

PRODUCT DESIGN FROM ZERO

End-to-End Product Design Checklist - For a new product, major new product line, or true zero-to-one initiative

How to use this checklist: This is a reference framework, not a rigid waterfall process. Activities can overlap, repeat, or be skipped when evidence shows they are unnecessary. The principle is to make decisions deliberately and preserve the chain from evidence to product outcomes.

0. Product Kickoff & Strategic Context

Understand the business context

- ☐ Clarify why the product should exist and what triggered the initiative.
- ☐ Understand the company, market, business model, strategic goals, and constraints.
- ☐ Identify key stakeholders, decision-makers, subject-matter experts, and delivery partners.
- ☐ Clarify expected business outcomes without treating proposed features as validated requirements.
- ☐ Capture known constraints: budget, timeline, regulation, technology, geography, operations, and dependencies.

Separate facts from assumptions

- ☐ Document what is already known and supported by evidence.
- ☐ Document stakeholder assumptions, beliefs, proposed solutions, and open questions.
- ☐ Create an initial assumptions / risks log.
- ☐ Identify the highest-risk assumptions that could invalidate the product direction.

Define initial success direction

- ☐ Clarify what success could mean for users and for the business.
- ☐ Identify candidate outcome metrics, while keeping them provisional until research provides context.
- ☐ Agree on how design, product, engineering, data, and business will collaborate.

1. Discovery & Research Planning

Frame the discovery space

- ☐ Write an initial problem-space statement without prematurely defining the solution.
- ☐ Identify target markets, contexts, and possible user groups.
- ☐ Map existing knowledge gaps.
- ☐ Define research objectives and research questions.

Plan the research

- ☐ Choose appropriate qualitative and quantitative methods for each research question.
- ☐ Define participant criteria and a recruitment plan.
- ☐ Create an interview / observation / survey plan as appropriate.
- ☐ Define sample rationale rather than choosing participant counts arbitrarily.
- ☐ Prepare consent, privacy, recording, and data-handling procedures where relevant.
- ☐ Plan how research notes, recordings, evidence, and insights will be stored.

Secondary research

- ☐ Review market and industry context.
- ☐ Analyze direct, indirect, and substitute competitors.
- ☐ Review relevant regulations, standards, trends, and domain constraints.
- ☐ Identify existing behavioral or market data that can inform primary research.

2. Primary User Research & Data Gathering

Conduct research

- ☐ Run user interviews focused on behaviors, goals, context, workarounds, and past experiences.
- ☐ Avoid leading questions and solution validation disguised as discovery.
- ☐ Observe real workflows or contexts where possible.
- ☐ Capture quotes, behaviors, triggers, barriers, motivations, and contradictions.
- ☐ Collect quantitative evidence when it can answer scale, frequency, or behavioral questions.

Research quality checks

- ☐ Separate what participants say from what they actually do.
- ☐ Look for disconfirming evidence, not only evidence supporting the initial idea.
- ☐ Track emerging patterns without stopping research too early.
- ☐ Record researcher assumptions and possible biases.

3. Synthesis & User Understanding

Organize evidence

- ☐ Clean and structure research notes.
- ☐ Code or tag evidence consistently.
- ☐ Use affinity mapping or another synthesis method to identify patterns.
- ☐ Distinguish observations, interpretations, insights, and recommendations.

Model user needs

- ☐ Identify goals, needs, pain points, motivations, behaviors, triggers, and constraints.
- ☐ Identify meaningful behavioral segments.
- ☐ Create personas only when evidence supports stable, useful segments.
- ☐ Use Jobs To Be Done when it better explains motivation and context.
- ☐ Map the current journey / workflow and critical moments.
- ☐ Identify unmet needs, workarounds, breakdowns, and opportunity areas.

4. Problem Framing & Opportunity Definition

Frame the right problems

- ☐ Separate signals, symptoms, causes, problems, and proposed solutions.
- ☐ Write evidence-based user problem statements.
- ☐ Define the corresponding business problems.
- ☐ Identify root causes where evidence is sufficient.
- ☐ Map assumptions that still require validation.

Define opportunities

- ☐ Translate research insights into opportunity statements.
- ☐ Create How Might We questions where useful.
- ☐ Map opportunities to user and business outcomes.
- ☐ Prioritize opportunities using evidence, impact, risk, strategic fit, and feasibility.
- ☐ Define which problem(s) the first product version should solve - and which it should not.

5. Product Definition & Experience Strategy

Define product direction

- ☐ Clarify the product value proposition.
- ☐ Define primary users, core use cases, and key jobs.
- ☐ Define product principles to guide future decisions.
- ☐ Define the minimum valuable experience rather than simply the minimum number of features.
- ☐ Clarify scope, non-goals, dependencies, and major risks.

Define success

- ☐ Create a measurement framework linking user outcomes to business outcomes.
- ☐ Choose primary, secondary, and guardrail metrics.
- ☐ Define baseline needs and how metrics will be instrumented after launch.
- ☐ Identify qualitative signals that will complement quantitative metrics.

6. Ideation & Concept Exploration

Explore before converging

- ☐ Generate multiple solution directions for prioritized opportunities.
- ☐ Use sketches, workshops, competitive patterns, and analogies to broaden the solution space.
- ☐ Challenge obvious solutions and feature requests.
- ☐ Consider operational, policy, content, and service solutions - not only interface changes.

Evaluate concepts

- ☐ Assess desirability, usability, feasibility, viability, risk, and strategic fit.
- ☐ Identify assumptions behind each concept.
- ☐ Prototype the riskiest assumptions at the lowest useful fidelity.
- ☐ Select concepts based on evidence and explicit trade-offs.

7. Information Architecture & Core Flows

Structure the product

- ☐ Define the product's conceptual model.
- ☐ Create the information architecture and navigation model.
- ☐ Define entities, relationships, hierarchy, and terminology.
- ☐ Validate labels and grouping when ambiguity is high.

Design end-to-end flows

- ☐ Map primary user flows and task flows.
- ☐ Include alternate paths, permissions, empty states, error states, edge cases, and recovery paths.
- ☐ Consider cross-device / cross-channel behavior where relevant.
- ☐ Review flows with product, engineering, and domain stakeholders before polishing UI.

8. Wireframing & Interaction Design

Design progressively

- ☐ Start with low-fidelity wireframes focused on structure and behavior.
- ☐ Define information hierarchy, actions, states, feedback, and progressive disclosure.
- ☐ Apply established interaction patterns intentionally.
- ☐ Review accessibility and inclusive-design needs early.
- ☐ Iterate before investing in visual polish.

Design system thinking

- ☐ Identify reusable patterns and components.

- ☐ Use or establish foundations for typography, spacing, color, layout, and states.
- ☐ Define component behavior, variants, and responsive rules.
- ☐ Document product-specific patterns that should scale.

9. Prototyping & Usability Validation

Prepare validation

- ☐ Define what must be learned from the test.
- ☐ Choose representative participants and realistic tasks.
- ☐ Create a prototype with only the fidelity needed to answer the research questions.
- ☐ Define success criteria and observation notes before testing.

Run usability testing

- ☐ Moderate without teaching the interface.
- ☐ Observe task success, errors, hesitation, comprehension, confidence, and recovery.
- ☐ Capture qualitative evidence and relevant usability measures.
- ☐ Identify issue severity and frequency.
- ☐ Distinguish usability problems from preference feedback.

Iterate

- ☐ Prioritize findings by impact, severity, evidence, and product risk.
- ☐ Revise flows and interface decisions.
- ☐ Retest critical changes when risk justifies it.
- ☐ Document what changed and why.

10. High-Fidelity Design & Accessibility

Finalize the experience

- ☐ Apply visual hierarchy, typography, spacing, color, imagery, and brand appropriately.
- ☐ Design all relevant states: default, hover, focus, selected, loading, success, warning, error, disabled, empty.
- ☐ Design responsive behavior and platform-specific adaptations.
- ☐ Review content and microcopy for clarity and consistency.

Accessibility

- ☐ Check contrast, focus order, keyboard behavior, target sizes, labels, errors, and status communication.
- ☐ Consider screen-reader semantics with engineering.
- ☐ Review against relevant WCAG requirements.
- ☐ Include accessibility in acceptance criteria rather than treating it as a final audit.

11. Delivery & Engineering Collaboration

Prepare for implementation

- ☐ Review technical feasibility before final handoff.
- ☐ Provide clear flows, states, responsive rules, components, content, and interaction specifications.
- ☐ Clarify business rules and acceptance criteria with Product and Engineering.
- ☐ Resolve ambiguous edge cases before development where possible.
- ☐ Keep design files and documentation organized and traceable.

Collaborate during build

- ☐ Stay involved during implementation.

- ☐ Answer questions and make trade-offs collaboratively.
- ☐ Perform design QA on implemented work.
- ☐ Verify responsive behavior, states, accessibility, and content.
- ☐ Track intentional deviations from design.

12. Launch Readiness

Before launch

- ☐ Confirm analytics / event instrumentation for the agreed metrics.
- ☐ Validate critical flows in production-like environments.
- ☐ Review operational readiness, support content, permissions, data, and migration needs.
- ☐ Define launch risks, monitoring plan, and rollback / mitigation paths where relevant.
- ☐ Confirm who will monitor user feedback and product performance after release.

13. Post-Launch Measurement & Learning

Measure outcomes

- ☐ Compare actual behavior against the baseline and success metrics.
- ☐ Analyze funnels, adoption, activation, retention, errors, and other relevant product metrics.
- ☐ Collect qualitative feedback and support signals.
- ☐ Segment results when aggregate metrics hide important differences.

Learn and iterate

- ☐ Compare outcomes with original hypotheses.
- ☐ Identify unintended consequences and guardrail-metric changes.
- ☐ Prioritize follow-up opportunities based on evidence.
- ☐ Feed learning back into the product strategy and roadmap.
- ☐ Maintain a decision log so future teams understand why key choices were made.

14. Case Study & Professional Documentation

Document the work

- ☐ Explain context, role, team, constraints, and product challenge.
- ☐ Show how evidence changed your understanding of the problem.
- ☐ Show key decisions and rejected alternatives - not every artifact produced.
- ☐ Connect research insights to product decisions.
- ☐ Show iterations and what was learned through validation.
- ☐ Connect the final solution to user and business outcomes.
- ☐ Be explicit about what is measured, estimated, simulated, or not yet launched.
- ☐ Reflect on what you would do differently with more evidence or time.

QUICK GATE CHECKS

Gate	Question
Before User Research	Do we know what we need to learn, from whom, and why?
Before Problem Definition	Do we have evidence beyond stakeholder assumptions?
Before Ideation	Can we clearly explain the user problem, business problem, and opportunity?
Before Committing to a Solution	Have we explored alternatives and identified the riskiest assumptions?
Before High-Fidelity UI	Are the core flows, architecture, states, and interaction model coherent?
Before Development	Have critical usability and feasibility risks been reduced enough?
Before Launch	Can we measure whether the product actually creates the intended outcomes?
After Launch	Are we learning from real behavior and feeding evidence back into decisions?

Core mindset: Do not confuse a signal with a problem, a problem with a cause, an assumption with evidence, or a proposed feature with a validated solution.