CREATIVE & DESIGN TOOLS

UI/UX Design Principles

The thinking behind good design decisions — UX/UI/interaction design distinctions, user research, information architecture, interaction design laws, visual design and Gestalt principles, color and typography, usability heuristics, wireframing and prototyping, iterative usability testing, and a closing capstone designing, prototyping, and testing a real small app feature from research through iteration.

10 Chapters | Philip Osztromok | Generated with Claude

CHAPTER 1: WHAT IS UX/UI DESIGN?

UI/UX Design Principles

Chapter 1 · What Is UX/UI Design?

This site's own Creative & Design Tools subject already teaches Figma, Photoshop, and Blender — the tools. This course covers something none of those do: the actual thinking behind a good design decision. It's a companion to those tool courses, not a replacement for them — you'll still open Figma to build something, but this course is about knowing *what* to build and *why* before you open it.

> **SCOPE NOTE**
>
> Accessibility gets a light mention throughout this course as a design consideration, but the real depth on that topic already lives in this site's own dedicated **Web Accessibility (a11y)** course — that course is the one to reach for when accessibility itself is the focus.

UX vs. UI vs. Interaction Design

These three terms get used almost interchangeably in casual conversation, but they describe genuinely different layers of a real design problem:

TERM	WHAT IT ACTUALLY COVERS	A USEFUL TEST QUESTION
UX (User Experience)	The entire end-to-end experience of using a product — research, structure, flow, emotion, outcome	"Did this actually solve the person's real problem?"
UI (User Interface)	The specific visual and interactive surface a person touches — buttons, layout, color, type	"Is this screen clear, consistent, and pleasant to look at?"
Interaction Design (IxD)	How a person's actions and the system's responses are choreographed over time	"When I do this, does the system respond the way I'd expect?"

UI is a real subset of UX — a beautifully designed screen (good UI) can still sit inside a confusing, poorly-researched product (bad UX). This course covers principles from all three layers, since a genuinely good design decision usually has to work at all of them at once.

A Real, Corrected History of "User Experience"

A POPULAR CLAIM WORTH CORRECTING

It's often said that cognitive scientist **Don Norman coined the term "user experience."** The real, verifiable history is narrower than that: in 1993, when Norman joined Apple, he took the title **"User Experience Architect"** — the first documented use of the phrase "User Experience" in a job title. That's a real, significant milestone (it's a big part of why the term entered the professional mainstream), but it isn't the same claim as having invented the term itself, which the historical record doesn't actually support.

Norman's own real, larger contribution predates that job title: his 1988 book *The Design of Everyday Things* popularized ideas like **affordances** (what an object's own design suggests you can do with it) and **feedback** (how a system confirms an action was received) — concepts this course will build on directly in Chapter 4.

The broader field itself is older still. Xerox PARC's own research in the 1970s produced the graphical user interface concepts (windows, icons, a mouse) that Apple's 1984 Macintosh brought to a mass consumer audience — decades of real interface-design thinking before "UX" existed as a job title at all.

The Design Thinking Process

Design thinking is a real, widely-taught problem-solving framework, most associated today with the design firm IDEO and Stanford's own d.school. Its real intellectual roots go back further — creativity research in the 1940s-50s, L. Bruce Archer's 1960s "Systematic Method for Designers," and Horst Rittel and Melvin Webber's 1970s concept of **"wicked problems"** (problems too complex and interconnected to solve with a single, linear pass) — but it was IDEO and Stanford who packaged it into the five real, named stages most design teams use today.

1

Empathize

Research your users' real needs

2

Define

State the real user need and problem

3

Ideate

Challenge assumptions, generate ideas

4

Prototype

Build cheap, testable versions

5

Test

Try solutions with real users

A REAL, IMPORTANT NUANCE

Design thinking is explicitly **not a strict, linear pipeline**. Stanford's own materials describe it as carried out "in a more flexible and non-linear fashion" — teams can run stages concurrently, repeat one, or circle back to an earlier stage the moment testing reveals the original problem definition was wrong. Treating it as a rigid five-step checklist misses the entire point of the framework.

Hands-On Exercises

EXERCISE 1

A colleague says "I already redesigned the UX of our checkout button — I made it bigger and changed its color." Explain, in your own words, why this description more accurately describes a UI change than a UX change, and what a genuine UX change to checkout might look like instead.

View solution

EXERCISE 2

A teammate claims "Don Norman invented the term user experience in 1993." Explain, in your own words, what the real, verifiable historical record actually supports instead, and why the distinction matters.

View solution

EXERCISE 3

A team finishes user testing (the Test stage) on a new onboarding flow and discovers users don't actually understand the core problem the product is trying to solve for them. Explain, in your own words, what the design thinking process says they should do next, and why that's only possible because the process is non-linear.

View solution

CHAPTER 1 QUICK REFERENCE

- **UX** — the whole experience; **UI** — the visual/interactive surface (a real subset of UX); **IxD** — how actions and responses are choreographed over time
- Real correction: Don Norman was the first to use "User Experience" in a **job title** (Apple, 1993) — the historical record doesn't support him having coined the term itself
- Norman's real, larger contribution: *The Design of Everyday Things* (1988) — affordances and feedback
- The GUI itself predates "UX" as a term — Xerox PARC research in the 1970s, brought to consumers by the 1984 Macintosh
- Design Thinking's five real stages: **Empathize → Define → Ideate → Prototype → Test** — explicitly non-linear and iterative, not a rigid checklist

CHAPTER 2: USER RESEARCH METHODS

UI/UX Design Principles

Chapter 2 · User Research Methods

Chapter 1 covered the Design Thinking process's first stage, **Empathize**, only by name. This chapter makes it concrete: the actual methods teams use to learn what real users need, rather than guessing — interviews, surveys, and the artifact both feed into, the persona.

The Real Gap Between What Users Say and What They Do

> **A POPULAR QUOTE WORTH CORRECTING**
>
> A quote often used to justify skipping user research altogether — "*If I had asked people what they wanted, they would have said faster horses*", attributed to Henry Ford — is widely considered **apocryphal by historians**. No documented source in Ford's own writings or speeches has ever been found; it appears to be a later invention that stuck because it made a clean-sounding point.

The correction matters because the quote is usually used to argue the opposite of good UX practice: "don't bother asking users, they don't know what they want." The real, evidence-based lesson researchers actually rely on is narrower and more useful: what people *say* they want in an interview and what they actually *do* when using a product often diverge — not because research is worthless, but because self-reported preference is only one kind of data. Good research combines what users say (interviews, surveys) with what they actually do (usability testing, analytics, contextual observation).

User Interviews

A one-on-one interview is the real workhorse of qualitative research — open-ended questions, asked to a real user, about their own actual experiences and behavior rather than hypothetical future features.

- **Ask about the past, not the future** — "Tell me about the last time you did X" gets real behavior; "Would you use a feature that did Y?" gets a guess.

- **Avoid leading questions** — "Don't you find this confusing?" primes the answer you're looking for.
- **Follow up on specifics** — a vague answer ("it was fine") is worth a genuine "can you walk me through exactly what happened?"

How Many Users Do You Actually Need? Nielsen's Real Formula

Jakob Nielsen's own widely-cited claim — that testing with just **5 users** finds around **85% of usability problems** — comes from a real formula, not a rule of thumb pulled from nowhere:

```
Problems found = N × (1 − L)ⁿ

N = total usability problems in the design
L = proportion of problems a single user finds (typically ≈31%)
n = number of test users
```

Plugging in `L ≈ 0.31` and `n = 5` gives roughly 85% of problems found — the source of the famous number.

> **THE REAL CAVEATS NIELSEN HIMSELF STATES**
>
> - This applies to **qualitative** usability testing — a **quantitative** study (measuring metrics like task completion time) needs roughly **20 users** instead, for statistical reliability.
> - If your product has **several genuinely distinct user groups**, the 5 users need to be split per group (e.g. 3-4 per group for two groups) — 5 users total across two very different audiences won't cover either one well.
> - Nielsen recommends **three separate 5-user studies**, testing → fixing → retesting each time, over one single 15-user study — a redesign introduces new problems a single big study never gets the chance to catch.

Surveys

A survey trades depth for breadth — closed-ended, structured questions sent to many users at once, well-suited to quantitative questions ("how satisfied are you, on a 1-5 scale") but poorly suited to discovering something you didn't already think to ask about. Interviews find the unknown unknowns; surveys measure how common a known issue actually is.

Personas: A Real, Corrected History

Personas — fictional, composite users built from real research, used to keep a design team focused on actual human needs rather than an abstract "the user" — are most associated with designer **Alan Cooper**, who began using prototype personas as early as 1983, built from informal interviews with 7-8 real users. The technique reached a wide audience through his 1999 book, *The Inmates Are Running the Asylum*.

A GENUINE, LESSER-KNOWN PRECURSOR

Cooper popularized personas specifically within software design, but he wasn't first to the underlying idea: marketer **Angus Jenkinson** developed a real, similar concept in 1993-94 — "day-in-the-life" customer archetypes called *CustomerPrints*, adopted internationally by an advertising agency before Cooper's own software-focused formalization. Two real, independent threads converging on the same underlying idea, in two different fields.

Example: "Maria, the Time-Pressed Parent"

A persona built from real research, not assumption

- **Goal:** Complete grocery ordering in under 5 minutes, from a phone, often one-handed
- **Frustration:** Re-entering a delivery address every single order
- **Quote (from a real interview):** "I don't need it to be pretty, I need it to be fast"

THE REAL PITFALL

A persona is only as good as the research behind it. A persona invented from a team's own internal assumptions, with no real interviews or data behind it, isn't a research artifact — it's a guess wearing a research artifact's clothing, and it can actively mislead a team more than having no persona at all.

Three Research Methods, Compared

METHOD	BEST FOR	REAL LIMITATION
Interviews	Discovering the unknown — real motivations, context, unexpected problems	Small sample, time-intensive, relies on self-report

METHOD	BEST FOR	REAL LIMITATION
Surveys	Measuring how common a known issue is, across many users	Can't discover something you didn't think to ask about
Usability Testing	Watching actual behavior with a real interface (say vs. do)	Needs a real or prototype interface to test against

Hands-On Exercises

EXERCISE 1

A product manager cites the "faster horses" Henry Ford quote to justify skipping user research entirely for a new feature. Explain, in your own words, what's wrong with both the quote itself and the conclusion being drawn from it.

View solution

EXERCISE 2

A team runs one single usability study with 15 participants and, separately, has three genuinely distinct types of users for their product. Explain, in your own words, two real ways this study design deviates from what Nielsen's own research actually recommends.

View solution

EXERCISE 3

A team wants to quickly build a persona in a single meeting, using only what everyone already assumes about their typical user. Explain, in your own words, why this wouldn't count as a genuine persona per this chapter, and what real step would fix it.

View solution

CHAPTER 2 QUICK REFERENCE

- The "faster horses" Ford quote is real, widely considered **apocryphal** — the genuine lesson is that stated preference (say) and real behavior (do) are different, complementary kinds of data, not that research is worthless
- Interviews ask about the **past**, avoid leading questions, follow up on vague answers
- Nielsen's real formula: `N × (1 − L)ⁿ`, giving ~85% of problems from 5 users — but only for **qualitative** studies, split per distinct user group, ideally as 3 rounds of 5 rather than one round of 15
- Surveys measure how common a *known* issue is; interviews discover *unknown* issues
- Personas: popularized by **Alan Cooper** (1999 book, prototype use since 1983), with a genuine independent 1993-94 precursor from Angus Jenkinson — only real if built from actual research, never from assumption alone

CHAPTER 3: INFORMATION ARCHITECTURE & NAVIGATION DESIGN

UI/UX Design Principles

Chapter 3 · Information Architecture & Navigation Design

Chapter 1 drew the line between UX and UI. Information architecture (IA) sits closer to the UX end of that line — it's the underlying structure and organization of a product's content, before a single button or color gets designed. Navigation is what a person actually uses to move through that structure. Get the architecture wrong, and no amount of good visual design in Chapter 5 will make the product easy to use.

A Real, Named Origin

Designer **Richard Saul Wurman** is credited with coining the term "information architecture" itself. His own book, *Information Architects*, helped establish it as a named discipline in the 1990s — a real, deliberate framing of organizing information as its own design problem, distinct from visual design or engineering.

Wurman's Real Organizing Principles

Several of Wurman's own real, named principles are directly useful for structuring a real product today:

Objects

Treat pieces of content as distinct, structured entities — not undifferentiated text on a page.

Choices

Offer meaningful navigation options — too few choices limits users, too many overwhelms them.

Disclosure

Reveal information progressively — enough to inform the next choice, not the entire structure at once.

Front Doors

Design multiple real entry points — most visitors won't arrive at your intended starting page.

Growth

Structure the system so it can genuinely expand later without a full redesign.

Navigation Patterns

PATTERN	STRUCTURE	BEST FOR
Hierarchical	A single tree — broad categories branching into narrower subcategories	Most content-heavy sites; matches how people already expect to browse
Hub-and-spoke	One central screen, each destination reached and returned to from there	Task-focused apps with genuinely separate, unrelated features
Sequential	A fixed, ordered path — step 1, then step 2, then step 3	Onboarding flows, checkout, anything with a real required order

The Real "Magical Number Seven" Correction

It's extremely common advice to keep a navigation menu to "7 items, plus or minus 2" — citing psychologist George Miller's famous 1956 paper. The real paper doesn't actually support that specific application.

WHAT MILLER ACTUALLY FOUND — AND SAID HIMSELF

Miller's 1956 paper covered two genuinely different, only coincidentally similar-looking findings: how many distinct levels people can reliably tell apart on a single scale (roughly 4-8), and how many items people can hold in short-term memory right after seeing them (roughly 7). Miller himself stated plainly that the correspondence between these two numbers was "only a coincidence," and that there is "nothing magical" about the number seven — he used the phrase rhetorically, not as a hard cognitive limit.

Later research complicates the number further: memory span genuinely varies by what's being remembered (roughly 7 digits, but closer to 5 words), and psychologist Nelson Cowan has proposed working memory capacity is closer to **4 chunks**, not 7. None of this research was ever about how many items a person can comfortably scan in a navigation menu — a completely different cognitive task from recalling a list from memory.

WHAT THIS ACTUALLY MEANS FOR NAVIGATION DESIGN

A navigation menu isn't a memory-recall task — a user reads the menu, they don't memorize it first. The real design guidance isn't "cap it at 7 items" at all; it's to group related items sensibly, label them

clearly, and let real usability testing (Chapter 2) — not a 1956 memory-span number — tell you whether a specific menu is actually too long.

The Three-Click Rule: A Real, Debunked Myth

A second widely-repeated rule — that users should be able to reach any content within **three clicks**, or they'll leave — was popularized by Jeffrey Zeldman's own 2001 book. Real usability research has since challenged it directly: studies have found the number of clicks needed affects **neither user satisfaction nor task success rate**. The Nielsen Norman Group has publicly labeled the rule false.

WHAT ACTUALLY MATTERS INSTEAD

What genuinely predicts whether a user finds what they need isn't the click count — it's whether each individual click is a **confident, correctly-informed one**. Five clicks where each choice is obvious beats two clicks where either one is a genuine guess. Rigidly enforcing "three clicks max" can actively hurt a real site — an e-commerce site forced into overly broad top-level categories to satisfy the rule makes browsing genuinely harder, not easier.

Card Sorting: Testing an IA With Real Users

Card sorting is a real, direct IA research method building on Chapter 2's own toolkit: real users are given cards labeled with real content topics and asked to group them in a way that makes sense to them (open card sort), or to sort them into categories you've already defined (closed card sort). The result is real evidence for how your actual users mentally organize the content — not a guess made in a planning meeting.

Hands-On Exercises

EXERCISE 1

A teammate insists a navigation menu must be trimmed to exactly 7 items because "that's the limit of human memory, per Miller's research." Explain, in your own words, what's wrong with this reasoning, citing what Miller's own paper actually found.

View solution

EXERCISE 2

An e-commerce team forces every product into one of only three top-level categories, specifically to satisfy the three-click rule, even though several products genuinely don't fit any of the three well. Explain, in your own words, what real research says about this tradeoff, and what should guide the decision instead.

View solution

EXERCISE 3

A team wants to figure out how their real users would naturally group 40 different content topics into categories, before designing any navigation menu. Explain, in your own words, which method from this chapter fits this need, and why it's more reliable than the team guessing themselves.

View solution

CHAPTER 3 QUICK REFERENCE

- **Information architecture** (term coined by Richard Saul Wurman) — the underlying structure of content, distinct from navigation and visual design
- Navigation patterns: **hierarchical** (most content), **hub-and-spoke** (separate features), **sequential** (a required order)
- Real correction: Miller's 1956 "7±2" finding was about memory recall, not menu scanning — Miller himself called the number "not magical," and it doesn't actually justify a hard 7-item menu cap
- Real correction: the three-click rule is debunked — click count doesn't predict satisfaction or success; what matters is whether each click is a confident, well-informed one
- **Card sorting** — a real IA research method for learning how actual users mentally group content

CHAPTER 4: INTERACTION DESIGN PRINCIPLES

UI/UX Design Principles

Chapter 4 · Interaction Design Principles

Chapter 1 mentioned affordances and feedback only in passing, as Don Norman's own real contribution to the field. This chapter goes deep on both, plus two genuinely useful, real predictive laws (Fitts's and Hick's) and the cognitive-science concept underneath all of it: cognitive load.

Affordances: A Real, Two-Step Correction

"Affordance" wasn't Norman's own term originally. Psychologist **James J. Gibson** introduced it in his 1966 book, developing it further in his 1979 work *The Ecological Approach to Visual Perception* — an affordance, to Gibson, is what an object objectively *offers* or *provides*, whether or not anyone notices it. A chair affords sitting whether a person has spotted it or not.

NORMAN'S 1988 ADAPTATION — NARROWER, ON PURPOSE

In *The Design of Everyday Things* (1988), Norman adapted Gibson's concept for design, deliberately narrowing it: a **perceived affordance** is an action possibility a person can actually, visibly recognize — not just an objective fact about the object. This shift made the concept genuinely useful for designers (you can design for what's perceivable), at the cost of no longer matching Gibson's own original, broader meaning.

A REAL, HONEST SELF-CORRECTION

Norman later acknowledged that this narrowing was unintended, and introduced a separate term, **"signifier,"** specifically for the visual cue that communicates a possible action (a button's own drop shadow, an underline on a link) — distinct from the affordance itself. A flat, un-styled clickable element can still technically afford clicking; without a real signifier, most users will never discover that it does.

Feedback

Where a signifier tells a user what's possible *before* an action, feedback confirms what actually happened *after* one — a button's own visible press state, a loading spinner, a confirmation message. Missing feedback is one of the most common real sources of user uncertainty: did that click actually register?

Fitts's Law: Predicting How Long a Click Takes

Psychologist Paul Fitts published a real, formal model in 1954 for how long it takes a person to point at a target, based on the target's distance and size:

```
ID = log₂(2D / W)          (Index of Difficulty)
MT = a + b·ID              (Movement Time)

D = distance to the target's center
W = the target's own width/tolerance
```

A larger, closer target is measurably faster to reach than a small, distant one. This real formula has direct, verifiable UI consequences: bigger buttons for frequent actions, related controls placed close together, and the real, named **"infinite edges"** effect — an element pinned to a screen edge has effectively infinite size in that one direction, since the cursor can't overshoot past the edge. **"Magic corners"** (where two edges meet) are effectively infinite in both directions — the real reason macOS anchors its menu bar to the very top of the screen, and why older Windows versions placed the Start button in a screen corner.

Hick's Law: Predicting Decision Time

William Hick (1952) and Ray Hyman found that decision time increases with the number of choices — but logarithmically, not linearly, so doubling the choices doesn't double the time:

```
T = b · log₂(n + 1)

T = decision time
n = number of choices
```

A REAL CAVEAT — DON'T APPLY IT BLINDLY

Hick's Law genuinely holds for unfamiliar or randomly-ordered choices. But real research found the relationship changes with familiarity and structure: searching a **randomly ordered** list scales *linearly*, not logarithmically, while an **alphabetically ordered** menu lets a user who already knows the item's

name find it in genuinely logarithmic time. The same underlying "don't misapply a cognitive-science number without checking the real conditions it was measured under" lesson from Chapter 3's own Miller correction applies here too.

Cognitive Load: The Resource Every Design Decision Spends

Educational psychologist John Sweller developed **Cognitive Load Theory** in the late 1980s, describing the real, limited mental effort available in working memory. It splits into three real, distinct types:

Intrinsic Load

The task's own inherent difficulty — fixed, not something a designer controls.

Extraneous Load

Unnecessary burden created by how information is presented — genuinely controlled by the designer.

Germane Load

Effort spent actually building understanding — the "good" load, freed up when extraneous load is low.

THE REAL JOB OF UI DESIGN

A designer can't reduce a task's own inherent difficulty (intrinsic load) — booking a genuinely complex multi-leg flight is inherently harder than checking a single balance. What a designer *can* control is extraneous load: confusing layout, unclear labels, inconsistent patterns, all real, avoidable tax on a user's own limited mental effort. Reducing extraneous load is, in a real sense, most of what good interaction design actually is.

Three Real Principles, Compared

PRINCIPLE	WHAT IT PREDICTS	DIRECT UI TAKEAWAY
Fitts's Law	Time to reach and click a target	Bigger, closer targets for frequent actions; use screen edges/corners
Hick's Law	Time to decide among choices	Fewer, well-organized choices — but check familiarity/ordering first
Cognitive Load Theory	Total mental effort a task consumes	Minimize extraneous load — the only one of the three a designer controls

Hands-On Exercises

EXERCISE 1

A teammate says "affordance and signifier mean the same thing — Norman uses them interchangeably." Explain, in your own words, why this is inaccurate, using the real distinction the chapter describes.

View solution

EXERCISE 2

A designer wants to place a "Save" button that will be clicked constantly throughout a long editing session. Using Fitts's Law, explain, in your own words, two real, concrete placement/sizing decisions that would make this button genuinely faster to use.

View solution

EXERCISE 3

A team cuts a genuinely necessary step out of a multi-step tax-filing flow, arguing it reduces the user's cognitive load. Explain, in your own words, which specific type of cognitive load this actually would (or wouldn't) reduce, and why removing a genuinely necessary step is a different move than reducing extraneous load.

View solution

CHAPTER 4 QUICK REFERENCE

- **Affordance** — coined by James Gibson (1966/1979); Norman narrowed it to "perceived" in 1988, then introduced **"signifier"** separately once he recognized the narrowing was unintended
- **Fitts's Law** (1954): $MT = a + b \cdot \log_2(2D/W)$ — bigger, closer targets are faster; screen edges/corners are effectively infinite-sized
- **Hick's Law** (1952): $T = b \cdot \log_2(n+1)$ — fewer choices decide faster, logarithmically, but familiarity/ordering changes the real relationship

- **Cognitive Load Theory** (Sweller) — intrinsic (fixed), extraneous (designer-controlled), germane (useful) load; good UI design minimizes extraneous load specifically

CHAPTER 5: VISUAL DESIGN FUNDAMENTALS

UI/UX Design Principles

Chapter 5 · Visual Design Fundamentals

Chapter 4 covered the cognitive resources every interaction spends. This chapter is about the concrete visual choices that either spend that resource wastefully (extraneous load) or spend it well — how grouping, contrast, alignment, and hierarchy let a person understand a screen at a glance rather than having to work it out.

Gestalt Principles: How the Brain Groups What It Sees

Gestalt psychology — founded in the early 20th century by **Max Wertheimer**, alongside Wolfgang Köhler and Kurt Koffka, with Wertheimer's real 1912 publication on the "phi phenomenon" marking its formal start — studies how the mind organizes visual elements into a coherent whole, rather than perceiving them as separate, disconnected pieces. Five of its real principles are directly, constantly at work in UI design:

Proximity

Elements placed close together are perceived as one group — the real reason related form fields need real, deliberate spacing.

Similarity

Elements sharing shape, color, or size are perceived as related — the real reason every primary button on a site should look the same.

Closure

The mind perceives an incomplete shape as whole — the real reason a partial icon or outline still reads clearly.

Continuity

Elements aligned along a line or curve are perceived as connected — the real reason a straight edge of aligned text feels orderly.

Figure-Ground

The eye separates a foreground "figure" from its background — the real reason a modal needs real visual separation from the page behind it.

Contrast, Repetition, Alignment, Proximity

Designer Robin Williams' own influential book, *The Non-Designer's Design Book* (1994), distilled visual design down to four real, practical principles — often remembered by the acronym CRAP (or reordered as CARP):

PRINCIPLE	WHAT IT MEANS	WHAT HAPPENS WITHOUT IT
Contrast	Make genuinely different elements look genuinely different — size, weight, color	Everything looks equally important, so nothing actually stands out
Repetition	Repeat visual choices consistently across a design — same button style, same spacing rhythm	A product feels disjointed, like several different designs stitched together
Alignment	Every element lines up with something else on the page, not placed arbitrarily	A page feels visually messy even if the content itself is fine
Proximity	Group related items close together, separate unrelated ones — the same Gestalt principle above, applied deliberately	Users can't tell which label belongs to which field

WHERE "HIERARCHY" FITS

Visual hierarchy — the sense of what to look at first, second, third — isn't a fifth, separate principle. It's what genuinely applying contrast, repetition, alignment, and proximity *together* actually produces. A page with strong contrast on its heading, consistent repetition of its own patterns, clean alignment, and correct proximity grouping will already read with a clear hierarchy, without anything extra being added.

Visual Hierarchy in Practice: The Real F-Pattern

Nielsen Norman Group's real 2006 eye-tracking study, covering 232 users across thousands of web pages, found a consistent scanning shape resembling the letter **F**: a horizontal scan across the top of the content, a second, usually shorter horizontal scan further down, then a vertical scan down the left edge. A real follow-up study 11 years later, in 2017, confirmed the same pattern still held.

THE REAL, DIRECT DESIGN IMPLICATION

Because users scan rather than read every word, the words genuinely most likely to be seen are the ones at the **start** of a heading, subheading, or bullet point, and the content in the first couple of paragraphs. Front-loading the most important word or phrase in a heading (rather than burying it at the end of a longer sentence) is a real, direct, evidence-backed design decision — not a stylistic preference.

Hands-On Exercises

EXERCISE 1

A form places a field's label 40 pixels above the input box, but the next field's label sits only 8 pixels below that same input box. Explain, in your own words, which Gestalt principle is being violated here, and what a user is likely to misunderstand as a result.

View solution

EXERCISE 2

A page uses five different button styles across its various screens, each with different colors, corner radii, and sizes for what is functionally the same "primary action" button. Explain, in your own words, which of the four CRAP/CARP principles is missing here, and the real consequence for the user.

View solution

EXERCISE 3

A team writes a page heading as "Learn About Our Company's 20-Year History and Mission" and wants it to be quickly understood by someone scanning the page rather than reading closely. Using the real F-pattern finding, explain, in your own words, what's wrong with this heading's own word order, and how you'd rewrite it.

View solution

CHAPTER 5 QUICK REFERENCE

- **Gestalt principles** (Wertheimer, Köhler, Koffka, from 1912): proximity, similarity, closure, continuity, figure-ground — how the mind groups visual elements
- **CRAP/CARP** (Robin Williams, 1994): Contrast, Repetition, Alignment, Proximity — four practical principles that, applied together, produce visual hierarchy
- Visual hierarchy isn't a separate, fifth principle — it's the real result of applying the other four well

- Real, confirmed finding (NN/g, 2006, reconfirmed 2017): users scan pages in an **F-shaped pattern** — front-load important words in headings and bullets

CHAPTER 6: COLOR THEORY & TYPOGRAPHY FOR INTERFACES

UI/UX Design Principles

Chapter 6 · Color Theory & Typography for Interfaces

Chapter 5 covered how grouping, contrast, and alignment create structure. This chapter covers the two specific materials that structure gets built from on screen: color and type.

Color: The Accessibility Numbers That Actually Matter

Beyond aesthetics, color choice on an interface has real, testable, legally-referenced accessibility requirements — this course's own dedicated Web Accessibility course goes far deeper, but the core numbers belong here too, since they directly shape color decisions.

REAL WCAG CONTRAST REQUIREMENTS (LEVEL AA)

- **Normal text:** a contrast ratio of at least **4.5:1** against its background
- **Large text** (18pt/~24px or larger, or 14pt/~18.5px bold): at least **3:1**

The stricter Level AAA standard raises the bar further — 7:1 for normal text. These aren't rounded either: the real standard states 4.499:1 does not meet a 4.5:1 requirement.

COLOR BLINDNESS: REAL, VERIFIED PREVALENCE

Congenital red-green color blindness — the most common form — affects roughly **1 in 12 men (about 8%)** and **1 in 200 women (about 0.5%)**, a real, documented gap caused by the relevant genes sitting on the X chromosome. The direct design rule this produces: never use color as the *only* way to convey information (a red vs. green status dot with no label or icon is genuinely unreadable to a meaningful share of real users).

Color Harmony, Briefly

A few real, standard color-relationship patterns from color theory show up constantly in interface palettes: **complementary** colors (opposite each other on the color wheel, for strong contrast), **analogous** colors (next to each other, for a harmonious, low-contrast feel), and **triadic** colors (evenly spaced, for a balanced but vibrant palette). Screens use additive color

(RGB/hex) rather than print's subtractive CMYK — worth knowing if you ever hand a design off to a print vendor.

Typography: A Real Correction to a Common Claim

> **"SANS-SERIF IS ALWAYS MORE LEGIBLE ON SCREENS" — OVERSTATED**
>
> It's extremely common to hear that sans-serif fonts are simply more legible on screens than serif fonts. The real research is far less conclusive: one study found **serif** typefaces were actually more legible on screen, even though users didn't prefer them; another found sans-serif slightly faster for individual word comprehension. Typography researcher Alex Poole's own real conclusion: "most reasonably designed typefaces in mainstream use will be equally legible."

The real, practical reason sans-serif dominates online isn't necessarily inherent legibility — it's rendering. At small pixel sizes (roughly 9-20px), a serif's own fine details render poorly on lower-resolution displays, which made sans-serif the safer practical default for a long time. Modern high-density screens have significantly reduced that specific constraint, which is exactly why serif body text has become genuinely more common in web design again.

Real Typographic Fundamentals

TERM	WHAT IT MEANS	PRACTICAL GUIDANCE
Line length (measure)	How many characters fit on one line of text	Roughly 50-75 characters per line for comfortable reading — much longer and the eye loses its place returning to the next line
Line height (leading)	Vertical space between lines of text	Roughly 1.4-1.6× the font size for body text is a common, comfortable starting point
x-height	The height of a lowercase letter like "x", without ascenders/descenders	A larger x-height generally reads more clearly at small sizes on screen

Hands-On Exercises

EXERCISE 1

A designer sets normal body text at a contrast ratio of 4.3:1 against its background and argues this is "close enough" to WCAG's requirement. Explain, in your own words, whether this passes Level AA, citing the real, exact standard.

View solution

EXERCISE 2

A dashboard shows server status using only a red or green dot, with no label, icon, or text difference between the two states. Explain, in your own words, why this is a real, measurable accessibility problem, citing this chapter's own prevalence data.

View solution

EXERCISE 3

A colleague insists a design must switch from a serif font to a sans-serif font because "serif fonts simply aren't legible on screens, that's an established fact." Explain, in your own words, what this chapter's own research actually shows, and what the real, more accurate way to phrase the tradeoff would be.

View solution

CHAPTER 6 QUICK REFERENCE

- Real WCAG Level AA contrast: **4.5:1** normal text, **3:1** large text (18pt/~24px, or 14pt/~18.5px bold) — not rounded
- Real color blindness prevalence: ~8% of men, ~0.5% of women (red-green, the most common form) — never convey information with color alone
- Complementary / analogous / triadic — standard color-wheel relationship patterns; screens use RGB, print uses CMYK
- Real correction: "sans-serif is always more legible on screens" is overstated — research is mixed, and the real historical driver was low-resolution rendering, not inherent legibility

- Practical type guidance: ~50-75 characters per line, ~1.4-1.6× line height, larger x-height reads better small

CHAPTER 7: USABILITY HEURISTICS

UI/UX Design Principles

Chapter 7 · Usability Heuristics

Chapter 2 covered testing with real users. This chapter covers a real, different, and complementary method: expert review against a fixed, well-established checklist — Nielsen's own 10 usability heuristics.

A Real, Dated History

Jakob Nielsen and Rolf Molich originally developed a set of usability heuristics in **1990**. Nielsen refined them into their current, final form in **1994**, via a real factor analysis of 249 actual, documented usability problems. The resulting 10 heuristics have remained unchanged since 1994 — a genuinely long track record for a design-research artifact.

The 10 Heuristics

1

Visibility of System Status

Keep users informed about what's happening, with timely feedback.

2

Match Between System & Real World

Use familiar language and real-world conventions, not internal jargon.

3

User Control & Freedom

A clearly marked exit for unwanted actions — real undo and redo.

4

Consistency & Standards

Don't make users wonder whether different words mean the same thing.

5

Error Prevention

Design out error-prone conditions rather than relying on error messages.

6

Recognition Rather Than Recall

Make options visible — don't force users to remember information.

7

Flexibility & Efficiency of Use

Accelerators for expert users, without punishing novices.

8

Aesthetic & Minimalist Design

Every irrelevant unit of information competes with the relevant ones.

9

Help Users Recognize & Recover From Errors

Plain-language errors that state the problem and suggest a fix.

10

Help & Documentation

Searchable, task-focused help, available in context when needed.

Heuristic Evaluation as a Method

A **heuristic evaluation** is the real, structured process of one or more expert reviewers independently assessing an interface against these 10 heuristics, flagging every real violation they find. Nielsen's own real, specific recommendation: **three to five independent evaluators**, since any single reviewer — no matter how experienced — reliably misses a real share of the issues another evaluator would catch.

> **TWO DIFFERENT "5"S — DON'T CONFUSE THEM**
>
> Chapter 2 covered a real, separate finding: testing with **5 real users** finds around 85% of usability problems. This chapter's **3-5 expert evaluators** is a genuinely different number, measuring a genuinely different thing — expert judgment against a fixed checklist, versus real people actually attempting real tasks. It's an easy, understandable mistake to conflate the two just because both numbers cluster around 5, but heuristic evaluation cannot replace real user research — each method catches real problems the other one structurally can't.

Two Methods, Compared

METHOD	WHO'S INVOLVED	WHAT IT'S GOOD AT	WHAT IT STRUCTURALLY MISSES
Heuristic Evaluation	3-5 UX experts, independently	Fast, cheap, catches well-known usability patterns early	Real user context, motivation, and genuinely novel problems no heuristic anticipates
Usability Testing (Ch. 2)	Real users, attempting real tasks	Real behavior, real confusion, real task success/failure	Slower to run; needs a real or prototype interface to test against

Hands-On Exercises

EXERCISE 1

An interface shows the exact same word ("Delete") for two functionally different actions on two different screens — one permanently removes data, the other only hides it temporarily. Explain, in your own words, which of Nielsen's 10 heuristics this violates, and why.

View solution

EXERCISE 2

A manager says "we don't need to bother with real user testing — we already had one senior designer do a heuristic evaluation and they're very experienced." Explain, in your own words, why this reasoning doesn't hold up, using this chapter's own real recommendations.

View solution

EXERCISE 3

A colleague says "testing with 5 users and doing a heuristic evaluation with 5 experts are basically the same thing, since they use the same number." Explain, in your own words, why this is incorrect, citing the real, different thing each method actually measures.

View solution

CHAPTER 7 QUICK REFERENCE

- Nielsen & Molich's real 1990 heuristics, refined by Nielsen in 1994 via factor analysis of 249 real usability problems — unchanged since
- 10 heuristics: system status, real-world match, user control, consistency, error prevention, recognition over recall, flexibility, minimalism, error recovery, help/documentation
- **Heuristic evaluation** — 3-5 independent expert evaluators reviewing against the heuristics, since any one evaluator misses real issues
- Real, important distinction: heuristic evaluation's "3-5 evaluators" and Chapter 2's "5 test users" are different concepts measuring different things — heuristic evaluation can't replace real user research

CHAPTER 8: WIREFRAMING & PROTOTYPING

UI/UX Design Principles

Chapter 8 · Wireframing & Prototyping

Chapter 1's Design Thinking stages named "Prototype" only briefly. This chapter covers what actually goes into one — and the real, deliberate spectrum of how rough or polished it should be at each stage of a project.

Three Real, Distinct Terms

These three words get used loosely and interchangeably in casual conversation, but they describe genuinely different real artifacts:

TERM	WHAT IT ACTUALLY IS	INTERACTIVE?
Wireframe	A structural skeleton — boxes, placeholder text, layout only, no real visual design	No
Mockup	A static, visually finished design — real color, type, imagery — but not clickable	No
Prototype	A version (rough or polished) that simulates a real flow — clickable, navigable	Yes

A prototype can be built at any fidelity level — a clickable paper sketch is a real, low-fidelity prototype; a fully interactive, pixel-perfect Figma file is a real, high-fidelity one. Fidelity and interactivity are two separate, independent decisions.

The Fidelity Spectrum

Low fidelity — rough, quick, deliberately unpolished (hand sketches, boxes-and-labels wireframes). **High fidelity** — close to (or identical to) the finished product's real visual design, spacing, and interaction detail. Neither end is universally "better" — each fits a different real stage of a project.

> A REAL, PRACTICAL REASON TO START LOW-FIDELITY

A rough sketch represents genuinely less time and effort invested — which makes it genuinely cheaper and faster to throw away and redo once real feedback reveals a problem. Design practitioners commonly argue this cuts both ways in practice: a visibly rough sketch also tends to invite more open, unguarded critique from a viewer than something that already looks close to finished — though this specific psychological effect is closer to well-established design-practice wisdom than a single, citable research study, worth stating honestly rather than as a hard, proven fact.

Paper Prototyping: A Real, Named Method

Carolyn Snyder's own real 2003 book, *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*, established a genuinely standard method: hand-sketched screens on real paper, tested with real users, a facilitator physically swapping sheets in and out to simulate the interface responding to clicks.

REAL, DOCUMENTED LIMITATIONS

- Participants need real imagination to interpret an interaction a piece of paper can't literally show
- A facilitator's own judgment in swapping screens introduces a genuine risk of bias
- It requires real in-person participation, limiting how geographically diverse the test group can be

Choosing the Right Fidelity for the Stage

A PRACTICAL RULE OF THUMB

Use low fidelity when the real question is still "*is this the right structure/flow at all?*" — changes are still cheap. Move to high fidelity once the structure is validated and the real question shifts to "*does this specific visual/interaction design work?*" — details that only a polished version can actually test (real contrast, real spacing, real typography from Chapters 5-6).

Hands-On Exercises

EXERCISE 1

A teammate shows you a fully interactive, pixel-perfect Figma prototype and calls it a "mockup." Explain, in your own words, why this terminology is inaccurate, using this chapter's own real definitions.

View solution

EXERCISE 2

A team is still deciding between two genuinely different navigation structures for a new app, early in the project. Explain, in your own words, why building a high-fidelity, pixel-perfect prototype at this stage would be a real, avoidable waste of effort.

View solution

EXERCISE 3

A team runs a paper-prototype usability session with participants spread across five different countries, conducted entirely by video call with no facilitator physically present with each participant. Explain, in your own words, which real, documented limitation of paper prototyping this setup runs into.

View solution

CHAPTER 8 QUICK REFERENCE

- **Wireframe** (structural, no visual design) vs. **mockup** (visual, static) vs. **prototype** (interactive, any fidelity level) — three genuinely different artifacts
- Fidelity and interactivity are independent — a clickable paper sketch is a real, low-fidelity prototype
- **Paper prototyping** (Carolyn Snyder, 2003) — real, documented limitations: requires imagination, facilitator-bias risk, in-person only
- Rule of thumb: low fidelity while the structure itself is still in question; high fidelity once testing the specific visual/interaction design

CHAPTER 9: USABILITY TESTING & ITERATIVE DESIGN

UI/UX Design Principles

Chapter 9 · Usability Testing & Iterative Design

Chapter 2 introduced research methods; Chapter 7 covered expert-based heuristic evaluation. This chapter goes deep on the real thing both of those set up: watching a real person actually attempt real tasks with the prototype from Chapter 8, then feeding what you learn back into another round of design.

Moderated vs. Unmoderated Testing

TYPE	HOW IT WORKS	REAL TRADEOFF
Moderated	A facilitator guides the session live, asking follow-up questions in real time	Richer, deeper insight; slower and more resource-intensive to run
Unmoderated	A participant completes tasks alone, often via a remote testing tool	Faster, cheaper, more participants; no chance to ask a real-time follow-up

The Think-Aloud Protocol

Asking a participant to verbalize their thoughts continuously while using the interface — "I'm looking for a way to change my email... I don't see it, so I'll try this menu" — surfaces the real reasoning behind a click, not just the click itself. It's the single most common technique used inside a moderated usability test.

Real Task Metrics

- **Completion rate** — the real percentage of participants who finished the task successfully
- **Time on task** — how long it genuinely took, compared against a reasonable benchmark
- **Error rate** — how many real mistakes were made along the way, even in a task eventually completed

A Real Correction: Does Being Watched Change Behavior?

It's extremely common to hear that recording or observing a usability test participant meaningfully distorts their real behavior — often citing the **Hawthorne effect**, the idea that people change behavior simply because they know they're being observed.

THE REAL STUDIES BEHIND THE POPULAR CLAIM

The original "Hawthorne effect" studies (Western Electric's Hawthorne Works, 1924-1932) are now considered, by later analysis, genuinely weaker evidence than the popular version of the story suggests. Economists Levitt and List re-examined the real data and found only "slight evidence for the Hawthorne effect... in no way as drastic as suggested initially." Critic Richard Nisbett went further, calling the whole effect "a glorified anecdote." One of the original studies was itself described as "methodologically poor, uncontrolled" — no firm conclusion should ever have been drawn from it in the first place.

WHAT THIS MEANS FOR USABILITY TESTING

The underlying concern — that awareness of observation can shape behavior somewhat — is still worth being mindful of. But treating it as a large, well-proven, near-universal distortion (the popular version of the Hawthorne effect) overstates what the real evidence actually supports. A moderated, observed usability test is still a genuinely valuable, real source of evidence, not a fundamentally compromised one.

The Iterative Design Loop

A single test round is rarely the end of the process — the same non-linear spirit from Chapter 1's own Design Thinking stages applies directly here:

1. **Test** — Run the session with the current prototype (Chapter 8's own fidelity level).
2. **Learn** — Identify real, specific problems from what was observed.
3. **Redesign** — Change the design to address what was actually learned.
4. **Retest** — Confirm the fix actually worked — and didn't introduce a new problem.

This loop is exactly why Nielsen's own real research (Chapter 2) recommends three separate 5-user rounds over one large 15-user study — the loop only works if there's a real retest after each redesign.

Hands-On Exercises

EXERCISE 1

A team needs deep, real-time follow-up on why participants hesitate at a specific step, but also needs to test with 50 participants within one week. Explain, in your own words, why this creates a real tension between moderated and unmoderated testing, and how the team might resolve it.

View solution

EXERCISE 2

A stakeholder argues that recording usability test sessions should be avoided entirely, since "the Hawthorne effect proves people behave completely differently on camera." Explain, in your own words, why this overstates what the real research on the Hawthorne effect actually supports.

View solution

EXERCISE 3

A team redesigns a checkout flow based on one usability testing round, ships the change immediately, and moves on without testing again. Explain, in your own words, which real step of the iterative design loop this skips, and what real risk that creates.

View solution

CHAPTER 9 QUICK REFERENCE

- **Moderated** (live, deep, slow) vs. **unmoderated** (fast, scalable, no real-time follow-up) testing
- **Think-aloud protocol** — verbalizing reasoning while completing a task, surfacing the "why" behind a click
- Real task metrics: completion rate, time on task, error rate
- Real correction: the original Hawthorne effect studies are now considered weak evidence by later analysis (Levitt & List; Nisbett's "glorified anecdote") — being observed is worth being mindful of, but isn't the large, proven distortion the popular version claims

- The iterative loop: **Test** → **Learn** → **Redesign** → **Retest** — skipping the retest defeats the point of the loop

CHAPTER 10: CAPSTONE — DESIGNING, PROTOTYPING & TESTING A SMALL APP FEATURE

UI/UX Design Principles

Chapter 10 · Capstone: Designing, Prototyping & Testing a Small App Feature

This capstone runs one small, concrete feature — **"Add a Task with a Due Date"** for a fictional to-do app, TidyTask — through every prior chapter's own real tool, in the same order a real project would actually use them.

STEP 1 — CHAPTER 1: DESIGN THINKING

Framing the Problem

Empathize/Define first, before anything gets built: TidyTask's own support inbox shows a recurring real complaint — users forget to add a due date when creating a task, then miss it entirely. Define stage: "Users need due-date entry to be fast enough that they don't skip it when adding a task quickly." Ideate, Prototype, and Test come next — and per Chapter 1's own real finding, this loop stays non-linear: if testing later reveals this problem statement was wrong, the process returns here.

STEP 2 — CHAPTER 2: USER RESEARCH

A Real, Grounded Persona

A short round of user interviews (asking about the *last time* someone forgot a due date, not a hypothetical) surfaces a real pattern: most users add tasks in a rush, between other activities. A persona built from that real data — "Sam, the Between-Meetings Task Adder" — goal: add a task with a due date in under 10 seconds, one-handed, on a phone.

STEP 3 — CHAPTER 3: INFORMATION ARCHITECTURE

Where the Feature Lives

TidyTask uses hierarchical navigation. The "Add Task" entry point needs to be a real front door reachable from the main task list directly — not buried two levels down, since Sam's own real goal is speed. No click-count target is enforced (per Chapter 3's own three-click-rule correction) — the entry point's own clarity matters more than counting clicks.

STEP 4 — CHAPTER 4: INTERACTION DESIGN

Applying Fitts's and Hick's Law

The "Add Task" button is placed in the bottom-right corner of the screen (Fitts's Law's own "magic corner" — effectively infinite target size, reachable one-handed with a thumb). The due-date picker offers a small, curated set of real choices — Today, Tomorrow, This Weekend, Pick a Date — rather than an open calendar by default, applying Hick's Law's real logarithmic relationship to keep decision time low for the common cases.

STEP 5 — CHAPTER 5: VISUAL DESIGN

Gestalt Grouping & CRAP

The task-text field and its due-date picker sit close together (proximity) and share consistent input-field styling (similarity), signaling they belong to the same task. The "Add" button gets real contrast against the surrounding form (Contrast); every date-choice chip repeats the exact same shape and spacing (Repetition); every element aligns to one shared left edge (Alignment).

STEP 6 — CHAPTER 6: COLOR & TYPOGRAPHY

Accessibility Checked, Not Assumed

The "overdue" task label is checked against WCAG's real 4.5:1 contrast requirement before shipping — measured, not eyeballed. Overdue status is shown with both a red color *and* a text label ("Overdue"), never color alone, per Chapter 6's own real color-blindness finding. Task text uses a real ~50-65 character line length on mobile for comfortable scanning.

STEP 7 — CHAPTER 7: HEURISTIC EVALUATION

A Quick Expert Pass

Before real user testing, 3 team members independently review the wireframe against Nielsen's 10 heuristics. One real finding: the due-date picker gives no visible confirmation

once a date is chosen (a real **Visibility of System Status** violation) — fixed by highlighting the selected chip before this ever reaches a real test participant.

STEP 8 — CHAPTER 8: WIREFRAMING & PROTOTYPING

Low-Fi First, Then High-Fi

A rough, low-fidelity wireframe (boxes and labels) tests whether the overall structure and flow make sense first, since the structure is still the open question at this stage (Chapter 8's own real rule of thumb). Once validated, a high-fidelity, interactive prototype is built — real spacing, real type, real color from Steps 5-6 — genuinely clickable, ready for a real test.

STEP 9 — CHAPTER 9: USABILITY TESTING & ITERATION

Test → Learn → Redesign → Retest

A moderated, think-aloud session with 5 real participants (Chapter 2's own real formula) finds a genuine problem: 2 of 5 participants didn't notice the "Pick a Date" option existed at all, defaulting to "Today" even when that wasn't accurate. **Learn:** the custom date option needs a real, distinct visual signifier. **Redesign:** add a small calendar icon to that one chip specifically. **Retest:** a second round of 5 confirms the fix — all 5 now find it, with completion rate rising from 60% to 100%.

Chapter Attribution

STEP	CHAPTER	WHAT IT CONTRIBUTED
1. Framing	Ch. 1	Design Thinking's own non-linear Empathize→Define structure
2. Research	Ch. 2	A real, interview-grounded persona (Sam)
3. Structure	Ch. 3	Front-door placement, no arbitrary click-count target
4. Interaction	Ch. 4	Fitts's Law placement, Hick's Law choice-limiting
5. Visual grouping	Ch. 5	Gestalt proximity/similarity, CRAP contrast/repetition/alignment
6. Color/type	Ch. 6	Measured WCAG contrast, never color-alone status, real line length

STEP	CHAPTER	WHAT IT CONTRIBUTED
7. Expert review	Ch. 7	A real heuristic violation caught before user testing
8. Prototyping	Ch. 8	Low-fi structure test, then a real high-fi interactive build
9. Testing	Ch. 9	A real test-learn-redesign-retest cycle with measured improvement

What This Course Doesn't Cover

- Motion and animation design
- Design systems and component libraries at organizational scale
- Deep, platform-specific design guidelines (iOS Human Interface Guidelines, Material Design in full depth)
- Enterprise/B2B-specific UX patterns
- Accessibility beyond the core numbers covered in Chapter 6 — this site's own dedicated Web Accessibility (a11y) course goes far deeper

Hands-On Exercises

EXERCISE 1

A teammate suggests skipping the low-fidelity wireframe in Step 8 and going straight to a high-fidelity prototype, to "save time." Explain, in your own words, using this capstone's own Step 8 and Step 9 findings, why this is a real, risky shortcut.

View solution

EXERCISE 2

Someone proposes removing the retest in Step 9, arguing "we already know what the fix should be, so testing it again is unnecessary." Using this capstone's own real completion-rate numbers, explain, in your own words, why the retest was worth running.

View solution

EXERCISE 3

Pick one real principle or law from any single chapter of this course (other than Chapter 1) and explain, in your own words, how it was concretely applied somewhere in this capstone — not just restating the principle, but pointing to the specific design decision it produced.

View solution

COURSE COMPLETE

- All 9 prior chapters applied in sequence to one real, small feature — research through iterative testing
- A real, measured improvement: completion rate rose from 60% to 100% after one redesign-and-retest cycle
- **UI/UX Design Principles is now complete — 10/10 chapters.**