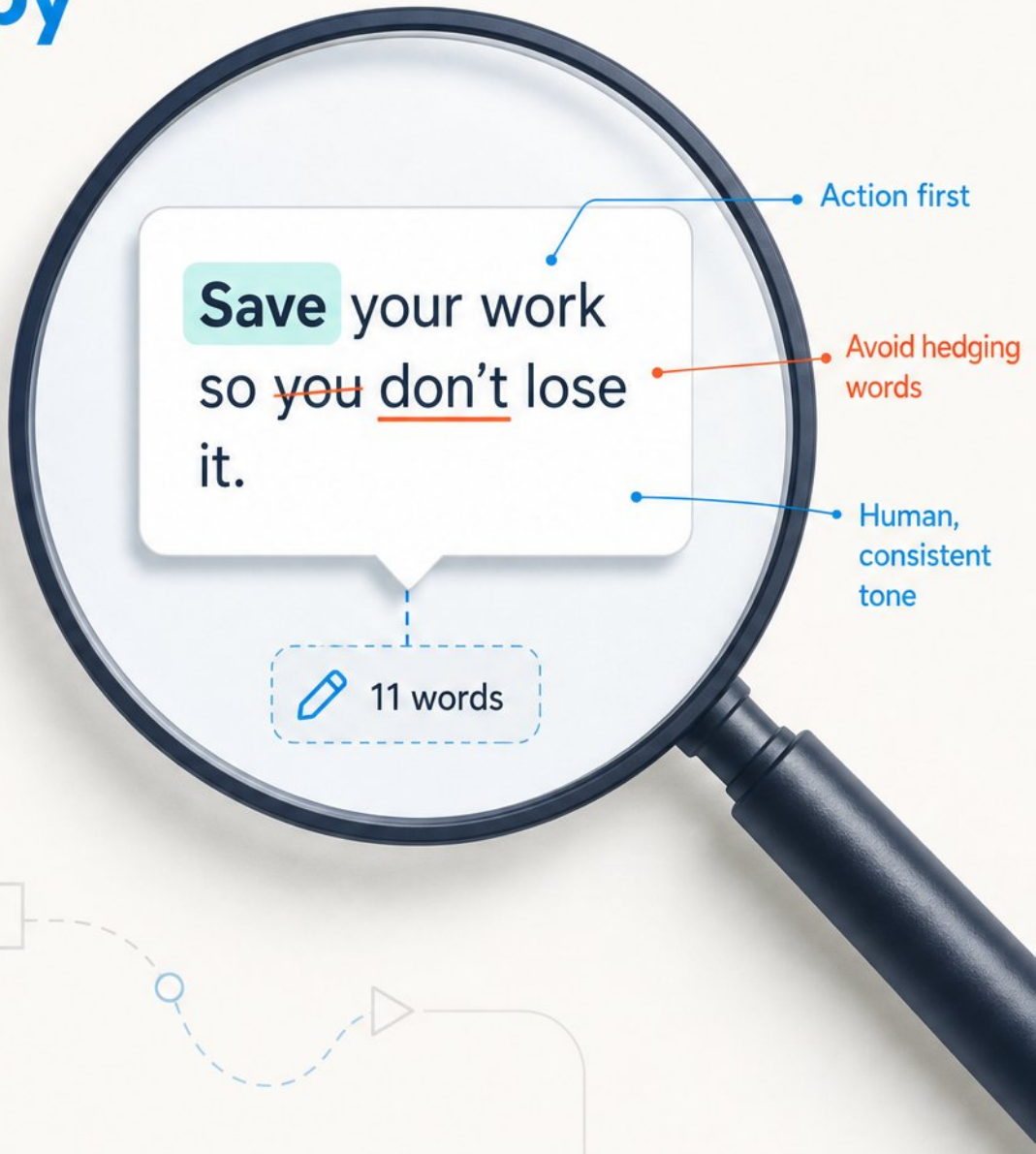
CONTENT (TEXT LAYER)
Clear, concise, and easy to scan

BOUNDARY / SURFACE
Subtle boundary contrast separates from background

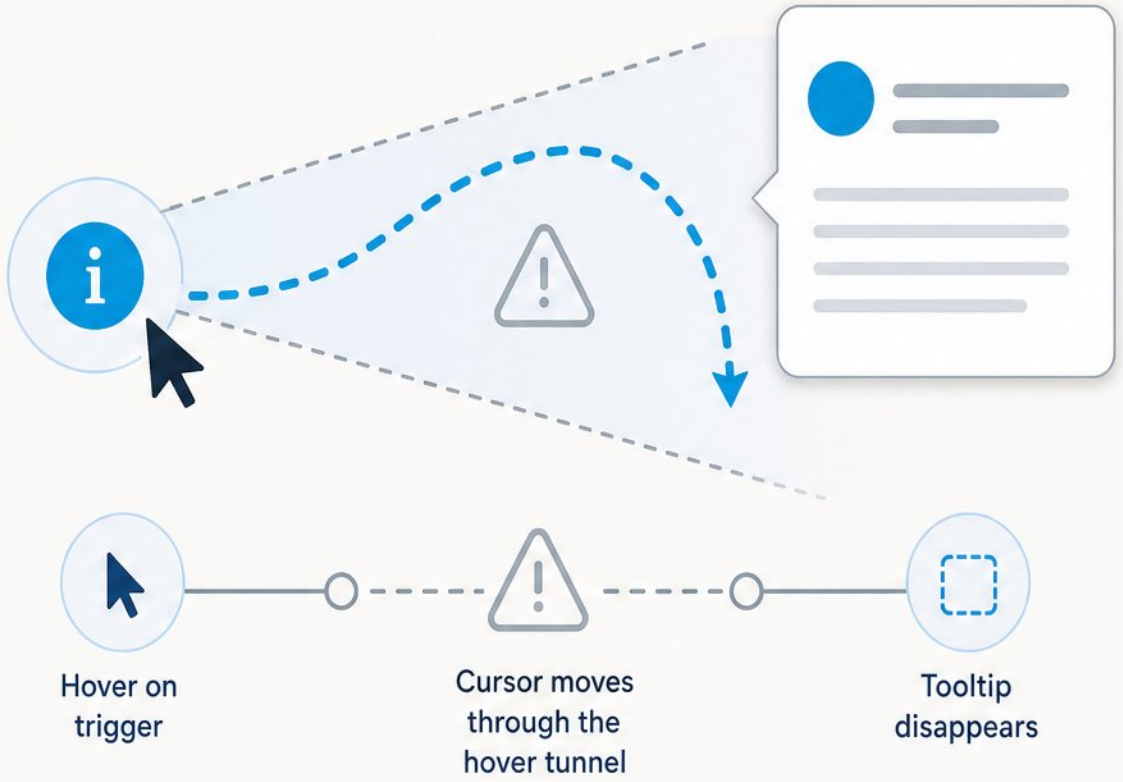
Blueprint principles: clarity, hierarchy, contrast, consistency, and accessibility.

Writing Clear, Scannable Microcopy

- Keep it to 1–2 lines, under 150 characters
- Put the most important word or action first
- Write in active voice with a human, consistent tone
- Never repeat the trigger's visible text verbatim
- Cut hedging words like 'you can' or 'in order to'



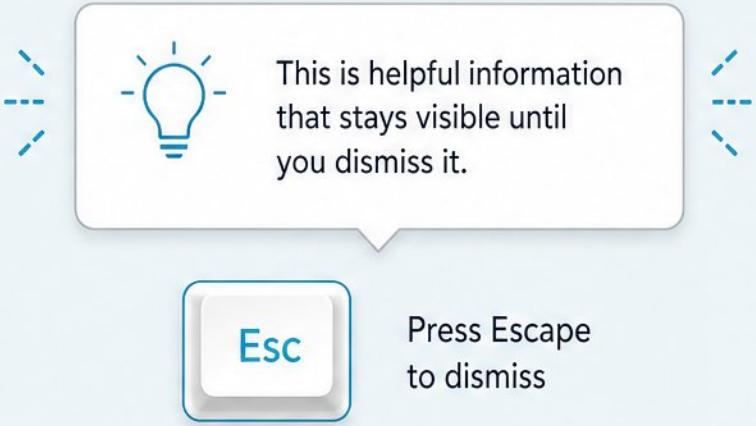
Triggers, Timing, and the Hover Tunnel Problem



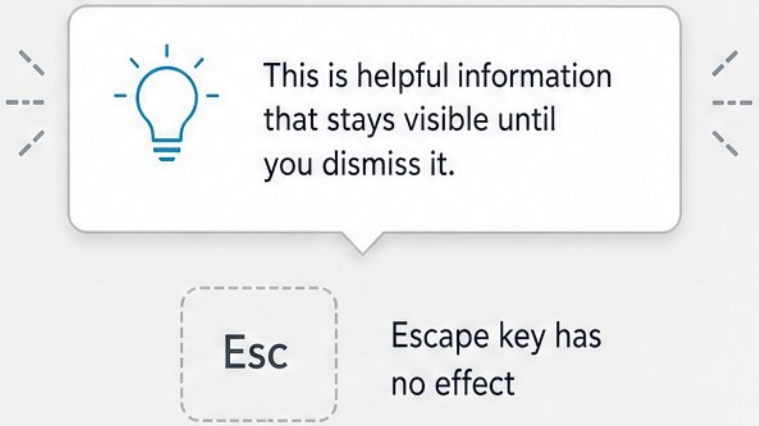
- Hover-only tooltips fail on touch devices and are inaccessible
- The hover tunnel: tooltip disappears when cursor moves toward it
- Solution: keep visible on hover (WCAG 1.4.13), add hide delay
- Touch adapt: long-press, dismiss button, or replace with popover

Dismissal Behavior and User Control

DISMISSES WITH ESCAPE



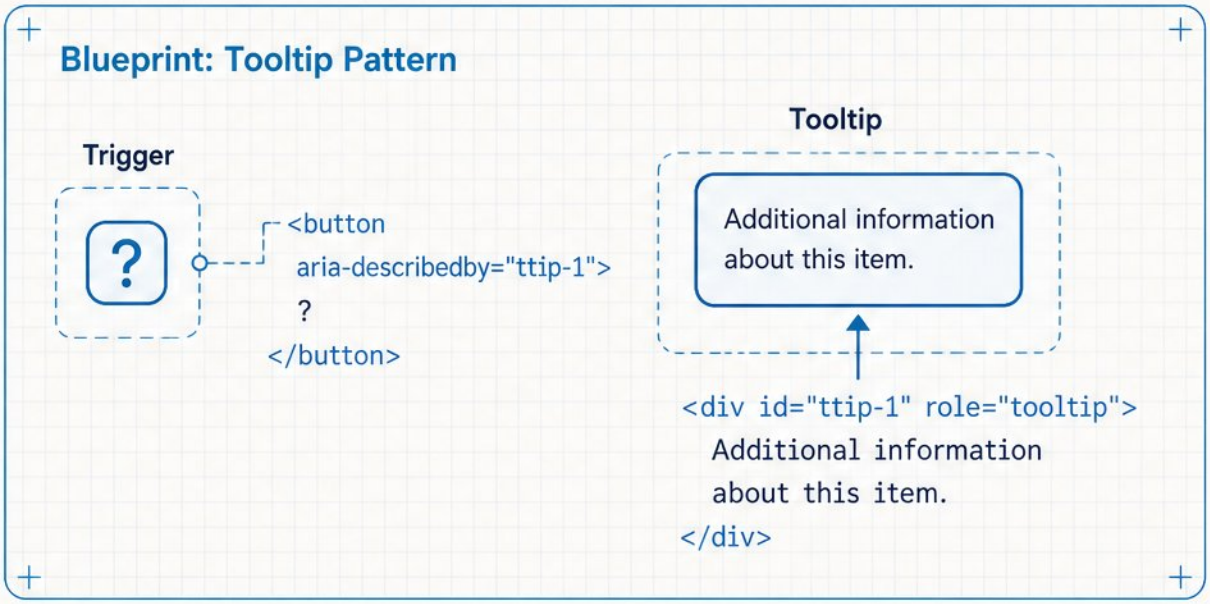
DOES NOT DISMISS WITH ESCAPE



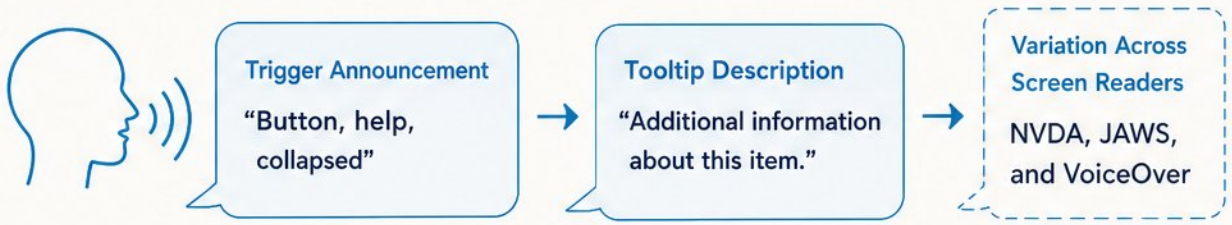
- Escape key must dismiss tooltip without moving focus or hover
- Dismiss on click outside, focus loss, or hover away
- Avoid auto-dismiss timers that fire too quickly
- Persistent content stays until the user takes action
- Predictable dismissal supports users with attention or motor impairments

Accessibility Foundations: WCAG, ARIA, and Screen Readers

- Map critical WCAG 2.2 criteria: 1.4.13 (Dismissible, Hoverable, Persistent), 2.1.1 Keyboard, 1.3.1 Info and Relationships, 1.4.3 Contrast.
- Use `role='tooltip'` on the container and `aria-describedby` on the trigger pointing to the tooltip's ID.
- `aria-describedby` supplements the accessible name; it never replaces `aria-labelledby`.
- Screen readers announce the description after the trigger's name and role, with minor variations across NVDA, JAWS, and VoiceOver.



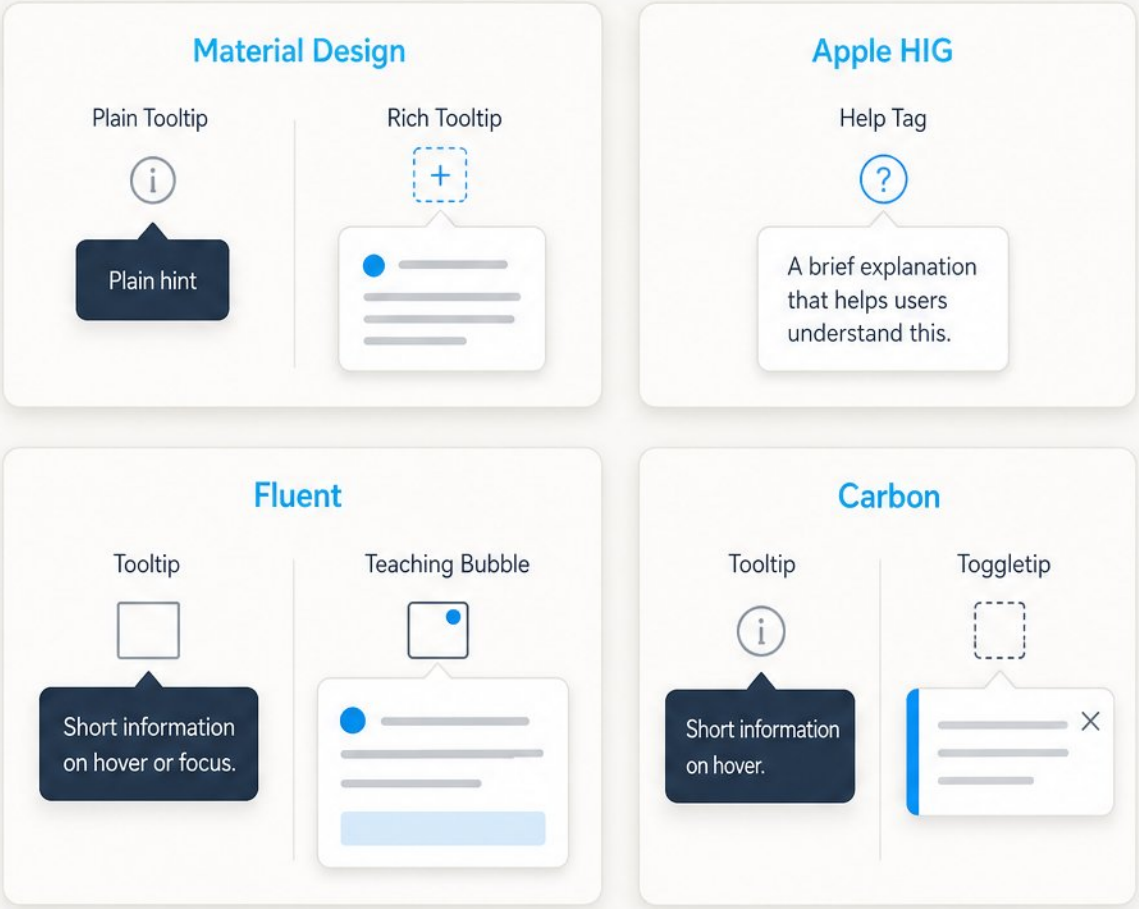
Screen Reader Announcement Flow (Example)



Inclusive Design Beyond Screen Readers

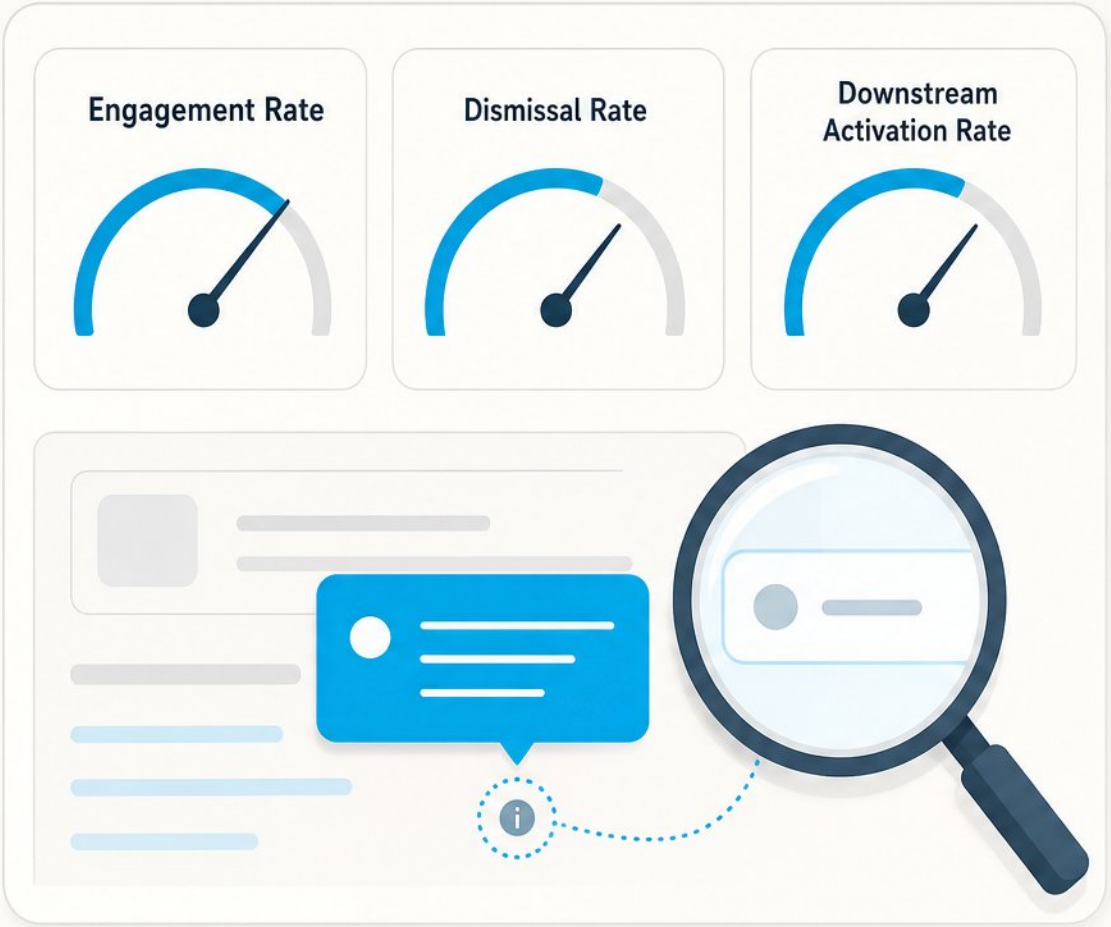
- Tooltips must appear on keyboard focus, never trap focus
- Ensure 4.5:1 text contrast and 200% zoom support
- Use generous hit targets and Escape-key dismissal
- Use plain language; keep content under 150 characters
- Never hide essential instructions inside a tooltip

Tooltips in Design Systems and Component Libraries



- Material Design: plain vs. rich tooltips; Apple HIG: help tags
- Fluent: tooltip vs. teaching bubble; Carbon: tooltip vs. toggletip
- Modern CSS anchor positioning replaces heavy JS positioning libraries
- Minimal spec: 60–200 char limit, placement rules, ARIA attributes
- Consistent usage across the product builds user trust and discoverability

Measuring Tooltip Effectiveness and User Impact



- **Define key metrics:** engagement rate, dismissal rate, and downstream activation rate
- **Investigate with qualitative methods:** usability tests, session recordings, and microsurveys
- **A/B test** tooltip presence, copy, and timing; optimize for downstream activation
- **Real-world case:** tooltip intervention recovered 75% of lost feature visits after redesign

When and How to Retire a Tooltip

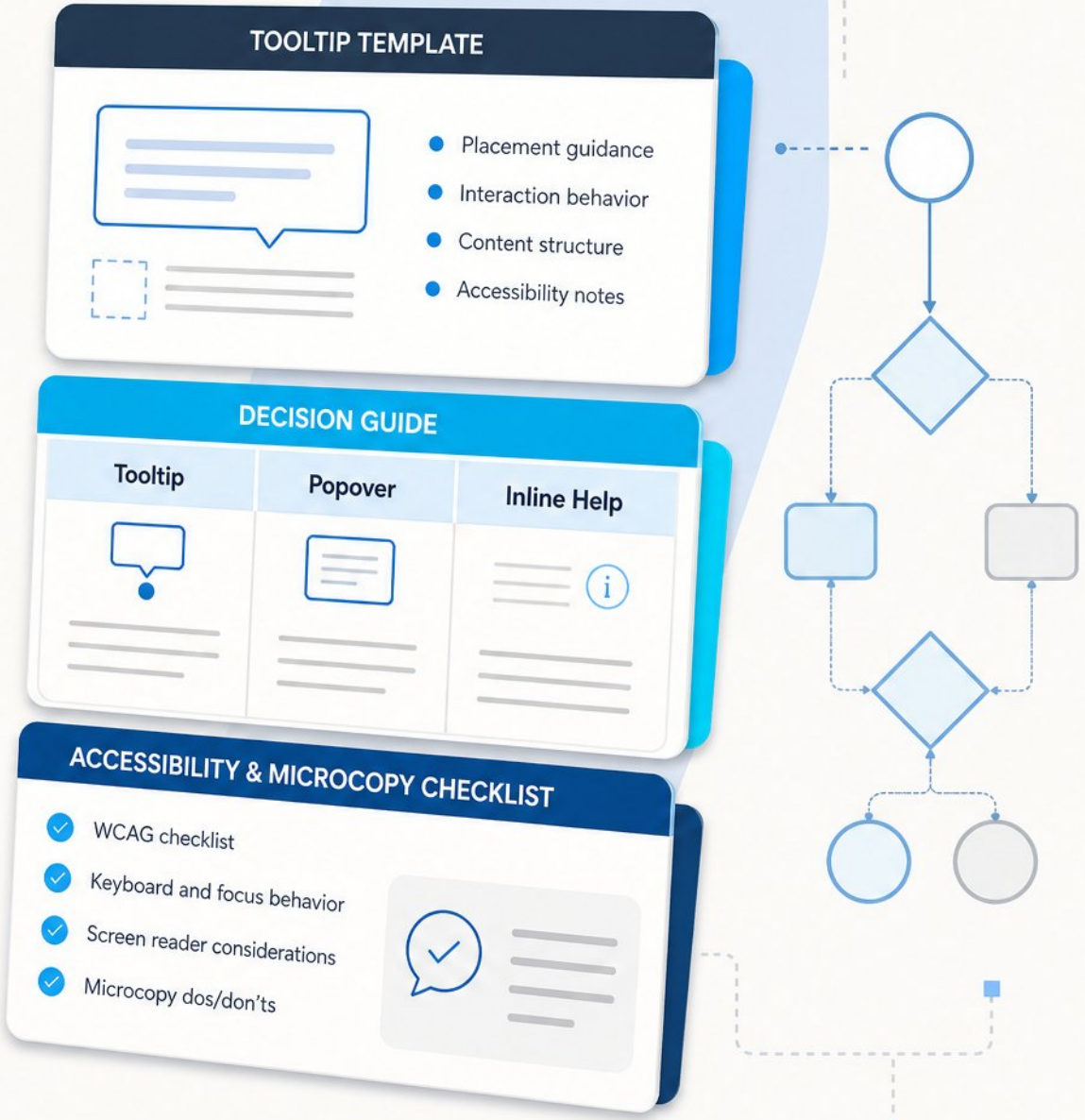
- Retire tooltips with 50%+ dismissal rates after two iterations
- Convert critical tooltips to inline text, popovers, or help panels
- Avoid 'help fatigue': too many tooltips teach users to ignore all
- The best tooltip makes itself unnecessary after one reading

Practical Workshop: Auditing Existing Tooltips

- **Apply a five-point audit checklist:** needed, findable, clear, dismissible, accessible
- **Redesign a weak tooltip:** improve trigger, copy, placement, and behavior
- **Peer-review** a real tooltip from your product using the checklist
- **Common anti-patterns:** redundant labels, hidden info, broken keyboard, blocking triggers

Building a Tooltip Design Kit and Next Steps

- **Build a reusable kit:** copy templates, placement rules, interaction specs, and accessibility requirements
- **Use a quick-reference card:** tooltip vs. popover vs. inline help, WCAG checklist, and microcopy dos/don'ts
- **Inventory all existing tooltips,** run the audit, and prioritize the top three pain points
- **Treat tooltips as a measurement** exercise—track metrics, iterate, and remove what doesn't help



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/28f1fdae/public