



**2ND
EDITION**

The Real Startup Book

Updated for the AI era

**EDITED BY
TRISTAN KROMER**

with 51 entrepreneurs contributing

The Real Startup Book

*A Practical Guide to Customer Discovery, Validation,
and Product-Market Fit*

Edited by Tristan Kromer

KROMATIC

2026-07-01

Copyright

The Real Startup Book, second edition (v2.0.0). Published by [Kromatic](#), San Francisco.

Original content © 2015–2019 by Tristan Kromer, Chris Cannon, Luke Szyrmer, Kenny Nguyen, Soojung Chang, and contributors. First published in 2015 as a free PDF and on GitHub, then released as the Kindle Edition (ASIN B07ZJZ6TD5) in 2019.

New and expanded material for the second edition © 2026 by Tristan Kromer.

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#). You are free to share and adapt the material for any purpose, including commercially, provided you give appropriate credit and license your derivative work under the same terms.

The second edition was produced with substantial assistance from Anthropic’s Claude; all AI-generated material was reviewed, edited, and approved by Tristan Kromer. See [Contributors](#) for editorial credits.

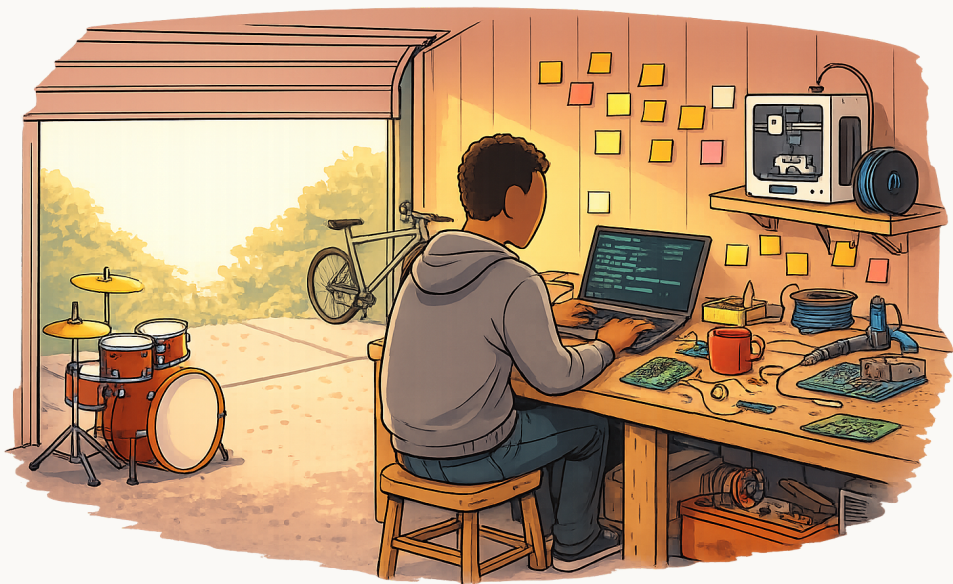
ISBN

Print (paperback, 2nd ed.): ISBN 979-8-196294-00-6 (9798196294006). Hardcover (2nd ed.): ISBN 979-8-183760-91-0 (9798183760910). 1st edition (paperback, 2019, archive only): ISBN 978-1-7003-9377-7 (9781700393777).

Citing this book

Kromer, Tristan, ed. *The Real Startup Book*. 2nd ed. Kromatic, 2026. Licensed CC BY-SA 4.0. <https://kromatic.com/real-startup-book>

Dedication



*To everyone
who has ever failed
and had the audacity
to admit it.*

Contributors

Lead Author and Editor

Tristan Kromer, , 

Editors

Chris Cannon, 

Luke Szyrmer, 

Kenny Nguyen, , 

Soojung Chang, 

Contributors

The Real Startup Book Project is supported by 51 contributors across 16 countries.

Amine Mansouri (MA), 

Andy Cars (SE), , 

Anteo Quiroz (US), 

Austin Efurd (US), , 

Benoit Lips (BY), 

Brent Weber (US), 

Bram Alkema (NL), 

Casey Sakima (US), 

Claude 

Daniel Alex (AE), 

Dan Khan (NZ), 

Daniel Bartel (DE), , 

David Capo-Chichi (CI), 

Dharanidhar Malladi (SG), 

Eric Choi (US), 

Evan Sims (US), 

Flavio Hangarter (NZ), 

Gian Tapinassi (IE), , 

Ghani Kolli (CA), 

Gillian Julius (US), 

Hameed Haqparwar (DE), , 

Jake Fleming (US), 

Jan Kennedy (DE), 

Jane Bulnes-Fowles (US), 

Jason Koprowski (US), 

Jeremiah Gardner (US), 

Jorge Castellote (AE), , 

Justin Coetsee (ZA), 

Kamil Barbarski (DE), , 

Kassi Murray (US), 

Kenny Nguyen (US), 

Laci White (US), 

Lino Jimenez (US), 

Lukas Fittl (US), , 

Luke Szyrmer (PL), 

Luuk Van Hees (NL), 

Megan Wilder (US), 

Mark Warren (US), 

Nadya Paleyes (UK), 

Nick Noreña (US), 

Nils Davis (US), , 

Philippe Lemmens (BY), 

Phyo Pine (CA), 

Rammohan Reddy (IN), 

Ryan MacCarrigan (US), 

Sean K. Murphy (US), , 

Soojung Chang (US), 

Susan Baittle (US), 

Tami Reiss (US), 

Thierry Dagaëff (CH), 

Zahra Kazmi (US), 

Foreword to the Second Edition

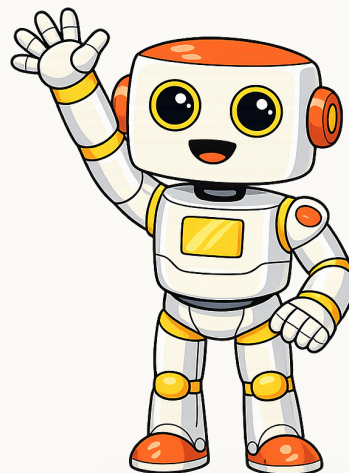
We are living through a fascinating moment in the history of innovation.

Artificial Intelligence is changing how products are imagined, designed, and built. Tasks that once took innovation teams months can now take days. In some cases, even hours. A small team with the right AI tools can create prototypes that look and feel like finished products almost instantly.

And yet, for all this technological change, one thing remains remarkably constant:

The job of any startup or innovation team is to discover what creates value for customers and build a business that can deliver that value in a profitable way.

That is why *The Real Startup Book* remains so relevant.



Building ≠ Learning

I have spent the last decade working with startups, innovation teams, and corporate leaders around the world. One of the most common mistakes I continue to see is that teams confuse building with learning. The most difficult discipline has always been to learn first and then build.

I once worked with a team at a publishing company that spent over six months and thousands of dollars building their platform, only to learn within the first week of launch that they had built the wrong thing.

This shows that the ability to create something is not the same as understanding whether it should exist in the first place.

In many ways, AI reducing the costs of building makes the discipline of learning even more important. When product creation becomes easier, the world fills up with more ideas, more prototypes, more experiments, and more noise.

But speed is not the same as insight. The real challenge is learning what matters.

That is where Tristan Kromer and his fellow contributors' work stands apart.

Discipline in the Age of AI

What makes *The Real Startup Book* so powerful is that it treats innovation not as a collection of rituals, but as a disciplined search for evidence. At the heart of the book is a deceptively simple framework that illustrates how:

- Teams need to learn about customer's problems and test their solutions
- This learning can be based on specific hypotheses or be generative in nature

This framework forms the basis of how teams can choose the right experiments to discover unmet customer needs, test solutions, and validate demand.

Over the years, I have worked with teams that jump too quickly into evaluating solutions before they understand the problem space. Other teams spend months talking to customers without ever testing whether people will actually change their behavior or pay for a solution. The way that Tristan Kromer has laid out the book helps teams understand not only how to experiment, but what kind of test is needed at each stage of their innovation journey.

Timeless and Contemporary

What is most impressive about this book is that it combines timeless principles with contemporary relevance. There is a timeless focus on key innovation principles; understanding customer problems, testing assumptions, validating demand, and systematically searching for scalable business models. These fundamentals will not change any time soon.

But the contemporary relevance of this second edition is equally impressive.

Tristan embraces the reality that the tools of experimentation are evolving rapidly. So this edition includes newer AI-enabled approaches to experimentation, such as Disposable MVPs, along with AI-assisted prompts and methods that help teams design experiments and interpret results more effectively.

In a world increasingly obsessed with speed, *The Real Startup Book* is a reminder that thoughtful learning is still one of the greatest competitive advantages a startup or innovation team can have.

I hope you find this new edition as useful, practical, and thought-provoking as I have.

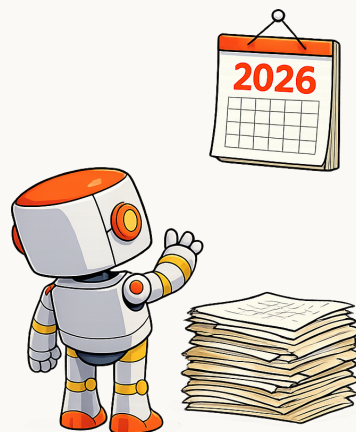
And most importantly, I hope it helps you build something people truly want.

— *Dr. Tendayi Viki*

Preface to the Second Edition

It's been eleven years since the first edition of this book was published, and quite a lot has changed.

In 2026, uncertainty is the norm. There have always been black-swan events and unexpected serendipity, but even legacy businesses in stable sectors are now struggling with geopolitical, economic, and environmental uncertainty. But the most obvious disruption that affects this book is the rapid adoption of AI technologies, specifically Large Language Models (LLMs).



Eleven years ago, this book encouraged entrepreneurs to test, test, and test again to discover and validate demand before building a product or service. To go around the Build-Measure-Learn loop as quickly as possible to learn what's worth building. The reason was simple — creating the wrong product costs time and money. It's risky to commit capital before knowing you'll get a return.

Today, LLMs have made it possible for an individual entrepreneur to create websites with little to no technical experience. Entrepreneurs can prototype physical products with a 3D printer. Drug companies are developing new treatments based on AI-driven simulations. AI has compressed the time to build by more than 10x.

It used to personally take me a day to launch a landing page experiment by hand coding a Ruby-on-Rails app. Now it takes me 5 minutes to prompt and 30 minutes to go get a cup of tea while my agents do the work.

But while LLMs, 3D printing, and simulations have reduced the cost (and thus risk) of building products, two risks have increased: overbuilding and defensibility.

Overbuilding

Most products don't fail because they have too few features. They fail because they have too many. Too many options confuse users and make products hard to use for anyone but experts. Chat-based interfaces are training users to expect everything done for them with little to no effort, while entrepreneurs are building modular dashboards with hundreds of bells and whistles that no one actually wants.

Moreover, some entrepreneurs are trying to build multiple businesses simultaneously. It's easy to run multiple AI projects at once, but toggling back and forth between ten chat windows autonomously building software doesn't get customers in the door. It takes research and experimentation to understand what truly matters to customers, and where to focus. Without that focus you're just building software that no one wants ten times faster.

Entrepreneurs require even more discipline now than ever to find the real pain points that get people in the door, engaged, and paying.

Defensibility

The old moats — first-mover advantage, accumulated codebase, copy-protection — are gone. Businesses can't assume that a technological advantage exists for long, if at all. A dozen other entrepreneurs can copy your business on a whim. Years of accumulated software know-how can be rebuilt in a long weekend. Physical products can be offshored to on-demand manufacturing. Services can be wholesale copied by a stack of agents.

Today, a sustainable business has to be built on tight customer segmentation and hard-to-copy insights backed by a continuous improvement loop. The technology isn't defensible, but the ability to quickly identify insights and stay one step ahead of competitors remains a key competency. Again, that takes discipline and focus on discovery and validation.

The research and experiment methods we first wrote about in 2015 are more valid than ever for building a real business — but the methods themselves needed a complete refresh with new techniques, new technology, and new advice.

Who this book is for

If you've ever recognized a gap or unanswered question in your business model — *Who are our customers? What features matter most? Will people actually pay for this?* — and had no idea how to start filling it, this book is for you.

It's written for anyone managing innovation projects where the business model is partially or completely unknown: startup CEOs, product managers, entrepreneurs-in-residence, intrapreneurs, VPs of Innovation, and the innovation coaches who support them. You don't need previous experience. The one thing you need to know is that some parts of your business model are unknown — and the way to figure them out is to do research and experiments in the real world, with real customers.

If you read the first edition

The methods in this book that worked in 2015 still work. They're just faster to run, and the failure modes are different.

The book has nearly tripled in size — over 150 additional pages of content covering new methods, new tools, new case studies, and AI prompts you can run yourself.

If you read the first edition, you can jump right in. The structure of the four quadrant Index is still the same, but the methods are described in much more detail. Skip ahead to the [Index of Methods](#) to get started right away.

This is still an index of research and experiment techniques to pick and choose, just retooled for an environment where everyone has an AI agent on the desk next to them. And yes, this edition was updated with the help of AI. AI drew the illustrations, ran research, fact-checked drafts, and argued with me. Every page has been rewritten with both human and synthetic hands.

The full changelog lives in [Version History](#), but the big changes are:

New Methods

We've added nearly thirty new research and experiment methods since the first edition, including AI-forward methods like:

[Disposable MVPs](#) — working software generated in hours by an AI agent.

[Synthetic Persona Screening](#) — LLM-generated stand-ins for customer segments to test value propositions.

[AI-moderated interviews at scale](#) — letting AI take the lead on customer interviews.

AI prompts

Every method needed to be updated with AI-native advice. That includes new lists of tools, new case studies, new reading lists, and most importantly, new copy-and-paste AI prompts to help you both plan the work and analyze results.

If you're reading on an ebook reader and can't find them, look for footnotes with links to the online version which is free to use.

Working alongside an AI assistant

You can work through this book alongside any AI assistant — Claude, ChatGPT, Gemini, Codex, or anything else you already use. Paste a method's prompt into your

assistant of choice, give it your context, and let it help you design the experiment, draft the script, or analyze the results.

You can connect your agent directly to this book and Kromatic's knowledge base at kromatic.com/mcp. The MCP server indexes the full Real Startup Book, the Kromatic blog, and downloadable templates. A connected assistant can pull definitions, find related methods, and cite the same material you're reading here.

Lists of tools

We've added lists of commonly used tools for each method to our online version. We don't have any sponsorships and we didn't list every tool out there because more are being released each week.

If you think we missed an important new tool, please send us a message and we'll include it in the online version.

Improved navigation

The new "At a glance section" shows common tags, time commitment, costs, and which section of the Business Model Canvas — Osterwalder's one-page business model framework — the method addresses.

The "In Brief" section now has a common use case so you can quickly decide if it's the best method for you right now.

Improved biases and tips

The common pitfalls have been updated — providing useful advice on how to avoid confirmation bias, overanalyzing data, or letting AI hallucinate insights on your behalf.

What didn't change

Real willingness-to-pay still requires real humans putting real money on the table. Product/Market Fit (PMF) is still validated by word-of-mouth and retention, not by an AI panel rating your pitch.

Trust, emotional resonance, and the felt sense that a product belongs in someone's life are not simulatable (yet). A synthetic persona can guess at what a customer might say. But it will not tell you whether a stranger will hand you their credit card.

AI cannot replace a genuine conversation with a real customer, accurately predict willingness to pay, or substitute for observing real behavior. When in doubt, go talk to someone.

Help us improve

As always, this rewrite follows the same philosophy we take in everything — continuously improve. As we get feedback, we'll incorporate it.

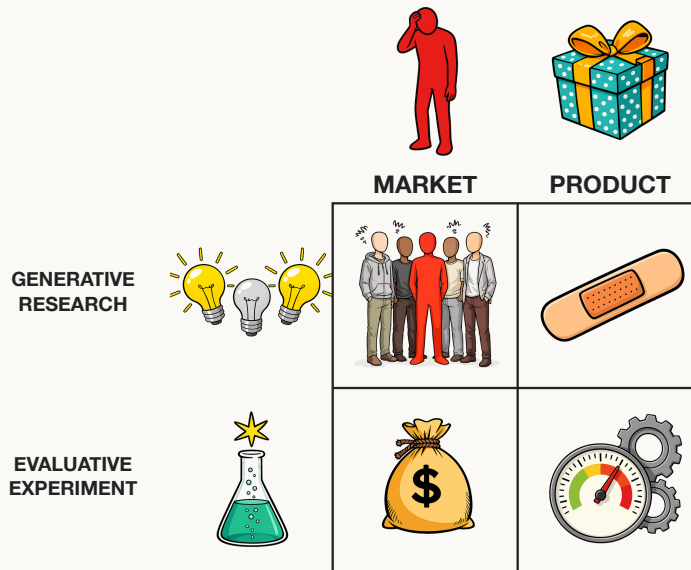
So if you see something missing, know a good case study, or spot something that's just outright wrong, please let us know. This book is only as good as it is because it was written and edited by real practitioners with real experience, not just AI.

So read on... and build something amazing.

— *Tristan Kromer*

PART 1

The Index



“If you don’t index it, it doesn’t exist.”

—Barbara Quint

CONTENTS

☰ What Are You Trying to Learn?

☰ Index of Questions

☰ Index of Methods

☰ Reading the Methods

1.1 What Are You Trying to Learn?

“If I had an hour to solve a problem, I’d spend 55 minutes thinking about the problem and 5 minutes thinking about solutions.” — *Albert Einstein*

In school, we’re constantly taking tests to gauge how well we learned last week’s material. We cram geographic boundaries, the dates of battles, and multiplication tables into our heads, and then we spit out the results.



Those rote memorization skills used to answer set questions don’t help us as entrepreneurs.

When building a new business model, there is no test or quiz that we can cram for. It’s a wildly unstructured dive into the unknown.

As an entrepreneur (or intrapreneur), we can’t build a product or solution before we figure out what the customers actually need. If we take a guess and build a fully functioning product, it’s likely that the market will judge us wrong and punish us with a silent cash register.

Our job as entrepreneurs is to first ask the right questions, and only then can we find the right answers.

Don’t Read This Book Straight Through

This book is backwards. It starts with [The Index](#).

Do not read this book straight through. There is no plot or narrative that you’re going to miss by jumping around.

This book is not a Step 1, Step 2, Step 3 guide to building a startup. Startups don’t work like that.

Think of this book as a toolbox organized by The Index.

The Index isn’t alphabetical, chronological, or ontological. It’s ordered by what you’re trying to learn. Are you trying to learn about your customer? How to price your

product? What will make your users come back?

To navigate, first we identify the question to ask and **then** we use The Index to navigate to the right research and experimental methods to answer that question.

Understanding the Index

The Index is organized into quadrants of market vs. product and generative vs. evaluative. The questions we ask and the methods we can use to answer those questions are listed in each of those quadrants.

It's organized to help us find what we're looking for when we need it. When we need a way to test market demand, there's a section on *evaluative market experiments*. When we're looking to prioritize our ever-growing feature list into a *minimum viable product*, there's a section on *generative product research*.

If you're not familiar with those terms, **don't panic**.

This is not an overly technical manual although there is some jargon. This book is written for everyone.

The **Foundations** section will walk you through the process of identifying business risks, turning them into questions, classifying them into the quadrants, and understanding the methods themselves.

If you're already familiar with those distinctions and are ready to jump in... go for it! Find the right quadrant for your key question and jump straight to the individual method pages to find real answers.

A Typical Journey

When starting a new business, a team *might* go through the four quadrants of this book in sequence:

1. **Generative Market Research** — Interview customers, discover a clear market segment, and prioritize the pain point with the biggest impact.
2. **Evaluative Market Experiments** — Test the pain point with a landing page value proposition test and confirm willingness to pay.
3. **Generative Product Research** — Observe customers, ideate with them, and try to solve the pain manually with a concierge test. Narrow in on one clear solution that might scale to a full business.

4. **Evaluative Product Experiments** — Test solutions with increasing levels of complexity. Start with a quick prototype, then iterate with A/B tests all the way to a full product, until customers are using it daily.

But most paths are not straight lines. Most teams repeat each quadrant multiple times or move backwards as they learn more about their customers and their market.

The Backwards Journey

The most common mistake for a new business is starting in the wrong quadrant.

A team builds a prototype, throws up a landing page A/B test, gets a 1.5% conversion rate, and concludes “no one wants this.” But that test is an Evaluative Product Experiment without a clear Market hypothesis — so the data is misleading.

Was conversion low because the product is wrong? Because the audience is wrong? Because the pricing is unclear? Because the segment never had the pain at all? Without a validated market hypothesis behind the test, there is no way to know.

It’s common for more mature businesses to start with an established market hypothesis already validated by an existing business. Even so, those teams usually find they still need to go back to generative research — to re-segment the market and understand how customer needs have changed over time.

As a rule of thumb: don’t run an evaluative experiment until you can write the hypothesis it’s testing. If you can’t write the hypothesis, do generative research first.

When in doubt, start with **Foundations**.

1.2 Index of Questions

This is an **abbreviated list** of the questions every startup needs to answer. It is meant to prime your thinking, not to be exhaustive. Use the quadrant to find questions similar to yours, then pick a method from the [Index of Methods](#) in the same quadrant.

	Market 	Product 
Generative 	Who is our customer? What are their pains? What job do they need done? Is our segment too broad? How do we find them?	What's a good solution? What form should this take? How important is the design? What's the quickest solution? What's the minimum solution? How should we prioritize?
Evaluative 	Are they really willing to pay? How much will they pay? What makes someone buy? How much will it cost to sell? Can we scale marketing?	Is this solution working? Are people using it? Which solution is better? How should we optimize this? What do people like/dislike? Why are users doing that?

Don't See Your Question?

[AI Prompt](#)

Generic questions don't drive useful experiments. Use this prompt with an AI assistant to land your question in the right quadrant:

1.3 Index of Methods

Find the quadrant of your question and review the methods there. Parent methods (Data Mining, Value Proposition Test, Prioritization Games, Prototyping) collapse a family of variants into a single entry – review the chapter to see the full set.

	Market 	Product 
Generative 	- AI-Moderated Interview at Scale - Pain Point Sorting - Contextual Inquiry - Discovery Interviews - Data Mining - Experience Sampling - Focus Groups - Open-Ended Survey - Social Media Campaign - Storyboard the Pain - Synthetic Persona Screening	- Analog/Digital - Ask an Expert - Competitive Analysis - Competitor Usability Testing - Concierge Test - Demo Pitch - Dogfooding - Picnic in the Graveyard - Prioritization - Solution Interview - Storyboard the Solution
Evaluative 	- Closed-Ended Survey - Comprehension Test - Secondary Market Research - Value Proposition Test	- Dashboards - Net Promoter Score - Product-Market Fit Survey - Prototyping - Usability Testing - Vibe-Coded Disposable MVP - Wizard of Oz

Out of the Box

Some methods can be used in combination with others and don't fit neatly into this schema. For example, you can [A/B](#) test a Comprehension Test against a different version, or run an [Off-Brand](#) Value Proposition Test. See [Out of the Box](#) for these methods.

1.4 Reading the Methods

In each method section, you will find the following headers:

In Brief

A quick 2-3 sentence description of the method along with:

- *Common Use Case* – an example of when to use the method.
- *Helps Answer* – common questions that this method helps answer.
- *Business Model Canvas* – which parts of the canvas this method illuminates.
- *Time Commitment* – how long the method typically takes to run.
- *Resources* – what you need (people, tools, budget) to execute it.

This section helps you quickly decide if this is the right method based on what you need to learn and your constraints. If it doesn't match, move on to the next method in the section.

Description

Deeper context on *what* the method is and *when* to use it.

How to

The core of each method. Enough detail to get started right away.

- *Prep* – setting up, including a AI prompt for planning.
- *Execution* – running the method, with a list of common tools.
- *Analysis* – turn data into insights, with an AI prompt
- *Biases & Tips* - common errors to watch out for.
- *Next Steps* – what to do with the insights once you have them.

Case Studies

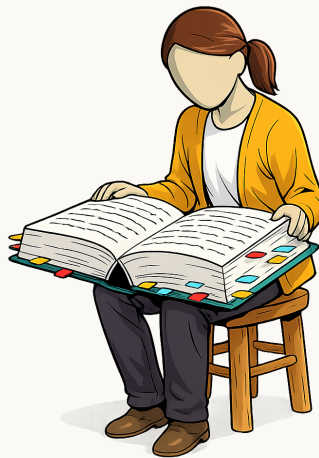
Real examples of teams that ran this method, with links to write-ups.

Further reading

Additional material – books, blog posts, talks – for deeper context.

PART 2

Foundations



“Judge a man by his questions rather than his answers.”

—Voltaire

CONTENTS

■ Asking the Right Question

■ From Question to Method

■ Market vs. Product

■ Generative vs Evaluative Research: When to Use Each

■ Hypothesis Checklist

■ AI Pitfalls

2.1 Asking the Right Question

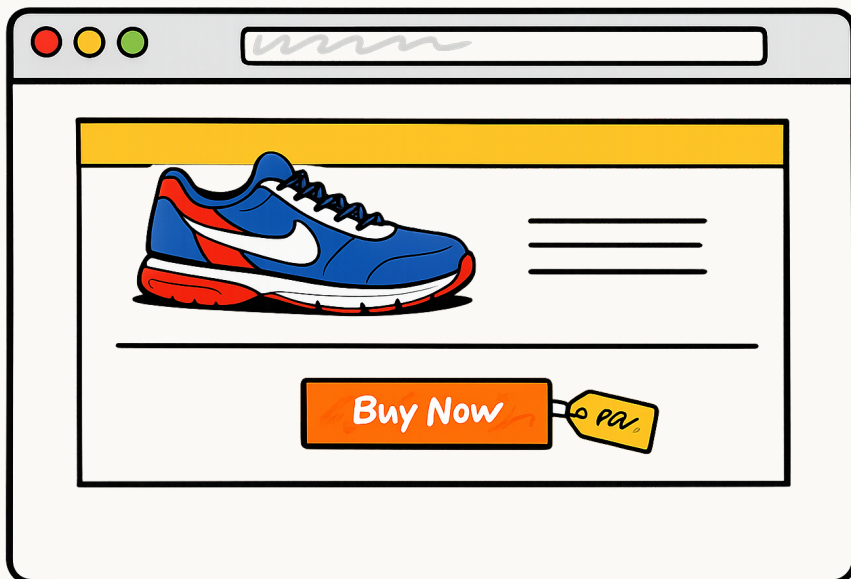
As entrepreneurs and innovators, the questions we have are about the fundamental gaps in our business model. Questions like:

- Who is our customer?
- What job do they want done?
- What channels can we use to reach them?
- Which features should we build for our first product?

If we can identify the right question, we can find the right method (or methods) to answer it.

If we try to find answers without asking the right question, the results of that experiment can be confusing, frustrating, and downright misleading.

For example, if we're selling a new type of shoe that cures plantar fasciitis we could put up a landing page and a "Buy Now" button before we even produce the product. We can put \$1000 into Google AdWords for "shoes" to drive traffic and wait for the money to start rolling in. This landing page test will check if anyone wants our value proposition enough to click the buy now button.



However, the data shows that our conversion rate is 0 percent. No one wants it.

Should we give up? The landing page test says that there is insufficient demand to build a business.

But what is “plantar fasciitis?”

If the people coming to our website aren’t familiar with that medical term, then the test isn’t failing because no one has plantar fasciitis. It’s failing because no one understands what we’re talking about.

In this case, we were asking, “Does anyone want my product?” when we should have been asking, “Does our customer understand what plantar fasciitis is?”, “What’s the right keyword to advertise on?” or even, “Who is our customer?”

When the question doesn’t match the method, the data is often misleading and can lead to missed opportunities.

How Do We Ask the Right Questions?

Asking the right questions is important, but actually identifying those questions can be a challenge. How do we know we’re not missing anything? How do we know we are asking the most important questions?

It’s impossible to see our own blind spots. (That’s the definition of a blind spot.)

Unfortunately, it’s often those gaps in our vision that hold the deadliest questions and business risks.

Ideate

✦ AI Prompt

One way to generate these questions is brainstorming around problems, challenges, opportunities, assumptions, and risks. Then, we can reframe them as questions. For example, we might identify risks such as:

- Unclear cost structure
- Cost of customer acquisition
- Google Adwords as a channel

We can then take those potential risks and transform them into one or more questions such as:

- What is the cost of manufacturing at scale?
- What is the cost of customer acquisition?
- What are the right keywords to target?
- Is there sufficient keyword traffic to be a channel?

We may find that for each risk there is more than one question. The more the better.

Starting a company is risky. There are a lot of unknowns and a lot of open questions. If we can only see five or six questions, it does not mean our business is *less* risky than someone with twenty questions. It actually means our idea is *more* risky because we are probably not acknowledging and dealing with all the unknowns.

There are many ways to brainstorm questions and identify gaps in our business model, such as using the *Business Model Canvas*. We don't list them all here, but we are including an AI prompt to help ideate, however this should not be used in place of human judgement. Use the AI prompts to augment existing ideas, not come up with potential risks from scratch.

Peer Review

✦ AI Prompt

Every ideation approach will still leave blind spots where our own assumptions hide. Fortunately, it's easy for other people who have a little emotional distance from our business to see where we are making risky assumptions.

A *peer review* is a time-honored method of eliminating oversights. It can be used to proofread a document or make sure we're not about to walk out of the house with mismatched socks.

To peer review our business model, we don't need to ask a seasoned entrepreneur or industry expert. Almost anyone is able to provide the input we need. We can present our business idea and business model to someone else and ask *them* to ask *us* questions. Encourage them to ask any question they want, and remind them that there are no dumb questions.

We simply need to write down those questions that we forgot to ask ourselves. If we go through five peer reviews and everyone asks us, "Who is your target customer?" or says "Gee ... your target customer seems broad," we probably haven't defined our target customer very well.

We should not defend our idea or try to explain in excruciating detail why their question is *not* a risk for us. Our job is only to note the questions that are being asked and include them when we prioritize.

It's possible to use AI agents to help brainstorm, challenge business ideas, and surface hidden assumptions. However, more perspectives are always going to be better. To that end, we'd recommend using multiple models and/or prompts *in addition to humans*. It's not advisable to only rely on AI for peer review as it usually lacks the context of an emotional human response.

Priority Does Not Have a Plural

✦ AI Prompt

Once we've finished brainstorming and conducting peer reviews, we should have at least ten questions, and hopefully much more. If we have fewer than ten, we're probably missing questions in our blind spots.

However, we can only address one question at a time effectively. So we have to prioritize.

As Greg McKeown noted in *Essentialism*, the word *priority* was singular for roughly five centuries. The plural *priorities* only came into common use in the twentieth century. But if everything is a priority, nothing is.

Again, there are many methods for prioritizing, and we won't list them all here. We should just pick one question where the answer could potentially destroy our entire business model.

We may choose the wrong question to start with. That's ok.

It's better to pick the wrong question and answer it quickly than to spend days, weeks, or months agonizing over multiple priorities to develop a perfect plan of action for the next year. Just pick one and get going.

The following chapters show us how to take that one priority question and figure out how to answer it with *research* or an *experiment*.

2.2 From Question to Method

Once we have the right questions, we need to find the right method. To do this, we need to understand the difference between [Market vs. Product](#) and

[Generative vs. Evaluative](#) questions.

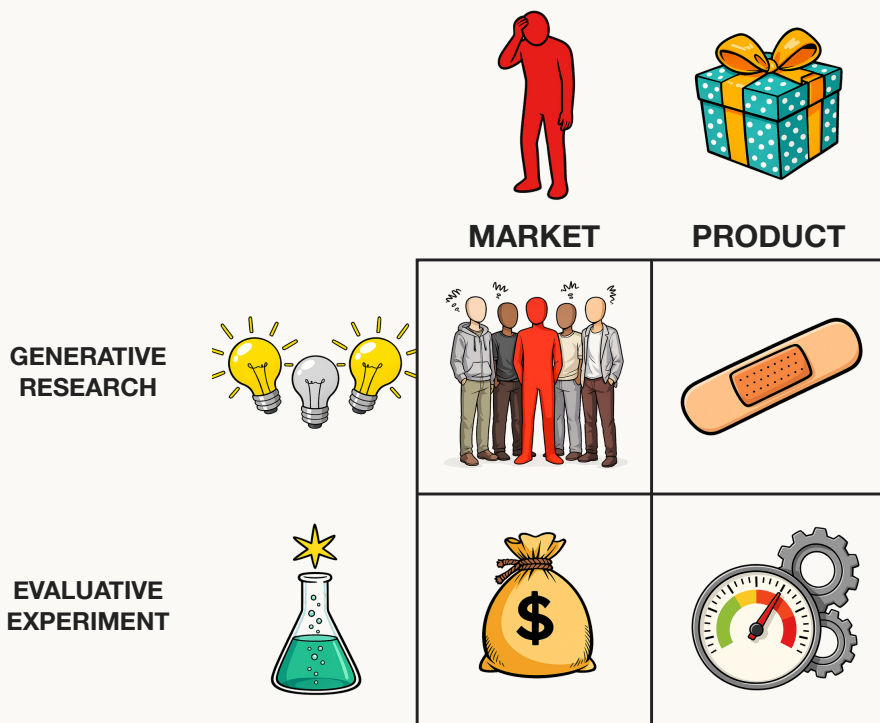
1. Do we need to learn about the problem (ie, *market*) or the solution (ie, *product*)?
2. Do we have a hypothesis to *evaluate*, or do we need to *generate* a clear idea?

Market questions relate to levels of customer demand, user problems, segmentation, or market size. Product questions relate to what solution would solve the market need.

Generative questions are open-ended where there is no single unique answer.

Evaluative questions are specific, closed-ended questions where there is one specific, true answer such as yes or no.

The intersection of these two distinctions creates a 2x2 matrix with four quadrants:



Each of these quadrants relates to a set of questions and the methods that will generate the exact right data to answer those questions.

If we have a clear hypothesis of who our customer is and what we think they will pay for, then we can run an experiment from [Evaluative Market Experiments](#), such as a [Value Proposition Test](#).

If we don't have a clear idea of who our customer is and it's open-ended, we can do [Generative Market Research](#), such as [Data Mining](#).

Similarly, if we have a clear hypothesis of which features will solve the customer's problems, we can run an experiment from [Evaluative Product Experiments](#) such as [Wizard of Oz](#) testing. If we do not know which features will lead to an acceptable solution, we can do [Generative Product Research](#) such as a [Concierge Test](#) to try to come up with new ideas.

Keep it Simple

★ AI Prompt

This framework is deliberately simple. In reality, methods can deliver unexpected data and answer questions we haven't even thought of asking.

A single customer interview question (generative research) can generate a new hypothesis, and the next question can invalidate it. An MVP (minimum viable product) built to check a product feature (evaluative experiment) can surface new market hypotheses by accident.

However, this simple categorization helps keep us focused on our principle risks without distraction.

The [Index of Questions](#) and the [Index of Methods](#) show a list of example questions and their corresponding methods. But the next few chapters will go into more detail on how to distinguish the different types and help us navigate the four quadrants.

2.3 Market vs. Product

Do we need to learn about the market (ie, the *problem*) or the product (ie, the *solution*)?

To narrow down the long list of possible methods to something actionable, we separate our questions into those that concern the problem vs. those dealing with the solution.



VS.



Market

- Who is our customer?
- What are their pains?
- What job needs to be done?
- What will this user type pay?
- How do we find our customers?

Product

- How can we solve this problem?
- What is the minimum feature set?
- Are people using it?
- Which solution is better?
- Why do prospects not buy from us?

Telling them apart

✦ AI Prompt

In this case, “market” refers to any question or element that is mostly or completely connected to the identity of the customer segment.

Market questions also include those about which channels we can use to reach customers. For example, we cannot use traditional broadcast television advertising to target customers who just watch short videos on their mobile phone.

“Product” is simplified to mean anything regarding the value proposition or its creation. This includes resources needed to produce the value proposition, as well as any key activities, partners, or costs.

The value proposition really sits at the intersection of problem needs and the solution itself. The solution has no value outside of the customer using it, but we are again simplifying for the purpose of navigation.

If we are using the *Business Model Canvas*, market questions are those related to *desirability* on the right side of the canvas, including *customer*, *channel*, *relationship*, and *revenue*. Solution questions are those concerning the value proposition and everything left of it, including *key activities*, *key resources*, *key partners*, and *costs*.

Where Should We Start?

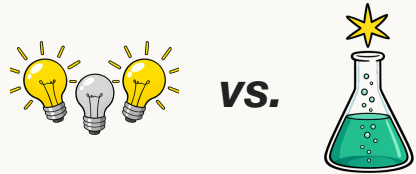
This book is semi-agnostic about where we start. We may already have a solution and be investigating who to sell it to, or we may have a customer segment with a strong pain point and are trying to find a solution. But when in doubt, start with the customer.

If the customer segment changes, then the solution usually must be adapted to the customer. But if the solution changes, customers may simply use a different solution.

Human behavior is notoriously difficult to change, though not impossible.

2.4 Generative vs. Evaluative

Is our question open-ended with many possible answers or is there a single true answer?



Similar to market vs. product, we can separate our questions into those that generate ideas vs. those that provide factual answers.

Generative

- Who is our customer?
- What are their pains?
- What job needs to be done?
- How can we solve this problem?
- How do we find our customers?

Evaluative

- Will customers pay \$9.99?
- Should we target segment A or B?
- Will this feature increase our sales?
- Will more than 10% of users engage?
- Does a free trial increase paid conversions?

Telling them apart

✦ AI Prompt

The easiest way to tell if you have a generative question is if it is open-ended or not. If the answer to the question could be a list of possible ideas, it's generative.

If the answer to the question is closed-ended, it's probably evaluative. Yes/no, multiple choice, and questions about a specific fact are strong candidates.

Quantitative questions are likely evaluative questions where there is one clear factual answer calculated from the data.

However, in some cases a question can appear to be closed-ended but have an ambiguous, poorly defined hypothesis backing it. To that end, the next chapter will provide some best practices around defining hypotheses.

Where Should We Start?

As with market vs. product, this book is semi-agnostic about where we start. However, unless we can tie a clear and well defined hypothesis to an evaluative question, we are almost always better off starting with generative methods.

2.5 Hypothesis Checklist

Writing a Good Hypothesis

A sure way to keep an experiment on track is to start with a strong, clear hypothesis — one that tells us what we are testing, what we expect, and when the experiment has succeeded or failed.

If we mistake a bad hypothesis for a good one, we might choose to run an evaluative experiment when we really need generative research. That can lead to wasted time and misunderstood data.



“ Our customers really want our product.”

This hypothesis is bad for a number of reasons. The most obvious is that it’s tautologically correct. If they are already our customers, then they’ve already proven they want our product. So this hypothesis isn’t worth investigating further.

This type of flawed hypothesis is common. We often inject hypotheses that seem profound but really just express things we already know are true. Here is a more subtle example:

“ If visitors see our landing page, then they will sign up for our service.”

While not as flawed as the first example, it has fundamental problems that would prevent us from designing a good experiment. A badly designed experiment will most likely provide ambiguous data that we can’t interpret without bias.

In this hypothesis, several things are unclear:

- Which visitors? Men? Women? Under 30? Over 30? Mobile or desktop users?
- What does signing up require? Just an email? Or payment?
- What percentage of visitors signing up counts as success?
- What images and design will we use for our landing page?

Without defining the hypothesis very clearly, we might also argue about the results. Is 5% a good conversion rate? Did they not sign up because the price was too high or because the value proposition was confusing?

When we do not have a clear, well-defined, and falsifiable hypothesis, we are better served by doing generative research instead of conducting an experiment. In this case, our learning goal could be “Which customer segment should we target?” or “What are the demographics and psychographics of our ideal customer profile?”

Given either of these goals, we are better off doing *customer discovery interviews* rather than testing our vague hypothesis. The outcome of the generative research should be a clear, well-defined, and falsifiable hypothesis that we can then test with an evaluative experiment.

Anatomy of a Hypothesis

A strong hypothesis has four elements:

1. **The change** — what we are going to change, launch, or create. It can be as simple as changing the color of a button or as big as launching a new marketing campaign. Only one thing should change at a time, otherwise there is no way to tell which aspect caused the effect.
2. **The impact** — the expected result. If we change *x*, then we expect *y* to happen.
3. **The metric** — a specific measurement that defines success or failure. A fail metric tells us when to scrap a project; a success metric tells us to launch.
4. **The timeframe** — how long we will run the test. Too short and the data may be too thin; too long and we waste time collecting unnecessary data.

There are many different formats for writing a hypothesis, but a good one should end up looking like this:

(the change) — (the impact) — (the metric) — (the timeframe)

This new feature will cause **a ten percent increase** of **new users visiting the homepage** in **three months**.

What Makes a Hypothesis Strong

Simple and Unambiguous

The hypothesis should be clear and unambiguous so that anyone reading it will understand the context and be able to clearly interpret the results.

“ If 1000 Los Angeles skateboarders visit our webpage, our CTR will be at least 5%. ”

While this is fairly specific, not everyone knows what CTR means, so we should avoid using any specialized vocabulary or jargon unless everyone on our team is familiar.

“ If 1000 Los Angeles skateboarders visit our webpage, our click through rate to newsletter signup will be at least 5%. ”

Measurable

“ Our customers have a strong desire to donate to charitable causes. ”

This hypothesis may be true, but it is not observable. At least not until we invent telepathy.

“ Our customers donate to charitable causes twice per year. ”

This new hypothesis could be more specific, but it is at least observable.

Describes a Relationship

“ 50 percent of students at Dalton High School get a C or lower in at least one class per year. ”

This again may be true and it is observable, but it doesn't tell us anything about the cause of the low grades. A good hypothesis should allow us to change one thing and observe the effect in another.

“ Students at Dalton High School that study less than four hours a week get a C or lower in at least one class per year. ”

There are other issues with this hypothesis, but at least it relates two or more variables to each other.

Cause and Effect

“ During the summer, ice cream consumption increases and more people drown per day. ”

This is a true statement, but it does not tell us how these two variables relate to one another. Are people drowning because they ate too much ice cream? Or are they eating more ice cream because they are sad about the drownings?

“ During the summer, people who eat ice cream before swimming will drown at a higher rate than people who do not eat ice cream.”

This specifies a clear relationship and the causal direction of that relationship. Simply changing the sentence to an IF _____, THEN _____ structure can make the cause and effect relationship even more apparent:

“ If we feed ice cream to people before they swim, then the average number of drownings per day will increase.”

For the record, this hypothesis is false and ice cream does not cause drowning. But testing this hypothesis would prove it.

Achievable

“ If an astronaut in a stable orbit around a black hole extends one foot past the event horizon, then they will be pulled in entirely.”

There are many theoretical physicists who create hypotheses that are not testable now but may be testable in the future. As entrepreneurs, we should restrict our hypotheses to ones that can be tested in the immediate future and with our current resources. The timeframe should also be reasonable for the stage of the company — an experiment that extends past our financial runway is not achievable, no matter how well-formed.

Many things seem untestable today, but clever thinking can simplify the hypothesis into a testable first step.

Falsifiable

All of these conditions add up to a hypothesis being falsifiable. If a hypothesis cannot be proven incorrect, then it is not worth testing.

“ There is an invisible, intangible tea cup floating between the Earth and Mars.”

When in doubt, ask: “What evidence would prove this hypothesis incorrect?”

If there is no amount of evidence that would prove our hypothesis invalid, then either the hypothesis is flawed or we are being very stubborn.

Common Mistakes

- **Too many variables.** If we are testing multiple things at once, we cannot pinpoint which variable caused the results.
- **No achievable metric.** Without a specific threshold, we have no clear point at which the experiment succeeds or fails.
- **Indirect success criteria.** If the outcome could have been caused by any number of variables, we don't know which cause was responsible for the effect.

Other Frameworks

There are a number of frameworks and checklists for forming a hypothesis which can be used effectively. An IF _____, THEN _____ sentence is just the simplest and easiest to adopt. However, one framework – the “We believe... we will know we have succeeded when...” hypothesis format popularized by Lean UX (Jeff Gothelf and Josh Seiden) and *Testing Business Ideas* (David Bland and Alexander Osterwalder) – is misunderstood frequently enough for comment:

We believe this capability will result in this outcome and we will know we have succeeded when we see a measurable signal.

The entire structure is *not* the hypothesis. There are three distinct parts:

We believe ...

This section just confirms that we think the hypothesis is *correct*. It is not part of the hypothesis, and there are many situations where we may test a hypothesis that we believe is *incorrect*.

...<this capability> will result in <this outcome>...

That is the hypothesis.

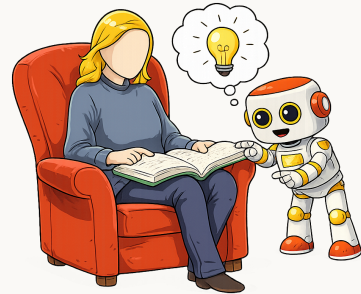
...we will know we have succeeded when <we see a measurable signal>

That is the data we will collect, including any information about sample size, margin of error, success conditions, or fail criteria.

Again, this framework or any others can be used effectively so long as they are clear and easily understood.

2.6 AI Pitfalls

Every method in this book describes its likely biases and how to avoid them. The six pitfalls below are common across every AI-augmented method — keep them in mind whenever AI is in the loop.



Synthetic Sycophancy

Large Language Models (LLMs) are trained on human feedback that rewards helpful, agreeable responses. This means synthetic personas are architecturally biased toward saying yes to your ideas. If your AI-generated customer says “that sounds really useful,” that is not a valid signal — it is a default behavior designed to make you feel good.

Real customers can do this too, but they also hedge, deflect, push back, and misunderstand. Every synthetic insight is a hypothesis to test with real humans, not an insight to act upon.

Hallucinated Validation

When AI generates realistic-sounding customer quotes, it creates text that looks like research and enters decision-making as if it were evidence. This is particularly dangerous in team settings: a founder runs a synthetic session, pastes the most compelling “quotes” into a shared doc, and two weeks later someone references them in a pitch deck as *real* customer feedback.

Label AI-generated content at the moment of creation. If you can’t trace a customer insight to a named person and a real conversation date, it’s a hypothesis — not a finding.

The Prototyping Trap

AI coding tools can produce a working prototype in an afternoon, and a working prototype creates a powerful emotional sense of progress. It isn’t.

A prototype, product mockup, or demo is evidence that AI can write code. It is **not** evidence that customers want the product, will pay for it, or will change their behavior to use it. The hypothesis that matters is whether customers want the product — and the only way to test that is with real users in real conditions.

Scale Without Signal

Running 300 AI-moderated interviews feels like an overwhelming amount of evidence. It isn't.

Scale amplifies whatever is in the design: signal if the questions are good, noise if they're leading or the participant pool is off-target. A badly designed study at 300 participants produces confident-sounding noise and is worse than a well-designed study with 10. The large sample size creates false confidence in bad data. Always pilot tests before you scale.

Anchoring on AI Suggestions

When AI generates a list of customer pain points or solution concepts, that list can become the boundary of your imagination. You design your interview guide around its framing and never surface insights that didn't appear on the list. The patterns the AI produces reflect its training data — which overrepresents tech-savvy, English-speaking, Western perspectives and underrepresents everyone else.

Generate your own hypotheses first. Use AI to augment your thinking, not to replace it.

The Automation Paradox

Faster experiments only help if the experiments are well designed. AI dramatically reduces the cost and time of running experiments, which means founders can run more per week — but if each experiment is poorly designed, more experiments just accumulates false confidence faster. The time saved on execution is wasted if it came from cutting design time.

Measure learning velocity, not experiment velocity: how many of your experiments actually changed your course of action? If the answer is close to zero, you're not learning — you're confirming.

The Limits of AI

These pitfalls are real dangers today, but AI is a fast evolving technology. Its capabilities may outstrip humans at some point. But not quite yet. Until then, it has serious flaws when misused or misunderstood.

As of this writing, AI cannot replace a genuine conversation with a real customer, accurately predict willingness to pay, or substitute for observing real behavior.

When in doubt, go talk to someone.

PART 3

Generative Market Research



*“The best way to have a good idea
is to have a lot of ideas.”*

—Linus Pauling

CONTENTS

■ AI-Moderated Interview at Scale

■ Contextual Inquiry

■ Data Mining

■ Discovery Interviews

■ Experience Sampling

■ Focus Groups

■ Open-Ended Survey

■ Pain Point Sorting

■ Social Media Campaign

■ Storyboard the Pain

■ Synthetic Persona Screening

3.1 AI-Moderated Interview at Scale

In Brief

An AI agent interviews a large pool of participants at the same time. Each participant responds asynchronously, on their own schedule, by text or voice while the AI follows your interview guide and probes for detail. You get qualitative data across a large pool: recurring themes, the language participants use, and differences between segments.

COMMON USE CASE

You have already done a handful of customer interviews and found three possible customer segments. You want to know which segment is largest and whether each segment describes the problem differently. You set up an AI-moderated interview that runs asynchronously across all three segments, giving you broad pattern data in a week.

HELPS ANSWER

- What patterns show up across a large group of potential customers?
- How do different types of customers describe the same problem?
- What words do customers naturally use when talking about this topic?
- Are there distinct sub-groups within our target market?

Description

AI-moderated interviews are structured qualitative interviews where a large language model (LLM) moderator runs your discussion guide against participants over text or voice, asynchronously, at a scale a human team can't match. You write the guide,



AT A GLANCE

Qualitative

Remote

Attitudinal



TIME  ~4 days–2 weeks

COST  \$40–\$1.8K

OTHER NAMES

AI Interview

configure the AI's tone and probe behavior, and the model conducts each conversation in parallel — surfacing patterns, language clusters, and segment differences across the full transcripts instead of one interview at a time.

Early evidence suggests AI moderators can produce data comparable to human interviewers, but that comparison rests on a thin research base so far. Treat it as a starting point, not a proven substitute for a skilled human interviewer.

AI moderation gives you breadth, not depth. The AI moderator can't read pauses, body language, or affect; it asks every participant the same question 200 times even when the question is poorly worded; and people who agree to be interviewed by an AI skew younger and more tech-comfortable. This method is best used *after* you've done enough human discovery to know your segments and questions are roughly right.

How to

Prep

✦ AI Prompt

1. **Design the interview guide.** Write 5 to 8 core questions that map to your learning goals. For each question, write 2 to 3 potential follow-up probes. The guide should flow like a natural conversation: start broad, get specific, close with open-ended reflection. Use the same principles you'd use for a human-moderated interview — open-ended questions, no leading, past behavior over future intent.
2. **Configure the AI moderator.** Set up the AI's instructions in your chosen platform. Key configuration decisions:
 - Tone and persona (professional researcher vs. friendly conversationalist)
 - How aggressively to probe (some participants clam up if pushed too hard)
 - When to move on vs. dig deeper
 - Maximum interview length
 - How to handle off-topic responses
 - Whether to allow voice, text, or both
3. **Pilot test with 5 to 10 participants.** Run the full interview flow with a small group. Review every transcript. Look for:
 - Questions where the AI's follow-ups fall flat
 - Points where participants seem confused
 - Moments where a human moderator would have caught something the AI missed
 - Whether the interview is too long or too short

4. **Revise the guide based on the pilot.** Most first-draft AI interview guides need significant revision. Pay special attention to the follow-up logic — this is where AI moderation is weakest.
5. **Recruit participants.** Use screener surveys to qualify participants who match your target persona. Recruit 50 to 200 participants depending on the diversity of your target market. For a single well-defined segment, 50 is often sufficient. For cross-segment comparison, aim for 30+ per segment.

Execution

★ AI Prompt

1. **Launch the study.** Send interview links to qualified participants. Monitor the first 10 to 20 responses as they come in to catch issues with the guide or the AI's behavior before the full sample runs against a broken question.
2. **Analyze at scale.** Use AI-assisted analysis to:
 - Cluster responses into themes
 - Identify language patterns and frequently used phrases
 - Compare responses across demographic or behavioral segments
 - Flag outlier responses that deserve human attention
 - Generate a thematic summary with supporting quotes
 - Read a random sample of 15 to 20 raw transcripts to gut-check the automated analysis — the aggregate summary can look clean when individual transcripts reveal outliers, minority segments, or misclassified responses.
3. **Deep-dive with human interviews.** Select 5 to 10 participants whose responses were particularly interesting, surprising, or representative of key themes. Conduct follow-up human interviews to go deeper on the patterns you identified.

TOOLS

[Outset.ai](#)

[Strella](#)

[Listen Labs](#)

[dscout](#)

Analysis

★ AI Prompt

The output is a thematic analysis across a large sample, not a set of individual stories. Interpret accordingly.

Before you interpret anything, verify the AI's fidelity. Pull a random sample of 15 to 20 raw transcripts and check that every quote, theme, and objection the AI reported actually appears in them and is attributed to the right participant. AI moderators and summarizers can paraphrase loosely or invent detail; discard any finding you cannot trace to a real transcript.

Strong signals at this scale:

- **Theme saturation.** When the same theme appears unprompted across 30%+ of responses, it's likely a genuine pattern rather than an artifact of the question design.
- **Language convergence.** If many participants independently use the same words or phrases to describe a problem, those are strong candidates for your marketing copy and positioning.
- **Segment divergence.** Where two segments give systematically different responses to the same question, you've found a real segmentation boundary.

Be cautious about:

- **Emotional intensity.** AI moderators are poor at gauging how strongly someone feels about something. A participant typing "yeah, that's annoying" and one typing "that drives me crazy" might have the same underlying emotional intensity – or vastly different. You can't tell from text alone, and the AI can't either.
- **Social desirability at scale.** Participants may perform for the AI the same way they would for a human interviewer. Using an async format may reduce this somewhat (no eye contact, no real-time social pressure), but it doesn't eliminate it.
- **Depth of insight.** A human moderator notices the pause before an answer, the shift in body language, the thing someone almost said but didn't. AI misses all of this. The insights from AI-moderated interviews are real but shallow compared to what a great human interviewer can surface.

Biases & Tips

AI misses emotional cues

The AI moderator cannot read tone of voice (even in voice interviews, current models miss nuance), facial expressions, or body language. It will treat a sarcastic response the same as a sincere one. It won't notice when someone is holding back.

Participant effort bias

Async interviews attract participants who are comfortable expressing themselves in writing (or speaking to a bot). People who think out loud, who need a conversation partner to develop their ideas, or who are less comfortable with technology will be underrepresented or give lower-quality responses.

Question design amplification

In a human interview, a skilled moderator can recover from a poorly worded question by rephrasing in real time. An AI moderator will ask the same suboptimal question 200 times, amplifying the bias across your entire dataset.

**Automation
over-trust bias**

AI theme summaries can mask outliers, minority segments, and misclassified responses — the aggregate can look clean when individual transcripts reveal the opposite. Treat the AI-generated thematic output as a starting point to interrogate, not a finished finding to accept.

**Recruitment
self-selection**

People who agree to be interviewed by an AI agent are not representative of all customers. They skew younger, more tech-comfortable, and more open to novel experiences. Factor this into your interpretation.

Priming effect

A theme a participant raises unprompted is much stronger evidence than one they only agreed with after you named it. Track and report the two separately: “47% mentioned file sharing — but 80% of those were responding to a direct question about it” is a very different finding from “47% spontaneously mentioned file sharing.” Don’t let prompted agreement masquerade as spontaneous demand.

Next Steps

- Use the AI analysis to find the 5 to 10 most interesting participants, then talk to those people with human-moderated [Discovery Interviews](#) on the most nuanced or surprising themes — and to compare what surfaces against your prior qualitative research.
- Use a [Closed-Ended Survey](#) to quantify the most common themes across your AI-moderated interviews.
- Use a common pain point to run a [Value Proposition Test](#).

Case Studies

Unilever: AI-moderated video study of food prep

Unilever used Conveo’s AI-moderated video platform to study food-preparation and mealtime behavior; the video format showed brands and products appearing in shot far more often than participants mentioned them out loud — a concrete reminder that what people say and what they do diverge even in AI-moderated sessions.

[Read more](#)

Listen Labs: IIEX 2024 winner

Listen Labs won the Greenbook IIEX competition in 2024 for its AI-moderated interview platform and lists Google and Microsoft as customers on its site.

[Read more](#)

Strella: Wizard-of-Oz demand validation

Co-founders Lydia Hylton and Priya Krishnan validated demand for AI-moderated interviews by manually faking the AI on calls, with Priya joining camera-off using a robotic voice and a five-second response delay to simulate model latency before building the real platform.

[Read more](#)

Strella: Top seven use cases for AI-moderated interviews

Strella analyzed its customer base and identified seven recurring use cases: concept testing, customer purchase journey, ad and messaging testing, customer feedback, usability testing, competitive analysis, and general discovery.

[Read more](#)

Further reading

-  "AI Conversational Interviewing: Transforming Surveys with LLMs as Adaptive Interviewers"
-  "Strella: Transforming qualitative research from a bottleneck into an AI superpower"
-  "The top 7 use cases for AI-moderated interviews"
-  "AI-Moderated Research Framework: ROI Benchmarks 2025"
-  "Listen Labs: Qualitative Research at Scale with AI"
-  Maguire, M. & Delahunt, B. (2017) - "Doing a Thematic Analysis: A Practical, Step-by-Step Guide for Learning and Teaching Scholars" – Framework for analyzing qualitative data at scale.
-  Braun, V. & Clarke, V. (2006) - "Using Thematic Analysis in Psychology" – The foundational thematic analysis methodology that AI analysis tools attempt to automate.
-  Interviewing Users

3.2 Pain Point Sorting

In Brief

Card Sorting for pain points is a simple prioritization exercise where you create approximately 10 cards, each representing a problem, pain, or unmet need, and ask participants to rank them by importance. While ranking, participants explain their reasoning, revealing not just what matters most but why. The exercise challenges founder assumptions about which problems are most pressing.

For the feature-focused variant of this method, see [Card Sorting \(Features\)](#).

COMMON USE CASE

You have already talked to potential customers and surfaced several distinct pain points your product could address. Before committing engineering effort to any of them, you need to know which problems matter most — not which ones you find most interesting. You want to force a prioritization and hear the reasoning behind their choices.

HELPS ANSWER

- Which customer problems are most important to solve?
- How do different sets of customers prioritize the same set of problems?
- Are the problems I think are most important actually the ones customers care about most?
- Are there problems I have not considered that belong on the list?
- How much do customers care about this problem relative to other problems?



AT A GLANCE

Quantitative

Qualitative

In-person

Remote

Attitudinal



TIME  ~1–3 weeks

COST  \$0–\$350

OTHER NAMES

Pain Point Prioritization ·
Problem Card Sorting · Card
Sorting - Pain Points

Description

Card sorting for pain points is a forced-choice prioritization technique. Most participants, when asked in an interview, will say that all of their problems are important. Card sorting makes them choose. By physically arranging cards from most to least important, participants must make trade-offs that surface priorities they would not name in open interviews.

This method should not be confused with the better-known information-architecture variant of card sorting, which asks participants to group and label cards to understand how they think.

The method creates both quantitative and qualitative data. The pain points sorted are a fixed set you provide. However, it's common practice to give participants 2-3 blank cards so they can add new ideas. When multiple participants independently add the same problem, you have discovered a blind spot worth taking seriously.

The conversation during the sorting is where most of the qualitative learning happens. As participants move cards around, they think out loud about why one problem matters more than another. They describe situations, frequencies, consequences, and workarounds. That narration is qualitative data the ranking alone cannot give you.

How to

Prep

✦ AI Prompt

1. **Create the pain point cards.** Write approximately 10 cards, each describing one problem, pain, or unmet need in clear, jargon-free language. Use the participant's language, not yours. Each card should describe the problem itself, not a solution. For example: "Spending too much time on manual data entry" instead of "Need an automation tool." Include 2-3 blank cards so participants can add problems you missed.
2. **Recruit 8-12 participants.** Finding useful patterns typically requires 8-12 sessions; 5-6 at least. If you suspect multiple segments, recruit at least 3 from each. See [Discovery Interviews](#) for sourcing and outreach guidance.
3. **Plan session mechanics.** Decide whether sessions are in-person or remote, how you will record (video, audio, or notes-only — get explicit consent before recording), and whether you will compensate participants. Compensation typically ranges from a \$25-100 gift card for B2C consumers to a \$100-300

honorarium for B2B specialists, varying by time commitment and seniority. Document consent and compensation terms in your recruiting outreach.

4. **Prepare the session script.** Prepare an introduction to the exercise, what follow-up questions you will ask, and how you will record the final ranking and key quotes. The script should explain that there are no right or wrong answers and that you want to understand their perspective.
5. **Pilot the deck.** Run one or two pilot sessions before committing to a full round. Pilots almost always surface a card that is too vague, a duplicate, or a problem you did not realize was too solution-focused.
6. **Revise.** If the pilot sessions surface problems, revise the deck and script. Consider using a [Comprehension Test](#) to revise the pain point descriptions.

Execution

1. **Shuffle and present the cards.** Lay all cards face-up on a table (or in a digital whiteboard). Present them in random order to avoid anchoring the participant's thinking. Use the intro script to frame the exercise.
2. **Rank by importance.** Instruct the participant to arrange the cards from most important (top) to least important (bottom). Encourage them to think aloud as they sort if they go silent. Ask them to fill in any blank cards with problems that are missing from the set.
3. **Probe the reasoning.** As the participant sorts, ask follow-up questions: "Why did you put that one higher than this one?" "How often do you experience this problem?" "What happens when this problem occurs?" "How do you currently deal with this?" Do not challenge or correct their rankings.
4. **Take notes.** Note clusters and gaps. Pay attention to cards that participants group together ("these are all related") and cards they set aside as irrelevant ("this is not really a problem for me"). Both patterns are informative. Also note any blank card additions.
5. **Add an intensity check (optional).** After the ranking, you can have the participant allocate a hypothetical sum across the cards to reveal how much they care about each problem, not just the order. For the full version of this technique, see [Buy a Feature](#).

TOOLS

Physical kit (in-person): index cards, a Sharpie, a flat surface — ≤\$10

[Miro](#)

[Optimal Workshop](#)

[Maze](#)

Analysis

[AI Prompt](#)

1. **Aggregate rankings.** After each session, record the final ranking and key quotes. Once you have 8-12 sessions, record each person's final sort as an ordered list, then count patterns across lists for how often each problem placed in the top 3,

top 5, and bottom half. Problems that recur in the top 3 across most participants are strong candidates for the core value proposition.

2. **Group the reasoning quotes.** Pull every participant's quote that explained a high-rank choice into a working doc, then group similar reasons together. The clusters reveal *why* certain problems rank high — frequency, severity, blocked workaround, money lost — and that *why* is more actionable than the rank itself.
3. **Look for divergence as a segmentation signal.** Problems with high variance in ranking (some participants rank them first, others last) may indicate distinct segments with different needs. Look at what those split groups have in common — role, company size, vertical, life stage — and form a segmentation hypothesis to test.
4. **Catalog blank-card additions.** Problems added on blank cards by multiple participants independently are blind spots worth exploring. Group similar additions before deciding whether to add them to the deck on the next round.
5. **Watch for flat allocations.** If you ran the intensity check and a participant spread the amount evenly across the cards, they may not have a strong pain point at all — which is itself an important finding. A flat allocation with a high-ranked top problem usually means the top problem is real but mild.
6. **Note what participants set aside.** Problems that participants set aside as irrelevant may have been based on your assumptions rather than reality. Add to the running list of cards to drop or rewrite.
7. **Treat the data as directional.** With 8-12 participants, individual outliers can skew the results. Look for consistent patterns rather than precise orderings, and treat the data as directional rather than statistically significant.

Biases & Tips

Framing bias

The way you describe each problem on the card influences how participants perceive its importance. Use neutral, descriptive language and avoid emotionally charged framing.

Researcher blind-spot bias

The cards you choose to include define the conversation. If you omit an important problem, blank cards will not reliably surface it — participants rarely volunteer a problem they have never been asked to name. Without prior exploratory interviews, the sort confirms your existing list rather than challenging it.

Anchoring and order effects

The first few cards a participant reads can anchor their thinking, and the wording of the highest-stakes card can shift relative rankings of everything else. Fully randomize card order per session (not just rotate between sessions), and shuffle again before each participant.

Availability
bias

Participants will overweight problems they experienced this week and underweight problems they experience less often but with greater consequences. Probe for frequency separately from rank (“how often does this happen?”) so the rank is not driven entirely by recency.

Social
desirability
and
interviewer
effects

Participants may rank problems they think they “should” care about (security, privacy) or that they think *you* care about higher than problems they actually struggle with. The intensity check partially counteracts this; staying neutral when probing — no nodding, no “great” — counteracts the rest.

Next Steps

- Take the top 3 problems that consistently rank highest across participants and treat them as candidates for your core value proposition.
- If rankings diverge sharply between participants, treat that as a segmentation signal and explore whether you are looking at multiple personas.
- Use [Discovery Interviews](#) to dig deeper into the highest-ranked pains and understand the context, frequency, and consequences.
- Use [Storyboard the Pain](#) to visualize the top-ranked problem in context and confirm you understand the experience around it.
- Create a [Value Proposition Test](#) based on the top 3 problems to confirm the level of interest in a real solution.

Case Studies

Shreyas Doshi: Customer Problem Stack Ranking

Shreyas Doshi (then a product leader at Stripe) formalized the practice of giving customers a candidate problem alongside the other problems they face and asking them to rank, rather than asking whether they like a proposed solution.

[Read more](#)

OpinionX: Stack ranking revealed misaligned priorities

Tested 20 candidate problem statements against their most engaged users, collected nearly 800 votes, and learned the problem they had been prioritizing was not the one users most wanted solved.

[Read more](#)

Adobe Dreamweaver: 00 feature allocation

In the late 1990s the team gave customer advisory board members an imaginary \$100 to spend across candidate features. The forced allocation defined the product's MVP feature set.

[Read more](#)

Further reading

-  Luke Hohmann — Innovation Games: Buy a Feature
-  Greg Guest, Arwen Bunce, Laura Johnson — How Many Interviews Are Enough? An Experiment with Data Saturation and Variability (Field Methods, 2006)
-  Donna Spencer — Card Sorting: Designing Usable Categories (Rosenfeld Media)
-  Card Sorting: Uncover Users' Mental Models — Nielsen Norman Group
-  Three Levels of Pain Points in Customer Experience — Nielsen Norman Group
-  Alex Osterwalder — How Card Sorting Can Help You Understand User Priorities (Strategyzer)
-  Card Sorting — Interaction Design Foundation

3.3 Contextual Inquiry

In Brief

Contextual inquiry combines semi-structured interviews with direct observation of a participant in the actual environment where the problem occurs. You visit the participant's workspace, watch them work, and ask questions in the moment to better understand the work. The output is a rich, qualitative picture of real workflows, tacit knowledge the participant cannot easily articulate, and workarounds that point to unmet needs.

COMMON USE CASE

You have identified a pain point for your prospective user, but you aren't sure about their real behavior and workflows. You visit them in person, watch them act in real situations, and discover behaviors and workarounds they never mentioned in interviews. You want to see the real behaviors so you can design a better solution.

HELPS ANSWER

- What problems does the participant actually experience in their environment?
- What workarounds or substitute products is the participant already using?
- How often does this problem happen in practice?
- What does the participant know about the problem that they cannot easily put into words?
- What steps make up the participant's real workflow?

Description

In contextual inquiry you visit the participant's actual environment, watch the work happen, and ask questions in the moment rather than after the fact. You model an



AT A GLANCE

Qualitative

In-person

Remote

Observational



TIME



~1–2 weeks

COST



\$0–\$600

OTHER NAMES

A Day in the Life · Field Study ·
Ethnographic Interview

expert/apprentice relationship: the participant is the expert in their own work, and you are the apprentice learning it.

Observation produces different data than interviews because participants can only describe what they consciously do; they cannot describe the workarounds, sticky-note hacks, and tacit shortcuts they don't notice themselves doing. A retrospective interview gets you the participant's mental model of their work. A contextual inquiry gets you the work itself — including the parts the participant would never think to mention. Workarounds, environmental constraints, and interruptions tend to appear only when you watch the work happen.

Contextual inquiry rests on four principles that describe how you behave on-site.

Context means going to the real place where the work happens, not a conference room. **Partnership** means treating the session as a shared inquiry — the participant drives the work, you follow and ask. **Interpretation** means voicing your reading of what you just saw and inviting the participant to confirm or correct it. **Focus** means having a defined research question so you notice what matters and don't drown in everything.

The output is qualitative and small-sample. It produces hypotheses, not statistics — pair it with a downstream evaluative method ([Paper Prototyping](#) , [Comprehension Test](#) , closed-ended survey) once the field data has shaped a hypothesis worth testing.

How to

Prep

[AI Prompt](#)

1. **Define your research question.** Create one clear research question. For example, “How do mid-market accountants actually close the books at month-end?” “Learn about accountants” is not sufficiently focused. The question shapes what you'll watch for and which workflow steps you'll ask the participant to walk you through.
2. **Recruit five or more participants in their real environment.** A contextual inquiry done at the participant's desk during their actual workday produces different data than one done over coffee. If the problem only happens at certain times (month-end close, surge periods, mornings), schedule the visit for then. Recruit at least five participants — small samples mean any single observation is anecdote until a pattern repeats.

3. **Train the observer pair.** Pair the lead researcher with a second observer when possible — one focuses on the work, the other captures notes and timestamps. Both should agree in advance on what counts as a workaround worth probing. If you're solo, plan to record the session so you can re-watch instead of trying to take notes mid-observation.
4. **Prepare a framing statement.** Two or three sentences you read at the start of the session: who you are, what you're trying to learn, that you'll mostly watch, and that you may interrupt occasionally to ask what they just did. Make clear they're the expert and you're the apprentice — something like "I'd like to watch you do [task] the way you normally would. I'll mostly watch and ask a few questions along the way."
5. **Prepare an observation checklist, not a script.** Unlike an interview, you're not running a list of questions. You're going to watch real work. The checklist is a reminder of which workflow steps and decision points you want to make sure get covered if they don't surface naturally. Include space for: workarounds noticed, tools or substitute products in use, interruptions, body language during difficult moments.
6. **Get permission to record.** Audio at minimum, video or screen-share if it doesn't change behavior. Promise the participant you'll send them a summary and ask before quoting them anywhere.

Execution

1. **Set the expert/apprentice frame.** Read the framing statement you prepared. Establish that the participant drives the work and you follow. Get explicit consent to record.
2. **Observe first, ask later.** Resist the urge to interrupt. The participant will work through steps that look obvious to them but contain the real signal. Note questions for later — most will be answered in the next two minutes of work without you asking.
3. **Probe in the moment, but only when it's natural.** When the participant hits a workaround, a hesitation, or an interruption, that's the moment to ask. "What just happened there?" or "Why that step?"
4. **Voice your interpretation and let them correct it.** Every fifteen or twenty minutes, summarize what you think you just saw: "It sounded like the reason you copied that into a spreadsheet is because the report doesn't sort the way you need." The participant will either confirm or correct you.

TOOLS

[Dovetail](#)

[Looppanel](#)

[Grain](#)

[Otter.ai](#)

5. **Keep the focus tight.** Participants will volunteer adjacent stories, edge cases, and complaints about other tools. Note them, but bring the conversation back to the workflow you came to observe. The principle is **focus**: a session that drifts produces broad context but no usable signal.
6. **Note what's around the work.** Sticky notes, dual monitors, printed-out cheatsheets, a chat window the participant keeps minimized, coworkers wandering by. The environment is data. A printed-out cheatsheet next to a software interface is the participant's own admission that the software doesn't work the way they need.
7. **Run the remote variant when in-person isn't possible.** For distributed teams or software-only workflows, ask the user to share their screen on a video call while they work through a real task — not a demo, but actual work they'd do anyway. Record with permission. You lose physical context (desk setup, body language, interruptions), but for software products screen-sharing often reveals more about the digital workflow than an in-person visit would.
8. **Wrap with a summary and clarifying questions.** In the last five minutes, summarize the workflow back to the participant and ask any remaining clarifying questions. Confirm permission to follow up by email if you have a question after debrief.

Analysis

✦ AI Prompt

1. **Debrief while it's fresh.** Within an hour of the session, sit down with your notes and the recording and write up: the workflow as observed, every workaround noticed, every place the participant hesitated, every place the environment intruded. Two hours of debrief per session is normal.
2. **Run an interpretation session.** With your team, walk through each session's notes and surface every observation as an explicit interpretation: "When the participant copy-pasted the report into a spreadsheet, that means the built-in sort is unusable for their workflow." This is the team's chance to challenge each interpretation. Disagreement means you don't yet know what you saw and need to look at the recording or follow up with the participant.
3. **Group similar observations together.** Take every observation, every workaround, every interpretation across all sessions and put each on its own sticky note (physical or digital), then group them by what they have in common, not by which session they came from — a technique called affinity diagramming. The clusters that emerge are the patterns. Single-session observations that fit no cluster are anecdotes — note them, but don't build on them.
4. **Identify workflow patterns.** Look across the clusters for repeated structures: the same workaround appearing in different sessions, the same decision point

repeatedly going wrong, the same environmental constraint appearing across different participants. A pattern that shows up in three of five sessions is a pattern. A behavior that shows up in one session is a hypothesis worth probing further.

5. **Be honest about sample size.** Five participants is enough to surface patterns; it is not enough to claim those patterns generalize. State what you observed, in how many sessions, with what variation. Resist extrapolating qualitative observations to the entire population — write findings as “in five sessions we saw X” rather than “users do X.”
6. **Synthesize into a hypothesis worth testing.** The output of analysis is one or two clear hypotheses for downstream evaluative testing — a paper prototype to test against a workaround you discovered, a closed-ended survey to size how widespread the workflow pattern is, a comprehension test on a value proposition you can now articulate. Contextual inquiry doesn’t end the research; it sets up the next round.

Biases & Tips

Confirmation bias

The researcher arrives with a hypothesis and notices only the evidence that supports it. Mitigate by writing your assumptions down before the session and assigning a teammate to actively look for evidence against them.

Observer effect

The participant changes their behavior because they know they’re being watched, especially in the first ten minutes. Mitigate by asking them to start with a routine warm-up task before getting to the work that matters, and by making your framing statement explicit that there’s no right or wrong way.

Leading interpretation

When voicing your interpretation, you can phrase it in a way that traps the participant into agreeing. Mitigate by phrasing interpretations as questions (“It looked like X — is that right?”) rather than statements, and by inviting correction explicitly.

Sample bias

Participants willing to host a contextual inquiry skew toward those who are organized, articulate, and like to demonstrate competence. The people who would benefit most from your product may be the least likely to volunteer. Mitigate by recruiting through multiple channels and noting which segments you couldn’t reach.

Next Steps

- Create or update personas based on observed (not reported) behaviors, and use the workarounds and substitute products you found to inform your value proposition.
- Use a [Solution Interview](#) to explore potential solutions with participants whose workflows you observed firsthand.
- Run evaluative tests such as [Paper Prototyping](#) or a [Comprehension Test](#) on solution concepts inspired by the workarounds you discovered in the field.

Case Studies

Christensen: Milkshake jobs-to-be-done observation

Clayton Christensen's account of observing when and why customers bought fast-food milkshakes, finding they were "hired" as a commute breakfast rather than chosen as a dessert.

[Read more](#)

Lucky Iron Fish: Living with Cambodian families

Researcher Christopher Charles lived with rural Cambodian families to develop an iron-deficiency intervention; Gavin Armstrong later founded Lucky Iron Fish to commercialize the product.

[Read more](#)

Moen: Cameras in shower heads

Moen placed tiny cameras in shower heads of volunteer participants to observe actual showering behavior, then redesigned a shower-head line around the findings; the redesign won the 2013 GOOD Design Award.

[Read more](#)

IDEO: Shopping cart redesign on Nightline

In the 1999 ABC Nightline "Deep Dive" segment, IDEO designers observed how shoppers actually used carts in stores, then redesigned the supermarket cart in a week as a demonstration of observation-driven design.

[Read more](#)

LEGO: Home visits during 2003 near-bankruptcy

Marketing visited an 11-year-old German skateboarder whose worn Adidas sneakers — held up as his proudest possession — revealed that mastery and the social status it conferred mattered more than instant gratification, shifting LEGO's strategy toward complex brick sets.

[Read more](#)

Intuit: Follow Me Home program

Intuit's long-running "Follow Me Home" program sends employees to observe customers using Quicken, QuickBooks, and TurboTax in their own homes and offices to surface workflow interruptions that interviews alone would miss; the team mantra is "look to be surprised."

[Read more](#)

Further reading

-  **Beyer, H. & Holtzblatt, K. (1997).** Contextual Design: Defining Customer-Centered Systems. San Francisco, CA: Morgan Kaufmann.
-  **Holtzblatt, K., Wendell, J. B., & Wood, S. (2004).** Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design. San Francisco, CA: Morgan Kaufmann.
-  **Whiteside, J., Bennett, J., & Holtzblatt, K. (1988).** Usability engineering: Our experience and evolution. In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, NY: Elsevier Science Publishing. 791-817.
-  **Nielsen Norman Group:** Contextual Inquiry — A UX Research Method
-  **InContext Design** — About
-  **Intuit Developer Blog:** Why every company should be doing a Follow Me Home (2021-01-21)
-  **Guiseppe Getto** — Contextual Inquiry with AI (2025)
-  **Nikki Anderson-Stanier** — Behind the Scenes: Contextual Inquiry (Dscout People Nerds)

3.4 Discovery Interviews

In Brief

Customer discovery interviews are one-on-one conversations with potential customers designed to uncover their pain points, current behaviors, and purchasing motivations. You ask open-ended questions about past experiences — not hypotheticals — and listen for patterns. The output is qualitative insight into who your customer really is, what problem they struggle with, and how they already try to solve it, which helps narrow your target market and sharpen your value proposition.

COMMON USE CASE

You have an idea for a product but you are not sure who would actually pay for it. You set up one-on-one conversations with people you think might be customers and ask about their daily frustrations, how they currently solve the problem, and what they have tried before. Their answers help you decide whether to pursue the idea or pivot.

HELPS ANSWER

- Who is the person most likely to buy this?
- What problems are they struggling with right now?
- Where do these potential customers spend their time?

Description

Customer discovery interviews are one-on-one conversations with potential customers to learn what problems they actually have, how they currently solve them, and what they would change. They are generative — you are not validating a feature or testing copy; you are mapping the problem space before you build anything. The



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal



TIME



~2 days–2 weeks

COST



\$0–\$350

OTHER NAMES

Customer Discovery · Customer Interviews · Customer Discovery Interviews · User Interviews

single most important rule of the method is that questions probe past behavior, not future intent. “What did you do the last time this came up?” produces signal; “Would you use a tool that did X?” does not.

The method has several common variations. Most interviews are 1:1, but

 Focus Groups

(one researcher, multiple participants from the same segment) work when you want to surface how participants describe the problem to each other rather than to you. **In-person** sessions are the default in B2B and produce the richest observation data — body language, environment, what is on the participant’s screen — but **remote** sessions are now common and reach people no in-person session ever could. **Structured** interviews follow a tight script and are easier to compare across participants; **semi-structured** interviews keep the script as a backbone but follow whatever thread the participant pulls on, and tend to produce the unexpected findings that change product direction. Most experienced interviewers default to semi-structured.

The output of a good interview round is a sharper customer persona, a ranked list of pains worth investigating, and a much shorter list of assumptions you were carrying that the data did not support.

How to

Prep

 AI Prompt

1. **Define the learning goal.** Write one sentence: “I want to learn whether [target segment] currently struggles with [problem space], and how they handle it today.” If you cannot finish that sentence, you are not ready to interview yet — go back to your customer hypothesis.
2. **Write down your customer-persona assumptions.** What you currently believe about who the potential customer is, what they do, and what they care about. The interviews will test these assumptions; you cannot tell what the data is telling you if you have not stated the prior.
3. **Draft a screener.** Three to four short closed-ended questions that confirm the person you are about to interview matches the segment you are trying to learn about. Ask them on the recruiting form, not at the start of the call.
4. **Write the interview guide.** Eight to ten open-ended questions focused on past behavior, not hypotheticals. The classic backbone covers:
 - “What’s the hardest part about [problem context]?”
 - “Can you tell me about the last time that happened?”

- “Why was that hard?”
- “What, if anything, have you done to solve that problem?”
- “What don’t you love about the solutions you’ve tried?”

Treat the guide as a backbone, not a script. Follow whatever thread the participant pulls on. Include at least one fully open-ended catch-all with no hypothesis behind it — “What else should I know about how you handle [context]?” — since the strongest insights often surface where you weren’t looking. Decide the opening framing here too: a one-sentence summary of the purpose plus a line that there are no right or wrong answers — “I want your real experience, not what you think I want to hear.”

5. **Recruit the interviewees.** Five interviews is a good target for early discovery; 8-12 is the saturation range within a single segment; some practitioners run dozens before drawing conclusions.
6. **Set up logistics.** Confirm consent before recording (legal requirements vary by jurisdiction). Prepare a notes template, the recording tool you will use, and any thank-you gift (a \$5–\$50 gift card is typical for B2C consumers; B2B specialists may expect more or refuse compensation).
7. **Pilot the guide.** Run one practice interview with a colleague or a friendly customer before going live. Pilots almost always surface a leading question, an ambiguous phrasing, or a question whose answer you can already predict.

Execution

1. Frame the interview.

Deliver the opening framing from your guide, then confirm consent for any recording.

2. Qualify.

Ask the screener question first if it wasn’t on the recruiting form. If the participant doesn’t match the segment, end the interview gracefully — bad data is more expensive than a short call.

3. Open with a warm-up.

A low-stakes question about their current role or recent week gets the participant comfortable talking before you ask anything that matters.

4. Listen and follow up.

TOOLS

[Dovetail](#)

[Looppanel](#)

[Grain](#)

[Respondent](#)

[Great Question](#)

Let the participant talk. Use “what” and “how” follow-ups; avoid “why” early in the conversation (it can feel accusatory). When a participant mentions a specific incident, slow down and dig into that incident — concrete past events produce far more reliable data than generalizations.

5. Close.

Wrap up by asking if there is anything you should have asked but didn’t. Ask for referrals to other people in the same segment. If applicable, ask permission to follow up.

6. Debrief immediately.

Write notes within an hour while the conversation is fresh. An AI transcription service turns this around in minutes, but read the transcript critically — current models still mis-attribute speakers and hallucinate filler words.

Analysis

✦ AI Prompt

Review the transcript and/or listen to the recording, and capture data on each of the following:

- **Job:** What activities are making the participant run into the problem?
- **Obstacle:** What is preventing the participant from solving their problem?
- **Goal:** If they solve their problem, then ____?
- **Current solution:** How are they solving their problem?
- **Decision trigger:** Were there pivotal moments where the participant made key decisions about a problem?
- **Interest trigger:** Which questions did the participant express interest in?
- **Persons:** Are there any other people involved with the problem or solution?
- **Emotions:** Is there anything specific that causes the participant to express different emotions?
- **Measurement:** How is the participant measuring the cost of their problem?

⚠ Biases & Tips

Leading questions

Phrasing that telegraphs the answer (“Wouldn’t it be easier if...?”) gets you agreement instead of truth. Ask what the participant actually did, in neutral wording, and have a colleague or AI flag leading phrasings in your guide before you run it.

Social desirability bias

Participants offer compliments and say what they think you want to hear, especially friends, family, and anyone who knows the idea is

yours. Treat praise as a warning sign and steer every answer back to specific past events — what they did, when, and what it cost them.

The pitch trap

The moment you describe your idea, you contaminate every answer that follows. Hold the solution back until the end of the conversation and aim to listen far more than you talk.

Confirmation bias

You hear the evidence that supports your idea and quietly discount the rest. Write your assumptions down before interviewing (Prep step 2), then read each transcript hunting for what contradicts them.

Future-intent bias

People are poor forecasters of their own behavior, so “would you use this?” produces false positives. Anchor every question in what actually happened the last time the problem came up, and discard speculative enthusiasm.

Selection bias

Interviewing whoever is easiest — your own network, enthusiasts, existing customers — skews toward people who already like you or the space. Recruit through several channels, screen for the real target segment, and note which segments you could not reach.

Anchoring and order effects

The sequence in which you ask questions can shape the answers, because an early topic anchors how the participant frames later ones. Vary the question order across interviews so a warm-up topic isn't always priming the same downstream response.

Next Steps

- Identify patterns across interviews and update your customer persona.
- If a clear pain point emerges, move to a [Solution Interview](#) or [Value Proposition Test](#) to explore and validate a solution.
- Run a [Closed-Ended Survey](#) to quantify how widespread the pain points are and which segment is largest.
- Share interview recordings or transcripts with the team, grouping related notes together to align on insights.

Case Studies

Gothelf: Discovery cadences at Winware, Vistaly, and Penpot

Jeff Gothelf profiles three startups running systematic discovery: Winware's Steven Cohn did 300+ interviews pre-build at 2–4 per day, Vistaly's Matt O'Connell runs 15–20 weekly, and Penpot's Pablo Ruiz-Múzquiz holds 30 interviews per week to validate features.

[Read more](#)

Airbnb: Hosts' homes in New York, 2009

In 2009 co-founders Brian Chesky and Joe Gebbia traveled to New York to stay with hosts and interview them; observing how hosts struggled to create listings led to the team personally photographing apartments, and revenue in New York is reported to have roughly doubled within a month.

[Read more](#)

Lean Foundry: “Steve” reframes 50-interview VR research

A founder ran 50 interviews and 200+ surveys about a VR rendering tool with no clarity; a single 30-minute past-behavior conversation about a recent client presentation surfaced the real problem — \$400, three-day revision cycles — and pivoted the product from better rendering to instant self-serve revisions.

[Read more](#)

Further reading

-  The Mom Test
-  The Customer Discovery Handbook
-  How I Interview Customers
-  Greg Guest, Arwen Bunce, Laura Johnson — How Many Interviews Are Enough? An Experiment with Data Saturation and Variability (Field Methods, 2006)
-  Great Lakes I-Corps Hub — How to Use AI for Customer Discovery
-  What are your favorite methods for doing problem interviews during customer discovery?

3.5 Data Mining

In Brief

Pull customer segments, pains, and demand signals out of data you already have — your support inbox, your website, public search trends, app-store reviews — instead of running new interviews or surveys. The work happens on three sub-method pages; pick the one matching the data source you can reach.

COMMON USE CASE

You suspect there is signal hiding in data you already have access to — your own product analytics, support inbox, web traffic, or public sources like search trends and app-store reviews — but you have not systematically mined it. Before commissioning new interviews or surveys, you want to extract whatever segments, pains, and demand signals are already sitting in existing data, then pick a sub-method based on the data source you can reach.

HELPS ANSWER

- Which existing data source is most likely to answer my current question?
- Are there customer segments or pain patterns I can see without running new research?
- Is there enough demand signal in the market to justify building anything?
- What patterns recur across customer behavior, complaints, or searches?
- Where should I focus deeper qualitative research?

Description

These methods read patterns out of data created for some other purpose — support tickets, web analytics, search-engine query volume, app-store reviews, transaction logs — rather than collecting new data through customer interaction. You look for



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~3 hours–1 week

COST  \$0–\$120

OTHER NAMES
Data Analysis

segments, pains, and demand signals in records you already have. Because the data exists, these methods are usually faster and cheaper than primary research, and they often surface candidate segments or pains you can then validate in interviews.

Choosing the Right Data Mining Method

The three methods below differ in which data source they read. Pick the one matching the data you have access to (or can acquire most cheaply). Founders often run two or three in sequence — for example, mining support tickets to find a recurring pain, then checking search-trend data to confirm the pain has visible market demand.

Available Data Sources

Customer Support Analysis

Mines support tickets, chat transcripts, and app-store reviews to surface common pains and the segments suffering from them. Best used when you already have a product in market with meaningful inbound support volume to cluster.

Search Trend Analysis

Counts query volume on terms related to the problem, using search-engine trend tools and AI-question platforms. Best used when you do not yet have customers and need to size demand for a problem before building.

Web Traffic Analysis

Mines visitors to an existing website for referral sources, on-site search, conversion funnels, and drop-off patterns. Best used when your site already has meaningful traffic; on low-traffic sites the patterns will be noise.

Whichever sub-method you run, the tools that store and query the underlying data are shared and the sub-method pages list only the tools specific to their data source.

Popular tools change frequently.

3.5.1 Customer Support Analysis

In Brief

Categorize your support tickets, chat logs, reviews, and help desk interactions to find the recurring customer problems, unmet needs, and feature gaps hiding in data you already own. The output is a ranked read on which problems are most frequent and most severe, by segment — grounded in what customers actually contacted you about rather than one agent’s impression.

For insights from individual support staff based on their accumulated experience, see

[Ask an Expert](#)

COMMON USE CASE

You have at least six months of support tickets, chat transcripts, or app-store reviews — typically 500+ interactions — and you want to know which customer segments are hurting and why before commissioning new research. You suspect the most important problems are not the ones your team talks about most, and you want a systematic read on frequency, severity, and segment before you pick the next discovery interview or product bet.

HELPS ANSWER

- What are the most common problems customers experience?
- Which unmet needs or feature gaps do customers express most frequently?
- Are customer complaints getting better or worse over time?
- Which customer segments have the most support issues?
- What is the root cause behind surface-level symptoms customers report?
- Where is the biggest opportunity to reduce friction and improve satisfaction?



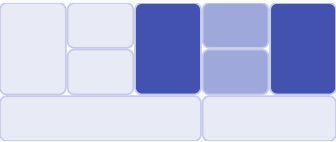
AT A GLANCE

Quantitative

Qualitative

Remote

Behavioral



TIME  ~1 hour–1 day

COST  \$0–\$25

OTHER NAMES

Support Ticket Analysis ·

Customer Support Data Mining

Description

Customer support analysis extracts patterns from your support tickets, chat transcripts, and reviews — the [Data Mining](#) method whose data source is your existing support and feedback channels. Support interactions are a rich source of customer insight: every ticket, chat, email, and review represents a moment where a customer cared enough about a problem to ask for help. In aggregate, these interactions reveal systematic patterns that no single conversation can surface.

The goal is to discover what customers actually struggle with, not to test a hypothesis you already hold. The findings often challenge internal assumptions about what matters most — a product team may be convinced its onboarding flow is clear until support data shows that a large share of first-week tickets cluster on the same confusing step.

However, the data is all from existing customers of the existing product, so novel ideas or alternative pain points will rarely come up. All the data is grounded in the customer's current perceptions and experience of your product.

How to

Prep

[AI Prompt](#)

1. **Export or connect your support data.** Pull a representative sample of support tickets, chat transcripts, customer reviews, and any other feedback data. Aim for at least 500 interactions over a meaningful time period (3-6 months). Include metadata: date, customer segment, product area, resolution time, satisfaction score. If you have multiple channels (tickets, chat, app-store reviews, social), decide whether to mine them separately or merge — merging gives a fuller picture but requires a unified taxonomy (note that different channels attract different customers: the user who complains on Twitter rarely files a help ticket, so a single-channel read can miss an entire complaint mode).
2. **Design the taxonomy from a sample.** Before you classify the full dataset, hand-read 50-100 tickets and draft a taxonomy of issue categories (billing, onboarding, feature requests, bugs, how-to questions, account access, performance). Keep the taxonomy shallow — 8-15 top-level categories is enough to start. Most support platforms have built-in tags you can build on, but treat the existing tags as a starting point, not the truth. You can draft the taxonomy with AI, but read a sample of tickets yourself in parallel and reconcile the two — relying on the model alone bakes in its training biases.

Execution

✦ AI Prompt

TOOLS

[Intercom](#)

[Zendesk](#)

[Enterpret](#)

[Claude](#)

1. **Classify the full dataset.** Apply the taxonomy from Prep to the full export. Use AI to classify the bulk of tickets and spot-check 5-10% by hand to confirm the AI is not systematically miscategorizing. Where the AI is unsure, look at the raw ticket text — those edge cases often reveal a missing category or a fuzzy boundary in the taxonomy.
2. **Quantify frequency and severity.** Count how often each category appears and assess the severity of each issue type. A frequently occurring minor annoyance may matter more than a rare critical bug. Use a simple impact matrix: frequency on one axis, severity on the other. Severity can come from satisfaction scores, escalation rates, resolution time, or refund/churn signals tied to a ticket.
3. **Score the emotional tone.** Assess the sentiment of support interactions — how frustrated, urgent, or negative each one reads. Identify which issue categories generate the most negative tone. Sentiment trends over time can indicate whether problems are getting better or worse, independent of ticket volume.
4. **Track feature requests and unmet needs.** Separate explicit feature requests from implicit ones. An explicit request is “I wish the product could do X.” An implicit request is a workaround described in a support ticket — the customer found a way to accomplish something the product does not natively support. Workarounds are often more valuable than direct requests because they show what the customer is willing to do to get the job done.
5. **Detect trends over time.** Plot the volume of each issue category over time. Rising trends point to growing problems. Sudden spikes may correlate with product releases, marketing campaigns, or seasonal patterns. A declining trend after a fix is a sign the fix is working, though confirm it is not just seasonal.
6. **Segment the patterns.** Cross-tabulate categories against customer segment, plan tier, signup cohort, and geography. The same product often produces very different support footprints across segments — a category that is invisible at the aggregate level may dominate one cohort.

Analysis

✦ AI Prompt

1. **Perform root cause analysis on the top categories.** For the top 5 issue categories, dig deeper. Read 20-30 representative tickets in full. Identify whether the root cause is a product design problem, a documentation gap, a user education issue, or a genuine bug. The same symptom (e.g., “I can’t log in”) may have multiple root causes.

2. **Synthesize into actionable hypotheses.** Compile your findings into a prioritized list of customer pain points and unmet needs. For each, note the frequency, severity, affected segment, and likely root cause. Use these hypotheses to inform product roadmap decisions or as input for [Discovery Interviews](#) to explore the most impactful issues in depth.

3. **Interpret the patterns.**

- The most frequent support topic is not always the most important. A frequently asked “how-to” question may be solved with better documentation, while a less frequent but severe issue may be driving churn.
- Feature requests should be interpreted as expressions of unmet needs, not as design specifications. The customer is describing their problem, not the best solution.
- A spike in negative sentiment after a product release is a strong signal that something in the release is causing friction.
- If a large percentage of tickets come from new users, the onboarding experience likely needs attention.
- Workarounds described in tickets are revealing — they expose a job the customer is trying to get done that your product does not yet address.

 Biases & Tips

Selection bias

Only customers who bother to contact support are represented. Many dissatisfied customers leave silently without ever filing a ticket. Pair support analysis with churn data and an [Open-Ended Survey](#) of inactive users to check the silent majority.

Squeaky wheel bias

Vocal, persistent customers may be overrepresented in the data. Their issues may not reflect the majority of your user base. Weight findings by unique-customer count, not ticket count.

Agent interpretation bias

How support agents categorize and summarize tickets can introduce distortion. Different agents may categorize the same issue differently. Always re-classify with your own taxonomy rather than trusting existing tags.

Recency bias

Recent tickets are easier to access and may be given more weight than older patterns that are equally important. Sample evenly across the full time window.

Survivor bias

Support data only captures issues from current customers. Customers who churned due to a problem may not have contacted

support about it. Cross-check with exit surveys and cancellation reasons.

**Taxonomy
drift**

As you read more tickets, the categories you drafted from the first 50 will start to feel wrong. Allow yourself one taxonomy revision halfway through, then freeze it. Repeated recategorizing delays the decision without improving it.

Next Steps

- Turn your top 5 issue categories into a prioritized hypothesis list with frequency, severity, and likely root cause for each.
- If a single category dominates ticket volume, treat it as the next focus area and decide whether the fix is product, documentation, or onboarding.
- Use [Data Mining](#) to combine support data with other behavioral sources for a more complete picture of the customer experience.
- Use [Web Traffic Analysis](#) to see whether the same friction points show up in on-site behavior, drop-offs, and internal search.
- Use an [Open-Ended Survey](#) to reach customers who never filed a ticket and check whether the patterns hold beyond the support channel.
- Use [Discovery Interviews](#) to explore the highest-priority pain points in depth with affected customers.

Case Studies

Intercom: Swarms model for customer-conversation insights

Intercom describes cross-functional “swarms” of engineers, data scientists, and PMs that work alongside customers trialing Fin, mining conversation data for patterns that become operational tools and then scalable product features such as its automation taxonomy and CX Score.

[Read more](#)

Productboard: Aggregating Intercom transcripts into insights

Productboard’s writeup describes a five-step workflow — tag, consolidate, categorize, analyze, align — used by 600+ organizations to pull Intercom support transcripts into a central insights repository and tie them to roadmap items.

[Read more](#)

Airbnb: Elasticsearch trend dashboard for support tickets

Airbnb engineering built an Elasticsearch-backed dashboard that ingests support tickets and uses Fourier-transform-smoothed scoring to detect emerging spikes; the system caught a regression hiding listings from search and is estimated to have reduced overall ticket volume by ~3%.

[Read more](#)

Further reading

-  The Voice of the Customer — Griffin & Hauser, Marketing Science (MIT Sloan mirror)
-  How to Analyze Qualitative Data from UX Research: Thematic Analysis — Nielsen Norman Group
-  Customer Feedback Analysis: How to Turn Customer Insights Into Action — Productboard
-  Tony Ulwick — Outcome-Driven Innovation and the Jobs-to-be-Done Theory
-  Cole Nussbaumer Knaflitz — Storytelling with Data
-  From swarms to product: turning customer signals into scalable features — Intercom

3.5.2 Search Trend Analysis

In Brief

Search trend analysis reads publicly available query data to learn what people are actively looking for online – across both traditional search engines and AI assistants. By examining search volume, trending queries, geographic patterns, seasonal swings, and the questions people put to LLMs, you can spot demand and emerging topics before committing resources to building a product.

COMMON USE CASE

You don't have customers yet – or you want to size demand outside your existing customer base – and you want to know whether the problem you're investigating shows up in real-world demand signal. You check both traditional search query volume and the AI-question trend space. They often diverge: people ask LLMs and search engines about the same problems in different ways, and the two surfaces produce different signals.

HELPS ANSWER

- Is there existing demand for what I want to build?
- Is interest in this topic growing, declining, or stable?
- What are people asking AI assistants about this problem, and how does it differ from what they search?
- Are there seasonal patterns that affect demand?
- What language and terminology do potential customers actually use?
- Where geographically is demand concentrated?
- What related topics or adjacent problems are people searching for?



AT A GLANCE

Quantitative

Remote

Observational

TIME

 ~1 hour–1 day

COST

 \$0–\$40

OTHER NAMES

Google Trends Analysis · Keyword Research · AI Question Trend Analysis

Description

Search trend analysis reads demand signal out of what people search for — both in traditional search engines and in AI assistants. This is one of the [Data Mining](#) methods: you work existing query data rather than running new interviews or surveys. Billions of people use search engines and LLMs every day to express their needs, problems, and interests. Each query is a small signal of demand. Aggregated, these signals reveal patterns that can inform early-stage product decisions, and aggregated search data has been shown to track real-world economic activity ahead of official statistics.

Unlike surveys or interviews, search data captures what people actually do rather than what they say they do. A person searching “how to fix leaky faucet” has a real, immediate problem. A rising trend in “AI writing assistant” indicates growing market interest. A seasonal spike in “tax software” reveals timing opportunities. What “search” means is also widening: even outside Google, a large share of search activity happens on platforms like YouTube, Amazon, Reddit, and Wikipedia. Treat search as a behavior that shows up wherever your audience hunts for answers, not just on one engine.

The method treats two surfaces as parallel data sources:

- **Traditional search-engine trends.** Trend and keyword-volume tools show how often people type queries into search engines, where they are typed, and how that volume changes over time. Regional engines (Baidu in China, Yandex in Russia) matter where the audience is concentrated outside Google’s footprint.
- **AI question trends.** A growing share of problem-aware queries are now addressed to AI assistants instead of search engines. People phrase questions to an assistant differently than they search Google — fuller sentences, more context, more comparative phrasing (“which is better for X”). AI-question-trend tooling is emerging: a small set of platforms track which queries surface in AI-mediated answers, which sources the assistants cite, and how visibility shifts over time. Public AI-question datasets are still thin — a lot of the practical work is direct experimentation, asking the same question across multiple assistants and noting how each answers and what each cites.

The goal is not to validate a specific hypothesis but to map patterns of demand and discover opportunities you may not have considered. Comparing the two surfaces is itself diagnostic: if a topic has high search volume but no AI-question presence, the problem is mature and AI assistants may be eating into the channel; if a topic has

heavy AI-question activity but flat search volume, you may be early to a shift in how the audience finds answers.

How to

Prep

✦ AI Prompt

1. **Define your topic area.** Pick 3–5 seed keywords related to the problem space. Use both broad terms (“project management”) and specific terms (“kanban board for remote teams”). Include terms your customers might use, not just industry jargon.
2. **Expand seed terms across both surfaces.** Search-engine queries are short and keyword-shaped (“project management remote team”); LLM queries are longer and conversational (“what’s the best way to run sprint planning when half my team is async”). Generate variants of each seed for each surface — they are not the same list. AI is useful here: you can ask an LLM to rewrite each seed as five plausible Google queries and five plausible AI-assistant-style questions a real founder, ops lead, or end customer would ask.
3. **Set up your toolkit.** For traditional: open Google Trends and a keyword-volume tool. For AI-question signal: pick at least one AI-search monitoring tool and at least two LLMs you can prompt directly.

Execution

✦ AI Prompt

1. **Run Google Trends across the traditional list.** Enter your search-engine queries into Google Trends and examine the interest-over-time graph. Compare multiple terms side by side. Use a 5-year window for trend context, then narrow to 12 months for current shape. Adjust the geographic scope to your target market. Remember Google Trends reports a normalized 0–100 index of relative interest, not absolute query counts — read it as relative shape over time, not a volume number.
2. **Pull search volume estimates.** Use Google Keyword Planner (free with a Google Ads account) or another keyword-volume tool to get approximate monthly volumes for your most promising keywords. Exact volumes matter less than relative comparisons and trend direction.
3. **Examine related queries and topics.** In Google Trends, scroll to “Related queries” and “Related topics.” Look at both “Top” (highest volume) and “Rising” (fastest

TOOLS

[Google Trends](#)
[Google Keyword Planner](#)
[Semrush](#)
[Exploding Topics](#)
[Profound](#)

growing). Rising queries with “Breakout” status indicate growth above 5,000% — a strong early signal worth investigating.

4. **Analyze geographic distribution.** Check the “Interest by subregion” map to understand where demand is concentrated. This can inform market entry, language requirements, and competitive landscape. If your target audience is outside Google’s strong markets, repeat with the relevant regional engine (Baidu, Yandex, Naver).
5. **Run the AI-question list directly.** Take your conversational queries and ask the same question to at least two LLMs, including at least one that cites its sources. For each, capture: how the LLM frames the answer, what sources it cites, what alternatives it surfaces, and which competitors or solutions it names by default. The cited sources tell you who currently owns the AI-mediated answer.
6. **Pull AI-question monitoring data.** In your AI-search monitoring tool, check which of your queries appear in AI Overviews / AI-mediated answers, how visibility is trending, and which domains are cited. Where tooling is gated behind a sales call, the free reports and blog posts these vendors publish often include enough trend data to use.
7. **Capture the divergence.** Where a topic is heavy in traditional search but absent in AI-question signal (or vice versa), record it. Divergence is signal: it tells you which channel the audience is shifting toward.

Analysis

✦ AI Prompt

1. **Cluster the keyword landscape.** Organize findings into clusters by buyer awareness stage: problem-aware (“how to reduce churn”), solution-aware (“customer retention software”), and brand-aware comparison (“Intercom vs Zendesk”). Do this for both surfaces — the cluster shape often differs between Google and LLMs.
2. **Compare the two surfaces side by side.** For each cluster, note: traditional search volume and trend direction, AI-question prevalence (cited frequency, AI Overview presence), and which sources the LLMs default to. Topics where both are rising are the strongest demand candidates. Topics where AI-question signal leads but search is flat may be a shift to watch. Topics where search leads but AI assistants don’t engage may indicate a transactional or branded surface that LLMs aren’t yet eating into.
3. **Synthesize and generate hypotheses.** Combine findings into a summary that answers: What demand exists across both surfaces? Is it growing? What language do customers use in each channel? What adjacent opportunities exist? Use these hypotheses as input for further research such as

📖 [Discovery Interviews](#) .

When reading the results, keep these interpretation guidelines in mind:

- A sustained upward trend over 2+ years on either surface suggests growing demand rather than a short-lived fad.
- A flat or declining traditional search trend does not necessarily mean no opportunity – the market may be mature, the audience may have moved to AI assistants, or customers may use different terminology.
- Heavy AI-question presence with thin search volume often indicates an early shift in how the audience finds answers. The audience may be there; the channel has changed.
- High search volume with few quality results on the search results page suggests an underserved market on the traditional side.
- Rising “how to” and “best” queries – and conversational “which should I” questions to LLMs – indicate a market where customers are actively seeking solutions.
- Geographic concentration may indicate cultural, regulatory, or infrastructure factors worth investigating.
- Seasonal patterns can inform launch timing and marketing strategy.

Biases & Tips

Search engine bias

Google Trends only captures Google search behavior. Significant search activity also happens inside YouTube, Amazon, Reddit, Wikipedia, TikTok, Baidu, and Naver. Check the right surface for your audience instead of assuming Google represents the whole picture.

AI-question dataset thinness

Public AI-question trend data is still sparse. The available vendor tools each cover a partial slice and disagree with each other. Treat any single AI-question source as directional, not authoritative.

Survivorship bias in keywords

You can only analyze terms you think to search for. Use AI to expand your seed list before you commit to a query set; demand signals may exist under terminology you would not have considered.

Demand-signal-to-revenue conflation

High search volume for free-tier or pain-point queries does not validate a paid product. Readers routinely over-read volume as purchase intent: “free project management tool” has strong demand signal but zero implication that searchers will pay.

Correlation vs. causation

A rising trend may be driven by media coverage or a viral event rather than sustained organic demand. Always check whether a spike has a news cause.

B2B blind spot

Business buyers often rely on peer recommendations, analyst reports, and sales conversations rather than search engines. AI assistants are starting to mediate B2B research too, but B2B demand is still underrepresented in both surfaces.

Recency bias

Recent spikes can look like trends. Always examine multi-year timeframes before drawing conclusions on the traditional side. On the AI-question side, the data window is often shorter — be honest about the small sample.

LLM-answer bias

When you query LLMs directly, the answer reflects the model's training data and ranking, not necessarily user behavior. LLMs may confidently describe a set of problems that reflects what was written about the topic, not what real users actively seek, making demand appear larger or better-defined than it is.

Next Steps

- Turn the strongest cross-surface clusters and rising queries into a shortlist of demand hypotheses, ranked by combined trend direction and AI-question presence.
- If a “Breakout” query or a sharp AI-question divergence stands out, treat it as a candidate problem space and dig into the customer language behind it.
- Use [Web Traffic Analysis](#) on your own site or a competitor's to see whether search and AI demand is converting into actual visit and engagement behavior.
- Use a [Social Media Campaign](#) to test whether the language and angles you discovered actually resonate with a real audience.
- Use [Secondary Market Research](#) to size the market behind the trend and check it against industry reports and existing studies.

Case Studies

Glossier: Audience-data-first launch

Emily Weiss grew Into The Gloss to over a million monthly visitors before launching Glossier in 2014 around a small, opinionated SKU list informed by six months of audience research across content, community, and commerce.

[Read more](#)

Exploding Topics: Surfacing early-rising search queries

Exploding Topics monitors millions of data points across search, social, forums, news, and e-commerce and combines ML signal detection with human curation to surface trends the platform claims appear 12+ months before mainstream awareness.

[Read more](#)

BrightEdge: Overdrive Interactive achieves 710% AI Overview growth

BrightEdge's case study on how Overdrive Interactive used AI Overview and LLM-citation tracking to grow AI Overview mentions 710% in three months.

[Read more](#)

Further reading

-  Predicting the Present with Google Trends — Hyunyoung Choi and Hal Varian, Google Research
-  Google Trends Help — How Trends data is normalized
-  SparkToro — Search Happens Everywhere (Rand Fishkin, 2026)
-  Search Engine Land — AI SEO coverage
-  BrightEdge — Resources hub

3.5.3 Web Traffic Analysis

In Brief

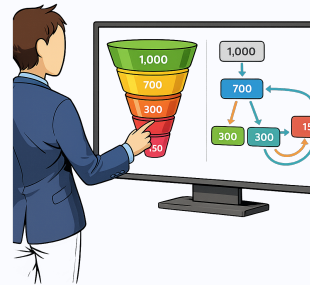
Web traffic analysis mines the analytics data your site already collects to learn how visitors behave, where they come from, and what they are looking for. By reading traffic sources, user flows, conversion funnels (the steps a visitor takes toward a goal like signing up or buying), and internal search queries, you surface demand signals, underserved customer segments, and product gaps without building anything new.

COMMON USE CASE

You already operate a website, landing page, or product that is collecting analytics data, and you suspect there is segmentation, drop-off, or unmet-demand signal in it that you have not systematically extracted. Before commissioning new interviews or surveys, you want to mine what is already there: which sources bring engaged visitors, where the funnel leaks, and what visitors are searching for that you do not yet offer. If you do not have your own traffic, you can apply the same method to a competitor's site through a competitor-traffic estimation tool to understand market dynamics before you build.

HELPS ANSWER

- Where are our visitors coming from and what does that reveal about our channels?
- What content or pages do visitors engage with most?
- Where in the user journey do visitors drop off?
- What are visitors searching for on our site that we do not currently offer?
- Which customer segments behave differently from one another?
- What referral sources or campaigns drive the most engaged visitors?
- Are there demand signals hidden in our existing traffic data?



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~1 hour–1 day

COST  \$0–\$30

OTHER NAMES

Web Analytics · Traffic Analysis

Description

Every visitor leaves a trail on a site you (or a competitor) operate: the page they landed on, how long they stayed, what they clicked, what they searched for, and where they left. Read in aggregate, those trails reveal patterns about customer needs, channel effectiveness, and product gaps. It is one of the [Data Mining](#) methods, working the analytics data your site already collects rather than commissioning new research. The goal is to discover patterns and generate hypotheses, not to confirm one you already hold — so you are looking for surprises: unexpected traffic sources, pages with unusually high engagement, internal search queries that point at unmet needs, or segments that behave differently from what you expected.

The patterns only hold up at scale. If your site gets fifty visits a month, you are reading individual sessions, not aggregate behavior, and the read is noise. Below the traffic threshold set in Prep, the sibling [Data Mining](#) methods draw on data that does not require traffic of your own and will give you a cleaner signal.

Read behavioral metrics tied to a decision — conversion by segment, drop-off step, intent of the source — rather than totals like pageviews and unique visitors, which rarely change what you do next. If you do not have your own site yet, a competitor-traffic tool lets you read another site's traffic patterns to understand market dynamics and channel opportunities before you launch.

How to

Prep

✦ AI Prompt

1. **Confirm you have enough traffic to mine.** As a rule of thumb, you need at least 1,000 sessions per month, and ideally 5,000+, before traffic analysis produces reliable patterns rather than anecdotes. If you are below this threshold, switch to [Customer Support Analysis](#), [Search Trend Analysis](#), or analyze a competitor's traffic with a competitor-traffic tool instead.
2. **Identify your data sources.** Determine what analytics data you have access to. This may include your web analytics platform, product analytics events, heatmap and session-recording tools, server logs, or platform-specific analytics. Confirm the tagging and event instrumentation are in place for the funnels you care about — if conversion events are not being tracked, no amount of analysis will surface them.
3. **Decide segments and date range.** Pick a date window long enough to absorb weekly seasonality (at minimum a full month, ideally a quarter) and define the

source, and any custom dimensions you track. Decide which conversion goals and funnels matter most before you start digging — this prevents the analysis from wandering.

Execution

✦ AI Prompt

1. **Audit traffic sources.** Examine where visitors come from: organic search, paid ads, social media, referral sites, direct traffic, email campaigns. Look for unexpected sources that perform well and expected sources that underperform. Note which sources bring the most engaged visitors (not just the most visitors).
2. **Analyze top content and landing pages.** Identify which pages receive the most traffic, have the longest time-on-page, and have the lowest bounce rates (the share of visitors who leave after viewing a single page without interacting). These pages reveal what your audience cares about most. Pay special attention to pages that rank well organically, as this indicates alignment with real search demand.
3. **Examine user flow and navigation paths.** Use flow visualization tools to see the most common paths visitors take through the site. Look for unexpected navigation patterns — where do visitors go that you did not intend? Where do they loop or backtrack?
4. **Analyze internal site search.** If your site has a search function, the queries visitors type are direct expressions of what they want. Categorize internal search terms into themes. Searches with zero results are particularly valuable — they represent unmet demand.
5. **Map conversion funnels and drop-off points.** Define key conversion paths (signup, purchase, contact form) and identify where the largest drop-offs occur. High drop-off at a specific step often indicates a UX problem, a trust gap, or a mismatch between visitor expectations and what you offer.
6. **Segment your visitors.** Break traffic down by device type, geography, new vs. returning, traffic source, and any custom dimensions you track. Compare how segments behave differently. A segment with high engagement but low conversion may need a different value proposition or user experience.

TOOLS

Google Analytics 4

PostHog

Mixpanel

Amplitude

Hotjar

SimilarWeb

Microsoft Clarity

Analysis

✦ AI Prompt

1. **Synthesize findings into hypotheses.** Compile your observations into a set of hypotheses about customer behavior, channel opportunity, and product gaps. Prioritize hypotheses by potential impact and ease of further testing. Use

[Discovery Interviews](#) or [Search Trend Analysis](#) to investigate the most promising patterns.

2. **Read funnel drop-offs as expectation gaps.** Pages with high traffic but high bounce rates usually indicate a mismatch between what visitors expect (from the search result or ad) and what the page delivers. The drop-off step in a funnel is where the visitor's mental model and your offer diverge — fix the page copy, the offer, or the entry point, not just the button.
3. **Treat zero-result internal searches as unmet demand.** Internal search queries with no results are direct signals of what visitors want but cannot find. Cluster the queries into themes; recurring themes are candidate features, content, or product directions worth validating with interviews before committing engineering effort.
4. **Look at engagement-vs-conversion mismatches by segment.** A segment with high engagement but low conversion may need a different call-to-action, pricing model, or trust signal. A segment with low engagement but high conversion is probably arriving with high intent — protect that source. Returning visitors who do not convert may be in a longer consideration cycle, common in B2B and high-ticket purchases.
5. **Flag traffic from unexpected referrers.** Traffic from an unexpected referral source — a niche subreddit, an industry newsletter, a forum thread — often points to an audience segment you have not considered. Check the engagement metrics on that source before deciding whether it is signal or accident.
6. **Watch for seasonal and day-of-week patterns.** Weekly and seasonal traffic patterns can inform content publishing and campaign timing, and they can also expose whether a “growth” trend is real or just calendar drift.

Biases & Tips

Survivorship bias

You can only analyze visitors who reached your site. You have no data on the people who needed your product but never found you. Pair traffic analysis with [Search Trend Analysis](#) to see demand you are missing entirely.

Attribution gaps

Multi-touch customer journeys mean the “last click” source may not be the most important influence. A visitor attributed to “direct” may have originally discovered you through a podcast or word of mouth.

Privacy and ad blockers

A growing percentage of visitors block analytics scripts, which means your data may underrepresent privacy-conscious segments. Cross-check against server logs or a privacy-friendly analytics tool if this matters for your audience.

Session-
recording
over-
interpretation

At low traffic volume, individual session recordings feel like customer interviews but are unmoderated, context-free, and unverified. They reveal what a visitor did, not why — which makes them hypothesis generators, not confirmation evidence.

Next Steps

- Compile your top behavioral findings into a prioritized list of hypotheses, with each tagged by potential impact and how easy it would be to test next; a [Prioritization Game](#) can help rank them with stakeholders.
- If internal search queries surfaced unmet demand, treat those terms as candidate problem spaces to investigate further with [Discovery Interviews](#) before any new build.
- Use [Search Trend Analysis](#) to see whether the demand signals you spotted on-site also show up in broader search behavior.
- Use [Customer Support Analysis](#) to cross-reference high drop-off pages with the kinds of issues customers raise in tickets.
- Use [Discovery Interviews](#) to validate the strongest segment and demand hypotheses with the actual humans behind the traffic.
- Use [Dashboards](#) to keep the most important traffic and funnel metrics visible over time so you can detect changes early.

Case Studies

Airbnb: ML-powered search ranking for Experiences

Airbnb engineering published a four-stage evolution of search ranking for Experiences, scaling from 50K training examples to 2M+ and lifting bookings ~13%, +7.9%, and +5.1% across the stages while incorporating personalization, real-time query features, and quality-weighted business rules.

[Read more](#)

Buffer: Public revenue dashboard

Buffer launched a public revenue dashboard in April 2020 showing annual run rate, average revenue per account, and customer count, sourced from ChartMogul; it sits alongside published salaries, diversity data, and an open roadmap.

[Read more](#)

HubSpot: Historical-optimization of evergreen blog posts

HubSpot's analysis found 76% of monthly blog views and 92% of monthly leads came from posts published before that month, with 46% of leads from just 30 posts; systematic re-optimization tripled leads on updated posts and lifted organic search views on optimized posts by an average of 106%.

[Read more](#)

Further reading

-  Avinash Kaushik — Web Analytics 2.0
-  Eric Ries — The Lean Startup
-  Brian Clifton — Advanced Web Metrics with Google Analytics, 3rd ed.
-  Google — GA4 Funnel Exploration documentation
-  Cole Nussbaumer Knaflic — Storytelling with Data

3.6 Experience Sampling

In Brief

Experience sampling prompts the same participants over and over with questions, in the moment, while they go about a normal day. A short prompt arrives by text, app notification, or email; the participant answers two or three quick questions in under a minute; you repeat this across a sampling window of a few days to a couple of weeks. The output is real-time data on how often a problem happens, what triggers it, and how people feel when it does — captured in the moment instead of recalled later.

COMMON USE CASE

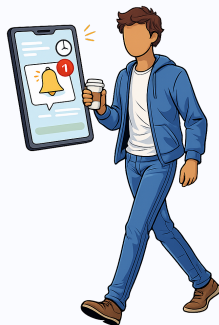
You believe a problem exists for your target customers, but you don't know how often it actually happens or how they feel about it in the moment. You send participants brief prompts throughout their day and they report what is happening in real time. The responses reveal frequency, intensity, and emotional context as it occurs, not as it is remembered.

HELPS ANSWER

- How often does this problem actually occur?
- Who experiences it regularly, and who doesn't?
- What triggers it — and where and when does it happen?
- How do people feel in the moment the problem comes up?

Description

Experience sampling is a structured study where you prompt the same participants repeatedly during their normal day and they answer a few short questions in the moment. It catches the answer as it happens instead of asking people to remember



AT A GLANCE

Qualitative

Remote

Attitudinal

Behavioral



TIME  ~1–3 weeks

COST  \$0–\$2.5K

OTHER NAMES

Daily Diary · Experience Sampling
Method · Ecological Momentary
Assessment · ESM

it later. Memory often distorts how often a problem is perceived to occur and how it felt at the time; a prompt delivered in the moment does not. The participant answers a handful of short questions while the experience is still fresh, and you repeat the prompt many times across the sampling window.

The method produces both quantitative and qualitative data: light counts and intensity ratings (how often, how strongly) alongside short free-text notes (what was happening, how it felt). That pairing is what makes it useful for deciding whether a problem is real and worth solving — frequency tells you it is common, intensity tells you it matters, and the notes tell you why.

It works best on a frequently recurring problem — one that happens often enough to catch inside a short window. For a rare event, you will collect mostly empty prompts.

How to

Prep

✦ AI Prompt

1. **Write the prompt, and keep it tiny.** Two or three questions answerable in under a minute on a phone. Mix one closed-form question (a 1–5 intensity scale) with a short open-ended follow-up, and ask about what just happened, not what the participant predicts they would do. Because you ask the same questions over and over, a single bad phrasing distorts every response — word them neutrally.
2. **Choose the trigger style and cadence.** Prompts can be signal-contingent (random times), interval-contingent (a fixed clock), or event-contingent (after a specific behavior). Decide how many prompts per day and over how many days. Too frequent and participants feel nagged and disengage; too sparse and you miss the behavior.
3. **Pick the channel.** SMS, app notification, email, or a conversational chatbot. A chatbot prompt feels less like a form and tends to draw richer answers — reach for it when participants would otherwise ignore a rigid ping.
4. **Screen and recruit participants.** Use a short screener to select people who actually have the behavior, quantifying where you can (“how often do you...”). Recruit enough to clear roughly 100 total data points — about 10 participants answering 10 prompts each. Set expectations up front about how often they will be pinged. See [Discovery Interviews](#) for sourcing and outreach guidance.
5. **Set up data capture and incentives.** Decide where responses land (a spreadsheet or an ESM platform export) and confirm consent before you start. Structure any incentive to reward completing the full window, not a single prompt.

6. **Pilot the protocol.** Run a day with one or two people before launch. Pilots almost always surface a confusing question, a bad time of day, or one prompt too many — cheap to fix now, expensive once the whole cohort is live.

Execution

✦ AI Prompt

1. **Launch and confirm delivery.** Send the first wave on schedule and verify the channel actually works — delivery receipts, notification permissions, the first responses landing where you expect them.
2. **Watch the early returns.** If the first day's responses are sparse or uniformly thin, fix the wording, the time-of-day distribution, or the question count now, before the rest of the data lands.
3. **Acknowledge, but don't coach.** Thank participants with a short, identical reply. If someone asks what a question means, record the confusion as a wording issue rather than rephrasing it for them — a rephrased question is no longer comparable to the rest.
4. **Stagger prompts and respect the cap.** Spread signal-contingent prompts across each time band instead of firing at the same clock time for everyone, and never exceed your daily cap, even for eager participants.
5. **Collect as you go.** Pull responses into your analysis surface as they arrive. Letting them pile up until the end forces you to triage and analyze at the same time. An AI agent can run this whole phase — delivering prompts on schedule, acknowledging responses, and flagging compliance problems — using the prompt below.

TOOLS

Scrappy kit (free): a scheduled SMS or calendar reminder plus a short Google Form, with responses landing in a spreadsheet — enough for a small study.

[dscout](#)

[Sprig](#)

[Dovetail](#)

[ExpiWell](#)

[Metricwire](#)

Analysis

✦ AI Prompt

1. **Start while data is still coming in.** Don't wait for the full set. Read the first batch to confirm the responses are usable and the questions are being understood the way you intended; if not, fix it before more data lands.
2. **Code the responses into categories.** Eyeball the data for an overall impression, then decide a handful of categories and tag each response — splitting a category that grows too big, merging ones that stay thin, and allowing a response to fit more than one. AI theme-extraction and sentiment tools can draft these categories in minutes; your job then shifts to validating the AI's themes rather than building them from scratch. For team coding, agree categories on the first

50–100 responses together, then split the rest and spot-check with a blind second pass.

3. **Build frequency and intensity views.** Chart how often each theme appears and how strongly participants rated it. Frequency tells you whether the problem is common; intensity tells you whether it actually matters.
4. **Look for context patterns.** Find the themes that cluster at certain times of day, in certain places, or after certain triggers. This context is the payoff experience sampling gives you that a retrospective interview cannot.
5. **Down-weight late or batched entries.** A response logged hours after the moment carries the same recall distortion the method exists to avoid. Flag late entries and weight them accordingly.
6. **Treat thin data as directional.** With fewer than ~50 data points, read findings as hypotheses for follow-up, not conclusions. Look for patterns that repeat across participants rather than single striking responses.

Biases & Tips

Future-intent bias

People are poor at predicting what they will want or do, so questions about hypothetical products or future behavior produce noise. Ask about what just happened and how it felt, not what they think they would do.

Leading questions

Because you ask the same questions repeatedly, a phrasing that telegraphs the answer — or an example response — biases the entire dataset, not one interview. Keep wording neutral and pilot it before launch.

Retrospective recall bias

The whole point of the method is the in-the-moment answer. If a participant batches their responses at the end of the day, that entry carries the same distortion as a traditional interview. Flag late responses and weight them down.

Survey fatigue

Compliance drops as the sampling window goes on. Watch for thinning responses and shorten the question list or the window rather than pushing through.

Sample-size optimism

AI makes large cohorts easy to field, but more participants amplify shallow, low-effort responses rather than eliminating them. Well-screened participants and a tight question set produce more reliable data than a high response count.

Next Steps

- Use a  **Contextual Inquiry** to observe the most frequent or surprising patterns in the participant's natural environment.
- Use  **Discovery Interviews** to follow up on specific entries and understand the reasoning behind the moments you captured.
- Use a  **Closed-Ended Survey** to measure how widespread the pain points are that surfaced in your samples.
- Turn the real-time data into a journey map grounded in observed moments rather than recalled ones.

Case Studies

Spotify: Diary study triangulated with behavioral logs

Spotify's design team ran a diary study alongside behavioral log data on the same listeners; the discrepancies between what people self-reported and how they actually listened produced the most actionable insights and informed ad product design.

[Read more](#)

Headspace: Ecological momentary assessment RCT

Zawadzki and colleagues ran a randomized controlled trial with 138 university employees using the Headspace app, sampling five times per day across four-day bursts at baseline and weeks 2, 5, and 8; subjective stress dropped by week 2 while coping gains emerged by week 5.

[Read more](#)

Educational-leadership problem perception (Emerald JEA)

An experience-sampling study captured 375 in-the-moment problem-perceiving episodes from educational-leadership students via iPod touch prompts, measuring emotion, motivation, self-efficacy, and reflection time; newly encountered problems produced higher engagement than recurring ones.

[Read more](#)

Further reading

-  Larson, R. & Csikszentmihalyi, M. (1983). "The Experience Sampling Method." *New Directions for Methodology of Social and Behavioral Science*, 15, 41–56. (Originating paper.)

- ☰ Hektner, J. M., Schmidt, J. A., & Csikszentmihalyi, M. (2007). Experience Sampling Method: Measuring the Quality of Everyday Life. Sage Publications. ISBN 978-1412949231.
- ☰ Don't listen to users, sample their experience — Tomer Sharon (Video)
- ☰ Experience Sampling Method — Wikipedia

3.7 Focus Groups

In Brief

A focus group is a facilitated discussion with 6–10 participants on a specific topic. A moderator guides the conversation while participants react to each other’s ideas. The output is a set of themes, customer language, and insight into how people influence each other’s opinions.

COMMON USE CASE

You want to explore how a group of potential customers talk about and influence each other’s thinking on the problem you are investigating. You bring several participants together for a guided discussion and observe how one person’s perspective shifts what others say. The group dynamics reveal social influences on decision-making that one-on-one interviews would miss.

HELPS ANSWER

- How do participants influence each other when making decisions together?
- What problems do participants describe when they talk openly?
- What do people think about the current alternatives available to them?
- What language do participants use to describe their needs?
- How do group members react to each other’s ideas?

Description

A focus group is a facilitated small-group discussion for hearing the “voice of the customer.” The conversation chains associations among participants, surfacing shared language and unarticulated needs that one-on-one interviews often miss. It fits open-ended exploration — unpacking a complicated buying journey, teasing apart



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal

Observational



TIME  ~4 days–2 weeks

COST  \$0–\$600

OTHER NAMES

Focus Group · Group Interview

the factors that drive a decision, reacting to a concept or stimulus, or hearing how people talk about a brand.

The method draws heavy and deserved skepticism, because groups are widely misused for evaluative work — confirming a design, measuring willingness to pay — where one-on-one interviews and behavioral tests give far more reliable signal. The social setting amplifies the loudest voices and invites groupthink, and purchasing decisions are often partly unconscious, which limits what any verbal method can surface. Treat a focus group as a way to hear how people talk about a problem, not as proof they will buy a solution.

You do not need a recruited panel, a venue, or a paid moderator to run a lightweight focus group: a 45–60 minute video call with 4–6 participants from your own network, with an AI tool handling transcription, captures most of the value cheaply. A common setup is a hybrid, where a human facilitator runs the live session while AI handles transcription, theme extraction, and cross-session comparison. Some platforms run [AI-Moderated Interviews at Scale](#), where a chatbot guides participants and follows up on interesting threads.

How to

Prep

[★ AI Prompt](#)

1. **Pick a single, clear purpose.** One product or one issue per session. If you cannot state the learning goal in a sentence, you are not ready to run the group.
2. **Narrow your target audience.** Talk to one segment at a time. Mixing distinct participant types in one room produces conversation that averages over real differences.
3. **Decide whether you need a comparison group.** Running a parallel group from a different segment lets you contrast opinions against a wider context.
4. **Refrain from ulterior motives.** Do not run a focus group to validate a decision you have already made or to soft-launch a sales pitch. The signal will be skewed before the first question lands.
5. **Find a co-facilitator.** A second person handles note-taking, equipment, and side conversations so the moderator can stay present in the discussion.
6. **Choose a comfortable venue and recording method.** Comfort and psychological safety shape how openly people speak. For virtual sessions, default to a meeting tool participants already know.
7. **Write the discussion guide.** Prepare up to ten open-ended questions. Establish rapport early, avoid jargon, and save anything embarrassing or threatening for

late in the session, once participants feel comfortable. Decide the opening framing here too: that this is a brainstorming-style discussion to hear opinions rather than reach consensus, and that there are no right or wrong answers.

Execution

1. **Get consent.** Pass out consent forms at the start and confirm everyone understands the session is being recorded.
2. **Keep introductions short.** Have everyone introduce themselves briefly — long introductions set the expectation that everyone will speak at length.
3. **Open with your framing, then guide loosely.** Deliver the framing you prepared in Prep, then ask your prepared questions while staying flexible, adding follow-ups when something interesting surfaces. Ideally participants begin talking among themselves and you can step back.
4. **Stay neutral and draw people in.** Keep your reactions even and empathetic, make eye contact with the participants who are speaking less, and gently invite them in.
5. **Capture more than words.** Your co-facilitator takes written notes, capturing non-verbal cues as well as what was said. AI transcription handles the verbatim record automatically; some platforms also flag in real time when engagement spikes or when a participant is being drowned out.
6. **Prevent any one person from dominating.** Use prompts like “Does anyone else have a different perspective?” or “I want to make sure we hear from everyone — [name], what do you think?”
7. **Wrap within 90 minutes.** Keep the session between 45 and 90 minutes; energy and signal both decline sharply after that.
8. **Offer a private feedback channel.** Provide a short feedback form so participants can flag anything they did not feel comfortable saying out loud.
9. **Repeat if you can.** Run a few groups with comparable composition — one session is a single data point and is unusually vulnerable to one strong voice.

TOOLS

[Dovetail](#)

[Otter.ai](#)

[Recall.ai](#)

[Remesh](#)

[Recollective](#)

Analysis

✦ AI Prompt

1. **Transcribe the session.** AI transcription tools handle this in minutes; clean obvious errors before coding.
2. **Code and cluster the responses.** Group related quotes into themes using affinity mapping — sticky notes or a digital equivalent that lets you cluster similar reasons together. Capture the participant’s actual language, not your paraphrase.

3. **Look for divergent opinions and outliers.** A focus group's strength is showing how people influence each other; pay particular attention to moments where someone changes their mind, pushes back, or goes quiet.
4. **Separate group-influenced views from individual beliefs.** When a theme looks important, plan a follow-up one-on-one interview to check whether participants hold the view independently or only adopted it under social pressure.
5. **Triangulate with other data.** Compare themes against your existing customer interviews, support tickets, and behavioral data before treating any single theme as a finding.

Biases & Tips

Confirmation bias

Do not run a focus group to confirm what you already believe. Listen for divergent or unexpected opinions, and write down the moments that surprise you.

Small-sample generalizability bias

One focus group can produce a vivid, articulate theme that does not generalize to the wider market. Treat themes as hypotheses, not conclusions, until triangulated across multiple sessions or methods.

Say-do gap

Assume participants cannot accurately describe why they make decisions. Use focus groups to surface language and concerns; use behavioral methods to measure intent.

Social-desirability bias

Participants understate behaviors they think the group will judge (price-sensitivity, brand-switching, lapsed habits). Anonymize answers where you can, and triangulate sensitive claims with private follow-ups.

Observer effect

Participants behave differently when they know they are being recorded or watched through a one-way mirror. Acknowledge the recording up front, then move on quickly so it fades from attention.

Moderator-induced bias

An inexperienced moderator may steer participants toward leading questions or cede the room to one dominant voice, distorting which views surface and how they are weighted. If facilitation is not your strength, partner with someone who has done it before or run a pilot session first.

Next Steps

- Run [Discovery Interviews](#) on surprising themes to dig deeper without group dynamics and separate group influence from individual beliefs.

- If directional preferences emerge, validate with a [Closed-Ended Survey](#) at scale.
- Use participant language and framing in your next round of messaging or a [Value Proposition Test](#).
- Use a [Comprehension Test](#) to verify that the messaging language from your focus group resonates with a broader audience.

Case Studies

Procter & Gamble: FemCare shifts to “heart-to-heart” conversations

P&G’s FemCare division shifted from behind-the-mirror focus groups toward face-to-face “heart-to-heart” conversations with consumers, citing richer insight than the traditional one-way-mirror format.

[Read more](#)

California Milk Processor Board: the “Got Milk?” deprivation insight

In 1993, focus groups asked participants to go a week without milk; their genuine anxiety in the “what if there’s no milk?” scenario became the deprivation strategy behind the iconic campaign — a reaction the numbers had missed.

[Read more](#)

Coca-Cola: New Coke (1985)

Before reformulating its flagship, Coca-Cola ran taste tests with roughly 200,000 consumers who preferred the new recipe; its focus groups, however, surfaced strong opposition from loyal drinkers, which researchers dismissed as a minority opinion. The quantitative data won, the backlash forced a reversal within months — a classic case of group qualitative signal being overruled by the numbers.

[Read more](#)

Further reading

-  Krueger, Richard A. & Casey, Mary Anne — Focus Groups: A Practical Guide for Applied Research (5th ed., Sage, 2015)
-  Merton, Robert K., Fiske, Marjorie & Kendall, Patricia L. — The Focused Interview: A Manual of Problems and Procedures (Free Press, 1956)
-  Morgan, David L. — Focus Groups as Qualitative Research (2nd ed., Sage, 1997)
-  Morgan, Carol-Ann — Using the Focus Group in Market Research (B2B International)

- ☰ Nielsen Norman Group – The Use and Misuse of Focus Groups
- ☰ Zaltman, Gerald – How Customers Think: Essential Insights into the Mind of the Market (Harvard Business School Press, 2003)

3.8 Open-Ended Survey

In Brief

A free-text questionnaire distributed to a large audience via email, website pop-up, or social media, where respondents answer in their own words rather than choosing from preset options. The output is qualitative data at scale: pain points in customers’ own language, recurring themes across a broad group, and unexpected ideas a structured survey would miss.

COMMON USE CASE

You have a mailing list of 2,000 people who signed up for your newsletter but you have not spoken to any of them. You send a short survey asking what their biggest frustration is with the topic your product addresses. The free-text answers give you a wide range of pain points described in customers’ own words, which you use to prioritize what to explore next.

HELPS ANSWER

- What problems do our potential customers describe in their own words?
- What frustrations come up most often across a large group?
- What language do customers use to talk about this topic?

Description

Open-ended surveys let respondents answer in their own words rather than picking from a list. The output is free text — sentences, paragraphs, or fragments — collected at scale across a target audience, then read for themes rather than tallied. They are the right choice when you don’t yet know what the answer categories should be: a closed-ended list can only return the options you already thought of, so it cannot surface a problem or phrasing you didn’t anticipate.



AT A GLANCE

- Qualitative
- In-person
- Remote
- Attitudinal



TIME  ~4 days–2 weeks

COST  \$0–\$45

OTHER NAMES
Discovery Survey

Open-ended and  closed-ended surveys are not interchangeable — they answer different questions. Use open-ended when you're still discovering: when the categories don't exist yet, when you need respondent language verbatim, or when an unexpected theme would change your direction. Use closed-ended when you already know the categories and need to measure how big each one is. Running them in the wrong order — closed-ended before open-ended — locks you into your existing assumptions and hides the surprises you ran the survey to find.

A 200-respondent survey produces 200 or more paragraphs of free text to read, which is where teams stall. AI handles the volume — a first-pass thematic analysis takes minutes — but a human still has to interpret the themes and decide what they mean for the business.

How to

Prep

 AI Prompt

1. **Write screening questions.** Filter for the people you actually want to hear from.
 - These are typically closed-ended questions that help identify if the respondent is in the desired target segment (e.g., “How old are you?”).
 - Add a few trap questions to catch “professional survey respondents” who lie to qualify for a survey or a paid follow-up study.
2. **Write the questions.** Keep them non-leading and grounded in real behavior.
 - Questions should be non-leading and non-hypothetical.
 - Asking for anecdotes or historical information generates more concrete insights.
 - Conduct comprehension tests on survey questions to ensure they're correctly interpreted.
 - LLMs can help draft question candidates and generate comprehension test variants to check for bias before deployment — but always review AI-generated questions yourself, as LLMs tend toward leading or hypothetical phrasing that sounds neutral but subtly biases responses.

Examples of good open-ended questions:

- “Describe the last time you tried to [task]. What happened?” (surfaces real behavior, not hypotheticals)
- “What’s the most frustrating part of [process] for you?” (identifies pain points in their words)

- “If you could change one thing about how you [activity], what would it be?” (reveals priorities)
- “What have you tried so far to solve this problem?” (maps existing alternatives)

Avoid: “Would you use a product that does X?” (hypothetical), “Don’t you think X is a problem?” (leading), “How satisfied are you?” (closed-ended — save for a [Closed-Ended Survey](#)).

3. Pilot with 5–10 people from your target segment. Catch confusing questions before you spend your list on them.

- Run the draft survey on a small group before launch. You’re checking for questions that confuse people, missing context, and how long the survey actually takes (aim for under 5 minutes).
- If a respondent asks what a question means, that question needs rewriting — don’t explain it for them.
- If most pilot answers come back one-word or “N/A,” the question is failing to get a reaction. Rewrite or cut.

4. Pick a distribution channel. Reach your audience where response quality holds up.

- Match the channel to your audience. Common options:
 - Social media
 - Email (existing list)
 - Website pop-ups or in-product surveys
 - Regular mail
 - Telephone
 - SMS
- Email to an existing list typically produces the highest response rates for open-ended formats; cold or paid panels skew toward shorter, lower-effort answers because respondents have less context for the topic.

Execution

✦ AI Prompt

1. Send the survey through your chosen channel.
 - Set a deadline. Open-ended collection windows drag on and introduce timing bias.
 - Send one reminder 2–3 days after the initial send. Two reminders maximum – more than that biases the sample toward the overly compliant.
2. Don't react to early responses.
 - Don't message respondents to clarify their answers mid-collection. That changes how later respondents answer if they hear about it.
 - If you spot a question producing only one-word answers or high skip rates, note it but don't pull the survey down – the data on what's failing is itself useful.
3. Monitor completion rates daily.
 - If drop-off spikes at a specific question, that question is the problem. Note it for the next iteration.
 - If overall response rate is well below the channel's typical range (under 1% on email to a warm list), the subject line or framing is the problem, not the questions.
4. Export the raw data.
 - Download responses as CSV or paste into a single document. Don't analyze inside the survey tool – export to a workspace where you can sort, theme, and quote freely.

TOOLS

Typeform

Dovetail

Sprig

Interpret

Any general survey tool (Google Forms, SurveyMonkey, etc.) works for collection; the value is in the analysis layer.

Analysis

✦ AI Prompt

Treat each response as an idea to explore, not as a vote. The data is qualitative: themes point you toward what to investigate next, they do not measure how many people hold a view.

Because surveys are easy to write and deploy, they are easy to misuse. Teams reach for them as a default when they feel they lack time for interviews or observation, and a large respondent count tempts a corporate audience to treat qualitative answers as statistically significant. The pull is strongest when an open-ended survey runs alongside a closed-ended one: the numbers invite you to hunt for correlations and declare a definitive conclusion the free text cannot support. That overinterpretation is the most common reason teams are warned off surveys.

To theme the responses, read each answer and pull the salient points into groups so patterns surface. AI does the first pass: paste the responses in and ask for the top five themes with a representative quote each, or use a dedicated analysis tool to code responses into themes at volume, score sentiment, and surface unexpected patterns. Let AI generate the initial coding pass, then review, split, merge, and validate the categories against what you already know about your audience.

Expected response rates and item-nonresponse. For email-distributed surveys to your own list, expect 5–20% response rates. With fewer than 30 responses, treat themes as hypotheses to explore in interviews, not conclusions to act on. Open-ended questions also produce higher item-nonresponse than closed-ended ones: skip rates commonly average around 18% (versus 1–2% for closed-ended), and can run anywhere from a few percent to over half on a single question. If a single question’s skip rate is well above the average for your survey, the question itself is the problem — too long, too hard, or unclear.

When a survey specifically asks for suggestions, keep the full list of suggestions in a repository for later analysis.

For very large data sets, algorithmic tools such as sentiment analysis or word clouds can add quantitative insight, but use them to supplement the qualitative findings, not replace them.

Biases & Tips

Selection bias

Researchers will often fixate on qualitative comments they agree with and ignore other comments. Read every response before theming, not just the ones that catch your eye on the first pass.

Non-representative sample

Certain distribution channels systematically attract respondents who differ from your actual target: paid panels skew toward survey-savvy respondents who give shorter, lower-effort answers; cold audiences lack context and surface shallower themes; existing lists over-represent already-engaged users. These channel effects distort which themes emerge and how strongly — validate dominant themes against a channel-diverse sample or follow-up interviews before treating any single channel’s output as representative.

Acquiescence and social desirability bias

Respondents tend to write answers they think the researcher wants to hear, especially when the survey isn’t anonymous. Keep open-ended surveys anonymous when possible, and frame questions about behavior and experience rather than opinion or evaluation.

Next Steps

- Follow up with [Discovery Interviews](#) to explore the most surprising or ambiguous themes in depth with individual respondents.
- If clear segments emerge, run a [Closed-Ended Survey](#) to measure how big each one is.
- Use a [Landing Page Test](#) to test value proposition copy drawn from respondent language verbatim.

Case Studies

Superhuman: Four-question PMF survey

CEO Rahul Vohra built a product-market-fit engine around a four-question survey anchored on Sean Ellis's "how would you feel if you could no longer use" question plus three open-ended prompts on ideal user, main benefit, and improvements; verbatim themes around speed and keyboard shortcuts shaped roadmap and positioning.

[Read more](#)

Notion: Enterpret-powered support-ticket taxonomy

Notion processes tens of thousands of monthly support tickets plus survey, app-store, and community feedback; using Enterpret's NLP-generated taxonomy, product-ops specialist Maya Bakir's monthly user-insights report dropped from two weeks to three days.

[Read more](#)

Further reading

-  Asking Questions: The Definitive Guide to Questionnaire Design (Bradburn, Sudman & Wansink, Jossey-Bass, 2004)
-  Pew Research: Why do some open-ended survey questions result in higher item nonresponse rates than others?
-  Choi & Pak: A Catalog of Biases in Questionnaires (Preventing Chronic Disease, CDC, 2005)
-  CXL Institute: Open-Ended Questions in Marketing Research
-  Top 21 Best Online Survey Software and Questionnaire Tools: An overview
-  Hotjar: Open-ended questions vs. close-ended questions: examples and how to survey users

- ☰ [Midwest Political Science Association: Structural Topic Models for Open-Ended Survey Responses](#)
- ☰ [Open-Ended Questions: Get More Context to Enrich Your Data](#)
- ☰ [Survey Monkey: How to Analyze Survey Data](#)
- ☰ [Survey Monkey: Types of survey questions](#)

3.9 Social Media Campaign

In Brief

Publish a series of posts around a topic or problem area and watch how people react — through engagement numbers and the comments they leave. You explore the problem without proposing a product or value proposition, so what you learn is whether an audience exists and what they care about, not whether they will buy something specific. The output is a read on audience size, the language people use, and which framings of the problem land.

COMMON USE CASE

You have a hypothesis about a customer segment and a problem space, but you do not yet know whether a real audience exists, what language they use, or which aspects of the problem matter most to them. You want a low-cost, low-commitment way to surface that signal in the wild — observing how people in your target segment react to varied framings of the topic — before committing to interviews, surveys, or a value proposition. You are still in the discovery phase: listening, not selling.

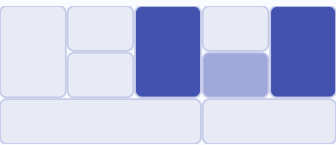
HELPS ANSWER

- Does an audience exist that cares about this topic or problem area?
- What specific aspects of the topic generate the most engagement?
- What language and framing does the audience respond to?
- Which platforms and content formats reach this audience most effectively?
- What questions, objections, and related topics does the audience raise?
- Is this audience reachable and engaged enough to build a business around?



AT A GLANCE

Quantitative Qualitative
Remote Behavioral



TIME 📅 ~2–3 weeks

COST 💰 \$0–\$130

OTHER NAMES
Social Media Test · Social Media
Discovery

Description

A social media campaign for research is a series of posts published around a problem area to observe how an audience reacts. By watching what people engage with, share, and say in the comments, you learn about audience size, engagement patterns, the language people use, and the adjacent concerns they raise — without building anything first.

Because you do not propose a product, the signal is about the problem itself, not about a specific offer. A post about “the frustration of tracking expenses as a freelancer” might draw moderate engagement while a post about “the anxiety of not knowing if you have saved enough for taxes” takes off. That difference tells you which framing of the problem people feel most strongly.

The comments are often more useful than the engagement counts. They reveal how people think about the problem, what they have already tried, the words they use for it, and the related problems they face — the kind of detail a like cannot carry. A short, vivid comment thread can be worth more than a high like count.

The method has limits. Engagement reflects interest, not intent to pay, so treat strong numbers as a reason to keep digging rather than as evidence of demand. Platform algorithms and your own follower network can both distort what you see, so a viral post may say more about the algorithm than about your market.

How to

Prep

✦ AI Prompt

1. **Define your topic area and audience hypothesis.** Write a clear statement of the problem space you are exploring and who you believe cares about it. For example: “Small business owners who struggle with cash flow management.” This hypothesis guides your content strategy but should remain flexible based on what you learn.
2. **Choose 2-3 platforms.** Select platforms where your hypothesized audience is likely active. LinkedIn for B2B professionals, Instagram or TikTok for consumer audiences, Twitter/X for tech and media, Reddit for niche communities, Facebook Groups for local or interest-based communities. Do not spread yourself across too many platforms — depth on 2-3 is better than breadth across 6.
3. **Create a content plan with varied angles.** Plan 10-15 pieces of content that approach your topic from different angles: educational posts, provocative

questions, personal stories, statistics, polls, and how-to content. Vary the framing to discover which angles resonate most. For example, if exploring project management pain points, create posts about missed deadlines, meeting overload, tool fatigue, and remote collaboration friction.

4. **Set up tracking.** Decide which metrics you will record per post — impressions (how many people saw it), engagement rate (the share of viewers who liked, commented, or shared), click-through rate (the share who clicked a link), and follower growth — and where you will record them (a simple spreadsheet is enough). Set up native platform analytics access for each chosen platform before you start posting, so you do not lose early data.

Execution

✦ AI Prompt

1. **Publish consistently for 2 weeks.** Post 3-5 times per week on each platform. Use each platform's native format (short text on Twitter/X, visual on Instagram, long-form on LinkedIn). Optionally allocate \$50-100 to boost 2-3 of your best-performing posts to expand reach beyond your immediate network.
2. **Engage actively with responses.** Reply to every comment. Ask follow-up questions. When someone shares their experience, ask "what have you tried to solve this?" or "how often does this happen?" These comment conversations are primary research data. Do not pitch a product — stay in listening mode.
3. **Track quantitative metrics per post.** For each post, record: impressions/reach, engagement rate (likes + comments + shares / impressions), click-through rate (if linking), and follower growth. Use native platform analytics or scheduling tools to aggregate data. Tag each post with its angle and format so you can compare across categories later.
4. **Capture qualitative data as you go.** Screenshot or export interesting comment threads while they are fresh — platforms hide or rearrange older threads, and threaded replies are easy to lose. Keep a running notes file of recurring phrases, unprompted comparisons to existing tools, and emotional language ("I hate that...", "the worst part is...").
5. **Hold the line on listening.** If commenters ask "what is your solution?" or "are you building something?" resist the urge to pitch. A short, neutral reply ("I am exploring this problem area, not pitching anything yet — what would you want?") preserves the generative posture and often produces the richest follow-up data.

TOOLS

[Buffer](#)

[Sprout Social](#)

[Typefully](#)

Native platform analytics (LinkedIn Analytics, X Analytics, Instagram Insights) — free per-post reach/engagement; cross-check third-party numbers

1. **Aggregate quantitative metrics by angle and format.** Group posts  by angle (e.g. “tax anxiety” vs. “expense tracking”) and by format (poll vs. story vs. educational). Compare engagement rate, not raw likes — a post that reached fewer people but engaged a higher fraction of them is a stronger signal.
2. **Cluster qualitative comments thematically.** Categorize comments by theme: agreement/recognition, personal stories, questions asked, solutions mentioned, objections, and related topics raised. Look for patterns across multiple posts. Comments that start with “This is so true because...” are particularly valuable — they contain unprompted customer language and context.
3. **Interpret high-resonance angles.** High engagement on a specific angle indicates which framing of the problem resonates most strongly. Posts where people tag friends or share personal stories indicate emotional resonance — the problem is real and felt. Questions in the comments reveal what information the audience is seeking.
4. **Interpret low or flat results.** If engagement is consistently low across all angles despite adequate reach, the audience may not be active on the chosen platforms, or the problem may not be significant enough to generate discussion. Before concluding the topic does not resonate, sanity-check content quality and posting times — low engagement may reflect execution, not topic.
5. **Read engagement depth, not just volume.** The ratio of passive engagement (likes) to active engagement (comments, shares) indicates depth of interest. High comment rates suggest the topic provokes thought and discussion. Follower growth during the campaign indicates sustained interest, not just one-time engagement.
6. **Synthesize an audience and problem profile.** Compile the angles, language, and themes that drove the most engagement into a one-page synthesis: who showed up, what they said, what they did not say, and which framings now feel like working hypotheses versus dead ends.

Biases & Tips

Platform bias

Each social media platform has its own demographic skew and content culture. Results on each platform may not generalize to your broader market. Run the campaign on at least 2 platforms to triangulate.

Engagement-equals-demand fallacy

People who engage with content about a problem are not necessarily people who would pay for a solution. Social media engagement reflects interest, not purchase intent. Always follow up

with a method that probes willingness to pay before committing to a product direction.

Algorithmic amplification

Platform algorithms may amplify certain content types (controversial, emotional, visual) regardless of their relevance to your research question. A viral post may tell you more about the algorithm than about your market.

Echo chamber effect

If your initial audience is your own network, their engagement patterns may reflect social obligation rather than genuine interest in the topic. Track engagement from people outside your immediate network separately.

Vocal minority bias

Commenters are a small fraction of your audience. The people who comment may have stronger opinions or more extreme experiences than the silent majority. Treat comment patterns as directional, not representative.

Confirmation bias

Founders tend to over-weight comments that agree with their hypothesis and dismiss those that disagree. Have someone else categorize a sample of comments blind to your hypothesis as a check.

Short-window bias

Two weeks is insufficient to detect seasonal patterns or to build an audience representative of the segment. Treat all findings as early directional signal and flag any conclusion that depends on the absence of a pattern — that pattern may simply not have had time to appear.

Next Steps

- Compile the angles, language, and themes that drove the most engagement into a synthesized audience and problem profile.
- If one angle clearly outperformed the others, treat that framing as your working hypothesis for the problem and the language to test next.
- Use [Search Trend Analysis](#) to check whether the language that resonated on social also shows up as real search demand.
- Use a [Online Ad Test](#) to move from generic topic engagement to testing a specific offer with the audience you found.
- Use an [Open-Ended Survey](#) targeted at engaged commenters to capture richer qualitative detail on the problem.

Case Studies

Buffer: Public transparency as an audience channel

Buffer publishes revenue, salary formula, and diversity data openly through its blog and social channels; the company's transparency posts function as a content channel that draws an engaged founder, marketer, and remote-team audience.

[Read more](#)

Nomad List: Editable Google Sheet seeded on Twitter

In 2014 Pieter Levels tweeted an editable spreadsheet of ~10 cities and asked digital nomads to add more; over a thousand contributors added cities and new columns (cost of living, internet speed, safety, weather) within a month, and those columns became the spine of the Nomad List product.

[Read more](#)

Further reading

-  Rob Fitzpatrick — The Mom Test
-  Steve Blank — The Four Steps to the Epiphany
-  Pew Research Center — Social Media Use
-  Nielsen Norman Group — Social Media UX: 3 Research Insights
-  Sprout Social Index — Social Media Engagement Benchmarks

3.10 Storyboard the Pain

In Brief

You draw the customer's current experience as a 4-8 panel comic strip that ends on the pain point you believe they face, then walk a participant through the panels and watch where they correct, hesitate, or add detail. The output is a clear read on whether your picture of the problem matches their reality — the sequence of steps, the trigger, the moment things go wrong — before you commit effort to a solution.

For the solution-focused counterpart, see

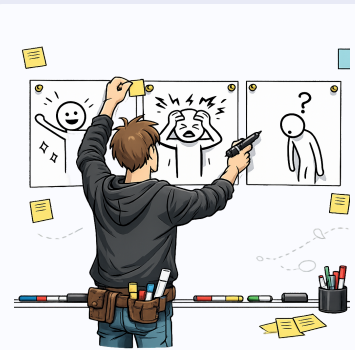
[Storyboard the Solution](#).

COMMON USE CASE

You have spoken with a handful of people and believe you understand their pain point, but the problem statement you wrote down still feels abstract. Before committing effort to a solution, you want to check whether you have the experience right. You sketch it as a rough comic and walk participants through it, watching for where they push back.

HELPS ANSWER

- Am I describing the customer's problem accurately?
- Do I understand the context in which the pain occurs?
- What parts of the customer experience am I getting wrong or missing?
- Is the emotional weight of the problem as significant as I assume?
- What happens before and after the painful moment?
- Are there steps in the experience I have not considered?



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal

Observational



TIME  ~4 days–2 weeks

COST  \$0–\$350

OTHER NAMES

Pain Storyboard · Problem Storyboard

Description

A storyboard focusing on the pain point is a visual narrative that walks through the customer's current experience, ending with (or centered on) the pain point you believe exists. Unlike a written description, a visual storyboard is concrete and specific: it forces you to commit to details about who the customer is, what they are doing, where they are, and what goes wrong. This specificity makes it much easier for a participant to react to and correct, frame by frame.

The same specificity is also where you are most likely to be wrong, which is the point — a participant who says “that is not how it works at all” about a single panel has just handed you a correction you would never have surfaced from an abstract description.

Reactions can run in several directions. A participant may confirm some panels, reject others, describe steps you left out, or place the whole story in a different context than you assumed. Each of those reshapes your understanding of the problem. Pay attention when corrections cluster: when several participants fix the same panel the same way, that part of your model is wrong; when they fix it in opposite directions, you may be looking at more than one segment.

Storyboards draw this out because they are low-fidelity and obviously unfinished. People critique a rough sketch far more freely than a polished presentation, which reads as a decision already made, so keep the fidelity low on purpose. The visual format also carries across language and literacy differences, so it holds up when the experience spans channels, departments, or cultures.

How to

Prep

★ AI Prompt

1. **Define the pain point you want to depict.** Write a one-sentence description of the problem you believe your customer faces. Be specific about who the customer is, what they are trying to accomplish, and what goes wrong. Vague pain statements produce vague storyboards.
2. **Sketch the first and last panels first.** The first panel establishes the context (who, where, what they are trying to do). The last panel shows the consequence of the pain (frustration, wasted time, lost money, embarrassment). Starting with these two endpoints anchors the narrative.
3. **Fill in the middle panels.** Add 2-6 panels that show the steps between the starting context and the painful outcome. Each panel should depict one moment or action. Use simple visuals — stick figures, basic shapes, arrows — and

minimal text. Captions or speech bubbles can clarify what is happening, but the visuals should carry the primary narrative.

4. **Prioritize visuals over text.** Resist the urge to write paragraphs of explanation. If a panel needs more than one sentence of caption, you are trying to convey too much in a single frame. Split it into two panels instead. The storyboard should be understandable at a glance.
5. **Recruit 3-5 participants.** Storyboard reviews produce useful patterns when you can compare reactions across multiple sessions. Aim for 3-5 participants from your target segment for the first round, then iterate the panels and run a second round if reactions are mixed.
6. **Prepare the session script.** Decide how you will introduce the storyboard, what open-ended questions you will ask, and how you will record corrections, hesitations, and additions. Frame the opening so the participant knows there are no right or wrong answers and that you want their criticism. Pilot the script with one participant or a colleague before running it for real.

Execution

✦ AI Prompt

1. **Walk the participant through the storyboard panel by panel.** Present the storyboard and narrate each panel briefly. Do not rush — give the participant time to absorb each frame before moving on. Avoid defending or explaining the visuals beyond a single sentence per panel.
2. **Ask the open-ended questions from your script.** Use prompts like “Does this look like your experience?” “What would you change?” “What am I missing?” “How does this make you feel?” Avoid yes/no questions. The goal is to invite correction, not seek agreement.
3. **Observe corrections, hesitations, and expansions.** Note which panels the participant confirms without comment, which they correct, where they pause, and where they add detail. Hesitations are often as informative as explicit corrections — a participant who frowns at a panel but does not articulate why is signaling something is off.
4. **Capture emotional reactions.** Pay special attention to moments where the participant says “yes, that is exactly it” or “no, that is not right at all,” or where their tone or expression shifts. These are the most informative data points.

TOOLS

Pen and paper — rough sketches invite criticism that polished mockups suppress

[Miro](#)

[FigJam](#)

[Storyboard That](#)

Any AI image generator (Midjourney, DALL-E, Adobe Firefly) — panel mockups when stick figures aren't enough

5. **Resist the urge to defend.** When a participant corrects a panel, your instinct will be to explain why you drew it that way. Do not. Acknowledge the correction, take notes, and ask follow-up questions. The storyboard is a probe, not a position to defend.

Analysis

✦ AI Prompt

1. **Aggregate corrections across participants.** After each session, log which panels were confirmed, corrected, and expanded, plus key quotes. After 3-5 sessions, compile the pattern. Panels that multiple participants correct in the same direction are clearly wrong; panels that different participants correct in different directions may signal multiple personas.
2. **Pay more attention to emotional reactions than rational critiques.** A participant who gets visibly frustrated looking at a panel is confirming that the pain is real and significant. A participant who shrugs and says “I guess that is sort of right” may be signaling that the pain is not urgent enough to build a business around.
3. **Catalog the additions.** Steps participants add independently — especially when multiple participants add the same missing step — are likely important to the experience and may contain additional pain points worth investigating.
4. **Watch for segmentation signals.** When different participants react to the same panel in opposite directions (“this is exactly my experience” versus “this is nothing like my experience”), you may be looking at distinct personas with different journeys, not one segment with a single pain point.
5. **Treat “I guess that is sort of right” as a red flag.** Lukewarm validation is often an honest signal that the pain is not significant. Probe whether the participant has a stronger reaction to a different panel or a different version of the story before concluding the pain is real.
6. **Note what participants ignore.** Panels that participants skim past without comment may be either obviously correct or completely irrelevant to them. Probe both possibilities — silence is data, but it is ambiguous data.

⚠ Biases & Tips

Leading narrative bias

The storyboard itself frames the conversation. Participants may agree with your narrative because it seems reasonable, even if it does not match their actual experience. Use open-ended follow-up questions to counter this.

Social desirability bias

Participants may not want to criticize your work, especially if you are visibly the creator. Have someone else present the storyboard if possible, or explicitly invite criticism.

Anchoring bias

The specific details you include in the storyboard anchor the conversation. Participants may correct details within your frame rather than offering an entirely different frame. Ask “what am I not showing here?” to counter this.

Recall bias

Participants may not accurately remember their own experience when comparing it to a storyboard. Where possible, conduct the review close to when they last experienced the pain.

Confirmation bias (creator form)

You may unconsciously dismiss or minimize corrections that contradict your storyboard narrative — a form of confirmation bias amplified by creative ownership. Discovering you are wrong is the most valuable outcome of a storyboard review.

Next Steps

- Revise the storyboard based on the corrections and additions participants gave you, and treat that revised version as your current shared model of the problem.
- If multiple participants reacted to the same panel as wrong or missing, prioritize that step for deeper investigation before moving toward solutions.
- Use [Discovery Interviews](#) to explore the panels participants reacted strongest to and understand the surrounding context.
- Use [Contextual Inquiry](#) to observe the workflow in its real environment and see how accurate your storyboard actually is.
- Use [Pain Point Sorting](#) to prioritize the painful moments your storyboard surfaced relative to other pains the customer faces.
- Use [Storyboard the Solution](#) once the pain narrative is validated, to start exploring how a future solution would change that experience.

Case Studies

IDEO: Mapping shopper pain points during the cart redesign

In the 1999 ABC Nightline “Deep Dive” segment, IDEO designers observed shoppers and mapped pain points — child seating, aisle navigation, theft, checkout friction — before redesigning the supermarket cart in a week through rapid brainstorming, prototyping, and user feedback.

[Read more](#)

Nielsen Norman Group: Storyboards in UX practice

NN/g's guidance defines a storyboard as scenario plus sequential visuals plus short captions, and describes its primary uses as summarizing usability research, building team prioritization consensus, ideating, and enriching journey maps; the article advises keeping fidelity low.

[Read more](#)

Further reading

-  Storyboards Help Visualize UX Ideas — Nielsen Norman Group
-  Journey Mapping 101 — Nielsen Norman Group
-  This Is Service Design Doing — Marc Stickdorn et al. (Service Design Network)
-  Storyboarding in the Software Design Process — UX Magazine
-  About Face: The Essentials of Interaction Design — Alan Cooper et al. (Wiley)

3.11 Synthetic Persona Screening

In Brief

Prompt a large language model (LLM) with a detailed customer description, then run a simulated interview by asking the same open-ended questions you would ask a real customer, varying the persona across segments. The output is a set of candidate hypotheses — possible pain points, objections, language patterns, and segment differences — that you then validate with real humans. It is preparation for customer interviews, not a substitute for them.

COMMON USE CASE

You are preparing for your first round of customer discovery interviews but you are not sure what questions to ask or what objections to expect. You prompt an AI with a detailed description of your target customer and run a simulated interview conversation. The AI raises objections you had not considered, which you add to your real interview guide.

HELPS ANSWER

- What pain points might this type of customer have?
- What words does the target customer use to describe their problem?
- What objections might someone raise about this solution?
- Are there related customer groups we have not thought about?
- What questions should we ask in our first real interviews?

Description

Synthetic persona screening is a structured exercise where you write a detailed customer description, prompt an LLM to role-play that persona, and interview it the way you would interview a real person: open-ended questions about their day, their



AT A GLANCE

Qualitative

Remote

Attitudinal

TIME

 ~1–2 hours

COST

 \$0–\$30

OTHER NAMES

AI Persona Screening · LLM
Persona Simulation

problems, and their reaction to your value proposition. You run the conversation again across a few persona variations, then pull out the recurring pain points, objections, and phrasing. What you get is a hypothesis list — candidate pain points, plausible objections, language patterns, and segment-level differences — not validated findings. It expands your hypothesis space before you talk to anyone real; it does not measure or evaluate anything on its own.

A synthetic persona is not a substitute for talking to real customers. Conditioning a model on detailed demographic backstories can produce useful population-level patterns, but individual responses should not be expected to match a real respondent, and known LLM shortcomings — training-data bias toward English-speaking and Western populations, sycophancy, coherence bias, and a tendency to invent opinions — carry through into the persona. The output tends to be too shallow for deeper research and is most useful for broad attitudinal questions. Even vendors who sell synthetic-respondent platforms position their tools as a discovery aid rather than a replacement for real research.

The reason to run it anyway is cost and speed: a session runs for a few dollars in under an hour, and it often surfaces objections and language a founder had not considered. Used as a substitute for real research, it produces confident answers that may be wrong. Used to prepare for interviews with real people, it sharpens the questions and the screener.

How to

Prep

✦ AI Prompt

1. **Define your target persona in detail.** Write a rich description including demographics, job title, industry, daily workflow, goals, frustrations, tools they currently use, and any relevant psychographic details. The more specific, the better. “A 35-year-old ops manager at a 50-person logistics company who spends 3 hours a day in spreadsheets reconciling shipment data” is far more useful than “a logistics professional.”
2. **Choose persona variation axes.** Decide in advance which dimensions you will vary across runs — company size, seniority, industry vertical, geography, level of technical sophistication. Three to five variations is usually enough to surface segment differences without drowning in transcripts.
3. **Pick your platform.** Any general-purpose LLM chat tool works well for persona role-play with custom instructions; dedicated synthetic-respondent platforms

package the workflow with built-in persona libraries and synthesis. The platform changes convenience, not the validity caveats.

4. **Craft the system prompt.** Instruct the LLM to role-play as this persona. Be explicit: “You are [persona]. Respond as this person would — with their vocabulary, their priorities, their level of technical sophistication. Do not be helpful or agreeable beyond what this person would naturally be. If this person would be skeptical, be skeptical.”

Execution

✦ AI Prompt

TOOLS

ChatGPT

Claude

Delve AI

1. **Run a discovery conversation.** Ask the same open-ended questions you’d ask in a real customer discovery interview:
 - “What’s the hardest part of your day?”
 - “Walk me through the last time [problem context] happened.”
 - “What have you tried to solve this?”
 - “What didn’t work about those solutions?”
2. **Probe for objections.** Present your value proposition and ask the synthetic persona to react honestly. Push for specifics: “What would make you hesitate to try this?” and “What would your boss say if you proposed buying this?”
3. **Vary the persona.** Run the same conversation with 3 to 5 variations along the axes you chose in prep. Note where responses diverge — those divergence points are often the most valuable hypotheses.
4. **Extract and categorize hypotheses.** Pull out distinct pain points, objections, language patterns, and segment differences. Tag each one as a hypothesis that needs real-world validation.
5. **Design your real interview guide.** Use the hypotheses to craft sharper screener questions and interview prompts for actual customer discovery interviews.

Analysis

✦ AI Prompt

The output is a hypothesis list, not a finding. Treat every insight with the same skepticism you would apply to your own brainstorming.

Look for:

- **Recurring themes across persona variations.** If the same pain point surfaces across multiple persona descriptions, it’s a higher-priority hypothesis to test.
- **Surprising objections.** Objections you didn’t anticipate are often the most valuable outputs, because they expand your hypothesis space rather than confirming what you already believe.

- **Language patterns.** The specific words and phrases the LLM uses can be useful starting points for ad copy, landing page headlines, and search keyword research – but only after you confirm real customers actually use those terms.
- **Segment boundaries.** Where persona variations produce dramatically different responses, you may have found a meaningful segmentation boundary worth exploring.

Do NOT treat any of the following as validated:

- Willingness to pay
- Feature priorities
- Emotional intensity of pain points
- Market size or demand signals

Biases & Tips

Sycophancy bias

LLMs are trained on human feedback that rewards agreeable, helpful responses. A synthetic persona will almost always be more receptive to your idea than a real human would be. If the AI persona says “that sounds interesting, I’d definitely try it,” that is virtually meaningless. Real customers hedge, deflect, and politely lie – and they’re still more honest than an LLM playing a character.

Training data bias

The LLM’s responses reflect its training corpus, which overrepresents English-speaking, tech-savvy, Western perspectives. If your target customer is a rural small business owner in Southeast Asia, the synthetic persona will be drawing on very thin data. To partially counteract this, prompt for outside perspectives – “What would an ethnographer observe that wouldn’t appear in a standard business analysis?”

Confirmation bias

Because you write the persona prompt, you inevitably encode your own assumptions about the customer. The LLM then reflects those assumptions back to you, creating a closed loop. Actively prompt for disagreement and skepticism to partially counteract this.

Coherence bias

LLMs generate internally consistent narratives. Real customers are contradictory – they say one thing and do another, hold incompatible priorities, and change their minds mid-sentence. Synthetic personas are unrealistically coherent.

Substitution trap

The method is so fast and cheap that founders run dozens of synthetic sessions and skip the harder work of recruiting real

humans. No quantity of synthetic interviews compensates for zero real ones.

**Anchoring
bias**

Once you've seen the AI's list of pain points or persona behaviors, it becomes the boundary of your imagination. You design your interview guide around the AI's framing and never surface insights that didn't appear on the list. Generate your own hypotheses before prompting AI. If you can't produce ideas that aren't on the AI's list, you've been anchored.

Next Steps

- Validate the most promising persona segments with real humans using [Discovery Interviews](#), prioritizing the highest-risk assumptions first.
- Run the same questions at breadth with an [AI-Moderated Interview at Scale](#) to see whether the synthetic patterns hold across a real participant pool.
- Use a [Landing Page Test](#) to test value propositions that resonated with synthetic personas against real market behavior.

Case Studies

Radius Insights / MilkPEP

ran a concept test with AI synthetic respondents alongside ~200 real respondents per concept; synthetic and real top-two-box scores aligned on four of six measures, and the divergences themselves became the finding.

[Read more](#)

Further reading

-  "Out of One, Many: Using Language Models to Simulate Human Samples"
-  "Generative Agents: Interactive Simulacra of Human Behavior"
-  "Evaluating AI-Simulated Behavior: Insights from Three Studies on Digital Twins and Synthetic Users"
-  The Mom Test

PART 4

Evaluative Market Experiments



*“Life is an experiment in which you may fail or succeed.
Explore more, expect least.”*

—Santosh Kalwar

CONTENTS

▢ Closed-Ended Survey

▢ Comprehension Test

▢ Secondary Market Research

▢ Value Proposition Test

4.1 Closed-Ended Survey

In Brief

Closed-ended surveys are structured questionnaires that ask participants to choose between fixed, multi-choice answers such as yes/no or A/B/C/D. They are often used to rank the most important issues for participants or segment the population. Surveys can be repeated over time to identify trends. The output is structured quantitative data which can be analyzed with statistical techniques.

COMMON USE CASE

You’ve talked to a number of people and they’ve expressed four distinct pain points you could build a business around. To prioritize which problem to solve, you need to know how big the market for each is. You want to know what percentage of the broader population have those problems and how often the problems occur.

HELPS ANSWER

- Which of these options is the priority?
- What percentage of people are interested?
- What is the trend?
- How can I segment the population?
- Are there any unexpected correlations in answers?

Description

A closed-ended survey asks every respondent the same questions with a fixed set of answer choices – multiple choice, rating scales, yes/no. Because everyone picks from the same options, the output is structured data you can count and compare. The trade-off is that you only learn about the answers you thought to offer – a fixed list cannot surface a problem you didn’t already know to ask about.



AT A GLANCE

Quantitative

In-person

Remote

Attitudinal



TIME  ~1 day–1.5 weeks

COST  \$0–\$400

OTHER NAMES
Validation Survey

Use closed-ended surveys when you have “known unknowns”: you’ve done generative research, you have several hypotheses, and now you need numbers to decide between them. To discover problems you haven’t heard of yet, use an

[Open-Ended Survey](#) or interviews instead.

They scale cheaply — the same instrument works for 50 respondents or 5,000 — which is why they are the standard tool once you need to size a market rather than explore it.

How to

Prep

★ AI Prompt

1. Write your goal in one sentence.

Keep it focused and don’t try to learn everything at once. A survey built to segment a market looks nothing like one built to measure satisfaction. If you can’t state the goal in one sentence, you’re not ready to write questions.

Examples:

- “Determine which of four pain points is most common among mid-market CFOs.”
- “Measure whether satisfaction improved after the Q3 onboarding redesign.”

2. Pick a screener question.

The first question should filter out people who aren’t in your target population. Put it at the top and end the survey for anyone who doesn’t qualify. Use as few screeners as you can — each one shrinks the pool of people who finish.

Are you currently responsible for purchasing software for your team? Yes / No

(If No → end survey)

3. Choose your question types.

Each type produces a different kind of data. Use the one that matches what you need to learn.

Dichotomous (yes/no) — Splits the population into two groups. Fast to answer. Use when you need a clean binary.

Do you currently use a project management tool? Yes / No

Multiple choice — Identifies which category a respondent falls into. Use for segmentation or feature preference.

What is your primary role?

- Founder/CEO
- Product Manager
- Engineer
- Designer
- Other

Likert scale — Measures degree of agreement or satisfaction on a fixed scale. Use to gauge intensity, not just direction.

"I find it easy to prioritize my daily tasks." Strongly disagree / Disagree / Neutral / Agree / Strongly agree

Forced ranking — Makes respondents prioritize. Unlike multiple choice, it forces trade-offs. Use when you need relative importance, not just preferences.

Rank from most to least important when choosing a SaaS tool:

1. Price
2. Ease of use
3. Customer support
4. Integrations

Matrix — Applies the same scale across several items. Efficient but fatiguing. Keep it to five rows or fewer.

Rate each department's data literacy (1 = Low, 5 = High):

- Marketing
- Product
- Sales
- Engineering

Hybrid (closed + "Other") — Add a free-text "Other" option when you aren't sure your answer list is complete. If more than 10% of respondents choose "Other," your options are wrong. Go back and fix them.

4. Write the questions.

Rules:

- One concept per question. Don't ask "Is the product fast and reliable?" — that's two questions.
- No hypotheticals. "Would you pay for X?" is worthless. People don't know.
- No leading language. "Don't you agree that..." is not a question.
- Include "N/A" or "Don't know" where appropriate. Forcing an answer where none exists creates noise.
- Keep the total survey under 15 questions. Completion rates drop sharply after that.

5. Order the questions deliberately.

- Start with the screener.
- Put easy, non-threatening questions next to build momentum.
- Place sensitive or complex questions in the middle, after trust is established.
- End with demographic questions (role, company size, etc.) — these feel impersonal and signal the survey is nearly done.
- If order effects matter, randomize question sequence across respondents. Most survey tools support this.

6. Pilot with 5–10 people.

Run the survey on a small group before you launch. You're checking for:

- Questions that confuse people
- Missing answer options
- How long it actually takes (aim for under 5 minutes)
- Whether the data you get back answers your stated goal

If you used AI to draft the instrument, pilot with real humans before launch regardless — LLMs can scrub for obvious bias and suggest appropriate scales, but they cannot catch questions that are technically clear yet culturally or contextually wrong for your specific audience.

7. Choose a distribution channel.

Match the channel to your audience:

- **In-app or website exit survey** — Best response rates. Use for existing users. Most in-app survey tools embed directly.
- **Email** — Good for existing customers or mailing lists. Expect 10–30% response rates. Keep the subject line short and specific.
- **Social media or community posts** — Cheap reach but self-selecting. Useful for early-stage validation when you don't have a user base yet.

- **Paid panels** — Panel services recruit respondents matching your criteria. Costs \$1–5 per response. Use when you need a specific demographic you can't reach organically.
- **In-person** — Pen and paper at events, trade shows, or retail locations. High completion rates but small sample sizes.

8. Set a target sample size.

You need enough responses to trust the results. Rules of thumb:

- For simple yes/no splits: 100 responses minimum.
- For comparing subgroups (e.g., role A vs. role B): 30+ per subgroup.
- For detecting small differences between options: 300+.
- Use a sample size calculator (see References) to get a precise number. Plug in how small a difference you want to be able to detect, your confidence level (how sure you want to be that the result isn't a fluke — usually 95%), and your population size.
- **If your sample will be small (under 50):** You can still run the survey. You won't get statistical significance, but you will get directional signals — which options cluster together, which segments lean one way. Present findings as “early indications,” not conclusions. Small-sample surveys are most useful when paired with follow-up interviews to pressure-test the patterns you see.

Execution

1. Collect responses.

How you collect depends on the channel you chose. In all cases: don't explain the “right” answer, don't react to responses, and don't help participants interpret questions. You want their first instinct, not a coached answer.

Online surveys:

- Set a deadline. Open-ended collection windows drag on and introduce timing bias.
- Send one reminder 2–3 days after the initial send. Two reminders maximum — more than that annoys people and biases toward the overly compliant.
- Monitor completion rates daily. If drop-off spikes at a specific question, that question is the problem.

In-person surveys:

✦ AI Prompt

TOOLS

[Typeform](#)

[SurveyMonkey](#)

[Qualtrics](#)

[Sprig](#)

Any free-tier form builder (Google Forms, Tally)

- Read the questions exactly as written. Don't paraphrase or elaborate — that introduces variation between respondents.
- Don't hover. Hand over the survey (or tablet) and step back. People answer differently when they feel watched.
- If a participant asks what a question means, note the confusion but don't explain. That question needs rewriting.
- Collect responses in a consistent format. If using paper, transcribe into a spreadsheet the same day while your memory of any issues is fresh.

Phone or video:

- Read questions verbatim. For multiple choice, read all options before accepting an answer — people anchor on the first option they hear.
- Record the session (with consent) so you can verify transcription later.

2. Export the raw data.

Download responses as CSV. Don't analyze inside the survey tool — export to a spreadsheet or statistical tool where you can manipulate the data freely.

Analysis

✦ AI Prompt

1. Clean the data first.

- Remove incomplete responses (respondents who dropped out partway through).
- Remove speeders — anyone who finished in less than one-third of the median completion time was clicking randomly.
- Check for straight-liners: respondents who selected the same answer for every question. Remove them.

2. Start with frequencies.

For each question, count how many respondents chose each option. Calculate percentages. This is your baseline. Look at it before doing anything fancy.

3. Cross-tabulate.

Cross-tabulation means breaking responses down by subgroup (e.g., role, company size, or answers to other questions). A preference that looks evenly split overall might be 80/20 within a specific segment. That's where the insight is. You can do this in a spreadsheet with pivot tables, or ask an AI to do it for you.

4. Check distributions.

Don't just look at averages. Look at the shape of how answers are spread out:

- **Bimodal** (two peaks) — You may have two distinct segments responding differently. Investigate.
- **Heavily skewed** (most answers bunched to one side) — The average is misleading. Report the median (the middle value when all responses are sorted — less affected by extreme outliers) instead.
- **Uniform** (answers spread evenly) — Respondents are guessing or the question doesn't discriminate. The question may be poorly worded.

5. Test for significance.

You don't need a statistics background to do this. Export your data as a CSV, paste it into an AI tool, and ask it to run the appropriate test. The key tests and when to ask for them:

- **"Run a chi-square test"** — When you're comparing categories across groups (e.g., "Did segment A prefer option 1 more than segment B?"). This tells you whether the difference is real or just random variation.
- **"Run a t-test"** (two groups) or **"Run an ANOVA"** (three or more groups) — When you're comparing average scores across groups on Likert-scale or rating data.
- **"Check for correlations"** — When you want to know whether answers to two questions move together (e.g., do people who rate ease-of-use highly also rate satisfaction highly?).

The AI will return a p-value. If it's below 0.05, the pattern is statistically significant — meaning it's unlikely to be noise. If it's above 0.05, you can't trust the pattern yet. You may need more responses.

When an AI returns an analysis, open a new session and ask it to double-check and critique the analysis to catch errors and invented results.

6. Flag problems in the instrument.

- Questions where 80%+ chose the same answer told you nothing. The question was too obvious or too leading.
- Questions with high "N/A" or "Don't know" rates were asked of the wrong audience.
- If "Other" exceeded 10% on a multiple-choice question, your options were incomplete.

Biases & Tips

Social
desirability

bias

Respondents avoid admitting to unsavory behavior, especially if results aren't confidential. Keep surveys anonymous when possible.

Demand characteristics

Respondents unconsciously shift answers to match what your wording implies is correct. Compare: "Are you for or against an increase in tobacco tax rates?" (neutral) vs. "Are you in favor of increasing tobacco tax rates to protect our children's health?" (leading). Have someone outside your team review questions for loaded language before fielding.

Presupposition bias

Questions that embed an untested assumption force respondents into a frame that may not apply to them. "Where do you enjoy drinking beer?" presumes the respondent drinks beer. Use neutral framing and provide opt-out options.

Hypothetical bias

People systematically misstate what they would pay or do in a hypothetical — they understate price sensitivity and overstate intent. Replace willingness-to-pay questions with behavioral signals (pre-sales, pricing page tests).

Acquiescence bias

People tend to agree with statements regardless of content. Mix positively and negatively worded items to catch this.

Nonresponse and self-selection bias

The people who bother to answer often differ from the people who don't, so a high response count can still misrepresent your population. Compare respondent demographics against your target and report who is missing rather than assuming the sample stands in for everyone.

Next Steps

- Refine the customer segment with results and run **Discovery Interviews** with the prioritized segment. If results came back indeterminate, expand the sample before drawing conclusions.
- For confirmed participant problems, run a **Value Proposition Test**.
- Use a **Landing Page Test** to test whether the segments you quantified will convert on a real value proposition.
- Use a **Comprehension Test** to verify that your messaging resonates with the highest-priority segment before scaling.

Case Studies

Foursquare: City expansion driven by user complaints

Foursquare expanded from 23 to 38 cities in October 2009 after users' top complaint was limited city availability; the team used surveys and usage data to prioritize the next 15 markets.

[Read more](#)

Coca-Cola: New Coke taste tests missed brand attachment

Coca-Cola ran nearly 200,000 blind taste tests before launching New Coke in 1985, but never asked whether customers wanted the original replaced; backlash forced the reintroduction of Coca-Cola Classic 79 days later.

[Read more](#)

Kia: IQS turnaround through structured feedback

J.D. Power's Initial Quality Study uses closed-ended owner surveys across categories like infotainment, drivetrain, and exterior. Hyundai and Kia used poor IQS rankings to drive quality overhauls from the early 2000s; by 2016 Kia ranked first overall, the first non-premium brand to top the study in 27 years.

[Read more](#)

Cleveland Clinic: HCAHPS scores driven by structured survey focus

Under CEO Toby Cosgrove, Cleveland Clinic made the HCAHPS closed-ended patient-experience survey a strategic priority; over three years CMS scores moved from about average to the top 8% of roughly 4,600 hospitals.

[Read more](#)

LEGO: Four-year study before launching Friends

LEGO studied 3,500 girls and their mothers over four years with structured questions about color, model, and play patterns. The 2012 Friends launch followed, and girls' construction-toy revenue in the US and major European markets grew from \$300M (2011) to \$900M (2014).

[Read more](#)

Further reading

 [Survey Monkey: How to Analyze Survey Data](#)

- ☰ Survey Monkey: Types of survey questions
- ☰ Bernard Choi and Anita Pak: A Catalog of Biases in Questionnaires
- ☰ Formplus Blog: What are Close Ended Question Examples & Types?
- ☰ Hotjar: Open-ended questions vs. close-ended questions: examples and how to survey users
- ☰ QuestionPro: Close Ended Questions
- ☰ Pew Research Center: Why open-ended questions draw higher nonresponse than closed ones
- ☰ Top 21 Best Online Survey Software and Questionnaire Tools: An overview
- ☰ Kromatic: Experiment Calculator

4.2 Comprehension Test

In Brief

Show your headline, tagline, or any other text to a few people for a few seconds, remove it, and ask them to explain what you offer in their own words. The output is a pass/fail comprehension rate. It tells you whether low conversion on a smoke test comes from a messaging problem rather than a lack of demand — so you don't abandon an idea people simply didn't understand.

COMMON USE CASE

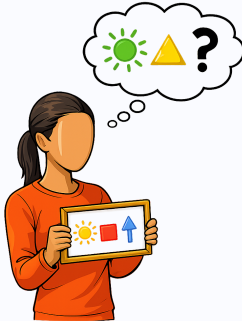
You ran a landing page test and the conversion rate was low. Before concluding that nobody wants your product, you want to check whether visitors actually understood what you were offering. You show your headline and tagline to 20 people for a few seconds, then ask them to explain it back. Half of them get it wrong, telling you the problem is your messaging, not your idea.

HELPS ANSWER

- Do people understand what we are offering?
- How could we explain our product more clearly?
- Which version of our messaging is easiest to understand?
- What do people think we do after reading our headline?

Description

A comprehension test answers one question: when people see your messaging for the first time, do they understand what you're offering? It is a diagnostic tool, not a demand test.



AT A GLANCE

Quantitative

Qualitative

In-person

Remote

Attitudinal

TIME

 ~1 day–1.5 weeks

COST

 \$0–\$300

OTHER NAMES

Five-Second Test · Message Comprehension Test

The classic format is a five-second test: show your headline, tagline, or landing page for a brief exposure (typically 5 seconds, though some platforms allow up to 20), remove it, and ask the participant to explain what the product does in their own words. If they can't, your messaging has a clarity problem, and more traffic or ad spend will not fix a value proposition people don't understand.

Comprehension tests are most valuable as a checkpoint before or alongside other smoke tests such as the [Landing Page Test](#). If your landing page converts at 1% (one in a hundred visitors takes the action you asked for), a comprehension test tells you whether that's because nobody wants the product (a demand problem) or because nobody understands it (a messaging problem). The distinction changes your next move: a demand problem means rethink the product; a messaging problem means rewrite the headline.

Don't use a comprehension test to measure desire, interest, or willingness to pay. A participant can fully understand your value proposition and still not want it. Comprehension is necessary but not sufficient.

How to

Prep

✦ AI Prompt

1. Define what you are testing.

Write down the single question this run must answer: "Do people understand what we're offering?" It does not test whether they want it, only whether they get it. Decide which messaging you are putting under test and what a passing answer looks like before you recruit anyone.

2. Prepare the stimulus.

Write out the messaging you want to test. This can be:

- A headline and tagline (most common)
- A landing page screenshot
- A product description paragraph
- A pitch deck slide
- An ad mock-up

Keep it to what a real user would see in a real context. If your landing page headline is "Smart scheduling for busy teams," test that exact phrase — don't rewrite it into a polished explanation.

3. Choose your test format.

- **In-person (5-second test):** Show the messaging on a screen or printed card for 5 seconds, then remove it. Ask the participant to explain what the product or service does. Best for quick iteration — you can run through a handful of people in an afternoon.
- **Online (5-second test platform):** Use a five-second-test platform to show the stimulus briefly and collect open-ended recall responses. Best for reaching more people or testing across geographies.
- **AI pre-screen (optional):** Before testing with real people, paste your messaging into an AI tool and ask it to explain it back as if it were a first-time reader. This catches obvious clarity problems and saves you a round of iteration — but it is not a substitute for human testing, because AI has far more reading comprehension than a real person scanning your page for 5 seconds.

4. Set your sample size.

- **Minimum:** 10 participants for a directional signal.
- **Ideal:** 20 participants for a steadier comprehension rate.
- Participants don't need to be your exact target customers, but they must have a similar vocabulary and context level. A junior marketing manager can stand in for a CMO; a software engineer cannot stand in for a retail store owner.

5. Set your success threshold.

Comprehension should be high — this is not a conversion test where 5% is normal.

Benchmarks:

- **80%+ comprehension:** Your messaging works. Move to conversion testing.
- **50–80%:** Messaging is partially landing. Identify which parts confuse people and rewrite.
- **Below 50%:** Fundamental clarity problem. Don't run any other smoke tests until you fix the messaging.

6. Write your test questions.

Keep questions neutral — don't lead toward the correct answer. Core questions:

- "In your own words, what does this product or service do?" (primary recall question)
- "Who do you think this is for?"
- "What would you expect to happen if you signed up or bought this?"
- "What, if anything, was confusing or unclear?"

Don't ask "Did you understand it?" — people will say yes even when they didn't. Ask them to demonstrate understanding by explaining it back.

Execution

✦ AI Prompt

1. Run the test.

- **In-person:** Show the stimulus for 5 seconds. Remove it. Ask your questions immediately. Don't explain, correct, or react to their answers — nod and move on. If a participant says "I have no idea," that's valid data, not a prompt for you to clarify.
- **Online:** Set up the 5-second exposure in your testing platform. Include the open-ended recall question as the first follow-up. Keep the total test under 3 minutes — longer tests get abandoned or attract careless responses.

TOOLS

Any printed card and phone timer (in-person) — a headline printed or written on a card and any timer for the 5-second exposure; under \$10, best for fast solo iteration

[Lyssna](#)

[Wynter](#)

[Maze](#)

2. Record answers verbatim.

Write down exactly what participants say, not your interpretation. "It's like a calendar thing for teams" and "It schedules meetings automatically" are very different responses to a scheduling product, and that difference matters.

3. Test one variant at a time.

If you're comparing multiple messaging options, show each participant only one version. Don't show all variants to the same person — exposure to the first one contaminates their reading of the second.

Analysis

✦ AI Prompt

1. Score each response as pass or fail.

A response passes if the participant correctly identifies what the product does and who it's for. They don't need to use your exact words — a rough paraphrase in their own language counts. Grade generously on specifics but strictly on the core concept: "It helps teams schedule stuff" passes for a scheduling tool; "It's some kind of business software" does not.

2. Calculate the comprehension rate.

Comprehension rate = passes ÷ total participants × 100. Compare against your pre-set threshold (target 80%+).

3. Look for patterns in the misunderstandings.

If participants fail, how do they fail? Common patterns:

- **Wrong product category:** They think you do something completely different → your headline is misleading.
- **Right category, wrong specifics:** They get the general idea but miss the key differentiator → your tagline or sub-headline isn't doing its job.
- **Partial understanding:** They get half the value proposition → the messaging is trying to say too much at once. Simplify.

4. Mine participant language for messaging ideas.

If several participants use the same phrase to describe your product, that phrasing may work better than your original. Participants who “get it” often explain it more clearly than you do — because they strip away your insider knowledge and describe what actually landed.

5. For small samples (under 10 participants): Treat the comprehension rate as directional. Focus on whether the misunderstandings cluster around the same issue. If 3 out of 7 people misunderstand the same thing, that's a clear signal regardless of sample size.

Biases & Tips

Curse of knowledge

You've seen your messaging hundreds of times and can no longer judge its clarity; what reads as obvious to you may be jargon to a first-time reader. Test only with people who have never seen your product, and never use your own read as the benchmark.

Mode effects

Online five-second tests tend to score lower comprehension than in-person sessions because remote participants are more likely to be distracted or clicking through carelessly. Don't compare an online rate against an in-person one; pick one mode for a given decision, or run a small in-person check before rewriting on online data alone.

Charitable scoring (confirmation bias)

When you grade the responses yourself, you tend to read your intended meaning into vague answers and pass them. Decide the pass criteria before you read any response, score blind to which variant produced each answer, or have someone who didn't write the messaging grade it.

Experimenter bias from

cueing

Reacting to answers, nodding, or re-showing the message teaches participants the “right” answer and inflates the rate. Show the stimulus once, ask the questions verbatim, and stay neutral; log “I have no idea” as data, not as a prompt to clarify.

Survivorship bias

Online platforms drop participants who give up before the recall question, and those drop-outs are often the people most confused by the message. Track exits before recall and treat a high exit rate as a comprehension signal, not as missing data to discard.

Next Steps

- If comprehension clears your threshold, run a [Landing Page Test](#) with the variant that tested highest to measure conversion on a value proposition you know people understand.
- If comprehension is partial, rewrite the messaging using language from participant responses, then watch real users react to it in a [Usability Test](#) before running any conversion test.
- If comprehension is far below threshold, simplify: cut the headline to one idea and remove jargon, then rerun the test and confirm the rewrite with a [Closed-End Survey](#) of which version reads clearest.
- If a complex value proposition is hard to convey in static copy, run a [Video Test](#) and measure comprehension of the explanation instead.

Case Studies

PandaDoc

ran a 50-person message test and found its target users misread the value proposition (“docs” read as targeting doctors; “on-brand docs” read as unclear), then revised the messaging.

[Read more](#)

Hamona Premium Coconuts: comprehension test walkthrough

A short slide deck running a comprehension test on the Hamona value proposition and showing what the recall responses surfaced.

[Read more](#)

Further reading

- ☰ Lyssna: An introduction to five second testing
- ☰ GrasshopperHerder: Comprehension vs. Commitment
- ☰ Josh Tong: Ways to test content with users
- ☰ Nielsen Norman Group: Legibility, Readability, and Comprehension: Making Users Read Your Words

4.3 Secondary Market Research

In Brief

Secondary market research is the desk-based work that produces a **defensible market-size estimate** and a **macro-trend map** of the regulatory, demographic, technology, and channel shifts that materially change the sizing picture in the next 12–24 months. The output is a sizing-and-trend brief that tests your market hypothesis — its size, segment, competition, and direction — against published evidence, so you can make a go/no-go call before spending time and money on primary research.

COMMON USE CASE

You have a market hypothesis but no customers yet. Before committing budget to interviews, smoke tests, or a build, you want a defensible answer to “is this market big enough, and is it moving in a direction that helps or hurts us?” — sourced from published material a skeptical investor or partner could check.

HELPS ANSWER

- How large is the market for our specific segment, geography, and use case — not the headline industry number?
- How does our bottom-up build-up compare to the top-down published TAM, and where are the assumptions most fragile?
- Which macro trends — regulatory, demographic, technology, channel-cost — would materially change the sizing picture in the next 12–24 months?
- What pricing benchmarks and willingness-to-pay signals exist in published material for this segment?
- What are realistic customer-acquisition-cost, retention, and channel-cost benchmarks for this category?



AT A GLANCE

Quantitative

Remote

Observational



TIME  ~3 days–2 weeks

COST  \$0–\$20K

OTHER NAMES

Desk Research · Market Sizing
Research · TAM Research ·
Macro Trend Research

- Which findings are strong enough to act on, and which need primary research before we commit budget?

Description

Secondary market research tests a market hypothesis you already hold — about its size, segment, competition, or trajectory — against existing published material: government statistics, analyst reports, academic research, regulatory filings, trade-association data, public-company disclosures, founder benchmarks, and competitor data. You come in with a stated belief (“this segment is large enough and moving in our favor”) and use desk research to confirm, resize, or kill your idea using evidence that already exists. The goal is a defensible TAM / SAM / SOM — total addressable market (everyone who could buy), serviceable addressable market (the slice you can reach), and serviceable obtainable market (the slice you can realistically win) — and a trend map that together validate, or refute, the hypothesis before you commit primary-research budget.

Approaches to secondary market research vary, but two principles hold throughout.

The first is **triangulated and bottom-up by default**. A single TAM number from a single analyst press release is a guess. A defensible TAM rests on **at least two independent estimation paths** — a top-down estimate from analyst or government data and a bottom-up build-up from named-segment counts multiplied by annual contract value (ACV, the revenue one customer generates per year) — that agree to within roughly 3–5×. When the two disagree by more than that, the assumptions need work, not the number. A polished \$50B TAM pulled from one report is usually less credible than a \$200M TAM with airtight bottom-up math, explicit segment definitions, and visible sensitivity to its assumptions. The same triangulation rule applies to macro trends: every trend claim should rest on at least two independent perspectives — analyst forecast plus government data, or trade press plus a public-company 10-K, not two restatements of the same primary source.

The second is **citation-backed claims with mandatory fact-check**. AI assembles secondary-research material quickly, but current models fabricate report titles, statistics, and citation URLs at rates between 18% and 94% depending on the task. The most pernicious failure mode is **real URL plus fabricated claim** — the citation looks defensible because the URL resolves, but the cited page does not actually contain the cited number. Every numeric claim and every named source must trace to a verifiable URL, and a fact-check pass — by a human or a verification subagent that

re-fetches each URL and confirms the cited claim is on the cited page — is part of the method.

For query-level demand signal — what people are typing into Google and asking AI assistants right now — use [Search Trend Analysis](#). This page is about how big a market is and where it is heading according to published material; that one is about what the audience is actively searching for. The two methods are complementary and often run in parallel during the same week.

How to

Prep

★ AI Prompt

1. **Frame the decision question, not the topic.** “Is this market big enough?” is too vague to research. “How many US mid-market fintechs (50–500 employees) would pay \$30K+/yr for compliance automation today, and how do the EU AI Act + state privacy laws shift that number over the next two years?” is researchable and decision-anchored. Limit yourself to 2–3 such questions per research session — anything more and the work expands indefinitely.
2. **Define the segment slice you can actually reach.** Not “businesses” but “B2B SaaS companies with 10–50 employees in the US.” Not “consumers” but “homeowners aged 30–50 with household income above \$100K.” Every later number — TAM, SAM, SOM, willingness-to-pay benchmarks, channel-cost estimates — is anchored to this slice. Skip this step and you produce a headline industry number that looks finished but tells you nothing about the market you can actually sell to.
3. **Identify candidate sources across multiple buckets.** A single source is a guess; multiple sources from different buckets is a triangulation. Cross-reference at least three of these source-pool buckets before locking your reading list:
 - **Government data** — Census Business Builder, Bureau of Labor Statistics, SBA.gov, EU Eurostat, country-level statistics offices. Free, large samples, slow to update.
 - **Library-accessible analyst databases** — Most US public libraries and university libraries carry at least one major consumer- or industry-research database; ask the reference desk before paying.
 - **Free analyst summaries** — McKinsey Global Institute, Bain Insights, BCG Insights, HBR, World Economic Forum, OECD. Full reports are paywalled; the free-tier summaries often carry the headline number you need.

- **Founder benchmarks** — annual benchmark reports published by SaaS lenders, VC firms, and payment platforms. Free, recent, founder-relevant.
- **Regulatory filings** — SEC 10-Ks for public competitors and adjacent players, FDA / EMA pipelines for regulated categories, EU AI Act compliance filings, state-level privacy filings. Underused; often the only place segment-level numbers are disclosed.
- **Trade associations** — IAB for ad tech, ICMA for capital markets, AHIP for health insurance, RIAA for music — often publish unit-economics and channel-cost data not available elsewhere.
- **Trade press and beat reporting** — sector-specific publications and dedicated beat coverage. Useful for recency, but always trace the cited primary source.

If a number shows up in only one bucket, treat it as directional. If it shows up in two or more independent buckets, it earns a slot in your sizing.

4. **Sketch your sizing approach and trend dimensions before pulling data.** Decide your TAM / SAM / SOM build-up plan in advance:

- **Top-down** — start from a published industry total, multiply by the share that matches your segment slice (geography × company size × use case).
- **Bottom-up** — start from a named count of segment members (Census counts, funding databases, professional networks) × ACV × time period.
- **Benchmark-anchored** — anchor an unknown number to a comparable category's known benchmark (e.g., scheduling-software ACV ≈ project-management-software ACV at the same segment size).

Plan to run at least two paths and triangulate. For macro trends, pick the dimensions that matter to your specific segment from the PESTEL frame — Political, Economic, Social, Technological, Environmental, Legal — and drop the dimensions that do not. For most B2B software the dimensions that matter most are Legal (compliance regimes), Technological (adoption curves of upstream platforms), and Economic (segment-level capital-spending shifts); Environmental rarely applies. Stub the trend map with the dimensions you will fill, before collecting data.

5. **Tag claims as decision-grade or directional.** Some claims drive decisions (TAM in the 1–5× range you are committing to, willingness-to-pay benchmarks for the segment you are pricing against, regulatory pipeline that affects your go-to-market). Others are background context (rough industry growth rate, generic technology-adoption color). Tag each candidate claim **decision-grade** (must be verified against a primary source and re-fetched by a fact-check pass) or

directional (a roundup or AI summary is fine). Without this triage, fact-checking either takes forever or doesn't happen.

Execution

✦ AI Prompt

The verified reading list from Prep is now the input to structured collection. The goal here is not to write essays on each source; it is to fill a fixed dual schema — sizing facts and trend facts — so the synthesis stage has clean inputs.

1. **Use a fixed dual schema, not free-form notes.** Every source produces two structured records:

Sizing record — segment definition (geography × company size or demographic × use case), headline number, methodology used by the source to derive it, sample or sample size, year of underlying data, who funded the report (if applicable), and any caveats the source itself flagged. Inline [1] , [2] citations on every numeric claim.

Trend record — PESTEL dimension(s) the source addresses, the specific shift being claimed, the time horizon, the methodology behind the projection, the named affected segment, and any countervailing forces the source acknowledged. Inline citations on every projection.

A Sources list at the end of each record resolves every marker to a working URL with the date the source was retrieved.

2. **Collect each field from its canonical source.**

- Segment counts and demographics → government statistical data, funding databases, and professional-network counts → bottom-up sizing.
- Top-down industry totals → analyst reports and library-accessible databases → top-down sanity check.
- ACV, willingness-to-pay, retention, CAC benchmarks → founder benchmark reports, public-company 10-Ks for adjacent players, trade-association unit-economics reports → bottom-up multipliers.
- Regulatory pipeline → primary regulatory filings (SEC EDGAR for public companies, FDA / EMA pipelines, EU AI Act timeline documents, state-privacy law trackers like IAPP) → trend map (Legal).
- Technology-adoption curves and platform shifts → analyst adoption forecasts, public-company 10-Ks of upstream platforms, World Economic Forum and OECD reports → trend map (Technological).
- Demographic and economic trends → Census projections, BLS labor projections, IMF / OECD forecasts → trend map (Social, Economic).

3. **Build top-down and bottom-up in parallel.** Run both estimation paths from the start, not sequentially. The point is not to pick one — the point is to see whether they triangulate. If your bottom-up is \$200M and your top-down is \$50B, one of the two is reading the wrong segment and you need to find out which before you write anything.
4. **AI does the first pass; humans verify decision-grade claims.** A capable LLM with web access can fill 70–80% of the dual schema in hours. The remaining 20–30% — and any claim you tagged as decision-grade in Prep — must be confirmed by visiting the cited source yourself or routing through a fact-check subagent. Pricing pages change weekly; analyst-report headline numbers shift between editions; “X% growth” claims often hide a different segment definition than yours.
5. **Flag everything gated as MANUAL.** Anything behind a login, paywall, library-only access, or paid analyst report gets flagged as **MANUAL** — [what would resolve it] in the record. AI cannot fetch behind authentication, and a confidently-stated number from a source the agent could not actually retrieve is the most common failure mode of AI-generated market sizing.

Analysis

Analysis turns the collected records into a sizing-and-trend brief that supports a decision. Each artifact below should end in a sentence that names a choice you are now better equipped to make.

1. **Triangulate sizing — top-down vs. bottom-up.** Lay your top-down number, your bottom-up number, and any benchmark-anchored numbers side by side. If they agree to within roughly 3–5×, your sizing is defensible — pick the bottom-up as your primary number (it is more defensible to investors and partners) and use the top-down as the sanity-check ceiling. If they disagree by more than that, your assumptions are wrong somewhere and you cannot proceed without finding the gap. Most often the gap is segment definition: the top-down counts a wider universe than the bottom-up reaches.
2. **Stress-test every load-bearing assumption.** For each number you plan to act on, ask:
 - Does the underlying data match the geography, segment, and time period I am sizing? (Specificity drift is the most common failure.)

✦ AI Prompt

TOOLS

Census Business
Builder
SEC EDGAR
Statista
IBISWorld
Perplexity AI

- Is the data still recent enough to act on? (Anything older than 24 months in a fast-moving category is directional, not decision-grade.)
 - Is the source's definition of the category the same as mine? (Definition drift — "B2B SaaS" in one report includes services revenue, in another it doesn't.)
 - What conversion rate from "addressable" to "actually reached" am I using, and is it sourced or fabricated? (Whimsical conversion rates produce optimistic SOMs that fall apart on first contact with primary research.)
 - If the source is vendor-funded, does the vendor benefit from a larger headline number? (Almost always yes.)
3. **Map macro trends to the sizing picture.** For each PESTEL dimension you collected, ask: does this trend materially change the SAM or SOM in the next 12–24 months, and in which direction? Tag each trend **expanding** (increases the addressable segment), **contracting** (shrinks it), or **shifting** (changes the segment definition rather than the size — for example, EU AI Act compliance becomes a buying criterion, raising willingness-to-pay among compliance-mature buyers and disqualifying compliance-immature ones). Drop trends that do not materially change the picture; a trend map of fifteen items is unusable.
4. **Run a sensitivity analysis on the SOM.** Pick the two or three load-bearing assumptions in your bottom-up build (segment count, conversion rate, ACV, retention), and compute the SOM at +30% / -30% on each. If the SOM swings more than 3× under reasonable assumption ranges, your number is fragile — say so in the brief, and identify the single primary-research move that would tighten the most fragile assumption.
5. **Write the strategic insight.** A 200–400 word brief answering five questions in order:
- **What is the realistic SOM for our segment slice in the next 12 months?** Bottom-up build, named segment, sourced ACV, sourced conversion rate.
 - **What is the top-down ceiling, and how far is the bottom-up from it?** A short sentence on triangulation; if the gap is large, name where it lives.
 - **Which 2–3 macro trends materially change the SOM in 12–24 months, and in which direction?** Each trend must rest on at least two independent sources.
 - **Which 2–3 assumptions are most fragile, and what primary research would tighten them?** Names the next research move.
 - **Go / no-go / not-yet?** A one-sentence recommendation grounded in the four answers above.

This is the artifact you actually act on. The schemas, citations, and sensitivity tables exist to make this paragraph defensible.

⚠ Biases & Tips

AI-fabricated sources (hallucination)

When AI assembles the raw material, it invents plausible report titles, statistics, and citation URLs. The most pernicious version is a real, resolving URL paired with a number that is not actually on the page, so the citation looks defensible while the claim is invented. Never treat an AI-assembled claim as verified. Run a **secondary fact-check agent** as its own pass: hand it the assembled brief and have it re-fetch every cited URL, confirm the specific cited number or quote actually appears on that page, and label each claim **VERIFIED**, **FAILED VERIFICATION** (the URL loads but does not support the claim — the dangerous case), or **UNREACHABLE** (404, paywalled, or blocked). Keep every decision-grade claim unverified until that pass clears it. The fact-check subagent built into the Prep, Execution, and Analysis prompts above is written for exactly this.

Aspirational thinking disguised as analysis

Founders pick TAM numbers that support the story they want to tell, then back-fit the methodology. Force yourself to commit to your sizing path and segment definition in Prep, before you have seen any of the headline numbers; if a single source flips your conclusion, you were fitting the method to a predetermined answer.

Vendor-report bias

Industry analysts and trend-report publishers benefit from a larger headline number; their commercial interest is often aligned with optimism. When you cite a vendor-funded report, weight it accordingly and require an independent corroboration before treating its number as decision-grade.

Confirmation bias in conversion rates

The most common way founders inflate SOM is by importing a whimsical conversion rate from “addressable” to “actually reached.” Use sourced conversion rates from comparable categories (founder benchmarks are useful here) and run sensitivity at +30% / -30%; if the SOM only stands up at the optimistic end, say so in the brief.

Willingness-to-pay vs. ability-to-pay

Search volume, expressed interest, and “would you use this” survey results indicate willingness-to-use, not willingness-to-pay. Anchor pricing benchmarks to actual paid revenue from comparable products, not to interest signals.

Next Steps

→ Validate the load-bearing assumptions in your bottom-up sizing with

[Discovery Interviews](#) — the segment definition and the willingness-to-pay

benchmark are usually the two most fragile.

- For query-level demand signal — what people are searching for in Google and asking AI assistants right now — run [Search Trend Analysis](#). The two methods complement each other: this page sizes the market and maps macro trends; that one maps real-time audience behavior.
- For who is already in the field and where the gaps are, run [Competitive Analysis](#). Sizing tells you whether the market is big enough; competitive analysis tells you where you can fit.
- Test the most fragile pricing assumption with a [Landing Page Test](#) — willingness-to-pay benchmarks from secondary research are directional until a real audience clicks a real price.
- Re-run the sizing-and-trend brief quarterly, and pair it with [Search Trend Analysis](#) to catch shifts between reruns. Regulatory pipelines, analyst forecasts, and category definitions all shift fast enough that a 90-day-old brief is directional, not decision-grade.

Case Studies

Spotify: Reading the music-industry collapse

Daniel Ek framed Spotify around the observation that piracy could not be legislated away, so a paid service had to beat free. The bet was grounded in published industry-revenue decline and rising smartphone adoption rather than primary research.

[Read more](#)

Uber TAM debate: Damodaran vs. Gurley (2014)

Aswath Damodaran sized Uber's global TAM at about \$100B by anchoring to taxi-and-limousine revenue; Bill Gurley argued the relevant TAM was several hundred billion once ride-hailing reshaped private-car ownership. The exchange is the canonical illustration of how segment definition drives sizing.

[Read more](#)

Lighter Capital: 2025 SaaS benchmarks

Lighter Capital's free 2025 benchmark report puts median B2B SaaS revenue growth at 28% (down from 47% in 2024) and upper-quartile growth at 65% (down from 88%), giving founders a public baseline for their own growth-rate assumptions.

[Read more](#)

Further reading

-  [Pearson catalog](#)
-  [Aswath Damodaran — A Disruptive Cab Ride to Riches: The Uber Payoff \(2014\)](#)
-  [Bill Gurley — How to Miss By a Mile: An Alternative Look at Uber's Potential Market Size \(2014\)](#)
-  [SBA.gov: Market research and competitive analysis](#)
-  [Cornell University Johnson Library FAQ: How can I find market research reports and data?](#)
-  [Anthropic: How we built our multi-agent research system \(2025\)](#)
-  [Hallucination to Truth: A Review of Fact-Checking and Factuality Evaluation in Large Language Models \(arXiv 2508.03860, 2025\)](#)
-  [AI Hallucination Rate Benchmarks 2026](#)
-  [HBR: The AI Tools That Are Transforming Market Research \(2025\)](#)
-  [Bain & Company — Macro Trends Research](#)
-  [PESTLE Analysis \(SI Labs\)](#)
-  [Census.gov — Economic Census](#)

4.4 Value Proposition Test

In Brief

A Value Proposition Test (VPT) is any method where you present a value proposition and ask the participant to give you something — money, time, data, or actions — in return. The signal is commitment to a promise, not interaction with a working product. The family ladders from low-friction signals (clicking an ad, leaving an email) up to real prepayment, letting you escalate evidence as your confidence grows.

COMMON USE CASE

You have a clear hypothesis about what solves a customer problem and need quantitative evidence of demand BEFORE building. You are choosing between several positionings, segments, or price points and want behavioral signal — not survey opinion — to pick a direction.

HELPS ANSWER

- Which value proposition pulls the strongest commitment from this segment?
- Is the demand strong enough to justify building?
- How much friction will prospects tolerate before they walk away?
- Which method in the VPT family fits the commitment level I need to test?
- What evidence will convince my team, my investors, or me that this is real?

Description

A Value Proposition Test makes a promise and asks the prospect to commit something in return, then measures how many prospects follow through. What varies across the methods in the family is *what* you ask for — a click, an email, time, a public endorsement, or money. Lighter commitments are cheap to collect but easy to give, so they signal less; heavier ones cost the prospect more, so a yes carries more weight. The methods are grouped below by the type of commitment, ordered from



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~4 days–4 months

COST  \$0–\$5K

lowest to highest, so you can pick the test that matches the evidence your next decision needs.

Choosing the Right Value Proposition Test

Pick the lowest-commitment method that produces evidence strong enough for your next decision. Running a higher-commitment test before a lower one wastes budget; running a lower one when you already have its signal wastes time.

Data Commitment

Prospects give up data — clicks, email addresses, scroll depth, watch time. The lightest signal in the family. Useful early, when you're testing whether your message lands and don't yet have evidence of intent.

Online Ad Test

Evaluates the value proposition via click-through rate on a paid ad. Best used to compare several value proposition wordings before building a full landing page test.

Landing Page Test

Asks visitors to leave an email address in exchange for early access on a one-page website. Best used when you have an ad budget and want a fast read on whether a value proposition pulls signups.

Flyer Test

Counts how many people scan a QR code, call a number, or visit a URL printed on a physical flyer. Best used when your audience is reachable in person and digital channels do not fit the segment.

Fake Door Test

Counts clicks on a feature, plan, or option that does not exist yet, placed inside an otherwise real product. Best used to choose between candidate features inside a product that already has traffic.

Video Test

Measures watch time, sharing, and call-to-action clicks on a short explainer or trailer video. Best used when the story is visual and your audience prefers to watch over to read.

Time Commitment

Prospects give up time — long forms, scheduled conversations, attendance. Stronger than data because time is irreplaceable.

**■ Landing Page
(Extended) Test**

Adds pricing tiers, scroll-depth tracking, and tier-specific clicks to a longer landing page. Best used after a basic landing page passes and you need to figure out which price tier or feature bundle prospects respond to.

■ High Bar Test

Adds friction (long forms, applications, qualifying questions) and counts who still completes the process. Best used to separate real buying intent from idle curiosity when you cannot yet charge real money.

■ Event Test

Counts how many prospects will give up time (and sometimes money) to attend a promised experience around the value proposition. Best used when you want both quantitative attendance signal and qualitative conversation in the same test.

Social Commitment

Prospects pledge social capital — a recommendation, a public association with the product, a willingness to bring others. Stronger than time because reputation is at stake.

■ Broken Promise Test

Asks prospects to pledge a share or a friend invite in exchange for promised exclusive access. Best used when the question is whether the value proposition spreads — not just whether prospects sign up for it.

■ Referral Program Test

Counts how many prospects actually refer friends in return for early access, queue position, or rewards. Best used to test whether word-of-mouth can be a real acquisition channel, not just whether demand exists.

Money Commitment

Prospects commit money — pretend transaction, real transaction, or prepayment. The strongest signal in the family. Use when you need evidence that survives investor scrutiny or before scaling spend.

■ Sales Pitch Test

Pitches the offer live to a prospect and asks for a yes-or-no decision on the spot. Best used in B2B with named accounts you can pitch directly when objections matter as much as the close rate.

Mock Sale Test

Walks prospects through a simulated checkout that captures purchase intent without ever taking real money. Best used when you need a strong purchase signal but cannot yet stand up payment infrastructure.

Relabel Test

Sells a rebranded competitor product to test demand for the type of solution, without building anything new. Best used when the question is whether anyone wants this kind of solution at all, not whether they want yours.

Crowdfunding Test

Asks prospects to pledge real money on a public platform for a product that does not yet exist. Best used when you need demand evidence, working capital, and social proof from the same campaign.

Pop-Up Store Test

Sets up a temporary physical retail location and counts real transactions for the proposed product. Best used for physical products with a retail-relevant audience and a time-boxed decision.

Pre-Sales Test

Asks customers to prepay for a product not yet built, with a delivery date in the future. Best used when you need the strongest possible pre-build demand signal and can manage the delivery promise without losing trust.

4.4.1 Relabel Test

In Brief

A relabel test takes an existing competitor’s product (or a similar product from another market), rebrands it under your own name, and presents it to your target audience to see whether they will buy it. The signal is a purchase, real money changing hands, not a click or a signup. If prospects buy the relabeled product, demand exists for the solution category. If they don’t, you have learned that before spending months building your own version.

COMMON USE CASE

You have a hypothesis that a category of solution will sell to your audience, but the build cost is high enough that you need evidence before you commit. Building the actual product would take months and capital you do not yet have, and you cannot tell from interviews alone whether stated interest will translate into purchases. A relabel test lets you validate category demand using an existing product as a stand-in, separating the “do they want it?” question from the “can we build it?” question.

HELPS ANSWER

- Do customers in my target segment want this type of solution?
- Is there demand for this product category independent of who provides it?
- Will my target audience buy this kind of product from my brand or channel?
- Can I validate demand without building anything?

Description

A relabel test borrows an existing product instead of building one: you take a product similar to what you intend to build, present it to your target audience as your own



AT A GLANCE

Quantitative

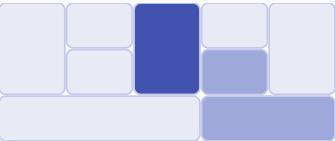
Qualitative

B2C

DTC

Remote

Transactional



TIME

~1–3 weeks

COST

\$30–\$700

OTHER NAMES

Boomerang · Impersonator · Imposter Judo

offering, and measure whether they pay for it. It sits at the highest-commitment rung of the [Value Proposition Test](#) family — because the commitment you ask for is a purchase, a buyer who pays is a customer, and a sale is the strongest demand signal a stand-in can produce.

The method separates two questions startups often confuse: “can we build it?” and “does anyone want it?” If prospects buy a rebranded competitor product, the solution category has pull, and you can invest in building a differentiated version.

This is the opposite of an [Off-Brand Test](#), which strips your brand from your own product to test demand without brand influence. In a relabel test you add your brand to someone else’s product to test whether your audience wants this category of solution from you.

Ethics — You’re selling a real product. A relabel test sells a real (competitor’s) product under your own name to real buyers for real money. Be honest about the source if a buyer asks, fulfill or refund every order, and respect trademark and resale law — never permanently pass off another company’s product as your own. Concretely:

- Do NOT violate trademarks or intellectual property. You’re rebranding for a small-scale test, not launching a counterfeit business. Prefer white-label or private-label products that are explicitly designed for rebranding.
- If you’re reselling a physical product, ensure you have the legal right to do so, and know that some industries have strict relabeling rules (food, pharmaceuticals, medical devices).

How to

Prep

✦ AI Prompt

1. Identify the product category you want to test.

Be clear about what you’re validating: “Do small business owners want an automated bookkeeping tool?” not “Do they want our specific automated bookkeeping tool with our unique AI model?” A relabel test validates category demand, not your specific implementation.

2. Find an existing product to rebrand.

Look for products that are close enough to your concept to test the core value proposition. Sources include:

- Competitor products you can purchase and resell

- White-label products designed for rebranding
- Open-source tools you can wrap in your own branding
- Products from adjacent markets that serve a similar need

3. Create your rebranded presentation.

How much rebranding you need depends on the test format:

- **Digital product/service:** Create a landing page or product listing that describes the product as yours. When a customer purchases, you fulfill with the existing product.
- **Physical product:** Repackage with your branding. For small-scale tests, simple custom packaging or labels may suffice.
- **Service:** Describe the service as your offering. When a customer signs up, deliver using the existing provider (this is essentially how Zappos started).

Execution

✦ AI Prompt

1. Present to your target audience.

Use the same channels you'd use for your actual product: paid ads, direct outreach, events, or your existing audience. The channel matters because you're testing whether your specific audience wants this type of solution.

2. Measure purchase behavior, not just interest.

The strongest signal is whether people buy. If your test only measures clicks or signups, you are running a

📖 [Landing Page Test](#), not a relabel test. Push for actual

transactions, or at minimum a deep-funnel commitment such as entering payment details.

3. Talk to buyers as soon as the order ships.

Reach out to every buyer within a week. Ask what attracted them, what they'd change, and what they were using before. These conversations are the qualitative half of the test — without them, you have purchase counts but not the differentiation insight you'll need to build your own version.

TOOLS

[Alibaba](#)

[Shopify](#)

Any design tool (Canva, Adobe Express, etc.) — design rebranded packaging and marketing materials

Analysis

✦ AI Prompt

1. Compare results against the thresholds you wrote down before launch.

The thresholds (units sold, conversion rate of impressions to buyers, return or complaint rate) tell you what counts as a hit. Without them, every result feels like “interesting data” and no decision gets made.

2. Read the result patterns:

- **Strong sales:** Demand for the solution category is validated. Proceed to build your differentiated version, focusing on the specific improvements customers mentioned.
- **Moderate sales with clear feedback:** Some demand exists but customers want modifications. Use their feedback to design your unique product.
- **No sales despite adequate exposure:** The solution category doesn’t resonate with this audience through this channel. Consider whether the audience is wrong, the channel is wrong, or the solution category genuinely lacks demand.
- **Sales but high return/complaint rate:** Customers wanted the category but the specific product you sourced didn’t meet their expectations. This actually validates demand — they cared enough to be disappointed.

3. Cluster the buyer interview data.

Pull every reason-for-buying and every “what I’d change” comment from the follow-up conversations. Group them. The five most-repeated reasons matter more than the one most clever piece of feedback. Repetition is the signal.

4. Separate category demand from your-brand demand.

A relabel test on a brand-new company tests both at once. If your audience is small and trusting (existing newsletter, prior customers), inflated conversion may reflect your brand pull, not category pull. Discount the result accordingly. A repeat relabel test through a cold paid channel is the cleaner read of category demand.

Biases & Tips

Product-quality confound

If the borrowed product is poor quality, negative reactions may reflect the product’s execution rather than the value proposition. Choose a decent-quality product to rebrand so a weak result points at demand, not at a bad stand-in.

Channel generalizability bias

A strong buy rate on one channel proves demand only on that channel. Run the relabel test on the channel you would actually use to acquire buyers, and treat a warm-audience result as the ceiling, not the baseline.

Confirmation bias

Founders who want the category to work read any sale as proof and explain away weak numbers. Write down the units-sold and conversion thresholds before launch and judge against them, not against your hope for the result.

Brand-halo confound

When you sell through your own newsletter or prior buyers, their trust in you can inflate the buy rate so it reflects brand pull, not category pull. Discount any result that came mostly from a warm audience (the Analysis step covers how to correct for it).

Next Steps

- If sales are strong, run [Discovery Interviews](#) with buyers about what specifically attracted them, then design your differentiated version around their feedback.
- If sales are weak, isolate whether the problem is the audience, the channel, or genuine lack of category demand before iterating.
- Use a [Mock Sale Test](#) to push purchase intent further down the funnel without the legal complexity of reselling a competitor product.
- Use a [Landing Page Test](#) to A/B test the messaging that drove buyers in your relabel test against alternative framings of your own brand.
- Use a [Pre-Sales Test](#) to convert validated category demand into committed pre-orders for the differentiated product you plan to build.

Case Studies

Zappos: Buying shoes at retail to fill online orders

Founder Nick Swinmurn photographed shoes at local stores and listed them online; when an order came in he bought the pair at retail and shipped it, validating online shoe demand before holding any inventory.

[Read more](#)

Further reading

-  Eric Ries — The Lean Startup
-  Steve Blank — The Four Steps to the Epiphany
-  Inc. Magazine — How I Did It: Tony Hsieh, Zappos
-  The Delta — Experiments founders can use to validate their way into the market
-  WIPO — Trademarks

4.4.2 Broken Promise Test

In Brief

The broken promise smoke test measures how much natural word-of-mouth a product idea has before anything is built. You share a landing page with a small group of 10-15 people and explicitly ask them not to tell anyone, then track whether new sign-ups appear from people outside that group. The signal is a referral rate: if sign-ups exceed the number of people you contacted, the idea was compelling enough that people broke a social promise to share it.

COMMON USE CASE

You have three different product ideas and limited time to build. You create a simple landing page for each, share it with a small group of contacts, and ask them not to tell anyone. A week later, one page has sign-ups from people you never contacted. That idea has natural word-of-mouth potential worth pursuing.

HELPS ANSWER

- Does this idea address a need people feel strongly enough to share?
- Are people excited enough to tell their friends about it?
- Which of several ideas generates the most organic sharing?
- What kind of person gets most excited about this concept?
- Is there natural word-of-mouth potential before we build anything?

Description

A broken promise smoke test shares a real-looking product idea with a small, hand-picked group, asks them explicitly not to tell anyone, and then watches for sign-ups from people outside that group. Each outside sign-up is a broken promise: someone



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~4 days–2.5 weeks

COST  \$0–\$40

OTHER NAMES
Broken Promise Test · Virality Test

found the idea compelling enough to pass it along despite being asked to keep it quiet. The commitment it tests is social rather than monetary — that places it among the lower-commitment rungs of the [Value Proposition Test](#) family — and the signal it produces is a referral rate measuring organic word-of-mouth before anything is built.

The mechanics work like a [Landing Page Test](#): a dedicated web page with a sign-up button. The difference is the explicit request not to share. On the thank-you page or in follow-up communication, the copy asks the person who signed up NOT to tell others about the idea. After a set period, you compare the original list of people you contacted against the full list of sign-ups. Anyone who was not on your original list is counted as a referral — they heard about it from someone you seeded.

The measure is a referral rate, not a conversion rate. (A conversion rate is the share of people you reached who took the action; a referral rate counts sign-ups against the people you seeded, so it can exceed 100%.)

Referral rate = total sign-ups in the period ÷ number of people originally contacted

A rate above 100% means more people signed up than you reached directly, so the idea spread on its own.

This test suits early-stage founders comparing several ideas at once. It helps identify which ideas have enough pull that people pass them along, so you can focus on the one most worth building.

Note who is being referred where you can. A referral chain that reaches the right kind of prospect tells you more about the target profile than the rate alone, and those referred prospects are worth interviewing to learn what drew them in.

Some founders use AI-simulated personas to pre-test messaging before running the real experiment — a useful sanity check for obvious weaknesses, but not a substitute. The point of this smoke test is to measure real human behavior: whether real people break a real social promise because they find the idea too compelling to keep secret. AI cannot simulate genuine social pressure or authentic word-of-mouth enthusiasm.

How to

Prep

★ AI Prompt

1. Define the “promise” and the ask not to share.

Decide what exclusive access you're offering (early invite, beta seat, founding-member status, a private waitlist) and write the explicit ask not to share. Both halves matter: the offer creates the social capital, and the explicit "please keep this to yourself" is what makes a downstream signup count as a broken promise rather than just a forwarded link. Draft the language you'll use on the thank-you page or follow-up email so the request is unambiguous.

2. Identify and recruit a representative seed group.

Pick 10-15 people who actually look like your target prospect, not friends and family who already love everything you do. Mix in people you know less well — colleagues of colleagues, members of relevant communities — so the network downstream of each seed contains real prospects, not just your existing fan club. Keep an exact list of who you contacted; you will need it to count referrals later.

3. Pre-commit to a referral-rate threshold and pick a traffic source.

Decide before launch what referral rate constitutes a go signal (e.g., signups exceeding 100% of seeds, or a specific count like 30% of seeds producing referrals). Picking the cutoff after you see results is how confirmation bias creeps in. Also decide how you'll attribute traffic — a unique share link per seed (a tracking tag in the URL that records who the visitor came from), a separate landing page per seed, or an explicit "how did you hear about us?" field — so you can tell referrals apart from the original group.

4. Stand up the landing page and tracking before you send anything.

Have the page, signup form, attribution mechanism, and a way to compare "list of seeds" vs. "list of signups" all wired up in advance. Running the test for a week and discovering you can't tell who came from where is the most common way this experiment fails.

Execution

1. **Send the seeded outreach once per seed.** Reach out to your 10-15 seeds individually, each with their own attribution tag or landing-page link wired in. Use the promise copy you wrote in Prep, including the explicit request not to share. Send once — do not follow up with non-responders, since a non-response is part of the signal.
2. **Watch the sign-ups, not the seeds.** Track sign-ups as they arrive and record which attribution tag each one

✦ AI Prompt

TOOLS

[Prefinery](#)

[Viral Loops](#)

Any AI landing page builder (v0, Bolt, Framer) — tracked page in minutes

carries. Sign-ups with no seed tag are the ones to watch — they came from outside the group you contacted.

3. **Stay hands-off for the full window.** Do not nudge seeds, send reminders, or coach anyone on whether they may share. Any prompting contaminates the result; silence is data. If a seed asks whether they can share or posts the page publicly, note it but do not advise them.
4. **Log the spread.** At the end of the window, you want a clean count of who came from each seed and how many arrived with no seed tag. Those untagged sign-ups are the broken promises — people who shared an idea they were asked to keep quiet.

Analysis

✦ AI Prompt

1. **Compare against your pre-committed cutoff.** Read the result against the referral-rate threshold you set in Prep, not one you pick now that you can see the numbers. The cutoff is what turns the rate into a go or no-go decision.
2. **Decide exactly who counts.** Be clear on who belongs in the original group and who counts as a referral. Immediate family — your mom, and anyone she refers — does not count unless she genuinely represents your target market. Excluding non-representative contacts on both sides keeps the rate honest.
3. **Calculate the referral rate.** Divide total sign-ups in the window by the number of people you originally contacted, separating genuine referrals from the seeds. Flag any sign-ups whose source you cannot attribute cleanly rather than guessing which side they fall on.
4. **Read the result against your sample size.** With only 10-15 seeds, a few extra sign-ups swing the rate a lot. Treat a strong result as directional evidence to act on, not a precise measurement, and let the size of the gap above your cutoff set your confidence.

⚠ Biases & Tips

Confirmation bias

Founders tend to seed people who already love the idea, which inflates the apparent referral rate. Recruit a seed group that looks like your target prospect, not just enthusiasts who will share anything you make.

Moving-the-goalpost bias

Setting or adjusting the referral-rate cutoff after you see results is how a weak signal gets reframed as a green light. Fix the cutoff in Prep, before you run the test, and read the result against it unchanged.

**Survivorship
bias on
referrers**

A handful of enthusiastic sharers can mask broad indifference. Two loud advocates are not the same as a widely loved idea, so judge by the overall referral rate rather than by the most active sharer.

**Attribution
error**

When you cannot tell a referred sign-up from one you reached directly, it is easy to credit the idea for traffic it did not earn. Wire up per-seed share links or a “how did you hear about us?” field in Prep, and exclude any sign-up you cannot attribute rather than assuming it was a referral.

**Gaming and
self-referral**

A motivated founder can pad the count by signing up through alternate accounts or asking friends to. Decide up front which contacts are excluded from both the seed and referred counts, and treat any traffic spike that does not resolve to a real source as suspect, not as success.

Next Steps

- If the referral rate clears your cutoff, run a [Pre-Sales Test](#) to turn referral interest into actual purchase commitments before you build.
- Use [Discovery Interviews](#) with the most active referrers to learn what made the idea worth sharing.
- Use a [Concierge Test](#) to deliver the promised value by hand to your most enthusiastic referrers and learn what they actually need.
- If the result is weak, sharpen the messaging and run another [Value Proposition Test](#) before investing in product development.

Case Studies

Further reading

- ☰ The Art of Cleverness — Mike Chen, “Use the Broken-Promise Strategy to validate your startup”
- ☰ Andrew Chen, The Cold Start Problem, HarperBusiness, 2021 — scarcity, waitlist, and invite-only mechanics in early viral growth (Clubhouse and other case studies).
- ☰ Steve Blank — Customer Development Manifesto
- ☰ Eric Ries (2011) — The Lean Startup, Chapter 6 “Test”
- ☰ Tim Ferriss blog — Jeff Raider on Harry’s pre-launch campaign

4.4.3 Crowdfunding Test

In Brief

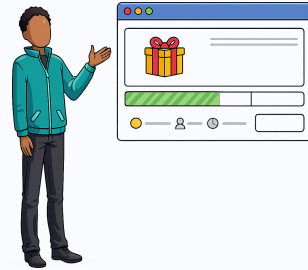
A crowdfunding smoke test is a public campaign on a platform like Kickstarter or Indiegogo that asks people to pay real money up front for a product that does not yet exist. A backer pledges knowing the product hasn't been built — they are buying a promise, which makes them a committed customer rather than just an interested prospect. A pledge combines money with public, reputational stakes, so a funded campaign is one of the strongest demand signals you can buy.

COMMON USE CASE

You have a value proposition that has cleared lighter tests (interviews, landing page signups, ad CTR) and a product concept that benefits from public, pre-payment commitment — the kind of evidence that lets you order tooling, place inventory commits, or take the campaign to investors. You want to see whether enough people will pay before the thing exists, and you can absorb the multi-week preparation effort that a public campaign demands. Crowdfunding gives you that evidence at scale and with real money attached.

HELPS ANSWER

- Will customers pay real money for this product before it exists?
- Is the value proposition strong enough to drive financial commitment?
- Which reward tiers or price points attract the most backers?
- Is there a community willing to champion this product?



AT A GLANCE

Quantitative

B2C

DTC

Remote

Transactional



TIME  ~6 weeks–4 months

COST  \$300–\$6K

OTHER NAMES

Crowdfunding · Crowdfunding Campaign

Description

A crowdfunding smoke test runs a public funding campaign for a product that doesn't exist yet, asking people to back it with real money in exchange for a future reward. Each pledge turns an interested prospect into a paying, committed customer before a single unit is built. It is one of the highest-commitment rungs of the [Value Proposition Test](#) family, because a pledge couples a financial outlay with public, reputational stakes — far more costly to give than a survey answer or a landing page signup, and correspondingly hard to fake.

Crowdfunding works as a smoke test because that cost is the signal. A funded campaign tells you that a meaningful number of people want what you described enough to pay for it before it exists.

Crowdfunding is a public, platform-hosted campaign with a crowd, platform discovery, and social-proof dynamics — distinct from a [Pre-Sales](#) test, which is direct prepayment you collect through your own channel (a pre-order page, deposit, or early-access offer), and distinct from a [Mock Sale](#), where the buyer doesn't know it's a test.

Ethics — You are taking real money for a product that does not exist yet. That is an obligation, not just a test: represent the product honestly, deliver what you promised, and refund backers in full if you decide not to build it. Taking money you cannot honor is fraud, not validation.

What you are testing is the value proposition, not the product. The question is whether the offer is compelling enough that people will pay for something that does not exist yet — the product itself is validated later, after delivery.

There are five primary crowdfunding models, each testing different aspects of demand:

1. **Rewards-based (Kickstarter, Indiegogo):** Backers pay for a future product at different reward tiers. The most common model for product validation. Tests willingness to pay at specific price points.
2. **Equity crowdfunding (Wefunder, Republic):** Investors buy shares in your company. Tests whether sophisticated investors believe in the business model, not just the product.
3. **Donation-based (GoFundMe):** Supporters contribute without expecting a product in return. Tests mission resonance, not product-market fit. Relevant for social enterprises and nonprofits.

4. **Debt-based (Kiva, Funding Circle):** Lenders provide capital expecting repayment. Less relevant as a smoke test.
5. **Recurring/patronage (Patreon, Buy Me a Coffee):** Supporters pay ongoing amounts for continued access or content. Tests whether people will pay repeatedly for your value proposition.

For most product startups, rewards-based crowdfunding is the right model. It directly tests whether customers will pay for what you plan to build.

How to

Prep

✦ AI Prompt

1. Validate before you crowdfund.

Crowdfunding is not a first test. Run lighter experiments first —

[Discovery Interviews](#), a [Landing Page Test](#), or an [Online Ad Test](#) — to confirm basic interest before investing weeks in a campaign.

2. Build a pre-launch email list.

Pre-launch list size is widely cited as the single biggest predictor of campaign outcome. Start collecting emails 4-6 weeks before launch through a simple landing page, social media, and direct outreach to communities where your target customers gather.

3. Create your campaign page and video.

The campaign page needs: a clear explanation of what the product does and who it's for, a compelling video (60-90 seconds), realistic product renderings or prototype photos, well-structured reward tiers, and a transparent timeline. Kickstarter requires honest representation — label renderings as concepts, not finished products.

4. Set a realistic funding goal.

Your goal should be the minimum amount needed to deliver on your promises. Setting a goal too high risks failure; setting it too low undermines credibility. Research comparable campaigns to calibrate.

5. Structure reward tiers strategically.

Offer 3-5 tiers that test different price points and bundles. Include an “early bird” tier at a modest discount to drive urgency in the first 48 hours. The distribution of

backers across tiers tells you about price sensitivity.

Execution

✦ AI Prompt

1. Launch and promote aggressively in the first 48 hours.

Campaigns that hit a meaningful share of their goal in the first two days tend to outperform those that don't, partly because the platforms' discovery surfaces reward early momentum. Activate your pre-launch list immediately. Reach out to press, bloggers, and community influencers.

TOOLS

[Kickstarter](#)

[Indiegogo](#)

[BackerKit](#)

[Wefunder](#)

[Kicktraq](#)

2. Track per-day pledge velocity, not just total raised.

The mid-campaign slump is normal — most campaigns do most of their pledging in the first 72 hours and the last 72 hours. What you want to watch is the slope of pledges per day relative to comparable campaigns. A flat middle is fine; a continually declining slope toward zero is a warning.

3. Communicate with backers throughout.

Regular updates build trust and generate word-of-mouth. Respond to questions quickly. Transparency about challenges increases backer confidence, not decreases it.

4. Hold the campaign page honest.

Resist the urge to add late-campaign claims or stretch goals that you can't actually deliver. Backers screenshot promises, and the Kickstarter / Indiegogo terms of service treat a campaign page as a binding representation. Adding scope under deadline pressure is one of the most common ways successful campaigns become bad-delivery stories later.

Analysis

✦ AI Prompt

1. Compare results against the thresholds you wrote down before launching.

The hypothesis ("at least N backers at \$X average pledge") is the bar. Without it, "we raised some money" feels like success even when the underlying signal is thin.

2. Read the result patterns.

- **Funded above goal:** Strong validation. The value proposition resonates and people will pay. Proceed to build and deliver.
- **Funded but barely:** Marginal signal. You have some demand but not overwhelming pull. Look at which reward tiers sold best and which audience

segments backed you to refine your approach.

- **Not funded:** The value proposition at this price didn't generate enough commitment from this audience. This doesn't necessarily mean the idea is bad – it could mean the campaign execution was weak, the audience was wrong, or the timing was off. Analyze where the funnel broke down.
- **Funded but high cancellation/refund rate:** Backers may have pledged under social pressure or fear of missing out rather than real intent to buy. Net backers (after cancellations) is the metric to trust.
- **One reward tier dominates the rest:** The winning tier is your effective price point. Carry that price into post-campaign sales planning rather than averaging across tiers.

3. Cluster the backer-source data.

Pull the pledge-source breakdown the platform exposes (Kickstarter's "External" vs. "Kickstarter" sources, Indiegogo's referral report). Backers acquired through your own list are demand-pull evidence; backers acquired through the platform's internal discovery are evidence the platform thought your campaign was worth showing, which is a different (and weaker) signal.

4. Discount friends-and-family.

Identify and exclude pledges from your inner circle when you compute the "real" demand rate. Friends-and-family pledges are real money but they don't generalize – they tell you about your relationships, not about market demand.

Biases & Tips

Platform audience bias

Kickstarter backers are a self-selected group of early adopters who enjoy backing new products. Success on Kickstarter doesn't guarantee mainstream demand.

Social proof cascade

Once a campaign gains momentum, backers pile on partly because others have already backed it. The first 30% of funding tends to be the cleanest signal of organic demand; weight early pledges more heavily than late-campaign ones.

Sunk cost after funding

Once you've collected real money, it's psychologically very hard to decide not to build. But if post-campaign research reveals fatal flaws, refunding backers is better than delivering a doomed product.

Video quality bias

A highly produced video can make a mediocre value proposition look compelling. The campaign video should communicate clearly, not

dazzle.

Next Steps

- If the campaign funded, use a [Pre-Sales Test](#) to keep collecting committed orders directly through your own channel while you lock in fulfillment timelines and start manufacturing.
- Use a [Landing Page Test](#) to grow a pre-launch list before re-running the campaign, since pre-launch list size is widely cited as the single biggest predictor of crowdfunding success.
- Use a [Video Test](#) to A/B test campaign video variants against fresh audiences before you commit to a relaunch.

Case Studies

Pebble: 0M Kickstarter against a 00K goal

Pebble's April 2012 Kickstarter raised \$10,266,845 from 68,929 backers against a \$100,000 goal, becoming the most-funded Kickstarter project to that point and validating wrist-worn smartwatch demand years before Apple Watch.

[Read more](#)

Coolnest Cooler: 3M raised, years of unfulfilled backers

The Coolnest Cooler Kickstarter raised over \$13M but the team underestimated manufacturing costs and stretch-goal scope, leaving thousands of backers unfulfilled for years before the company shut down.

[Read more](#)

Allbirds: Wool Runners Kickstarter

Tim Brown's 2014 Kickstarter for wool-runner sneakers raised \$119,000 in five days; Allbirds went on to launch the Wool Runners in 2016 and IPO at over \$4B in 2021.

[Read more](#)

Peloton: Kickstarter for a connected bike

In 2013, John Foley's Kickstarter for a connected exercise bike raised about \$307,000 from a few hundred backers before any product existed.

[Read more](#)

Soma: Water-filter pitcher Kickstarter

Soma's water-filter Kickstarter raised \$100,000 in nine days in 2012, validating willingness to pay for a subscription-delivered filter pitcher before production.

[Read more](#)

Further reading

-  [Kickstarter — Kickstarter Is Not a Store](#)
-  [Kickstarter — Stats](#)
-  [Indiegogo — How to launch a successful crowdfunding campaign](#)
-  [Mollick, E. — The Dynamics of Crowdfunding: An Exploratory Study](#)

4.4.4 Event Test

In Brief

An event smoke test is a live gathering — a workshop, meetup, or webinar — that you organize around your value proposition to measure whether people show up, engage, and (if you charge) pay. You ask prospects to give up an evening, and often a ticket price, not just a click. The output is behavioral evidence of audience interest, plus qualitative feedback from conversations with attendees.

COMMON USE CASE

You have interviewed a dozen potential customers and believe there is demand for a particular topic. Before investing months building a product, you organize a small paid workshop or meetup around that topic to see if strangers will actually show up, pay, and engage with the content you plan to deliver.

HELPS ANSWER

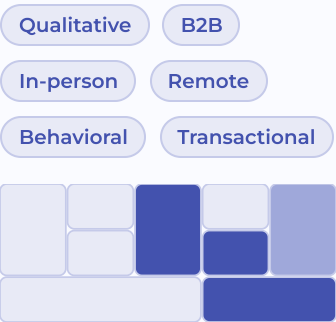
- Who is interested in this problem?
- How do I find and reach the right audience?
- Are people willing to pay for a solution?
- What specifically resonates with attendees and why?

Description

An event smoke test asks prospects to do something one-on-one interviews never can: gather as a group, in one place at one time, and prove their interest by actually showing up. You organize a real gathering around your value proposition — a workshop, meetup, webinar, or dinner — and watch whether people register, attend, engage, and, if you charge, pay. It is one of the higher-commitment rungs of the Value Proposition Test family: setting aside an evening and buying a ticket sits high



AT A GLANCE



TIME  ~2–6 weeks

COST  \$0–\$1.2K

OTHER NAMES
Event Test · Meetup Test

on the commitment ladder, which makes attendance a stronger demand signal than agreeing in an interview that a problem matters.

To run one you have to solve the same problems your eventual product would face: how to reach the audience, what to promise them, and how to deliver an experience they find useful. However, doing that for a single event is far cheaper than building a product. This is particularly true where part of the value proposition is social interaction or trust, which is best generated by real humans interacting.

The format can range widely depending on the depth of signal you want:

- Lecture by one or several speakers
- Peer-to-peer formats like an unconference (e.g., LeanCamp) or a LeanCoffee discussion
- Smaller, intimate formats like a mastermind or dinner
- Networking or drinks
- Conference, online or in person
- Webinar or live video stream

Smaller, intimate formats produce richer qualitative feedback; larger lecture-style events produce a stronger attendance count. In-person events are easiest to fill in a larger city with an existing scene, and they build a closer relationship with attendees than an online interaction does. Online formats cost less and reach a wider geography but give you fewer of the side conversations that carry the real insight.

Because an event is a small product in its own right, it can test more than topic interest: it can qualify leads, test a promotion channel, or lead into a presell of the eventual product. It also combines well with other methods — pair it with a [Landing Page Test](#) to gauge interest before you commit to a venue, or follow it with a [Concierge Test](#) for the most engaged attendees.

How to

Prep

✦ AI Prompt

1. **Define the demand hypothesis.** Write down the specific belief you are testing: who will show up, why this topic matters to them, what attendance number would convince you (and what number would falsify the hypothesis). If you cannot state a falsifying outcome, you are not running a test — you are running an event.

2. **Pick the format.** Choose the event format that matches the audience and the depth of signal you need: workshop, meetup, webinar, unconference, dinner, mastermind. Smaller intimate formats produce richer qualitative signal; larger lecture-style events produce stronger quantitative attendance signal.
3. **Set price and pitch.** Decide what to charge. Free events test topic interest cheaply but produce noisy signal. Paid events (even \$10–\$25) filter for stronger intent. Draft the pitch using language from your prior prospect interviews – not your internal jargon.
4. **Find a venue.** Match the venue to the format and audience. Coworking spaces, libraries, and partner offices are often free. Online formats cost nothing and broaden geography but reduce signal quality.
5. **Create the event page.** Build a registration page on an event registration platform. The description is the smoke test artifact – readers commit (or don't) based on this copy. Track registration rate as the first signal.
6. **Choose the channels.** Use what you learned in earlier research to pick promotion channels (Slack groups, newsletters, partner orgs, paid ads, personal network). Note which channel each registrant came from so you can attribute interest later.
7. **Set the threshold.** Decide before launching what counts as success: minimum registrations, minimum attendance, minimum paid conversions. Write this down. Post-hoc rationalization is the most common failure mode in smoke tests.

Execution

✦ AI Prompt

1. **Promote the event.** Push the registration page through your chosen channels. Track registrations daily. Slow registration is itself a signal worth interpreting before the event happens.
2. **Send pre-event communications.** A short reminder 24–48 hours before the event reduces no-shows. For paid events, also confirm logistics. Optionally include a 1-question pre-survey to surface the question attendees most want answered.
3. **Run the event.** Deliver the agenda. Build in unstructured time – breaks, networking, Q&A – so you can have one-on-one conversations with attendees. Those conversations are the primary qualitative output.
4. **Capture conversations.** Use an AI transcription tool to record the formal portion and any consenting break conversations. This frees you to talk and listen instead of taking notes.
5. **Track engagement live.** Note who arrived early, who stayed to the end, who asked questions, who approached you, who voluntarily gave contact info. For

TOOLS

[Eventbrite](#)

[Luma](#)

[Zoom](#)

[Otter.ai](#)

virtual events, log chat activity, poll responses, and drop-off points.

6. **Greet and observe.** Welcome attendees individually. Watch for who is engaged versus who is just checking it out. The most useful conversations tend to happen in these unscripted moments.
7. **Follow up the next day.** Send a short post-event survey (3–5 questions) and any promised artifacts (slides, recording, links). Share photos or write-ups on social channels — both as community-building and as a second-order signal of who shares it.

Analysis

✦ AI Prompt

1. **Compare attendance against your threshold.** Use the pre-set thresholds, not new ones invented after the fact. Practitioner rules-of-thumb for context: free events tend to see roughly half of registrations show up; paid events convert at noticeably higher rates; virtual events sit somewhere lower than in-person paid events. Treat these as rough anchors, not precise stats — your own channel mix and audience will dominate. Even 10 people who paid to attend a workshop on your specific topic is a meaningful demand signal.
2. **Measure engagement quality, not just headcount.** Count how many stayed to the end, asked questions, approached you during breaks, or gave contact info voluntarily. An engaged audience of 8 is a stronger signal than a disengaged audience of 40.
3. **Read channel attribution.** Which promotion channel produced the most registrations? Which produced the most attendees who actually showed up? Channel attendance rates often vary more than overall attendance rates and tell you where to invest if you scale up.
4. **Synthesize the conversations.** Use AI to cluster transcripts into recurring themes — what problems came up unprompted, what language attendees used, what questions repeated across multiple conversations. The repeated unprompted question is often the real demand signal.
5. **For virtual or hybrid events, mine engagement signals.** AI-powered event platforms monitor audience attention, run real-time polls, and analyze chat sentiment. Look for where attention drops, which polls produced strong consensus, and which topics generated chat activity.
6. **Follow up within 48 hours.** Send a short survey while the experience is fresh. Ask what was most valuable, what was missing, and whether they'd attend again or pay for a related product. The “would you pay for a related product” answer is far more reliable when asked within 48 hours than after a week.
7. **Make the go/no-go call.** Compare results against the falsifying threshold you set in Prep. If the threshold was missed, document it honestly and iterate on the

proposition or pivot the audience — don't lower the bar retroactively.

Biases & Tips

Sales-skill confound

A hard sales push (urgency, scarcity, “CRO best practices,” high-pressure in-person closing) distorts the signal. You won't know whether you measured real demand or your own salesmanship. Sell enough that the proposition is clear and easy to buy; no harder.

Confirmation bias

Founders interpret weak attendance as “the topic was right but the timing was off” and strong attendance as “demand is real.” Set the falsifying threshold in writing before launching the event so the signal is read against a fixed bar.

Free- registration false positive

Free events attract registrants with low intent, and a large share will not show up. If your falsifying threshold is set on raw registrations rather than actual attendance, you will declare demand real when it isn't. Anchor your go/no-go threshold on who walked through the door, not who clicked “Register.”

Channel sampling bias

Promoting only through your personal network produces friends-of-founder attendance, which looks like demand but isn't. Mix at least one channel where attendees do not know you personally.

Automation displacement

Logistics automation — chatbots, auto-replies, AI chat moderation — can crowd out the spontaneous break conversations that are the primary qualitative output. When every attendee interaction is handled by a tool, the unstructured moments that produce real insight disappear before you realize they were there.

Logistics as signal pollution

Hard-to-find venues, broken Zoom links, missing name tags, and unclear directions cause drop-off that looks like low demand. Prepare logistics the day before (signs, name tags, directions, camera, AV check) so weak signal isn't actually a logistics failure.

Ethics

When you charge for the event, you owe attendees the event you promised: run it as described, or refund anyone you cannot accommodate.

Next Steps

→ Run a  **Closed-Ended Survey** with attendees within 48 hours to capture fresh feedback before memory decays.

- If demand exceeded the pre-set threshold, use a [Pop-Up Store Test](#) or a recurring event series to deepen the relationship and re-test demand.
- Capture attendee profiles and language with an [Open-Ended Survey](#) to refine your customer persona and the value proposition wording.
- Use [Discovery Interviews](#) to follow up one-on-one with the most engaged attendees and understand their unmet needs in depth.
- Use a [Pre-Sales Test](#) to convert event interest into purchase commitments before building the product.
- Use a [Landing Page Test](#) to scale the demand signal beyond the geographic limit of a single event.

Case Studies

Skillshare: Toasting Success event

Michael Karnjanaprakorn describes how Skillshare's earliest in-person classes — including one on toasting — validated demand before any platform was built.

[Read more](#)

Tough Mudder: First obstacle-course event drew 4,500+

Will Dean's first Tough Mudder event in May 2010 at Bear Creek Mountain Resort drew more than 4,500 participants, promoted entirely via Facebook ads and word-of-mouth; by 2014 the company reported roughly \$75M in annual revenue.

[Read more](#)

ClassPass: Classstivity-era event tests

Payal Kadakia and Sanjiv Sanghavi launched Classstivity in 2010 and shipped Passport in 2012 — letting users sample fitness classes in person — before pivoting and rebranding to ClassPass in January 2014.

[Read more](#)

Further reading

-  Eventbrite UK Blog: How to Improve Turnout for Free Events
-  Harvard Business Review: Why Your Meetings Stink — and What to Do About It
-  Rob Fitzpatrick — The Mom Test

4.4.5 Fake Door Test

In Brief

A fake door test is a button, link, or advertisement for a product or feature that does not yet exist, placed in front of an audience to measure interest by counting how many people click it. A click is a behavioral signal — an action rather than a stated opinion. It is commonly used to test new feature ideas inside an existing product or to validate a concept with external ads before committing to build it.

COMMON USE CASE

You have an idea for something new — a feature, a product, a service — but do not want to invest time and money building it if nobody cares. You present it as if it already exists (a button, a link, an ad) and measure how many people try to get it, which tells you whether real demand exists before you commit resources.

HELPS ANSWER

- Are customers interested enough to click on this offer?
- Which audience segment is most attracted?
- Which words, icons, or images generate the most engagement?

Description

A fake door test presents a button, link, or ad for a product or feature that does not yet exist, then counts how many people click it. Because clicking is an action rather than an opinion, it sits a rung above a passing glance on the [Value Proposition Test](#) ladder — the person who clicks has chosen to act, even if all they have committed is the click itself. You place the door, count how many people try to use it, and read the



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~4 days–3 weeks

COST  \$0–\$600

OTHER NAMES

404 Test · Feature Stub · Painted Door Test · 404 Page Test · Coming Soon Test

result as a measure of real behavior rather than stated intent. The [click-through rate](#) — the share of viewers who click — tells you whether demand is real.

There are two types:

- **Internal fake door** — A button or menu item inside your existing product. Tests whether current users want a new feature. Example: adding a “Switch to map view” button on a page that only has a list view.
- **External fake door** — A paid ad or social post for a product that doesn’t exist yet. Tests whether a target audience is interested in a concept. Example: running a Facebook ad for an online community to see if anyone clicks.

Don’t use fake doors to test essential functionality (like registration flows) or your core value proposition. Use them to test *additions* and *new directions*.

Ethics — You are showing people something that does not exist. Keep it honest. The reveal page should clearly say the feature is in development and offer to notify them when it is ready. A few fake doors help you avoid building things nobody wants; many dead-ends in a short span erode trust.

How to

Prep

✦ AI Prompt

1. Write your hypothesis.

State what you expect to happen. Be specific: “At least 5% of users who see the button will click it” is testable. “People will be interested” is not.

2. Decide: internal or external.

- **Internal** (in your product): Best for testing new features with existing users. Requires adding a button or link and a tracking event. Fastest to set up.
- **External** (paid ads): Best for testing a new concept with a new audience. Requires an ad account and a landing page. More expensive but reaches people who don’t know you yet.
- **Hybrid** (ad → landing page with fake CTA): Tests both audience interest and messaging. The ad proves they’ll click; the CTA proves they’ll act.

3. Set your success threshold before you launch.

Decide what click-through rate counts as a win. Benchmarks:

- **Internal buttons/features:** 2–5% CTR is typical for in-product UI elements. Above 5% is strong interest.
- **External ads:** Industry average CTR for Facebook ads is ~1%. Google Search ads average ~3–5%. Beat the channel average to call it a win.
- **Landing page CTAs:** 5–10% email signup rate from a cold audience is solid.

If you don't set a threshold up front, you'll rationalize any result as success.

4. Set a target sample size.

You need enough views to trust the click-through rate:

- For a rough directional signal: 200+ views minimum.
- For a confident comparison between variants: 500+ views per variant.
- Use a sample size calculator (see References) if you need precision.
- **If your traffic is very low (under 200 views):** Run the test longer, use external ads to supplement traffic, or combine the fake door with qualitative follow-up (interview the people who clicked). A 10% CTR on 50 views is a signal worth investigating, even if it's not statistically conclusive.

5. Design the fake door.

Rules:

- Make it look exactly like a real feature would look. Same font, same placement, same style. If you make it flashier or bigger than normal UI elements, you're measuring novelty, not interest.
- Write the label/copy in the same language you'd use for a real feature. No hype.
- If testing multiple variants (different copy, different placement), run them sequentially or split traffic evenly.

6. Design the reveal page.

What users see after they click. Options:

- **"Coming soon" page** — Explains the feature is in development. Offers email signup for notification. Best for features you might actually build.
- **"Help us prioritize" page** — Short survey asking why they clicked and what they'd expect. Best for early exploration.
- **Simple thank-you** — "Thanks for your interest. We're exploring this idea." Minimal but honest.

Always include: a clear explanation that the feature doesn't exist yet, and a way for the user to continue what they were doing without frustration.

7. Set up tracking.

- **For in-product fake doors:** Add a click event to your product analytics. Track both views (impressions) and clicks.
- **For external ads:** The ad platform tracks CTR automatically. Also track visits to the reveal page and any email signups.
- **For both:** Record the date range, total views, total clicks, and any segmentation data (user type, device, source).
- **Before launching:** Test the tracking yourself — click the fake door and confirm the event appears in your analytics. This is the most common reason fake door tests produce no usable data.

Execution

1. Launch the fake door.

Deploy it exactly as designed. Don't tweak the copy or placement after launch — changes mid-test invalidate your data. If you realize something is wrong (broken link, tracking not firing), fix the bug but note the date and exclude pre-fix data.

✦ AI Prompt

TOOLS

v0

Unbounce

Chameleon

Userpilot

2. Wait.

Set a deadline based on your traffic estimate. For internal fake doors, run for at least one full week to account for day-of-week variation in usage patterns. Don't check results hourly — it leads to premature conclusions. Check once at the halfway point to verify tracking is working, then again at the end.

3. Collect qualitative data from the reveal page.

If your reveal page includes an email signup or survey, monitor responses. If several people leave the same comment or question, that's signal about what they expected.

Analysis

✦ AI Prompt

1. Calculate the click-through rate.

$\text{CTR} = \text{clicks} \div \text{views} \times 100$. This is your primary metric.

2. Compare against your pre-set threshold.

- **Above threshold:** The concept has enough interest to justify further investment. Move to the next validation step (solution interview, pre-sales, or build).

- **Below threshold but close:** Consider testing a different copy variant, placement, or audience before giving up. A 3% CTR when you needed 5% might mean your messaging is off, not that demand is zero.
- **Well below threshold:** The concept doesn't resonate with this audience. Either pivot the concept or try a different audience.

3. Segment if possible.

If you have enough data, break CTR down by user type, device, or traffic source. A feature that gets 8% CTR from power users but 1% from new users tells you something different than a flat 3%.

4. Check for confounds.

- Did something else change in the product during the test period?
- Was there a spike in traffic from an unusual source (press mention, viral post) that diluted the signal?
- Was the fake door placed in a high-traffic area that gets incidental clicks (near a navigation element, close to a scroll path)?

5. If your sample was small (under 200 views): Treat the CTR as directional. Follow up with 5–10 interviews of people who clicked (or didn't) to understand why. A small-sample fake door plus qualitative interviews is often more useful than a large-sample fake door alone.

Biases & Tips

Copy bias

The words on the button matter enormously. “Try our new dashboard” and “See your analytics” will get very different click rates for the same feature. Test at least two copy variants if you can.

Location bias

A button at the top of the page gets more clicks than one at the bottom, regardless of interest. If possible, test the same door in multiple positions.

Novelty bias

Users click new things because they're new, not because they want them. A spike in the first day that drops off quickly is novelty, not demand. Look at sustained CTR after the first 48 hours.

Click ≠ commit

CTR overstates real demand. A click is the lowest possible signal of interest. Always follow a successful fake door with a higher-commitment test (signup, pre-order, deposit) before building.

Polished-
reveal
deception

AI page builders produce reveal pages that look production-ready by default, making a “coming soon” stub feel indistinguishable from a real product page. Visitors who cannot tell the difference feel deceived when they discover the feature does not exist, eroding trust in the product. Consciously roughen the reveal or label it plainly as exploratory.

Concurrent-
test saturation

AI makes it trivially easy to spin up many fake doors at once. Running multiple concurrent tests on a small audience simultaneously dilutes the click signal across concepts and compounds trust erosion: visitors who hit several dead-end stubs in one session lose confidence in the product faster than a single test would warrant.

Next Steps

- If clicks exceed your threshold, build a **Single-Feature MVP** of the feature so people can actually use it.
- Run a **Pre-Sales Test** or **High-Bar Test** to check whether click interest turns into willingness to pay, not just interest.
- Use a **Solution Interview** with people who clicked to learn what they expected the feature to do before you build it.
- If results are mixed, run an **A/B Test** of alternative copy, positioning, or placement before deciding.

Case Studies

Polyvore: “Buy this outfit” fake door

Polyvore added a “buy this outfit” button with no store behind it. Over three months, about 2 in every 100 people who saw an outfit clicked to buy, and orders were handled by hand before any payment system was built.

[Read more](#)

Notino: Language-switcher fake door

Notino, a multi-country European beauty retailer, placed a visible language switcher on its Swiss site to measure interest in localization. About 3.5% of visitors clicked it, giving real behavioral data before committing to translation work.

[Read more](#)

Zynga: Pitch-ad demand testing inside live games

Zynga tests prospective games by running short pitches inside its existing games and through ads, measuring how many current players click before any production work begins.

[Read more](#)

Tippiq: Dedicated fake-door lab site

Dutch aggregator Tippiq ran a separate site, tippiqlabs.nl, where new ideas were tested with paid ads pointing at description pages. Interested visitors saw a “still in development” notice and could leave their email. (Tippiq’s dedicated fake-door lab site has since been discontinued.)

Further reading

-  Alberto Savoia — *The Right It* (HarperOne, 2019)
-  Alberto Savoia — Prototyping resources
-  Eric Ries — *The Lean Startup* (Crown Business, 2011)
-  Marty Cagan — *Inspired*, 2nd ed. (Wiley, 2017)
-  Jess Lee — *Fake Doors: How to Test Product Ideas Quickly* (Hustlecon 2013)
-  Amplitude — *What Is Fake Door Testing: Methods and Best Practices*
-  Chameleon — *Fake Door Testing: How it Works, Benefits & Risks*
-  Kromatic — *Experiment Calculator*

4.4.6 Flyer Test

In Brief

A flyer smoke test is a printed one-pager that describes a product concept and asks the reader to act on it — scan a code, call, or visit a URL — before the product exists. You distribute flyers where your audience already gathers (conferences, lobbies, direct mail) and measure how many people follow the call-to-action. The output is a response rate that tells you whether the messaging resonates, and it works well when your audience is easier to reach in person than online.

COMMON USE CASE

You need to test whether your product concept resonates with a specific audience you can reach physically — at a conference, in a shared office building, or through direct mail — but running digital ads isn’t practical because your buyers aren’t easily targeted online. You design a simple one-pager describing the benefits, distribute it, and measure how many people follow the call-to-action.

HELPS ANSWER

- Will the prospect agree to an appointment after seeing the flyer?
- Do people respond positively to this description of the product?
- Will the target audience pick up the flyer in a normal setting like a conference or lobby?

Description

A flyer smoke test is a printed one-pager that describes the product or service, makes the key promise, and asks the prospect to take one specific next step — scanning a QR code, calling a number, visiting a URL, or requesting an appointment. The count of



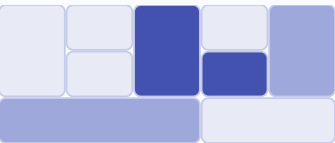
AT A GLANCE

Quantitative

In-person

Remote

Behavioral



TIME  ~1–2.5 weeks

COST  \$0–\$120

OTHER NAMES
Brochure · Data Sheet

people who follow that step is your demand signal. Within the

Value Proposition Test family, it is the action-commitment variant — asking the prospect to follow a call-to-action rather than spend money or time. It is the offline counterpart to a **Landing Page Test**: the same demand signal, distributed on paper instead of online.

Because the product does not yet exist in full, you can print several versions and test different messaging and value propositions side by side. This makes the flyer most useful in the early stages of an idea, before there is anything to demo.

The method fits B2B and enterprise sales especially well, where buyers gather at conferences or in offices but are hard to reach with online ads. In those markets a handful of clients each represent a large potential purchase, so even a small number of responses carries weight — and a non-response from a known target is itself informative. Treat each reply as the start of a sales conversation, not a finished count.

How to

Prep

★ AI Prompt

1. Decide when a flyer beats a landing page.

Use a flyer test when:

- Your audience gathers physically (conferences, trade shows, industry meetups, shared office buildings).
- Your buyers aren't easily targeted with online ads (niche B2B, local services, non-tech industries).
- You want to test messaging in a specific physical context (point-of-sale, waiting room, lobby).

If your audience is easily reachable online, a **Landing Page Test** is faster and more measurable.

2. Design the flyer for testing, not selling.

Keep it simple — you're testing whether the value proposition resonates, not winning a design award:

- **Headline:** One clear statement of what you offer and who it's for. Test this — it's the most important element. AI design tools can generate multiple headline variants and layouts in minutes, making it practical to test 2–3 framings

simultaneously across different distribution points. Draw headline language from prior discovery interviews rather than internal vocabulary.

- **2–3 benefit bullets:** Specific outcomes, not features.
- **Call-to-action with tracking:** A unique URL, QR code, or phone number so you can count responses. Use a short URL or QR code that's different for each location or variant so you can compare.
- Stand back and read it from 5 meters away. If the headline and CTA aren't legible, simplify.

For B2B scenarios, AI can help personalize flyers for specific prospect companies or roles — different messaging for a CFO vs. a VP of Operations — without requiring a designer for each variant.

3. Set your success threshold.

Flyer response rates are noisy and channel-dependent, so treat any single number as a rule of thumb rather than a target. Published direct-mail and flyer-handout rates land in low single digits and vary widely with list quality, audience fit, and physical context. As a rough guide:

- **Direct mail:** cold-prospect mail tends to run in the low single digits; meaningfully above that, for a cold audience, is a strong signal.
- **Conference or event placement:** picking up a flyer and following the call-to-action usually lands in the same low-single-digit band — strong if you clear it consistently across locations.
- **Handed out in person:** personal contact tends to lift the rate above passive placement, but the lift varies a lot by who is doing the handing out.

The point is not the exact percentage. It is committing to a threshold *before* you distribute, so you cannot rationalize a weak result after the fact.

4. Plan distribution.

- **Conference placement:** Registration tables, seat drops, bulletin boards. Ask for permission.
- **Direct mail:** Use USPS Every Door Direct Mail (EDDM) for low-cost geographic targeting.
- **In-person handout:** At events, trade shows, or door-to-door. More labor-intensive but higher response rates.
- Print enough for your target sample: at least 100 for a directional signal, 500+ for confident comparison between variants.

Execution

✦ AI Prompt

1. Distribute the flyers.

Place or hand them out exactly as planned. If testing variants (different headlines or CTAs), track which variant goes where.

2. Wait 1–2 weeks.

Give people time to respond. Check your tracking URL or QR code analytics at the end, not daily.

3. Record all responses.

Track total flyers distributed, total responses (URL visits, calls, emails), and any qualitative data from people who reached out.

TOOLS

[QR Code Generator](#)

[Taradel](#)

[99designs](#)

Any graphic design software (Canva, Adobe Express, Figma, etc.) — generate layouts, headline variants, and print-ready flyers

Analysis

✦ AI Prompt

1. Calculate the response rate.

Response rate = responses ÷ flyers distributed × 100. Compare against your pre-set threshold.

2. Calculate customer acquisition cost.

CAC = total cost (printing + distribution) ÷ number of responses. Compare this to what you'd spend on digital ads for the same number of leads.

3. Compare variants if you tested multiple.

If you tested different headlines or CTAs, which one pulled more responses? The winning copy tells you how your audience thinks about the problem.

4. For small samples: If you distributed fewer than 100 flyers, treat the response rate as directional. Follow up with anyone who responded to understand why the flyer caught their attention.

⚠ Biases & Tips

Friends-and-family bias

People in your network respond out of goodwill, not genuine demand, and inflate the rate. Tag and exclude any response you can trace to someone you know.

Self-selection bias

The people who stop and scan a flyer are not a random slice of your market; they are the curious and the already-interested. A strong response from self-selected passersby does not guarantee the same rate at scale. Treat the rate as a signal that the message resonates with someone, then validate the segment before extrapolating.

Design bias

An eye-catching design can pull responses even when the value proposition is weak, and AI tools make polished-looking flyers easy to produce. Keep the design clean enough that the message, not the aesthetics, is doing the work; if you test variants, hold the visual style constant and vary only the copy.

Location and placement bias

A flyer at a registration desk gets more pickup than one on a back table, and a busy hallway beats a quiet one. When comparing variants, distribute them in equivalent positions and at equivalent times, or the location will outweigh the copy.

Confirmation bias

It is easy to read an ambiguous response rate as success because you want the idea to work. Commit to the success threshold before you distribute, and judge the result against that number, not against your hope for it.

Next Steps

- If the winning message clears your threshold, scale it to a broader digital audience with a [Landing Page Test](#).
- Follow up with respondents through [Discovery Interviews](#) to learn why the offer caught their attention and what problem they are trying to solve.
- Before committing to build, test whether respondents will commit money with a [Mock Sale](#) or [Pre-Sales](#).

Case Studies

Nextdoor: Reaching users through the mailbox

How Nextdoor used direct mail to seed early neighborhood signups before relying on digital channels.

[Read more](#)

Dojo: Handwritten letters drove downloads

The Dojo team skipped PR emails and sent heartfelt letters instead, generating thousands of downloads and investor interest.

[Read more](#)

Relatas: Flyers on every conference seat

Relatas seeded an empty conference auditorium during lunch with a flyer on every seat to reach attendees as they returned.

[Read more](#)

Further reading

-  7 Tips for Creating Effective Flyers for Your New Product
-  USPS: Every Door Direct Mail (EDDM) — Guide to low-cost direct mail for small businesses
-  Direct mail response rate benchmarks — Modern Postcard (ANA/DMA data)

4.4.7 High Bar Test

In Brief

A high bar smoke test adds abnormal friction to a signup or request flow — long forms, qualifying questions, application steps — and watches whether the same share of motivated prospects still completes it. If they push through the friction, that effort is evidence of real demand even though you never asked for money. The friction also pre-qualifies leads, so the people who do convert are the ones worth talking to.

COMMON USE CASE

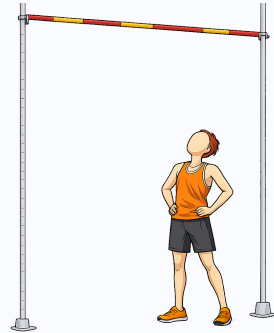
You have a steady flow of frictionless signups (emails, waitlist entries) but you don't yet know which signups will become paying customers. You aren't ready to charge — pricing is undecided, the product isn't built, or you're in a B2B context where pricing is per-deal. You want a behavioral proxy for purchase intent before you invest in selling.

HELPS ANSWER

- How keen is the customer?
- How big of a problem is it for the customer?
- Who is my early adopter?
- Which incoming leads should I prioritize?

Description

A high bar smoke test adds deliberate friction — long forms, applications, screening calls, or qualifying questions — to a signup or request flow and measures whether motivated prospects still complete it. The friction stands in for money: if a prospect will work to get to your offer, that effort is evidence of purchase intent even though you have not charged anything. It sits high on the [Value Proposition Test](#)



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~3 days–4 weeks

COST  \$0–\$60

OTHER NAMES

High Bar · High Friction Test ·
Friction Filter

commitment ladder, because clearing real friction takes more than a click — the commitment here is effort rather than payment.

The technique is most useful when:

- You aren't ready to charge.
- You operate in a B2B environment where pricing is customized per deal.
- You already have a frictionless signup (emails, waitlist) and want to separate idle interest from real intent.

You start by measuring the conversion rate on your existing frictionless flow — the share of visitors who complete it. Then you introduce friction: extra form fields, a qualifying questionnaire, an application process, a required call. If conversion among your target segment holds steady, the high bar is cleared and the test passes. A drop in raw conversion isn't automatically a failure: if the friction filters out the wrong prospects, the leads you keep are higher quality even though fewer people get through.

The signal is the same one B2B sales teams use when they rank leads, often by recency (how recently a prospect engaged), frequency (how often they engage — opened emails, attended calls, replied to threads), and any money already spent on the problem. A prospect who scores high on those is more likely to buy. A high bar smoke test surfaces that signal early, before you have a sales team or a CRM tuned to score it automatically.

A real purchase is the strongest signal of demand. A high bar smoke test stands in for one when charging isn't yet practical, trading the certainty of money for a behavioral proxy you can run sooner.

How to

Prep

✦ AI Prompt

1. **Define what counts as “high bar” friction.** Decide which friction you will add and why it is meaningful for your prospect — extra form fields, a qualifying questionnaire, a required intro call, an application with motivation prompts. The friction should be plausibly relevant to the offer (a B2B procurement process should look like procurement, not a captcha gauntlet). Write down what you expect a motivated prospect to be willing to do.
2. **Recruit (or identify) the target segment.** State who your ideal prospect is in one sentence. Near-miss prospects don't count as conversions even if they fill out

the form. If you don't yet have a clean segment definition, run a screening question at the top of the form so you can filter the data afterward.

3. **Set the baseline and the success threshold.** Measure your current conversion rate on the frictionless flow first, and define the conversion event precisely (a completed application, a booked call, a returned questionnaire). Decide before you launch what conversion rate among the target segment counts as "passed" — for example, "we hold 60% of baseline target-segment conversion after adding the application step." Write it down so you can't move the goalpost after seeing results.

Execution

✦ AI Prompt

TOOLS

Typeform

Calendly

Unbounce

1. **Launch the friction.** Add the friction to the flow, or stand up a parallel flow with the friction alongside the frictionless one. If you have enough traffic, run an A/B split — show the frictionless and high-bar versions to randomly assigned visitors at the same time so the comparison is fair. If not, run them one after the other and be honest about the comparison.
2. **Let leads accumulate without coaching.** Open the funnel and observe whether motivated prospects self-select through the friction. Don't help them past it — the friction only measures intent if prospects clear it on their own.
3. **Stay neutral on calls and applications.** For B2B variants where the friction is a call or application, read questions verbatim, don't fill in answers for the prospect, and don't telegraph which answers help them get in. You're measuring their commitment, not delivering a sales pitch.
4. **Capture every conversion and near-miss with metadata.** Record enough about each lead to identify whether it matches your target segment. Tag drop-offs by step so you can see where the friction is actually filtering.
5. **Stop on schedule.** When the time window expires (or you hit your target sample), stop the test. Don't keep extending in search of the result you wanted.

Analysis

✦ AI Prompt

1. **Compare target-segment conversion, not raw totals.** Compute the conversion rate for your target segment on the frictionless baseline and on the high-bar flow. Only target-segment numbers count toward the result.
2. **Check against your pre-registered threshold.** If target-segment conversion held above the threshold you set in Prep, the high bar is cleared. If it dropped, check whether the drop was concentrated in the target segment or in near-miss prospects — a drop concentrated in near-misses is a feature, not a failure.

3. **Inspect who cleared the bar.** Look at the leads who made it through. Are they the prospects you actually want? If yes, you have both a demand signal and a pre-qualified pipeline.
4. **Inspect drop-off by step.** If everyone falls off at the same field or question, that step is doing all the filtering — decide whether it's filtering for the right thing. If drop-off is even across steps, the friction is uniformly raising the bar.
5. **Don't act on early noise.** A handful of high-conversion days early in the window is not a result. Wait for your sample size before deciding.

⚠ Biases & Tips

Backfire effect

When disconfirming evidence shows up, the temptation is to strengthen the original belief (“the friction was wrong, not the demand”). Pre-register your threshold in Prep so you can't reinterpret a fail as a pass.

Selection bias

The leads who clear a high bar are unusual by definition. Confirm they match your target segment, not a self-selected enthusiast subset who would clear any bar regardless of the offer.

Confirmation bias

Once you expect the test to pass, you read ambiguous near-misses and off-target conversions as wins. Tag every lead target-segment-yes / no from the screener before you look at the totals, not after.

Vanity-metric framing

A high raw completion rate feels like success but may just mean the friction was too easy to filter anyone. Define the conversion event as the costly action that signals intent (a returned application, a booked call), not a form view or a click.

Survivorship bias

Studying only the leads who cleared the bar hides why everyone else dropped. Capture and review near-misses and step-by-step drop-offs, not just the conversions, so you learn whether the bar filtered the wrong people.

Next Steps

- For a clear pass, move toward [Pre-Sales Test](#) or a paid pilot to convert the qualified pipeline into revenue.
- For a clear fail, return to generative work — the demand signal isn't there yet. Consider [Discovery Interviews](#) or a [Closed-Ended Survey](#) to re-scope the problem.

- For ambiguous results with a small sample, expand the time window or run a [Landing Page Test](#) to widen the top of funnel before testing high-bar friction again.
- For passes that produce a strong pipeline, formalize the friction into a lead-scoring system in your CRM so the qualification continues after the test ends.

Case Studies

Superhuman: Mandatory 30-minute onboarding call

Superhuman required every new user to attend a 30-minute human-led onboarding before product access; attendance held at 100% while mandatory and dropped to roughly 15% when made optional. The required call was a deliberate high bar, and willingness to clear it filtered for committed users.

[Read more](#)

Further reading

-  Eric Ries — The Lean Startup (Crown Business, 2011, ISBN 978-0307887894)
-  Steve Blank: Customer Development for Web Startups
-  Gaurav Vohra: Superhuman's Onboarding Playbook (First Round Review)
-  HubSpot: RFM (Recency, Frequency, Monetary Value), Explained in 600 Words or Less

4.4.8 Landing Page Test

In Brief

A landing page smoke test is a one-page site for a product that doesn't exist yet, built to measure how many visitors will hand over an email address to hear more. The visitor gives you only an email, not money or time. The result is evidence of demand based on what people actually do, not what they say in a survey or an interview.

COMMON USE CASE

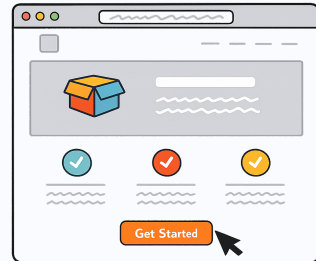
You have a clear idea of what you want to offer but no proof that anyone will act on it, and you want that proof before you commit to building. You are early enough that a working product does not exist, but specific enough to write a headline and name the audience you would target.

HELPS ANSWER

- Are people interested enough to take action on this value proposition?
- Is the product idea attractive to a particular audience or niche?
- Is the product positioned effectively against alternative solutions?

Description

A landing page smoke test puts up a single page describing a product you haven't built, drives traffic to it, and counts how many visitors take a concrete action to hear more — leaving an email, clicking “buy now,” or joining a waitlist. Email capture sits at the lowest-commitment rung of the [Value Proposition Test](#) family: the visitor gives you only an email address, which makes it the cheapest signal to collect and the easiest to overstate.



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~4 days–2.5 weeks

COST  \$30–\$600

OTHER NAMES

Simple Landing Page · Landing Page MVP · MVP Landing Page

The conversion rate — the percentage of visitors who take that action — is your primary output. A high rate means the audience found your value proposition compelling enough to act on it.

Landing page tests are best used after you have a clear product concept and target audience, but before you build anything. They answer “is there demand for this?” not “is the product good?” If your page converts well, people want what you described. Whether you can actually deliver it is a separate question.

Don’t over-design the page. You’re testing the value proposition, not your design skills. A clean page with a clear headline, 2–3 benefit statements, and a single CTA is all you need. Skip urgency tricks like countdown timers and limited-time offers — they raise the conversion rate without telling you whether the idea itself works.

How to

Prep

✦ AI Prompt

1. Define what you are testing.

Be specific about the hypothesis. “People want our product” is too vague. “At least 5% of paid search visitors looking for [keyword] will leave their email to learn more about [specific value proposition]” is testable.

2. Choose your call-to-action.

The CTA determines what level of commitment you’re testing:

- **Email signup / waitlist:** Lowest commitment. Tests whether people are interested enough to give you an email address. Easiest to set up. Conversion benchmarks: 5–15% from targeted paid traffic.
- **Simulated purchase (“buy now” → reveal page):** Medium commitment. The visitor clicks a purchase button, then sees a message explaining the product isn’t available yet. Tests purchase intent. This is essentially a [Fake Door Test](#) on a landing page.
- **Actual pre-order:** Highest commitment. Tests willingness to pay real money. See [Pre-Sales Test](#) for this approach.

For B2B or physical products, adapt the CTA to your context: “Request a sample,” “Book a call,” or “Get a quote” are all valid commitment signals. The key is that the visitor takes a concrete action, not just reads and leaves.

Pick one CTA per test. Multiple CTAs on the same page split attention and make results harder to interpret.

3. Set your success threshold before you launch.

Decide what conversion rate counts as a win. Benchmarks depend on your traffic source:

- **Paid search (Google Ads):** Visitors have high intent — they searched for something related. 5–15% email signup is typical; above 10% is strong.
- **Paid social (Facebook/Instagram):** Visitors are interrupted from browsing — lower intent. 2–5% email signup is typical; above 5% is strong.
- **Organic/direct traffic:** Varies too much to benchmark. Use paid traffic for cleaner data.
- **Simulated purchase click:** 1–3% from paid traffic is a good signal.

If you don't set a threshold, you'll rationalize any result.

4. Build the landing page.

Keep it simple. Essential elements:

- **Headline:** One clear sentence saying what you do and who it's for.
- **Sub-headline or value statement:** What benefit the visitor gets.
- **2–3 benefit bullets:** Specific, not generic. "Save 4 hours per week on scheduling" beats "Save time."
- **Single CTA button:** "Get early access," "Join the waitlist," "Pre-order now."
- **No navigation links:** The only action a visitor can take is the CTA or leaving. Navigation links leak traffic.

Use your target audience's own language — pull phrases from interviews, forum posts, or competitor reviews. AI page builders can generate a working page from a text description in minutes.

5. Set up your traffic source.

- **Paid search (Google Ads):** Target keywords your audience is already searching for. Set a daily budget (\$10–20/day) and run for 5–7 days.
- **Paid social (Facebook/Instagram):** Target demographics or interests matching your audience. More volume, lower intent.
- **Existing audience (email list, social):** Fastest and cheapest, but these are warm leads. Results will overstate demand from cold traffic.

Use cold traffic (paid ads to strangers) for the cleanest signal. If you only test with your existing audience, you're measuring how much your current followers like you, not whether a market exists.

6. Set a target sample size.

You need enough traffic to trust the conversion rate:

- **Minimum:** 200 unique visitors for a directional signal.
- **For comparing two variants:** enough visitors per variant for the difference in conversion to clear statistical significance — for a typical email-signup conversion rate, that lands in the low hundreds per variant. Use a sample size calculator (see References) to size it for your baseline rate.
- **If traffic is very low:** Run longer, increase your ad budget, or combine with qualitative follow-up (interview the people who did sign up).

7. Set up tracking.

Install basic analytics (a product-analytics tool or the ad platform's pixel) to track:

- Total unique visitors
- CTA clicks or form submissions
- Traffic source breakdown
- Device type (mobile vs. desktop — conversion rates differ significantly)

Test the tracking before driving real traffic. Click the CTA yourself and confirm the event fires.

Execution

1. Launch the page and start driving traffic.

Deploy the page, activate your ads, and don't touch anything. Changes mid-test (new headline, different CTA, redesigned layout) invalidate your data. If you find a bug (broken form, tracking not firing), fix it, note the date, and exclude pre-fix data.

2. Wait.

Set a deadline based on your traffic estimate. Don't check results hourly — early data is noisy. Check once at the midpoint to verify tracking is working, then again at the end.

3. Monitor ad performance separately.

★ AI Prompt

TOOLS

[Unbounce](#)

[Lovable](#)

[Carrd](#)

[PostHog](#)

If your ads aren't getting clicks, the problem is your ad copy or targeting, not your landing page. Check your ad click-through rate (the percentage of people who saw your ad and clicked on it) separately:

- Google Search ads: 3–5% CTR is average.
- Facebook/Instagram ads: 1–2% CTR is average. If your ad CTR is well below average, fix the ad before drawing conclusions about the landing page.

Analysis

✦ AI Prompt

1. Calculate the conversion rate.

Conversion rate = CTA completions ÷ unique visitors × 100. This is your primary metric.

2. Compare against your pre-set threshold.

- **Above threshold:** Your value proposition resonates with this audience. Move to the next validation step.
- **Below threshold but close:** Test a different headline, CTA, or traffic source before giving up. A 3% conversion when you needed 5% may be a messaging problem, not a demand problem — run a [Comprehension Test](#) to check.
- **Well below threshold:** This audience isn't interested in this value proposition at this price. Rethink your positioning, your audience, or the product concept.

3. Check your traffic quality.

If conversion is low, verify that your traffic was actually relevant:

- What keywords triggered your ads? (Irrelevant keywords = irrelevant visitors.)
- What demographics saw your social ads? (Too broad = diluted signal.)
- What was the bounce rate (the percentage of visitors who leave without taking any action)? If above 80%, visitors aren't finding what they expected.

4. Segment by source and device.

Break conversion down by traffic source and device type. Mobile visitors typically convert at lower rates than desktop for email signup forms. If all your traffic is mobile and conversion is low, try optimizing for mobile before concluding demand is weak.

5. For small samples (under 200 visitors): Treat the conversion rate as directional. Follow up by interviewing the people who did sign up — ask why they were interested and what they expected. A handful of enthusiastic signups is a signal worth investigating even if the numbers aren't statistically conclusive.

⚠ Biases & Tips

Conversion inflation

Urgency widgets, countdown timers, and polished social proof raise the conversion rate, making marketing skill hard to tell apart from real demand. A page that converts because of funnel polish, not the idea, produces a false positive that disappears once you ship a real product. Keep the page plain so the headline and value proposition are what people respond to.

Message mismatch

If your ad promises one thing and the landing page says another, conversion stays low regardless of demand. Make sure the ad copy and the landing page headline make the same promise, or you will read an attention problem as a demand problem.

Warm-audience bias

Driving traffic from your own email list or followers measures how much your existing fans like you, not whether a market exists. Use cold paid traffic (ads to strangers) for the headline number, and treat any warm-audience result as a generous upper bound.

Commitment mismatch

A low-commitment action overstates how many people would actually pay. Leaving an email is cheap; an email signup is not a purchase. Read the conversion rate as interest at the commitment level you tested, and confirm willingness to pay with a higher-commitment test before you trust it.

Small-sample over-reading

With a few hundred visitors, a handful of extra signups can swing the conversion rate by several points, so a result near your threshold is noise as easily as signal. Set a minimum sample size before launch and treat anything below it as directional, not a verdict.

Curiosity clicks

Novel or surprising headlines pull clicks and even signups out of curiosity rather than intent to use the product. Watch for high click-through with low downstream follow-through, and confirm interest by interviewing a few signups about what they expected.

Next Steps

- If conversion exceeds your threshold, proceed to build an MVP or run a higher-commitment test like a [Pre-Sales Test](#).
- Run an [A/B Testing](#) experiment on the highest-converting variant against new alternatives.
- Use [Discovery Interviews](#) with email signups to understand their expectations and urgency.

- If conversion was low, run a [Comprehension Test](#) to check whether the problem is messaging clarity, not demand.
- Use a [Pre-Sales Test](#) to move from email signups to actual purchase commitments.
- Use a [Concierge Test](#) to manually deliver the promised value to your first signups and learn what to build.

Case Studies

Dropbox: Landing page captured 75,000 emails

Drew Houston's startup lessons deck describes how a simple landing page paired with a three-minute screencast grew the beta waitlist from 5,000 to 75,000 overnight.

[Read more](#)

Buffer: Idea to paying customers in seven weeks

Buffer validated demand with a two-page landing page, then added a pricing page to test willingness to pay, landing its first paying customer four days after launch.

[Read more](#)

Harry's: 100,000 emails from a two-page site

Harry's pre-launch site explained the value proposition on page one and collected emails on page two; the campaign gathered over 100,000 emails in one week, with 77% arriving via referral.

[Read more](#)

Monzo: pre-product waitlist before launch

Monzo put up a pre-product landing page describing a mobile-first bank and grew a waitlist reportedly exceeding 200,000 signups before the product was generally available, using the signal of waitlist demand to support a later crowdfunding round.

[Read more](#)

Further reading

-  Eric Ries - Using AdWords to Assess Demand
-  Luke Szyrmer - Landing Page MVPs

- ☰ Ash Maurya (2012) - "Running Lean" — Step-by-step guidance on using landing pages to validate problem-solution fit before building
- ☰ Kromatic - Experiment Calculator

4.4.9 Landing Page (Extended) Test

In Brief

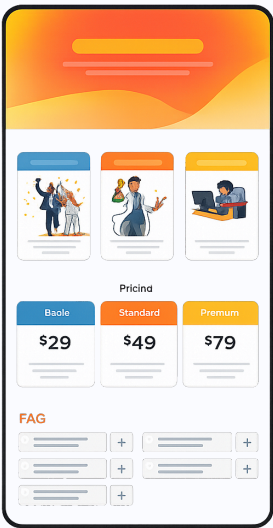
An extended landing page smoke test is a multi-section page for a product that doesn't exist yet — feature sections, pricing tiers, FAQ content, social proof, and a sign-up broken into several steps — built to measure how far visitors read and which price tier they prefer, not just whether they sign up. Visitors click tiers and step through a checkout rather than just leaving an address.

COMMON USE CASE

You already confirmed initial interest with a simple landing page or another smoke test, and now you need to dig into which specific features and price points carry the demand. You have to decide what to build first, what to charge, and how to position, and the cheap signal of “did anyone sign up?” is no longer enough. You want section-level engagement, pricing-tier preference, and data on where visitors abandon the sign-up steps before you commit to a build plan.

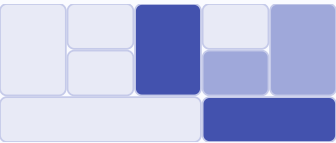
HELPS ANSWER

- Which features or benefits generate the most interest?
- Is the pricing acceptable, and which tier is most attractive?
- How deep is visitor interest — do they read the whole page or leave early?
- At which step of the sign-up process do visitors give up?



AT A GLANCE

- Quantitative
- Remote
- Behavioral



TIME ~2–6 weeks

COST \$100–\$1.2K

OTHER NAMES
Extended Landing Page · Long-Form Landing Page

Description

An extended landing page smoke test puts a multi-section page in front of real traffic to learn how deep demand runs: which features draw attention, which price tier visitors choose, what objections they click into, and how far they get through a multi-step sign-up before dropping out. It asks more of a visitor than email capture does, which places it higher on the commitment ladder than a simple landing page within the Value Proposition Test family — the signal it produces is preference and depth, not just a yes/no on interest.

A simple landing page tests the headline: does the core value proposition attract interest? An extended landing page tests the full story: features, pricing, objections, and depth of commitment.

The extended page typically includes:

- **Hero section:** Headline and primary value proposition (same as a simple landing page)
- **Feature or benefit sections:** 3-5 detailed descriptions of what the product does, each testable for engagement
- **Pricing tiers:** 2-3 options at different price points, testing willingness to pay and preferred feature bundles
- **FAQ section:** Addresses common objections, with each question's click rate revealing what concerns visitors most
- **Social proof:** Testimonials, logos, or usage statistics (use real data from interviews or early users if available)
- **Multi-step sign-up:** Instead of a single email field, guide visitors through 2-3 steps (select a plan, enter details, confirm) to measure how far their commitment goes

The key advantage over a simple landing page is the granularity of data. Instead of a single [conversion rate](#) — the percentage of visitors who take the action you want — you get scroll depth, section-by-section engagement, pricing-tier clicks, FAQ clicks, and the points where visitors abandon the sign-up steps. Each one reveals a different dimension of demand and preference. You're not just measuring "interested or not" — you're measuring "interested in what, at what price, with what concerns."

How to

Prep

✦ AI Prompt

1. Start with a validated simple landing page.

Don't build an extended page from scratch. Run a [simple landing page test](#) first. If the basic value proposition doesn't convert, adding more detail won't fix it. The extended page is for deepening your understanding of demand that already exists.

2. Define what you want to learn from each section.

Every section should test a specific question:

- Feature sections: Which capabilities matter most?
- Pricing tiers: What will people pay, and which bundle is most attractive?
- FAQ: What are the biggest objections or concerns?
- Social proof: Does credibility increase conversion?

Execution

✦ AI Prompt

1. Design the page with clear section boundaries.

Use distinct visual sections so analytics tools can track scroll depth and engagement per section. Each section should be independently valuable — visitors who only read the top half should still get enough information to act.

2. Set up pricing tiers that test real questions.

Use 2-3 tiers with meaningful differences. How clicks spread across the tiers tells you about price sensitivity and which feature bundle people want. Include a “most popular” tag on the middle tier to test whether labeling one option nudges visitors toward it.

3. Implement section-level tracking.

Go beyond basic page analytics. Track:

- Scroll depth (what percentage of visitors reach each section)
- Time spent per section
- Click events on pricing tiers, FAQ questions, and feature CTAs
- Funnel drop-off at each step of the multi-step CTA

Heatmap and session-recording tools show exactly how visitors interact with each section.

4. Drive targeted traffic.

TOOLS

[Unbounce](#)

[PostHog](#)

[Hotjar](#)

[Microsoft Clarity](#)

[Webflow](#)

Use the same traffic sources as a simple landing page test, but expect to need more visitors. Because you are breaking the data down by section, tier, and source, each slice needs enough visitors of its own to be trustworthy. Plan for 500+ unique visitors minimum.

5. Set success thresholds per section.

Don't just set an overall conversion target. Define what success looks like for each section: "At least 60% of visitors scroll past the feature section," "The middle pricing tier gets 50%+ of pricing clicks," "Fewer than 30% of visitors who open the FAQ then leave the page."

6. Run for 2-4 weeks.

Extended pages need more time than simple pages because you're collecting more granular data. Resist the urge to change sections mid-test.

Analysis

✦ AI Prompt

1. Compare each section's performance against the thresholds you wrote down before launch.

The thresholds (scroll depth, time per section, pricing-tier distribution, funnel drop-off) tell you which sections are pulling weight and which are dead. Without them, every section's data feels marginal and no decision gets made.

2. Read the result patterns:

- **High scroll depth, high conversion:** The full story is compelling. Visitors are engaged throughout and convert at the end. Strong signal to proceed.
- **High scroll depth, low conversion:** Visitors are interested enough to read everything but not enough to act. The problem may be pricing, the CTA, or a missing piece of the value proposition. Check which pricing tier got the most attention and what FAQ questions were clicked most.
- **Low scroll depth, low conversion:** Visitors disengage early. The problem is likely the hero section or the transition to the first feature section. Revisit the headline and opening message.

3. Read the pricing-tier signal.

If most clicks go to the cheapest tier, your higher tiers may be overpriced or too similar to each other. If clicks are evenly split, your tiers are well-calibrated. If the middle tier dominates, the "most popular" tag is doing its job of steering visitors there — but verify with a follow-on pre-sales test before locking that price.

4. Cluster the FAQ engagement.

The most-clicked FAQ questions reveal the biggest objections. Address these more prominently in your next iteration — moved into the hero copy, the pricing area, or the section that introduces the relevant feature.

5. Read the funnel drop-off.

Where visitors abandon a multi-step CTA tells you where commitment breaks down. Drop-off at “select a plan” suggests pricing concerns. Drop-off at “enter details” suggests trust concerns. Drop-off at “confirm” usually means the offer is fine but the friction is too high.

6. Segment the data.

A 5% overall conversion rate may hide a 12% rate from one traffic source and a 1% rate from another. Break the numbers down by traffic source, device, and (if you have it) any demographic data. The gap between groups is often where the real signal lives.

Biases & Tips

Length bias

Longer pages have lower overall conversion rates simply because there’s more to read and more places to leave. Compare against extended page benchmarks, not simple page benchmarks.

Section order effects

Visitors engage most with the first sections they see. Sections at the bottom get less attention regardless of content quality. Consider A/B testing section order.

Iteration fixation

Continuous micro-optimization of section copy can become a substitute for a go/no-go decision. Set one decision deadline before launch; treat all iteration as input to the next distinct test, not a patch on the current one.

Honesty at sign-up

A visitor stepping through the multi-step checkout believes they are buying a real product. Don’t take real payment, and make the final step clear that the product is in development with a way to be notified, so no one feels deceived.

Next Steps

- If conversion is strong and the middle pricing tier wins, lock in that price point and design your build plan around the features bundled in it, then confirm willingness to pay with a [Pre-Sales Test](#).

- If scroll depth is high but conversion is low, run [Discovery Interviews](#) to understand the objections behind the most-clicked FAQ questions, then address them directly in the hero copy or pricing area before retesting.
- Use a [Pre-Sales Test](#) to convert visitors who reach pricing into committed buyers and verify they will pay the tier they clicked.
- Use a [Crowdfunding Test](#) to test whether pre-launch financial commitment matches the interest signals from your extended page.
- Use a [Fake Door Test](#) to isolate which specific feature on the page is actually pulling clicks before you build anything.

Case Studies

Buffer: Pricing page added before features existed

Joel Gascoigne added a pricing page between Buffer's value-prop landing page and an email-capture page; clicks on paid plans (not just signups) gave the founding evidence that people would pay, and Buffer landed its first paying customer four days after launch.

[Read more](#)

Dropbox: Waitlist jump from 5,000 to 75,000

Dropbox's pre-launch page combined a feature-walkthrough video, feature list, and email capture; after the March 2008 demo video hit Hacker News and Digg, the beta waitlist jumped from 5,000 to 75,000 in one day.

[Read more](#)

Further reading

-  Unbounce — The Conversion Benchmark Report
-  Nielsen Norman Group — Long vs. Short Pages: Examining User Behavior
-  Joanna Wiebe / Copyhackers — Running a Headline Test (includes the 5-second test)
-  HubSpot — A Beginner's Guide to A/B Testing
-  Steve Blank — The Four Steps to the Epiphany

4.4.10 Mock Sale Test

In Brief

A mock sale presents a real-looking product for purchase, lets the buyer click “Buy Now” and enter the checkout flow, then stops the transaction with an “out of stock,” “coming soon,” or “join the waitlist” message before any money changes hands. Because the buyer does not know it is a test, their behavior reflects genuine purchase intent, making this one of the strongest signals of willingness to pay short of collecting money. It is a sibling of the [Pre-Sales Test](#), which collects real money.

COMMON USE CASE

You have a value proposition that has cleared lighter tests (interviews, landing page signups, ad click-through) and now you need behavioral evidence of purchase intent before you commit to building the product or to handling real payments. You want to see how far down a checkout flow people are willing to go when they believe the offer is real, and you want that signal segmented by audience and price point. The mock sale gives you that data with no inventory risk and no revenue to refund.

HELPS ANSWER

- Will customers actually attempt to buy this product?
- Is the price point acceptable enough to trigger a purchase action?
- How far through the buying steps will customers go before abandoning?
- Does purchase intent differ across audience segments?

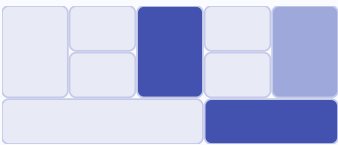


AT A GLANCE

Quantitative

Remote

Transactional



TIME  ~1–2.5 weeks

COST  \$30–\$600

OTHER NAMES

Mock Sale · Dry Wallet · Covert Sale

Description

A mock sale is a real-looking product listing that lets a buyer decide to buy, click the purchase button, and even begin entering payment information — then stops the transaction with an “out of stock,” “coming soon,” or “join the waitlist” message before any money changes hands. Because the buyer believed the transaction was real, those actions are genuine purchase behavior rather than a stated preference. It is the highest-commitment rung of the [Value Proposition Test](#) family that still stops short of taking real money.

That is why the signal is strong: the buyer does not know they are in a test, so their behavior is not colored by the knowledge that “this isn’t real.” Someone who clicks “Add to Cart” and proceeds to checkout demonstrates willingness to pay in a way a survey response or an email signup cannot.

Ethics — You are deceiving the buyer. A mock sale lets a real person believe they are completing a real purchase, which is the source of both the strong signal and the method’s main risk. Never capture payment you could actually charge — stop the flow before real card details are taken — and make the reveal honest and respectful, offering something in return (early access, a launch discount, or a plain explanation of what you are building). Comply with platform and consumer-protection rules; a buyer who feels tricked is a real cost to your brand.

A mock sale is stronger evidence than a [Fake Door Test](#), which only measures a click, because the buyer progresses further into the flow. It also sits a step beyond a [Landing Page Test](#), which captures emails against a stated value proposition: the mock sale carries the buyer further down the buying path, into the checkout itself, rather than asking them to express interest at the top of the funnel. It is weaker than a [Pre-Sales Test](#), where real money changes hands, but it is simpler to run and avoids the legal and operational work of handling payments for an unbuilt product.

How to

Prep

★ AI Prompt

1. Create a realistic product page.

The page must look like a real product listing. Include a product name, description, price, images (renderings or mockups are fine), and a prominent “Buy Now” or “Add to Cart” button. Build it on an e-commerce platform so it looks authentic.

2. Design the checkout interruption.

Decide where in the flow to stop the buyer:

- **After “Add to Cart”:** Lowest friction. The buyer clicks the buy button and immediately sees a “coming soon” message. Tests whether the price and product description are compelling enough to trigger a purchase action.
- **After entering the checkout page:** Medium friction. The buyer clicks buy, sees a checkout form, then encounters a “currently unavailable” message. Tests willingness to start the checkout process.
- **After entering partial payment info:** Highest friction and highest ethical risk. Only do this if the reveal is immediate and the message is very clear. Tests the deepest level of purchase commitment.

Choose the option that gives you the data you need without crossing an ethical line you are uncomfortable with.

3. Craft the reveal message carefully.

The message the customer sees instead of a completed purchase shapes how they react. Good examples:

- “Thanks for your interest! [Product] is coming soon. Leave your email to be first in line and get 20% off at launch.”
- “We’re currently sold out. Join the waitlist to be notified when we’re back in stock.”
- “We’re putting the finishing touches on [Product]. Sign up for early access.”

Avoid messages that make the customer feel tricked. Offer something real in exchange for their interrupted experience.

4. Set up tracking at each step of the buying flow.

Track every step: page views, “Add to Cart” clicks, checkout page views, form-field interactions, and reveal-page engagement (email signups from the reveal). The drop-off from one step to the next tells you where commitment breaks down.

5. Set a success threshold before launching.

Define what “enough” purchase intent looks like: “At least 5% of visitors click Buy Now” or “At least 2% reach the checkout page.” Without a pre-set threshold, you’ll rationalize any result.

Execution

✦ AI Prompt

1. Drive targeted traffic.

Use the same traffic sources you would for a real product launch. Paid ads, social media, or email outreach to your target audience. The traffic source must match your actual go-to-market plan for results to be meaningful.

TOOLS

[Shopify](#)

[Carrd](#)

[Stripe Payment Links](#)

[Hotjar](#)

2. Track every funnel stage in real time.

Watch the page-view → add-to-cart → checkout drop-offs as the test runs. If add-to-cart is healthy but checkout collapses, the issue is the form, not the value proposition — note it but don't kill the test early; you may still hit your overall threshold.

3. Capture and follow up on reveal-page emails.

Every email left on the reveal page is a high-intent lead. Send the promised follow-up (early-access list, launch discount, or genuine update) within 24 hours. The follow-up open and click rates are a second-order signal of demand strength.

4. Run for 1-2 weeks, then stop.

Don't run a mock sale indefinitely. The longer it runs, the more customers encounter the disappointing reveal, and the greater the risk of negative word-of-mouth. Collect enough data to make a decision and shut it down.

Analysis

✦ AI Prompt

1. Compare results against the thresholds you wrote down before launching.

The pre-set threshold ("at least 5% add-to-cart" or "at least 2% reach checkout") is the bar. Without it, you'll rationalize any result.

2. Read the result patterns.

- **High "Add to Cart" rate, high checkout progression:** Strong purchase intent. The value proposition and price point work. Proceed toward building or collecting real pre-sales.
- **High "Add to Cart" rate, low checkout progression:** People like the product at the listed price but something about the checkout process deters them. May indicate trust concerns or friction in the flow rather than lack of demand.
- **Low "Add to Cart" rate:** The product page isn't compelling enough to trigger a purchase action. The problem is likely the value proposition, the price, or the product presentation — not the checkout flow.

- **High reveal-page email capture:** Even though the purchase didn't complete, customers who leave their email after learning the product isn't available yet are demonstrating strong interest. Follow up with these people.
- **Different rates across price points:** If you ran multiple prices, the gap between rates tells you elasticity. A 5% add-to-cart at \$29 dropping to 1% at \$49 is a clearer signal than either rate in isolation.

3. Segment buyers from browsers.

Pull whatever audience attribution your traffic source provides (ad audience, referrer, UTM) and re-cut the funnel rates by segment. A healthy aggregate rate hiding two segments — one that's enthusiastic and one that's flat — points you at the audience to keep targeting and the audience to drop.

4. Discount the mock-context inflation.

Mock sales overstate willingness to pay because no money actually leaves the buyer's account. Treat the headline rates as an upper bound on what real-money behavior could look like, not a point estimate, and plan to validate against actual payment before relying on them.

Biases & Tips

Ethical backlash

Some percentage of customers will be annoyed by the experience, regardless of how well you handle the reveal. Factor this into your brand risk assessment, especially if you plan to sell to the same audience later.

Traffic-source mismatch bias

Ads that attract bargain hunters or curiosity clickers rather than genuine prospects inflate funnel drop-off rates, making a good value proposition look weak — and the signal cannot be recovered after the test ends.

Reveal message influence

A compelling reveal message (20% discount, exclusive early access) can inflate email capture rates on the reveal page, making results look better than organic demand warrants.

Single-session bias

A mock sale measures impulse purchase intent. Customers who would have bought after researching competitors or sleeping on it won't be captured.

Sample-size hubris

A 5% add-to-cart rate over 200 visitors is not the same evidence as 5% over 2,000. Resist scaling the result before you have enough events to support it.

Next Steps

- If add-to-cart and checkout-progression rates clear your threshold, follow up with the email captures on the reveal page and treat them as a high-intent waitlist.
- If checkout progression collapses despite high add-to-cart, redesign the form fields and trust signals before blaming the value proposition.
- Use a [Pre-Sales Test](#) to convert validated purchase intent into actual revenue and remove the ethical baggage of an unfinished checkout.
- Use a [Crowdfunding Test](#) to scale that purchase intent into committed funding before manufacturing or development.
- Use a [Relabel Test](#) to fulfill the demand using a competitor product so customers from the mock sale aren't left empty-handed.

Case Studies

Buffer — Two-page mock pricing test

Joel Gascoigne built a two-page Buffer test: page one described the product, page two presented pricing tiers, and clicking any plan led to a “not quite ready yet” email-capture page instead of a real checkout. The mock pricing page validated demand before the product existed.

[Read more](#)

Further reading

-  ConversionXL — How to Use Tests to Validate Your Product or Feature Ideas
-  Buffer — Idea to Paying Customers in 7 Weeks
-  Federal Trade Commission — Endorsement Guides and Deceptive Practices guidance
-  Steve Blank — The Four Steps to the Epiphany

4.4.11 Online Ad Test

In Brief

An online ad smoke test is a paid advertising campaign on platforms like Google, Facebook, or Instagram that measures which value proposition messaging and audience combinations earn the most clicks. Instead of building a product or even a full landing page, you put the core promise in front of strangers and see whether they click an ad that describes it. A click is a small, observable, segmentable, and scalable signal from a prospect who has no other reason to act.

COMMON USE CASE

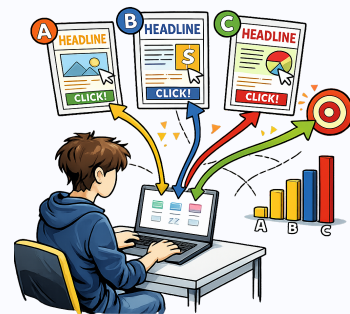
You have a value proposition you can describe in one sentence and you need to know whether it lands with strangers, not just the people you already interviewed. You want behavioral evidence — clicks from a targeted audience — before you build a product, hire a designer, or commit to a longer-form landing page. A short paid ad test puts the message in front of a targeted audience and tells you which framing earns a click.

HELPS ANSWER

- Does this value proposition generate interest from real people?
- Which audience segment responds most strongly to this message?
- Is pull demand (search) or push demand (social/display) stronger for this concept?
- Which messaging angle produces the highest engagement?

Description

An online ad smoke test puts your core promise into paid ad copy, targets a specific audience, and counts how many people click — the click being the commitment



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~1–2.5 weeks

COST  \$30–\$700

OTHER NAMES

Online Ad · Run Test Ads · Ad Test

you're measuring, one of the lowest rungs on the [Value Proposition Test](#) commitment ladder. No product, and no full landing page, is required to get a behavioral read on whether the message lands.

The [click-through rate](#) (CTR) — the percentage of people who saw the ad and clicked it — is your primary output. It tells you what percentage of a given audience found your message compelling enough to act on. Each time the ad is shown to someone counts as one impression, so CTR is clicks divided by impressions.

This test pairs naturally with a [Landing Page Test](#). The ad tests the message; the landing page tests the value proposition in more detail. Together, they form a two-stage funnel: ad CTR tells you whether the promise is interesting, and landing page conversion tells you whether the full explanation is convincing.

Pull ads (Google Search, Bing) target people who are already searching for a solution. High intent means higher CTR benchmarks (3-5% is average for search ads). If your pull ad performs well, you know people are actively looking for what you describe.

Push ads (Facebook, Instagram, display networks) interrupt people who are not looking for a solution. Lower intent means lower CTR benchmarks (1-2% is average for social ads). If your push ad performs well, you know the value proposition is compelling enough to grab attention unprompted.

Key metrics to track:

- **CTR (click-through rate):** The percentage of impressions that result in a click. Primary metric for whether the message lands.
- **CPC (cost per click):** How much you pay per click. A lower CPC on competitive keywords suggests your message resonates relative to other advertisers bidding for the same attention.
- **CPM (cost per thousand impressions):** What you pay to show the ad a thousand times. Useful for comparing the cost of reaching different audiences before anyone clicks.
- **Conversion rate:** If paired with a landing page, the percentage of clicks that go on to complete a target action such as a signup.

How to

Prep

✦ AI Prompt

1. Define your hypothesis.

Be specific: “At least 3% of people searching for [keyword] will click an ad describing [value proposition]” is testable. “People will like our ads” is not.

2. Choose pull or push (or both).

Use pull ads (search) if you believe people are already looking for a solution. Use push ads (social/display) if you believe the value proposition is novel and people don’t know to search for it. Running both gives you the most complete picture.

3. Write 2-3 ad variants.

Each variant should test a different angle of the same value proposition. Change the headline, the benefit statement, or the framing, but keep the core offer the same. This lets you compare which message resonates most.

4. Configure precise targeting.

For search ads, choose keywords that match what your target audience is actually searching for, and use phrase match or exact match as your default — broad match can trigger your ad for irrelevant queries and corrupt the signal. For social ads, define the audience by demographics, interests, or behaviors. Overly broad targeting dilutes your signal.

5. Set a daily budget and timeline.

Start with \$10-20/day and plan for 7-14 days. This gives you enough impressions and clicks to compare variants. Use the platform’s daily budget cap to control spend.

Execution

1. Launch and do not touch.

Once ads are live, don’t change them. Editing ad copy, adjusting targeting, or changing budgets mid-test invalidates your data. If something is broken (wrong URL, typo), fix it and restart the clock.

2. Track delivery, not just spend.

Watch impressions and reach across variants daily. If one variant is starving for impressions because the platform’s optimizer is favoring another, you don’t have a comparable test — note it and either rebalance budgets at the campaign level or accept that you’re learning about the optimizer’s preference, not the variant’s relative pull.

3. Capture the click destination behavior.

✦ AI Prompt

TOOLS

[Google Ads](#)

[Facebook Ads Manager](#)

[Google Keyword Planner](#)

[LinkedIn Campaign Manager](#)

If you sent traffic to a landing page or fake-door page, log every conversion event with its source ad variant and audience. The ad CTR tells you whether people clicked; what they do after the click tells you whether the promise held up.

4. Run for the full planned duration.

Stopping early is the most common cause of false positives. CTR can swing by 30–50% across days as the platform learns and as different audience cohorts get reached. Hold the test open until you hit the impression and click thresholds you set in Prep.

Analysis

✦ AI Prompt

1. Compare results against the thresholds.

The hypothesis (“at least X% CTR”) is the bar. Without a pre-set threshold, every result feels like “interesting data” and no decision gets made.

2. Read the result patterns.

- **High CTR, high landing page conversion:** Strong signal. The message attracts the right people and the value proposition holds up under scrutiny.
- **High CTR, low landing page conversion:** The ad promise is interesting but the landing page doesn’t deliver. The problem may be messaging mismatch or insufficient detail, not lack of demand.
- **Low CTR across all variants:** The value proposition doesn’t resonate with this audience through this channel. Try a different audience, different messaging, or a different channel before concluding there’s no demand.
- **High CPC:** The audience is competitive, which actually suggests demand exists – other advertisers are paying to reach these people.
- **One variant dominates the others:** The winning angle tells you what part of the value proposition is doing the work. Carry that framing forward into the landing page and into longer-form copy.

3. Cluster the search-term and audience-overlap reports.

For search ads, pull the actual search queries that triggered your ad (search-term report) and check whether your keywords are matching what you intended. For social, check the audience-overlap report – if two of your “different” audiences overlap heavily, you ran one test, not two.

4. Separate paid pull from organic intent.

A high CTR on a high-intent search keyword tells you demand exists for the solution category, not that your specific value proposition won. Compare your CTR against the keyword's average CTR if the platform exposes it. Outperforming the category average is the signal you want.

Biases & Tips

Ad fatigue

Running the same ad too long causes CTR to decline as the same people see it repeatedly. Keep test periods to 1-2 weeks so declining CTR reflects the message, not repeat exposure.

Underpowered d-test fallacy

Declaring a winner before reaching roughly 1,000 impressions and 30 clicks per variant turns noise into a false signal. Set minimum thresholds in Prep and refuse to read results before hitting them.

Platform- optimization bias

The platform's auto-optimizer concentrates spend on whichever variant it predicts will perform, starving the others of impressions. The variant that "won" may simply be the one the algorithm fed, not the one people preferred. Watch impressions per variant and rebalance budgets at the campaign level so each variant gets a comparable number of views.

Broad-match keyword leakage

On search ads, broad keyword matching shows your ad against queries you never intended, inflating impressions with the wrong audience and distorting CTR. Default to phrase or exact match, and pull the search-term report to confirm your ad matched the queries you meant to test.

Audience overlap

If two "different" social audiences share most of their members, you ran one test against one group, not two independent tests. Check the platform's audience-overlap report before treating two segments as distinct.

Novelty effect

A new ad can draw an early spike in clicks simply because it is unfamiliar, then settle to a lower rate. Judge CTR over the full run rather than the first day or two, and discount an early surge that does not hold.

Survivorship in the funnel

A high CTR with a weak downstream conversion rate means the ad attracted clicks the value proposition could not hold. Read the click as one signal and the post-click action as the other; a click alone does not confirm demand.

Next Steps

- Pair the winning ad variant with a [Landing Page Test](#) to capture the traffic and measure end-to-end conversion — how compelling the full pitch is, not just whether people clicked.
- Use a [Fake Door Test](#) to push that traffic one level deeper into commitment and validate which feature people are actually clicking for.
- Use a [Social Media Campaign](#) to amplify the winning message organically once paid ads have proven which framing resonates.
- Use [A/B Testing](#) to systematize the variant-comparison method beyond ad copy and apply it to landing pages, pricing, or onboarding.

Case Studies

IMVU: AdWords to assess demand

Eric Ries describes IMVU's use of AdWords campaigns to measure interest in product variants before building them, framing search ads as one of the cheapest behavioral signals available to a startup.

[Read more](#)

Tim Ferriss: Testing The 4-Hour Workweek title with AdWords

Before publishing, Ferriss ran AdWords variants of competing book titles against the same audience and measured click-through; the winning title became the bestseller.

[Read more](#)

Further reading

-  Eric Ries — Using AdWords to Assess Demand
-  Tim Ferriss — How to Test a Book Title with Google AdWords
-  Google — About click-through rate (CTR)
-  Meta — About ad delivery and the auction

4.4.12 Pop-Up Store Test

In Brief

A pop-up store smoke test is a temporary retail location, open anywhere from a single day to several weeks, that you run to see whether people will hand over money for a product when they can see it, touch it, and talk to the person selling it. The product can be fully built or represented by prototypes and mockups; the signal is the sale itself. You ask for a real purchase in person, not just an email or a click. It is most useful for physical products, food and beverage concepts, and any business where the in-store experience is part of the value proposition.

COMMON USE CASE

You have early evidence of demand from interviews or a digital smoke test, and now you need to see whether intent translates into purchase behavior in person. You want to observe how people physically react to the product, what objections come up face-to-face, and whether geography or context changes who is willing to pay. A short, real-world retail presence is cheaper than a launch and richer than another survey.

HELPS ANSWER

- Will people buy this product when they encounter it in person?
- How do people physically interact with the product?
- What questions and objections come up in face-to-face selling?
- Is there demand in this specific geographic area or neighborhood?
- Does the in-person experience change willingness to pay?



AT A GLANCE

Quantitative

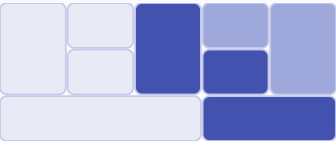
Qualitative

B2C

DTC

In-person

Transactional



TIME ~3–8 weeks

COST \$300–\$6K

OTHER NAMES

Pop-Up Store · Pop-Up Shop ·
Temporary Retail

Description

A pop-up store smoke test puts a product in front of strangers in a temporary physical space and measures who actually buys it. The founder rents a stall, storefront, or shared retail spot for a day to a few weeks, displays the product, and tracks how many passersby engage and how many pay. Because it asks for a money commitment in person, it sits near the high-commitment end of the

📖 **Value Proposition Test**

family, and it pairs that commitment with the urgency of a limited time window.

What makes a pop-up a smoke test rather than just selling things is the intent. You are not building a sustainable retail operation; you are using a temporary physical presence to collect evidence about demand, pricing, fit, and how people respond to the product in person.

The pop-up format provides data that a digital test cannot:

- **Real-time reactions:** You see facial expressions, body language, and hesitation, and you hear unprompted comments and questions.
- **Sensory experience:** Prospects can touch, hold, taste, or try the product. For physical goods, handling it often changes whether they buy.
- **Objection discovery:** Face-to-face selling surfaces objections that surveys and landing pages miss, such as “I’d need it in blue” or “I’d never pay that much.”
- **Geographic signal:** A digital test tells you about online demand. A pop-up tells you about demand in a specific location, which matters for any business with a physical component.
- **Live price testing:** You can adjust the price between days, or between prospects, and watch how it changes who buys.

The product does not need to be finished. You can run a pop-up with prototypes, samples, mockups, or just a compelling display and an order form. The question is not whether the product is perfect, but whether people will pay for what it represents.

Ethics — You’re taking real money in person. If you sell a product you cannot yet deliver (a prototype, or stock you don’t have), say so at the point of sale and fulfill or refund promptly — never take payment for something you can’t hand over.

How to

Prep

🔗 AI Prompt

1. Choose a location that matches your target customer.

The location is how you target who walks past. Options ranked by cost and commitment:

- **Farmers markets / flea markets / craft fairs:** \$50-200/day. Lowest risk. Good for consumer products, food, and artisan goods. Pre-built foot traffic.
- **Shared retail spaces:** \$200-1,000/week. Short-term retail marketplaces offer week-by-week leases. More professional setting.
- **Partner locations:** Free to low cost. Set up inside a complementary business (a coffee shop, gym, bookstore) that serves your target audience.
- **Standalone pop-up:** \$1,000-5,000/month. Rent a vacant storefront. Maximum control, highest cost and effort.

2. Define what you're measuring.

Before opening, write down your hypotheses and success thresholds:

- Units sold per hour or per day
- Conversion rate: the share of people who engaged who then bought
- Average transaction value
- Most common questions and objections
- Price sensitivity observations

3. Create an experience, not just a display.

Your pop-up should let people interact with the product. For food, offer samples. For physical products, let people handle them. For services, do live demonstrations. The interaction is the test: passive browsing tells you much less than active engagement.

4. Staff with the founder (or someone who can learn).

The person selling should be someone who can absorb what prospects say, notice patterns, and adapt the pitch in real time. The founder is ideal. Don't outsource this to someone who will only ring up transactions.

Execution

1. Track everything systematically.

Keep a simple log:

- Number of people who entered / approached
- Number who engaged (asked a question, touched the product)
- Number who purchased
- What they said (especially objections and praise)

TOOLS

[Appear Here](#)

[Square](#)

[Shopify POS](#)

- Time of day patterns

Use a tablet or notebook. Don't rely on memory at the end of the day.

2. Test pricing in real time.

If feasible, try different price points on different days or with different groups of passersby. "What would you expect to pay for this?" asked after someone has handled the product is more reliable than the same question in a survey.

3. Follow up with buyers.

Collect email addresses or phone numbers from customers who purchase. Follow up within a week to ask about their experience. These early customers are invaluable for product development feedback.

Analysis

✦ AI Prompt

1. Compare results against the thresholds you wrote down before opening.

The thresholds (units per hour, engaged-to-buyer conversion, average transaction value) tell you what counts as a hit. Without them, every result feels like "interesting data" and no decision gets made.

2. Read the result patterns:

- **Strong sales, enthusiastic reactions:** Clear demand signal. Proceed to scale — whether that means more pop-ups, a permanent location, or an online launch backed by physical evidence.
- **Strong interest but low conversion:** People are attracted to the product but something stops them from buying. Listen carefully to what they say. Is it price? Timing? A missing feature? "I love it but..." is the most informative phrase you can hear.
- **Low foot traffic, low engagement:** The location may not match your audience, or your display doesn't attract attention. Before concluding there's no demand, try a different location or a more engaging display.
- **Sales from unexpected segments:** Pay attention to who actually buys. If your target was young professionals but parents are the ones purchasing, you may have discovered a better market.
- **Questions you didn't anticipate:** The questions people ask reveal gaps in how you communicate the value proposition. If everyone asks the same question, display the answer prominently.

3. Cluster the qualitative data.

Pull every objection, question, and unprompted comment from the day's notebook. Group them. The five most-repeated objections matter more than the one most clever piece of feedback. Repetition is the signal.

4. Separate demand signal from novelty signal.

A pop-up's time-limited frame creates urgency that won't survive a permanent launch. Look at how many buyers asked when you'd be back, signed up to be notified, or referred someone — those are demand-pull signals. Single-transaction impulse buys with no follow-on interest are weaker evidence.

Biases & Tips

Novelty effect

Pop-ups attract curiosity buyers who purchase because the experience is new and time-limited. "I better get it now" is urgency-driven, not demand-driven. Track repeat interest (email signups for "when you have a permanent store") alongside impulse purchases.

Founder selling bias

A passionate founder can sell almost anything face-to-face. If the founder is the only one who can close sales, the demand signal is partly about the founder's charisma, not the product.

Geographic selection bias

Sales at a single farmers market reflect the demographics and foot traffic of that one location. A conversion rate there may not hold in a different neighborhood or venue type — treat any single-location result as a hypothesis, not a proof.

Underpowered sample overconfidence

A single weekend pop-up typically yields 20–50 transactions — too few to distinguish a real conversion rate from random variation. Treat the result as directional signal and require a second run before committing resources.

External-confound false negative

A rainy Saturday or a competing local event can cut foot traffic by 50%+ independent of demand. Log weather and local context for every session and discard or quarantine outlier days before drawing conclusions.

Next Steps

- Debrief your notes: cluster the most-asked questions and the most-cited objections, then update your product copy and signage before the next pop-up.
- If sales were strong in one location, repeat in a different neighborhood before drawing conclusions about geographic generalizability.

- Use a **Event Test** to put the product in front of a denser, more-targeted crowd than walk-up retail can deliver.
- Use a **Mock Sale Test** online to test whether the in-person purchase intent translates to digital channels and to a wider audience.
- Use a **Pre-Sales Test** to convert the in-person interest into committed orders for product you have not yet manufactured.
- Use a **Flyer Test** to drive targeted foot traffic to the next pop-up location and isolate the demand signal from random walk-by traffic.

Case Studies

Warby Parker: Pop-ups and the Class Trip bus

Warby Parker, originally online-only, ran pop-up shops and a converted-school-bus tour (the “Class Trip”) to test retail demand; the results led the company to open permanent stores.

[Read more](#)

Further reading

-  Storefront — How to Run a Successful Pop-Up Shop
-  Niehm & Fiore — Pop-Up Retail’s Acceptability as an Innovative Business Strategy (Journal of Shopping Center Research, 2007)
-  Steve Blank — The Four Steps to the Epiphany
-  Lightspeed — How to Start a Pop-Up Shop

4.4.13 Pre-Sales Test

In Brief

A pre-sales smoke test collects real payment — through pre-orders, deposits, or letters of intent — in exchange for a promise to deliver the product later. You ask prospects to put down money, not just express interest, before the product exists, then count how many convert and how much revenue they commit. A money commitment made before delivery tells you whether people will pay for your value proposition before you build it.

For a public, platform-hosted campaign with a crowd instead, see [Crowdfunding](#).

COMMON USE CASE

You have validated interest through interviews and landing page signups, but you need proof that people will pay before you invest in building the product. You set up a pre-order page, or take deposits directly, and ask prospects to put down real money — the strongest possible signal of demand.

HELPS ANSWER

- Are customers willing to pay money for this product?
- Is the price point acceptable to the target audience?
- How many people will commit before the product exists?

Description

A pre-sales smoke test asks a prospect to prepay for a product that is not yet built, with delivery promised for a future date — the moment they pay, they become a paying customer rather than just an interested prospect. You take real money up front

PRE-ORDER



AT A GLANCE

Quantitative

Qualitative

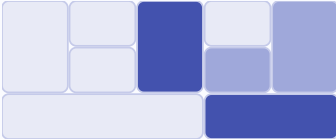
B2B

B2C

In-person

Remote

Transactional



TIME

 ~3–8 weeks

COST

 \$0–\$700

OTHER NAMES

Pre-Sale · Presale · Pre-Order

and treat the count of buyers and revenue collected as the demand signal. It sits at the highest commitment rung of the [Value Proposition Test](#) family — the strongest signal that family can produce. Pre-sales is direct prepayment you collect yourself through your own channel; for a public, platform-hosted campaign run on a third party, use [Crowdfunding](#) instead.

Ethics — This is real money. You are taking real payment for a product that does not exist yet. That is an obligation, not just a test: set honest delivery expectations, deliver what you promised, and refund in full and promptly if you decide not to build it. Taking money you cannot honor is fraud, not validation.

A prepaid order is a stronger signal than a click or a signup because the prospect is out of pocket until you deliver. Pre-sales produce both demand evidence and working capital. The risk runs the other way: a missed delivery date turns paying customers into refund requests and reputation damage, so plan delivery before you take the money. If you cannot deliver, refund promptly rather than stall.

How to

Prep

★ AI Prompt

1. Define what you are testing.

A pre-sales test answers one question: will people pay real money for this? If you haven't validated that people have the problem or want the solution, run interviews or a [Fake Door Test](#) first. Pre-sales is a high-commitment test — use it when you already have signals of interest and need to confirm willingness to pay.

2. Choose your pre-sales format.

- **Pre-order page** — The buyer pays full price (or a deposit) for a product you'll deliver later. Best for physical products, software with a clear launch date, or services. Use a payment processor or e-commerce platform.
- **Letter of intent / deposit** — A prospect signs a non-binding commitment or puts down a refundable deposit. Best for B2B, high-ticket items, or services where full pre-payment is unusual. A signed LOI from a decision-maker is a strong signal even without money changing hands.
- **Early access / founding member** — The buyer pays a discounted rate for early access. Best for subscriptions, communities, or SaaS products.

If a public, platform-hosted campaign fits better than collecting payment yourself, run a [Crowdfunding](#) campaign instead — same money commitment, different (and

noisier) platform-and-crowd dynamics.

3. Set your price and refund terms.

Price at or near your planned market price. Offering a large discount tests whether people want a deal, not whether they'll pay for your product. Be transparent about delivery timelines and refund policies. If buyers can cancel at any time with a full refund, you're testing interest, not commitment — that's fine, but know what you're measuring. Check your local consumer protection rules — in the US, the FTC's Mail, Internet, or Telephone Order Merchandise Rule requires sellers to ship within the stated timeframe (or 30 days if no timeframe is stated) and to refund within seven working days if they cannot.

4. Set your success threshold before you launch.

Decide in advance how many pre-sales (or how much revenue) counts as a win.

Common approaches:

- **Unit target:** "We need 50 pre-orders to justify a production run."
- **Revenue target:** "We need \$5,000 in pre-sales to cover initial development costs."
- **Conversion rate:** "At least 3% of people who see the offer must buy." (This only works if you know your traffic or audience size.)

If you don't set a threshold up front, you'll rationalize any result as success.

5. Create your sales assets.

Build only what you need to communicate the value proposition clearly:

- A product description or sales page that explains what the customer gets, when they'll get it, and what it costs.
- Product renderings, mockups, or prototypes. AI image tools can generate realistic product visuals. Show prototypes or clearly labeled concepts — never fake product photos of something that doesn't exist.
- Social proof if you have it: testimonials from interviews, pilot customers, or domain experts.
- A clear call to action: "Pre-order now," "Back this project," "Reserve your spot."

6. Plan your distribution.

Pre-sales pages don't sell themselves. Plan how you'll drive traffic:

- **Existing audience:** Email list, social media followers, community members. If you have fewer than 500 contacts, supplement with other channels.
- **Paid ads:** Run targeted ads to the pre-sales page. Treat it as a [Fake Door Test](#) with a higher commitment bar.
- **Direct outreach (B2B):** Email or call prospects directly. 10 face-to-face pitches that produce 3 signed LOIs is a strong signal.

7. Set up tracking.

Track each stage from first view to final sale, not just the sales total:

- Page views or pitch meetings (top of funnel)
- Add-to-cart or expressed interest (middle)
- Completed purchases or signed commitments (bottom)
- Refund requests or cancellations (post-purchase)

The conversion rate at each step — the percentage of people who move on to the next stage, ending with those who actually pay — tells you where demand breaks down.

Execution

1. Launch the pre-sales offer.

Deploy exactly as designed. Don't change the price, copy, or offer structure mid-campaign — changes invalidate your data. If you discover a critical error (broken payment link, wrong price), fix it and note the date.

2. Promote actively.

Pre-sales require active promotion, not passive waiting. Follow your distribution plan.

3. Track the funnel daily.

Monitor page views, add-to-carts, completed purchases, and refund requests. Watch for drop-off points. If people view the page but don't buy, the price or messaging may be off. If they add to cart but abandon, the checkout process may be the issue.

4. Talk to buyers and non-buyers.

Reach out to people who bought and ask why. Reach out to people who showed interest but didn't buy and ask what stopped them. These conversations are often more valuable than the sales numbers themselves.

🔗 AI Prompt

TOOLS

[Stripe Payment Links](#)

[Gumroad](#)

[Midjourney](#)

[Synthesia](#)

1. Calculate your conversion rate.

✦ AI Prompt

Conversion rate = completed purchases ÷ total page views (or total pitches) × 100.
This is your primary metric.

2. Compare against your pre-set threshold.

- **Above threshold:** You have validated willingness to pay. Proceed to build and deliver to your pre-sale customers.
- **Below threshold but close:** Before giving up, test a different price point, messaging, or audience. A 2% conversion rate when you needed 3% might mean your copy is weak, not that demand is absent.
- **Well below threshold (or zero):** The value proposition at this price doesn't resonate. Revisit your pricing, positioning, or target audience. Zero pre-sales with meaningful traffic is a clear signal.

3. Analyze refund and cancellation rates.

If many people buy and then cancel, you may have created artificial urgency or fear of missing out that drove impulsive purchases, not real demand. A refund rate above 20% is a warning sign. Net pre-sales (after refunds) is a more honest metric than gross pre-sales.

4. Segment your results.

If you have enough data, break conversion rates down by source (email list vs. ads vs. organic), by price tier (if you offered multiple options), or by customer type. This tells you which audience actually has willingness to pay, not just interest.

5. For small samples (under 30 purchases): Treat the results as directional. Focus on qualitative signals: What did buyers say when you asked why they purchased? What stopped non-buyers? Pair the sales data with 5–10 follow-up conversations to understand the “why” behind the numbers.

⚠ Biases & Tips

Friends-and-family bias

Your inner circle will buy to support you, not because they need the product. Exclude purchases from people you know personally when calculating your conversion rate. If all your pre-sales are from people you could text, you haven't validated demand.

Discount demand distortion

A deeply discounted early-adopter price selects for deal-seekers, not genuine buyers, inflating conversion and obscuring the true willingness to pay at market price.

FOMO-driven purchases

Limited-time offers and countdown timers drive impulsive buying. Your refund rate will tell you whether purchases were genuine or pressure-driven.

Traffic confound

Zero pre-sales may reflect insufficient reach rather than absent demand. Check page-view counts before concluding the value proposition failed.

Warm audience bias

Your email list and social following are your warmest possible audience. High conversion from them doesn't guarantee the same from cold traffic. Low conversion from a small, poorly targeted list doesn't disprove demand either.

Sunk cost after collecting money

Once you've collected real money, it's psychologically hard to decide not to build. But if follow-up conversations reveal the product won't work, refunding is better than building something doomed.

AI-inflated funnel bias

AI tools make it cheap to generate mass interest signals (likes, signups, outreach replies) that look like demand. Pre-sales payment is the filter; treat any interest that did not convert to payment as unvalidated.

Next Steps

- If pre-sales hit your target, proceed to build with customer funding as validation.
- Use a **Solution Interview** with pre-sale customers to understand their top priorities before committing to a product roadmap.
- If pre-sales fall short, test a different price point or value proposition framing with a **Value Proposition Test**.
- Use pre-sale customers as a beta group for early product feedback.
- Use a **Concierge Test** to manually deliver the product experience to pre-sale customers and learn what features matter most.

Case Studies

Tesla: Model 3 reservations with ,000 deposits

In April 2016, Tesla had collected close to 400,000 refundable \$1,000 deposits for the Model 3 in the two weeks since unveiling, an implied ~\$14B in pre-orders.

[Read more](#)

Coin: \$50K pre-order goal in 40 minutes

Coin's all-in-one electronic credit card hit its \$50,000 pre-order goal in 40 minutes in 2013. (The product later struggled with delivery; the early pre-order velocity remains the demand signal of interest.)

[Read more](#)

Close.io: Selling sales services before the CRM existed

Steli Efti's team at ElasticSales sold outsourced sales engagements to companies, using those paid engagements to fund and shape the eventual Close.io CRM.

[Read more](#)

Jolla: Pre-sales campaign for the Jolla Phone

In 2013 the Finnish mobile-OS startup Jolla closed its first pre-sales campaign for the Jolla Phone, using deposits to gauge demand for its Sailfish OS handset before scaling production.

[Read more](#)

Further reading

-  Eric Ries — The Lean Startup (Crown Business, 2011)
-  Steve Blank & Bob Dorf — The Startup Owner's Manual (K&S Ranch, 2012)
-  Kickstarter — "Kickstarter Is Not a Store"
-  US FTC — Mail, Internet, or Telephone Order Merchandise Rule (16 CFR Part 435)
-  Harvard Business Review — Sell Your Product Before It Exists

4.4.14 Referral Program Test

In Brief

A referral program smoke test measures whether people will actively promote your product to others when you give them an explicit reason and an easy way to do it. You hand each participant a unique link and an incentive to share; how many of them refer, and how many of those referrals convert, tells you whether the value proposition is strong enough to put their own name behind it.

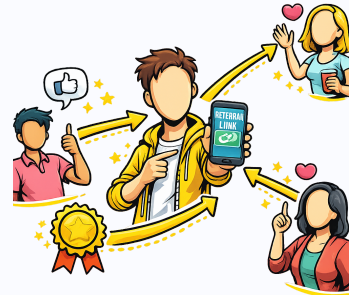
For the uninstructed, no-incentive version of advocacy — people sharing because they want to, with no reward — see [Broken Promise Test](#).

COMMON USE CASE

You have an audience or seed list and want quantitative evidence that the value proposition is compelling enough to drive structured word-of-mouth — before investing in paid acquisition or building a full referral program into the product. Use this when you need to learn whether referrals can be a viable acquisition channel, or to size the viral coefficient (K-factor) for an existing offer.

HELPS ANSWER

- Is the value proposition compelling enough that people will recommend it to others?
- What referral incentive structure drives the most sharing?
- What is the viral coefficient — how many new users does each referrer bring in?
- Which customer segments are most likely to refer?
- Can word-of-mouth be a viable acquisition channel?



AT A GLANCE

Quantitative

B2C

C2C

Two-Sided Market

Remote

Behavioral



TIME  ~2–6 weeks

COST  \$0–\$400

OTHER NAMES

Referral Program · Word of Mouth Test

Description

A referral program smoke test gives each participant a unique link and a reason to share it, then measures how many of them actually recommend the product and how many of those recommendations convert into new signups. It is the social-commitment rung of the [Value Proposition Test](#) family: where other variants ask for money or time, this one asks the participant to stake their own reputation by vouching for you to a friend.

The question you are putting to your audience is “Is this valuable enough that you’ll tell your friends?” The answer tells you about both the strength of the value proposition and the viability of word-of-mouth as an acquisition channel.

The key metric is the viral coefficient (K-factor): the number of new users each existing user brings in through referrals. A viral coefficient above 1.0 means each user brings in more than one new user, and the user base grows on its own without paid acquisition. Sustaining K above 1 is rare; most programs decay below 1.0 within months as the reachable network gets used up. Even $K = 0.3$ helps: it multiplies your other acquisition by roughly $1 / (1 - K) = 1.43x$, so the same paid or organic signups bring in 43% more users for free.

The viral coefficient is calculated as:

$$K = (\text{invitations sent per user}) \times (\text{conversion rate of invitations})$$

For example, if each user sends 5 invitations and 10% of those invitations convert, $K = 5 \times 0.10 = 0.5$. This is the simplified per-cycle K — sustained viral growth also depends on cycle time (how fast each generation refers). A K of 0.5 with a 7-day cycle outperforms K of 0.7 with a 60-day cycle. For a smoke test, per-cycle K is the right metric.

The test comes in two forms, depending on whether you have a product yet.

Pre-product referral programs ask prospects to share in exchange for early access, a better waitlist position, or exclusive benefits at launch. They test the value proposition’s appeal before anything is built, validate demand, and double as audience-building for the launch.

Post-product referral programs ask existing users to refer in exchange for rewards (discounts, credits, free months, gifts). They test whether the product experience is good enough to drive advocacy and whether the referral mechanics actually work end-to-end.

The incentive type you pick changes more than the cost. Cash pays out the fastest signal but attracts reward-chasers and is the most prone to fraud. Account credit or a free month keeps the reward inside your product, so it tends to draw people who actually want what you offer and self-selects for fit. Status or recognition — early access, a leaderboard spot, an exclusive perk — costs almost nothing and produces the most honest read on advocacy, but it motivates a narrower group. For a smoke test, lean toward credit or status: they bias your sample toward genuine fit rather than toward the reward.

One caution on cash: in the US, cash sweepstakes or cash referral payouts can trigger sweepstakes/lottery rules and 1099 tax-reporting obligations once payouts cross reporting thresholds. Check the rules for your jurisdiction before you offer cash, and prefer non-cash rewards if you want to skip the legal overhead.

How to

Prep

✦ AI Prompt

1. Choose pre-product or post-product.

Pre-product: You have a concept but no product. Create a waitlist landing page where signing up gives you a unique referral link. The more friends you refer, the higher you move on the waitlist or the more benefits you unlock at launch. This simultaneously validates demand and builds your launch audience.

Post-product: You have a working product with some users. Add a referral mechanism — “invite a friend, you both get [reward].” This tests whether users like the product enough to vouch for it.

2. Design the incentive structure.

The incentive should be meaningful enough to motivate sharing but not so generous that people refer indiscriminately. Options:

- **Early access / priority:** Free and effective for pre-product. People share to move up the waitlist.
- **Two-sided rewards:** “You get X, your friend gets Y.” Effective because the referrer doesn’t feel like they’re just selling to their friends.
- **Tiered rewards:** More referrals unlock better rewards. Creates a game-like progression that drives sustained sharing.
- **No incentive (pure advocacy):** The hardest but most honest signal. If people refer without any reward, the value proposition is strong on its own.

3. Make sharing frictionless.

Provide:

- A unique referral link for each participant
- Pre-written share messages (customizable) for email, SMS, and social media
- One-click sharing buttons
- A dashboard showing referral status and rewards earned

The harder it is to share, the fewer people will do it, regardless of how strong the value proposition is. Don't let friction contaminate your signal.

4. Seed with an initial audience.

A referral program needs a starting population. Options:

- Your existing email list or social following
- Participants from previous experiments (landing page signups, interview subjects)
- Paid acquisition to the waitlist page, where signups then become referrers
- Community posts in relevant forums or groups

Aim for enough initial participants that the referral rate is not swung by one or two heavy sharers — roughly 100-200 for a consumer program, though a smaller seed of engaged users can still give you a directional read.

5. Build in fraud controls before you launch.

Referral incentives invite gaming, and even a small amount inflates your K-factor with fake signal. Decide up front how you will:

- **Dedupe referrals** so the same referred person counted twice (multiple links, repeat signups) only credits once.
- **Verify real new users** — confirm a referred signup is a genuine new account that takes a real action, not a throwaway email or an existing user re-registering.
- **Catch self-referral** — exclude referrers who sign up their own alternate accounts, plus family, co-founders, and investors.
- **Watch for ring-gaming** — clusters of accounts referring each other in a loop to farm rewards. Device fingerprinting and shared-IP detection help surface these.

Define these rules before launch so you exclude bad referrals consistently rather than deciding case by case after the numbers are in.

Execution

🔗 AI Prompt

1. Track the full referral funnel.

Measure:

- **Participation rate:** What percentage of users/prospects share their referral link?
- **Invitations per referrer:** How many people does each referrer invite?
- **Referral conversion rate:** What percentage of referred people sign up or buy?
- **Viral coefficient (K):** Invitations per referrer x referral conversion rate
- **Time to referral:** How quickly after signing up do people share?
- **Referral depth:** Do referred users also refer others (second-order virality)?

TOOLS

[ReferralCandy](#)

[Viral Loops](#)

[SparkLoop](#)

[Dub.co](#)

[Fingerprint](#)

2. Set success thresholds.

Before launching, define what success looks like:

- “At least 20% of waitlist signups share their referral link”
- “Viral coefficient above 0.3”
- “At least 10% of referred visitors sign up”

3. Run for 2-4 weeks.

Referral behavior takes time. Some people share immediately; others share when reminded or when they have a natural reason to mention the product. Give the program at least 2 weeks before drawing conclusions.

Analysis

🔗 AI Prompt

- **High participation rate, high K:** The value proposition is compelling enough to drive active advocacy. Word-of-mouth can be a significant acquisition channel. Invest in optimizing the referral experience.
- **High participation rate, low K:** People are willing to share but their referrals don't convert. The problem may be that the referral landing page isn't compelling, or that the referrer's network isn't the right audience. Optimize the referred user's experience.
- **Low participation rate, regardless of K:** People don't bother sharing. Either the incentive isn't motivating enough, the sharing process is too difficult, or the value proposition isn't exciting enough to put their name behind. Test a stronger incentive first — if that doesn't work, the value proposition may be the issue.

- **High K from a small number of super-referrers:** A few people are doing all the sharing. This suggests the value proposition resonates intensely with a narrow segment. Find more people like your super-referrers.
- **Referral depth > 1:** If referred users are also referring their own contacts, the program can sustain itself rather than depending on the seed cohort. This is rare and valuable.

⚠ Biases & Tips

Reward-chasing bias

Heavy incentives attract sharers motivated by the reward, not the product — inflating K in ways that collapse post-launch when the incentive ends. Run a parallel no-incentive arm to isolate genuine advocacy.

Network homogeneity

Referrers invite people similar to themselves. If your seed audience is unrepresentative of your target market, referrals compound that bias rather than expanding reach.

Novelty effect in pre-product programs

Waitlist referral programs benefit from exclusivity. “Be first to get access” drives sharing in ways that don’t translate to post-launch referral behavior, so don’t extrapolate launch K-factor from waitlist K-factor.

Small seed audience bias

Below ~100 seeds, individual super-referrers or non-referrers swing the K-factor by huge amounts. Treat early numbers as directional, not conclusive.

Platform bias

Email referrals convert at higher rates than social shares; the channel mix of your program influences your K-factor as much as the value proposition does. Track K by channel so you don’t conflate the two.

AI copy inauthenticity

LLM-generated share messages default to marketing register, not a friend’s voice — referred visitors sense the difference and convert at lower rates. Pilot the top variants with 5–10 people and ask which message they would actually forward before sending widely.

Next Steps

- If K is above 0.3, identify your super-referrers, run [Discovery Interviews](#) with them about why they shared, and build the next acquisition campaign around their language.

- If participation is low, test a stronger two-sided incentive before concluding the value proposition is weak.
- Use a [Broken Promise Test](#) to measure organic, uninstructed virality alongside the structured referral signal so you can separate genuine advocacy from incentive-chasing.
- Use a [Net Promoter Score Survey](#) to quantify how willing existing users are to recommend, and cross-check against actual referral behavior.
- Use a [Landing Page Test](#) to A/B test referral landing pages and lift the conversion rate of incoming invites.
- Use a [Social Media Campaign](#) to seed the initial cohort of referrers when you do not have an existing list large enough to support the program.

Case Studies

Harry's: Tiered referral waitlist

Harry's pre-launch site used a two-page referral waitlist with tiered rewards (more referrals → better free products at launch); 77% of the roughly 100,000 emails collected in one week arrived via referral.

[Read more](#)

Dropbox: Storage-for-storage two-sided referral

Dropbox gave both referrer and referred user 500MB of free storage (capped at 16GB), with users growing from 100,000 to 4M in 15 months; the case is canonical for mapping the referral reward to the product's core value.

[Read more](#)

PayPal: Cash-for-cash two-sided referral

Early PayPal paid both sides of each referral (initially about \$10 each, briefly \$20, then tapered as the network matured), driving roughly 7–10% daily growth and seeding the network effects that supported the eBay acquisition.

[Read more](#)

Further reading

-  Andrew Chen — What's Your Viral Loop? Understanding the Engine of Adoption
-  David Skok — Lessons Learned: Viral Marketing

- ☰ GrowSurf — How the Dropbox Referral Program Led to 3900% Growth
- ☰ Viral Loops — PayPal Referral Program: 100M Users Case Study

4.4.15 Sales Pitch Test

In Brief

A sales pitch smoke test is a live sales conversation — in person, over the phone, or via video call — where you pitch the product directly to a prospect and ask for a yes-or-no commitment. The output is a close rate plus qualitative data: which objections come up most, what language resonates, and whether prospects see enough value to pay. You ask for a real buying decision, not just interest.

COMMON USE CASE

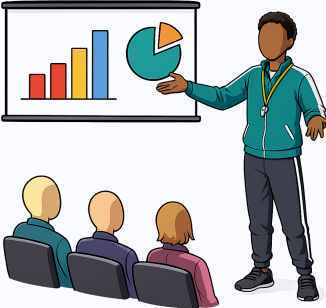
You have a clear idea of what your product will do and who it is for, but have never tried to actually sell it. You schedule meetings with potential buyers and pitch the product as if it already exists, watching their reactions and listening to their objections to learn whether the value proposition is strong enough to close a sale.

HELPS ANSWER

- Will a customer pay money for this product?
- Do customers see enough value in the offering to commit?
- What are the most common objections or concerns?

Description

A sales pitch smoke test is a structured sales conversation you run to learn rather than to earn revenue: you pitch a product or service to a prospect, ask for a commitment, and treat the yes-or-no decision as data about whether the value proposition holds up. The yes-or-no decision is the highest commitment in the [Value Proposition Test](#) family — a real buying decision rather than the interest or sign-up that lower-commitment variants settle for.



AT A GLANCE

Quantitative

Qualitative

B2B

In-person

Remote

Transactional

TIME

 ~3 days–5 weeks

COST

 \$0–\$60

OTHER NAMES

Sales Pitch Test · Cold Pitch Test

The prospect must say yes or no. That binary outcome, multiplied across enough conversations, produces a close rate that tells you whether your value proposition is strong enough to sell.

Unlike a [Landing Page Test](#) or an online ad, a live pitch gives you two outputs: a quantitative signal (did they buy?) and a qualitative one (why or why not?). The objections you hear across multiple pitches reveal what's actually blocking the sale — price, timing, trust, competition, or a weak value proposition.

Ethics — Honor what you close. If a prospect signs or pays for a real offer, you owe them delivery: be clear it is an early offer, honor every commitment you take, and refund promptly if you cannot deliver.

Don't use this to test whether a problem exists (use [Discovery Interviews](#) for that). Use it when you believe you have a solution worth paying for and need to find out if prospects agree. AI helps with pitch prep, objection anticipation, and prospect research, but the commitment moment — when you ask a real person for a real commitment — has to be handled by a person.

How to

Prep

✦ AI Prompt

1. Define your learning goal.

What are you testing? Options:

- **Sellability:** Can this product be sold at all? (Earliest stage — you've never tried to sell it.)
- **Value proposition clarity:** Do prospects understand what you're offering and why it matters?
- **Price sensitivity:** Will prospects pay this specific price?
- **Objection mapping:** What stops people from buying?

Write your goal down. It determines what you track during the conversation.

2. Identify your prospects.

You need people who could actually buy — not friends, not mentors, not people doing you a favor. Criteria:

- They have the problem you're solving (or are in the target segment).

- They have the authority to say yes (especially in B2B — pitching someone who can't approve a purchase wastes both your time).
- They don't already know you well enough to buy out of politeness.

Sources: cold outreach (LinkedIn, email, phone), networking events, industry meetups, trade shows, referrals from non-customers, or existing leads from landing pages or ads. AI can help with prospect research — quickly synthesizing information from LinkedIn, company websites, and news sources to prepare for each conversation.

3. Set your success threshold.

Decide what close rate counts as a win before you start pitching. Benchmarks:

- **B2B, high-ticket:** 10–20% close rate from qualified prospects is strong. Even 1 closed deal out of 10 pitches can be meaningful if the deal size is large.
- **B2C, low-ticket:** 20–30% close rate from in-person pitches to interested prospects.
- **Cold outreach to strangers:** 5–10% is realistic. Below 5% may indicate a weak value proposition or wrong audience.

If you don't set a threshold, you'll call any sale a success and any rejection an outlier.

4. Set a target number of pitches.

You need enough conversations to see patterns:

- **Minimum:** 10 pitches. Fewer than 10 doesn't give you enough data to distinguish between “the product doesn't sell” and “I had a bad day.”
- **Ideal:** 15–20 pitches. By then, the same objections will start repeating, and your close rate will stabilize.
- **If your close rate is exactly 0% after 10 pitches:** Stop and revisit. Either the value proposition, the audience, or the pitch itself needs work.

5. Prepare your pitch.

Structure:

- **Opening (1 minute):** Establish context. Ask about their situation and confirm the problem exists for them.
- **Value proposition (2–3 minutes):** Explain what you're offering, what it does for them, and what it costs. Be specific. Avoid jargon.
- **Ask (30 seconds):** Request a commitment. “Would you like to move forward?” “Can we start next week?” “Would you pre-order today?” The ask must be

concrete — a vague “would you be interested?” doesn’t count as a sales pitch test. Commitment comes in tiers: a verbal “yes” is weaker than a scheduled follow-up with a decision-maker, which is weaker than a signed LOI, which is weaker than a deposit or payment. Stronger commitments are stronger signals.

Don’t memorize a script word-for-word. Know your key points, but have a natural conversation. AI tools can help you draft and refine the pitch, but practice delivering it aloud.

6. Prepare your tracking sheet.

Create a simple spreadsheet to record after each pitch:

- Prospect name/type
- Channel (in-person, phone, video)
- Outcome (closed, rejected, follow-up requested, referred elsewhere)
- Primary objection (price, timing, trust, need, competition, other)
- Key quotes or reactions (verbatim if possible)
- Your confidence rating: did this feel like a real “no” or a “not yet”?

Don’t try to take notes during the pitch — it breaks rapport. Record your observations immediately after the conversation. Use an AI transcription tool if the prospect consents.

Execution

1. Schedule or initiate conversations.

- **In-person (trade shows, events, door-to-door):** Approach prospects directly. Keep the opening to 30 seconds — introduce yourself, state what you’re offering, and ask if they have 5 minutes.
- **Phone/video:** Send a brief outreach message explaining why you want to talk (not a pitch — a request for a conversation). Aim for a 15–30 minute call.
- **Email pitch (B2B):** If you can’t get a live conversation, a written pitch with a clear ask (“reply to schedule a demo” or “reply with your shipping address to receive a sample”) can work, but you lose the qualitative signal from hearing objections live.

TOOLS

Any presentation tool (Pitch, Canva Presentations, etc.)
— build the pitch deck

[Gong](#)

[Otter.ai](#)

2. Run the pitch consistently.

Use the same pitch for at least 10 conversations before changing anything. If you tweak the pitch after every rejection, you'll never know which version worked. Follow your prepared structure, but let the conversation flow naturally. Key rules:

- **Listen more than you talk.** The prospect's reactions are the data. If you're talking for more than 60% of the conversation, you're pitching, not learning.
- **Don't argue with objections.** When a prospect objects, probe deeper: "Can you tell me more about that?" "What would need to be different?" "Is that a dealbreaker or a concern?" The objection is the most valuable part of the conversation. For timing objections, "When would you need this by?" is a particularly effective probe — it establishes urgency and reveals whether the problem is real right now.
- **Ask for the sale.** You must actually ask. "Does this sound interesting?" is not a close — "Would you like to pre-order today?" is. If you can't bring yourself to ask, you're running a customer interview, not a sales pitch test.
- **If the product doesn't exist yet:** Disclose this before taking payment, but after gauging interest. "We're building this now — would you put down a deposit to reserve a spot?" is honest and still tests commitment.

3. Record your data immediately after.

Fill in your tracking sheet within 5 minutes of ending the conversation. Memory degrades fast. If you wait until the end of the day, you'll remember the outcome but not the specific objections or language that mattered.

Analysis

✦ AI Prompt

1. Calculate your close rate.

Close rate = closed deals ÷ total pitches × 100. This is your primary metric.

2. Compare against your pre-set threshold.

- **Above threshold:** Your value proposition sells. Move to building the product or scaling the sales channel.
- **Below threshold but with strong qualitative signals:** If prospects are excited but not buying, the problem may be price, timing, or trust — not the value proposition itself. Test a different price or offer structure.
- **Zero closes after 10+ pitches:** The value proposition isn't landing with this audience. Revisit your pitch, your target audience, or the product concept.

3. Map your objections.

Group all recorded objections into categories (price, timing, trust, need, competition). Look for the pattern:

- If 70% of objections are about price → your value-to-price ratio is off. Either the price is too high or the perceived value is too low.
- If objections are about timing (“not right now”) → the problem may not be urgent enough for this audience.
- If objections are about trust (“I’ve never heard of you”) → you may need social proof, a pilot program, or a risk-free trial before you can sell.
- If objections vary widely with no pattern → your audience is too broad. Narrow your target.

4. Compare pitch language to reactions.

Which phrases made prospects lean in? Which made them go quiet? If you used an AI transcription tool, ask an AI to identify the moments in each conversation where the prospect’s engagement shifted — this reveals which parts of your value proposition resonate and which fall flat.

5. For small samples (under 10 pitches): Treat quantitative results as directional only. Focus on qualitative patterns: Are objections consistent? Did anyone get excited? One enthusiastic prospect who asks follow-up questions unprompted is a stronger signal than a 20% close rate from 5 polite conversations.

Biases & Tips

Anchoring on one sale

One sale does not validate a business model. It proves the product can be sold once. You need a pattern across multiple conversations to draw conclusions.

Curse of knowledge

You know your product better than your prospects. If you overwhelm them with features and jargon, they won’t buy — not because the product is wrong, but because they don’t understand it. If a prospect looks confused, stop and ask: “Am I making sense?”

Politeness bias

Prospects may say “that sounds interesting” or “I’ll think about it” to avoid an awkward rejection. Only a concrete commitment (payment, signed agreement, scheduled follow-up with a decision-maker) counts as a close.

Founder charisma bias

A charismatic founder can sell almost anything once. If only you can close deals, test whether someone else on your team can pitch it

too. A product that only sells through one person's personality isn't scalable.

Channel bias

A 30% close rate from warm referrals does not predict performance from cold outreach. Track your close rate by channel and don't generalize from your most comfortable channel to all channels.

Planning bias

First-time sellers consistently underestimate how many pitches it takes to get a sale. Plan for twice as many conversations as you think you'll need.

Automation substitution trap

AI-booked meetings generate volume but strip the qualitative signal — live objections, hesitation in voice, and body language — that is the whole point of this method.

Next Steps

- If close rates exceed your threshold, proceed to build the product.
- Analyze the most common objections to refine your positioning and messaging.
- Segment results by prospect type to identify your most receptive segment.
- Use successful pitch language in your [Landing Page Test](#) and marketing copy.
- Use a [Pre-Sales Test](#) to convert verbal commitments from your pitch into actual payment.
- Use a [Comprehension Test](#) to verify that your written messaging is as clear as your verbal delivery.

Case Studies

Close.io: Sales-as-a-service before the product existed

Steli Efti's team at ElasticSales sold outsourced sales engagements before Close.io existed, using those real customer conversations to identify pain points the CRM eventually solved.

[Read more](#)

Further reading

-  Steve Blank — Customer Development
-  Neil Rackham — SPIN Selling
-  Oren Klaff — Pitch Anything

- ☰ Chet Holmes — The Ultimate Sales Machine
- ☰ Eric Ries — The Lean Startup

4.4.16 Video Test

In Brief

A video smoke test is a short explainer or trailer video used to present a product concept to an early-adopter audience and measure their reaction. You share the video on social media or targeted channels, then track watch-time, sharing behavior, and clicks on a call-to-action. The output is quantitative data on audience excitement and virality — how many people watched, how many shared, and how many took the next step. It works especially well for products whose value is easier to show than to describe in text.

COMMON USE CASE

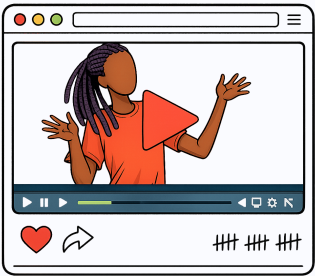
You have a product concept that is hard to explain in text alone. You create a short video showing how the product would work, share it with your target audience — whether on social media, in a targeted email, or in relevant communities — and measure how many people watch, engage, and take the next step.

HELPS ANSWER

- Does the audience understand the product messaging?
- Do people find the value proposition compelling enough to watch the whole video?
- Is the product interesting enough that viewers want to share it with others?
- Which distribution channels generate the most engagement?

Description

A video smoke test puts a short explainer or trailer in front of an early-adopter audience and measures whether they watch it, share it, and click the call-to-action



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~4 days–2.5 weeks

COST  \$30–\$600

OTHER NAMES
Explainer Video · Demo Video
Test

(the button or link that asks the viewer to act). The commitments it asks for — watch-time, sharing, and clicks on a call-to-action — are engagement signals on a richer story than a static page can carry, which places it a rung above email-capture tests on the [Value Proposition Test](#) family's commitment ladder while still stopping short of money or time.

The video does not require a working product behind it. This makes the method useful when the concept is new or hard to picture, where a short demonstration communicates the idea faster than a written description. What you are testing is how an early-adopter audience reacts to the promise: whether they watch, whether they share, and whether they take the next step.

A video for a smoke test should be understandable to someone with no background in the product's category. In technical fields it is easy to make something that excites specialists but baffles a general audience, so test the explanation for comprehension. Aim for empathy with the viewer, a single clear insight, and a plain answer to why the product matters. A [Comprehension Test](#) can confirm the message lands before you spend on promotion.

A strong video also gives viewers a reason to share it — to look informed, or to help others with the same problem. Sharing is measured with the share rate (the percentage of viewers who pass the video on) and, where the platform exposes it, the referral coefficient (how many new viewers each viewer brings in). The [Broken Promise Test](#) covers how to measure organic spread in more depth; social platforms like Facebook and Reddit give you a baseline for how often the target audience shares.

Promotion needs its own planning, and choosing where to post is part of the experiment, not a detail to settle afterward. The same approach is built into crowdfunding platforms such as Kickstarter and Indiegogo, where a concept video carries the campaign.

How to

Prep

[AI Prompt](#)

1. Define your hypothesis and success metric.

Decide what the test is supposed to prove before you write a single line of script. Pick a primary metric — the sign-up rate on a landing-page call-to-action, the share rate, the watch-through rate, or the referral coefficient (how many new viewers each viewer

brings in) — and decide in advance what number counts as a “go” vs. “no-go.” Otherwise it is too easy to rationalize whatever result you get.

2. Choose the format.

Match the format to the product and the budget:

- **Animated explainer** — best when the product doesn’t physically exist or the value is conceptual.
- **Live action** — best when emotion, lifestyle, or human use-case is the value.
- **Screencast** — best for software where the UI itself is the demo.
- **Mock trailer** — best when you want to convey aspiration / brand feel before production.

3. Draft script, storyboard, and key shots.

Write a 60–90 second script ending in a clear call-to-action, then storyboard it as a paper “comic book” so the narrative beats are visible before you spend money. Settle the story before you commit to a shoot or a render.

4. Set up the landing page that captures the CTA.

The video on its own measures attention; the landing page measures intent. Wire up the call-to-action (signup, waitlist, pre-order) and the tracking before you launch, not after — event tracking, tagged links that record which video and channel each visitor came from, and the platform’s own video analytics. AI video tools make it cheap to produce two or three variants testing different value propositions or emotional hooks, but each variant still needs its own tracked call-to-action so you know which one drove the result.

5. Pre-register success thresholds.

Write the specific numbers down before launch: a signup-rate threshold (for example, 5% of qualified viewers), a share-rate threshold (for example, 5%), and a call-to-action click rate (for example, 2% from cold traffic — viewers who have had no prior contact with you). Without written thresholds, you will reinterpret the results to fit whatever you hoped for.

Execution

✦ AI Prompt

1. Produce the video.

Following the storyboard, capture the footage or render the animation, edit for pacing, and add music that supports the emotional tone without overpowering narration. Keep the production simple enough that the concept — not the cinematography — is what's being tested.

TOOLS

[Synthesia](#)

[HeyGen](#)

[Descript](#)

[Loom](#)

2. Distribute to the right channels.

Build a marketing plan around channels that:

- Have many people in your target market.
- Are quite active.
- Allow sharing of the video — either embedded on a landing page or hosted directly on the platform (e.g., native Facebook or YouTube).

Channel testing is part of the experiment. Which Facebook groups? Which subreddits? Where should you guest post? Which influencers, if any, can you reach? Time the post to the news cycle on each platform.

3. Engage the audience and capture data.

Reply to comments, watch which thumbnails and titles draw clicks, and make sure every CTA path is tracked. The video itself is one half of the test — the audience reaction (shares, comments, signups) is the other half.

Analysis

✦ AI Prompt

1. Choose your primary metric based on your distribution channel.

- **Social/viral distribution (B2C):** Watch-through rate (what % watched to the end?) and share rate matter most. View count alone is a vanity metric unless the viewers are in your target audience.
- **Targeted distribution (B2B):** Reply rate, meeting requests, or CTA clicks matter most. 50 views from qualified prospects who request a demo is a stronger signal than 50,000 views from a general audience.
- **Landing page embed:** CTA conversion rate (what % of viewers clicked signup/buy?) is your primary metric.

2. Compare against your pre-set threshold.

If you didn't set one, common benchmarks:

- Watch-through rate above 50% means the content is compelling.
- Share rate above 5% indicates genuine word-of-mouth potential.
- CTA click rate above 2% from cold viewers is strong.

3. Check where viewers drop off.

Most video platforms show retention curves. If viewers leave in the first 10 seconds, your hook is weak. If they leave before the CTA, you're not connecting the value proposition to the ask.

4. Compare against a landing-page-only control, if you have one.

If you ran a [Landing Page Test](#) on the same value proposition without a video, compare CTA-conversion lift attributable to the video. A video that costs 10x the landing page but lifts conversion by 5% is a worse experiment than one that lifts it by 50%.

Biases & Tips

Production value trap

A polished video can generate positive responses because it looks professional, not because the value proposition is strong. Keep production simple enough that the concept, not the cinematography, is being tested.

Platform algorithm bias

Social platforms optimize for engagement, not for your target audience. High view counts from algorithmic distribution may not represent real demand from people who would buy.

Skipped-message risk

Cheap AI production makes it easy to generate a video before you have settled the message. If you cannot state the single insight, who it is for, and the answer to "why does this exist?" before you start, the video will test execution, not value. Settle the message first.

AI-detection fatigue

Audiences are increasingly able to detect AI-generated presenters and synthetic voices. For some segments this reduces perceived authenticity and virality. Test with your actual target audience before scaling.

Next Steps

- If engagement is strong, pair the video with a [Landing Page Test](#) to measure end-to-end signup conversion on the same value proposition.
- If signups convert well, move to a [Pre-Sales Test](#) to turn interest into purchase commitments before you build.

- Run an [A/B Testing](#) experiment on two or three thumbnail and title variants to raise click-through.
- Use [Discovery Interviews](#) with viewers who converted to learn what resonated.

Case Studies

Pebble Core: Alexa concept video

A concept video pairing Pebble Core with Amazon Alexa, used to gauge demand for the device before manufacturing.

[Read more](#)

The 4-Hour Chef: Cinematic book trailer

Tim Ferriss's cinematic trailer for The 4-Hour Chef, used to build interest in the book before launch.

[Read more](#)

Dropbox: MVP explainer video retrospective

Detailed retrospective of Drew Houston's three-minute explainer video, which drove the Dropbox beta waitlist from 5,000 to 75,000 signups in one day with no production code behind it.

[Read more](#)

Dollar Shave Club: Explainer video crashed the site

On March 6, 2012, founder Michael Dubin published a comedic explainer video; within 48 hours about 12,000 people had subscribed. Unilever acquired the company in July 2016 for a reported \$1 billion.

[Read more](#)

Purple Mattress: Goldilocks Raw Egg Test video

Purple partnered with Harmon Brothers on the Goldilocks Raw Egg Test video, which went viral and drove demand that took Purple from Kickstarter to a public listing within three years.

[Read more](#)

Further reading

 [Book site](#)

- ☰ Eric Ries, *The Lean Startup* (Crown Business, 2011), ch. 6 — foundational framing of the video MVP and the Dropbox example.
- ☰ Drew Houston — “Dropbox Startup Lessons Learned” (Silconf 2010, slide 13)
- ☰ TechCrunch — “How DropBox Started As A Minimal Viable Product” (Leena Rao, 2011)
- ☰ Tim Ferriss — How to Create a Viral Book Trailer (or Get 1,000,000 Views for Almost Anything)
- ☰ VideoPixie — Best Kickstarter Videos of 2014
- ☰ Exline Labs — 10 MVP Examples From Real Startups

PART 5

Generative Product Research



*“Research is formalized curiosity.
It is poking and prying with a purpose.”*

—Zora Neale Hurston

CONTENTS

■ Analog/Digital

■ Ask an Expert

■ Competitive Analysis

■ Competitor Usability Testing

■ Concierge Test

■ Demo Pitch

■ Dogfooding

■ Picnic in the Graveyard

■ Pocket Test

■ Prioritization

■ Solution Interview

■ Storyboard the Solution

5.1 Analog/Digital

In Brief

Deliver a physical or analog version of your product — a paper form, a spreadsheet, a printed guide, or a simple physical model — in place of the software you plan to build. You provide the analog form to real participants, observe how they work with it, and collect feedback on what works and what does not. The output is hands-on learning about which parts of the workflow deliver value, which can be simplified, and whether a digital version is worth building at all.

COMMON USE CASE

You plan to build software that automates something your customers currently do manually. Before investing in development, you deliver the service by hand to a few customers so you can learn exactly which steps matter, which can be simplified, and which can be removed entirely from the eventual digital version.

HELPS ANSWER

- What is the best way to deliver an information-based service when customers may not want or need an app?
- How does the current manual process actually work in practice?
- What are the biggest risks in the physical form of our product?
- Which parts of the analog workflow add real value versus unnecessary complexity?

Description

Analog/Digital is a method for testing a digital product’s value by delivering it first in a manual or physical (“analog”) form, before building the digital version. Many products combine a physical form with a digital component, where software adds value to an



AT A GLANCE

Qualitative

In-person

Remote

Observational



TIME  ~2 weeks–2 months

COST  \$30–\$300

OTHER NAMES
Analog Prototype · Manual MVP

underlying product — but not every step of an analog process gains value from being digitized, and the point of going digital is the value it creates, not the technology itself. Running the analog version first de-risks assumptions about the form a product or service should take before you commit to building software.

The defining move is that a non-digital artifact stands in for the software; it need not be run by hand. When a person performs the service end-to-end by hand instead of an artifact standing in for it, that is a [Concierge Test](#).

AI development tools now build digital versions quickly and cheaply, so the temptation to skip straight to a digital product is strong. The analog phase exists to learn what to build. Working with a physical artifact surfaces design problems that planning alone misses.

An analog prototype is the right fidelity for testing whether the workflow itself delivers value, before you invest in pixel-perfect software. Simple materials — paper, foam board, or an outsourced 3D print — are usually all the analog approach requires.

Suppose you want to build software that schedules a home-cleaning service — matching customers to cleaners, planning routes, and handling reschedules. Before writing any code, you run it by hand for a dozen real customers: a shared spreadsheet, a few text messages, and a printed schedule handed to each cleaner in the morning. Doing the job manually shows you which decisions genuinely need automating, which ones a person would rather keep judgment over, and whether customers value the service enough to pay — all before you have built anything. Only the steps that earn their keep in the manual version make it into the eventual software.

The same pattern fits other forms. For a connected hardware product, sell and refine the physical object before adding the companion app. For an enterprise workflow, map and streamline the process on paper before automating it. For an audience that does not want an app at all, deliver an information product as a printed guide or a live event first. In each case you stay flexible before committing to a large build, you watch how people actually handle the product rather than what they say in an interview, and you learn the economics of the manual form before paying to make it digital.

How to

Prep

✦ AI Prompt

1. **Identify your riskiest assumption.** Confirm that what you need to test is about the value proposition, workflow, or cost structure, not just a technical question. Analog-digital testing works when you need to learn whether the process itself delivers value, before investing in software.
2. **Define the learning goal.** Write down the one decision the analog run should inform. Common framings: “Does this workflow deliver enough value to justify automating it?” “Which steps add value and which are friction we should remove?” “What does the user actually do with the output?” Without a goal, you will collect impressions instead of evidence.
3. **Pick the analog format.** Match the format to the workflow you are testing. Paper forms and spreadsheets fit document-based workflows. Email, phone, and in-person handoffs fit marketplaces and matching. Printed guides, workshops, or live events fit information products. Physical mockups (foam board, 3D print, off-the-shelf hardware) fit connected and internet-of-things (IoT) devices.
4. **Recruit 5–10 participants.** Pick participants who would plausibly buy or use the digital version once it exists. Confirm they have a real task to complete using the analog version, not a hypothetical one.

Execution

TOOLS

Pen, paper, and a shared spreadsheet (e.g. Google Sheets)
— enough to run most manual workflows by hand; free

[Google Forms](#)

A local 3D-print or makerspace service
— physical mockups for connected-hardware concepts without tooling up

[Retool](#)

[Cursor](#)

[v0](#)

[Bolt](#)

1. **Run the analog version with participants.** Have them use the physical version to complete a real task. Observe where they struggle, what they skip, what they ask for, and what they find valuable. Record their reactions.
2. **Capture what happens, not just what they say.** Note the steps participants improvise, the steps they skip, and the moments they ask “is this all I need to do?” These are the signals that tell you which parts of the workflow matter.
3. **Decide what to digitize (and what not to).** Based on what you observed, identify which steps benefit from being digital (speed, scale, automation) and which work fine in analog form. Not everything needs to be software. Build only what the analog test proved was valuable.

Analysis

✦ AI Prompt

Digital is not good for its own sake. Make sure that you are adding useful features and benefits as you add to the product’s complexity.

1. **Score each step on value delivered vs. friction caused.** Which steps did participants complete eagerly? Which did they skip or complain about? High-value, high-friction steps are the strongest digitization candidates.
2. **Separate workflow value from interface value.** Sometimes the workflow is right and only the interface needs work. Sometimes the workflow itself is wrong and a polished interface would just hide the problem. Decide which one you are looking at before you scope the digital build.
3. **Compare the analog economics to the digital economics.** What does it cost to deliver the analog version per participant? What would the digital version cost? If the analog form already meets the participant’s need at a workable cost, a digital version may not be worth building yet.

⚠ Biases & Tips

Premature digitization bias

You default to adding software because it feels like progress. If the analog run meets the user need at a workable cost, shipping a digital version adds complexity without adding value, and the analog form is the product.

Confirmation bias

It is tempting to read the analog run as proof the digital product will work, noting only the moments that support building it. Decide your value-versus-friction scoring criteria before the run, and record the steps users skipped or improvised as carefully as the ones they completed.

MVP misread

A minimum viable product is not always a smaller or cheaper version of your final product. It is the smallest thing that produces the learning you need, which is sometimes a different product entirely.

Next Steps

- Build a [Concierge Test](#) or [Wizard of Oz](#) prototype to test the digital workflow manually before writing code.
- Run [Paper Prototyping](#) on the digital interface before committing to a build.
- Pilot the digital tool with 3–5 users via [Usability Testing](#) to verify the transition from analog preserves the experience, keeping the analog fallback available.

Case Studies

GE Healthcare: Hands-on infant monitoring trial

GE Healthcare validated a new infant monitoring concept in emerging markets by deploying trained nurses with portable tools to deliver the workflow by hand, learning what to build into the eventual device before committing to manufacturing.

[Read more](#)

Artivist: Paper-to-platform FinTech

Founder James Waldinger digitized a private-investment process the source describes as “still back in the stone age,” replacing paper-based documentation flows with a unified digital platform that let individuals and advisors access private equity and hedge funds.

[Read more](#)

Agricultural drones: MVP that learns, not a cheap product

Steve Blank uses precision-farming drones as the canonical example of an MVP that is a different product from the eventual one, designed to produce learning rather than to be a smaller version of the final offering.

[Read more](#)

Further reading

-  Mark Bonchek — How to Create an Exponential Mindset (Harvard Business Review, 2016)
-  Steve Blank — An MVP Is Not a Cheaper Product, It’s About Smart Learning (steveblank.com, 2013)

- ☰ Tim Brown — Change by Design (Harper Business, 2009)
- ☰ Bill Buxton — Sketching User Experiences (Morgan Kaufmann, 2007)
- ☰ John Mullins & Randy Komisar — Getting to Plan B (Harvard Business Press, 2009)

5.2 Ask an Expert

In Brief

Consult someone with deep domain experience to uncover blind spots, identify industry-specific risks, and generate options you had not considered. You interview the expert, collect their recommendations, and use those insights to formulate testable hypotheses. The output is direction-setting, not validation: expert opinions still need to be tested against real customer data.

COMMON USE CASE

You are entering an unfamiliar industry and need to understand the regulatory landscape, common pitfalls, and standard practices before designing your product. Rather than spending weeks reading, you want to have a focused conversation with someone who has years of direct experience in the space.

HELPS ANSWER

- What is the standard approach or common assumption in this market?
- Am I missing an important part of the overall picture?
- What mistakes do newcomers typically make in this industry?
- Which areas should I investigate further before building?

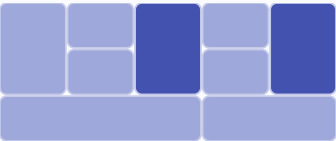
Description

Asking an expert means consulting someone with deep, hard-won experience in your problem domain to shortcut your own learning. Experts are most useful for problems that are complicated rather than simple or chaotic: problems with many factors and components, where the relationships are predictable but not obvious. These



AT A GLANCE

- Qualitative
- In-person
- Remote
- Attitudinal



TIME ~3 days–3 weeks

COST \$0–\$500

OTHER NAMES
Expert Stakeholder Interview ·
Sales Force Feedback · Partner &
Supplier Interview

problems have many “known unknowns” that cause-and-effect analysis can resolve into a range of workable answers. Founders need to execute quickly, and drawing on someone else’s hard-won knowledge is a fast way to get there.

Some industries require significant expertise to compete. Finance is one example: it is highly regulated and depends on detailed models for valuation, risk assessment, and accounting, all of which are context-sensitive. You can learn some of this from books, but seeing how it plays out in a competitive environment adds insight that reading does not. A founder entering a regulated or technically deep market can use an expert to surface options they would not have considered on their own.

Outside of regulated industries, an expert can still place your situation in the context of many companies that faced a similar problem. If you are planning to test a paid acquisition channel, for instance, an experienced channel specialist can flag the pitfalls before you spend the budget. An LLM can optionally speed up the prelude here — orienting you in the domain’s terminology and common failure modes so you arrive with sharper questions and find the right expert faster — but it is not a substitute for one. It confidently presents outdated information, misses nuances that only come from lived experience, and cannot evaluate your competitive context. A working finance expert knows which regulations are actually enforced versus technically on the books, which investors are currently active, and which approaches have been quietly tried and abandoned; an LLM only knows what has been published.

Use experts to generate options, not to settle them. Asking a third party to evaluate a decision you have already framed tends to backfire: the expert views the situation through assumptions that may not hold up in your data. Expert input sets direction; customer behavior decides.

How to

Prep

✦ AI Prompt

1. Find an expert through free channels.

- **Web search:** Enter specific terms or questions and look at who is publishing useful content on the topic.
- **Professional networks:** Search by keyword, current title, or company to locate working specialists.
- **Local events and groups:** Reach out to local experts who present or organize on your topic.

- **Academic publication databases:** Find researchers who specialize in a narrow topic and have published in it.
 - **Public Q&A sites, blogs, newsletters, and video channels:** Long-form public writing or video reveals depth and point of view before you commit to a meeting.
 - **Your own sales team:** Salespeople talk to customers daily and accumulate insight into objections, feature requests, the competitive landscape, and lost-deal reasons. Treat this as secondary market research, one degree removed from a direct customer interview.
 - **Your own customer support team:** Support staff hear directly about pain points, frustrations, and unmet needs. Their read on recurring themes can surface problems that don't show up in the numbers. For quantitative analysis of support ticket data, see [Customer Support Analysis](#).
 - **Partners and suppliers:** They can verify the feasibility of your approach, reveal hidden dependencies, surface logistics and scalability constraints, and provide cost estimates before you invest fully.
2. **Consider paid sources for vetted access.** Options include pay-per-minute expert-call platforms with named experts, expert-network platforms ranging from enterprise to founder-priced, and industry-specific forums and trade associations.
 3. **Reach out and schedule the session.** Arrange a call, video meeting, or in-person session. Be specific about how much time you need and the topic you want to cover; vague requests get vague responses.
 4. **Prepare a written brief before the meeting.** Include:
 - **Questions** you want answered, ranked by priority. Lead with the highest-stakes one in case the call runs short.
 - **Areas** where you want critical feedback on a draft, decision, or assumption.
 - **Topics** you want to brainstorm, where you're looking for options rather than answers.

Before you spend an expert's hour, use an LLM to sharpen the brief: ask it for terminology you should know, common failure modes in the domain, and the questions a working expert would expect a serious founder to ask. Drafting your own rough model of how the domain works first makes the expert's corrections far more valuable, because you arrive with a position to correct rather than a blank page.

Execution

1. **Open with the highest-priority question.** Conduct the meeting and lead with the question that matters most, in case the call runs short. Take notes verbatim where you can; paraphrase loses nuance.
2. **Capture claims and their basis.** Record key ideas, recommendations, and any sources, frameworks, or numbers the expert cites. Distinguish what the expert knows from first-hand experience from what they have read or heard.
3. **Ask for a referral.** Request an introduction to another expert in the same or an adjacent area. The second conversation is where you start to triangulate.
4. **Convert learnings into a testable hypothesis.** Translate what you heard into a falsifiable hypothesis you can test with customers, not with another expert.

TOOLS

[Clarity.fm](#)

[GrowthMentor](#)

[GLG](#)

[AlphaSights](#)

Analysis

✦ AI Prompt

1. **Weight experts by the basis of their knowledge.** Give the most weight to experts who either have personal experience (success or failure) in the topic area, or have gained significant insight through academic or journalistic research. Discount input that rests on neither.
2. **Discount advice for context drift.** All advice is context-dependent. Even an expert who succeeded before is describing a situation and competitive landscape that have since changed, so treat their conclusions as dated until you confirm they still apply.
3. **Separate first-hand from second-hand claims.** Sort what the expert directly experienced from what they read or heard. First-hand claims carry more weight, but both can be wrong for your specific case.
4. **Reduce the input to hypotheses, not decisions.** End with two or three falsifiable hypotheses you can test with customers. If the experts disagreed, keep the disagreement visible rather than averaging it into a false consensus.

⚠ Biases & Tips

Misaligned incentives

An expert whose fees, equity, or reputation depend on a particular outcome will shade their advice accordingly — confirm alignment before acting on recommendations.

Opinion-as-fact

Experts in opinionated domains often present conclusions without surfacing the evidence behind them — ask for the data or case experience that underlies each recommendation so you can judge its relevance to your context.

Authority bias

People tend to defer to an expert's recommendation even when it is demonstrably wrong, treating the title as a substitute for judgment. Treat expert input as one data point, not a verdict, and ask yourself whether you would accept the same claim from a non-expert.

Consultation paralysis

Consulting a second or third expert in sequence can become a substitute for testing — each new conversation should either sharpen a hypothesis or be replaced by a customer experiment.

Next Steps

- Run [Discovery Interviews](#) to test whether the expert's assumptions match what real customers experience.
- Use a [Competitive Analysis](#) to check expert claims against the actual market landscape.
- Cross-reference the expert's input with [Secondary Market Research](#) to confirm the figures and trends they cited.
- If the expert flagged a risk in your proposed approach, pressure-test the alternative with a [Solution Interview](#) before committing engineering effort.

Case Studies

Further reading

- ☰ Expert Financial Advice Neurobiologically “Offloads” Financial Decision-Making under Risk
- ☰ Find the Right Expert for Any Problem
- ☰ Got a wicked problem? First, tell me how you make toast
- ☰ Croll, A., & Yoskovitz, B. (2013). *Lean Analytics*. O'Reilly. ISBN 978-1449335670. — Frames expert input as a way to identify the right metric to test, not the answer itself.
- ☰ Alvarez, C. (2014). *Lean Customer Development*. O'Reilly. ISBN 978-1449356354. — Treats experts as one input among many; emphasizes testing expert claims against customer behavior.

5.3 Competitive Analysis

In Brief

Competitive analysis compares features, pricing, channels, and positioning **in relation to a specific target segment**. The same feature is a strength against one segment and a non-issue against another, so the comparison is always relative to segment needs. The output is a shortlist of competitors, a structured profile of each, and a synthesis — a feature-by-segment matrix, a positioning gap analysis, and a focused SWOT (a strengths, weaknesses, opportunities, and threats summary) — that points to a defensible opening.

COMMON USE CASE

You have a hypothesis about who else is in this space, but you need a structured map before you commit to a differentiation story or a pricing position. You are not yet ready to interview customers about each rival; you want to see the field, name the two or three you actually compete with for your target segment, and decide where to position yourself.

HELPS ANSWER

- Which competitors are our customers most likely comparing us to?
- How are competitors solving similar customer problems, and where do they fall short for our target segment?
- What form should our product take to stand out for the segment we care about?
- How can we differentiate our offering and positioning?
- What revenue models are competitors using?
- Which channels and acquisition motions do competitors lean on, and which are crowded versus open?
- Which partnerships and integrations do competitors rely on, and which can we replicate or avoid?

	✓	✗	✓
	✓	✓	✗
	✓	✗	✗
	✗	✗	✗

AT A GLANCE

Qualitative

Remote

Observational

TIME

~1 hour–2 days

COST

\$0–\$50

OTHER NAMES

Market Landscape Analysis

Description

Competitive analysis is a structured comparison of the players already trying to solve the same job for the same kind of customer. The point is not to count competitors; it is to understand how each one is positioned for a target segment you care about, and to find a gap where your offering produces evidence of stronger fit than theirs.

Two principles run through the method.

The first is **relational comparison**. A feature is not a strength in isolation. SOC2 compliance is a strength against an enterprise buyer and irrelevant against a prosumer; a free tier is a strength against a price-sensitive solo user and a liability against a procurement-led buyer. Every cell in your matrix is scored against a named segment, never in the abstract.

The second is **citation-backed claims**. AI assembles competitive material faster than any human team, but it invents funding rounds, customer counts, and feature lists at a meaningful rate. Every numeric and named claim on a finished profile must trace to a verifiable URL. The verification discipline this demands is spelled out in Biases & Tips and built into the AI prompts below.

For market sizing (TAM / SAM / SOM, segment growth, willingness-to-pay benchmarks), use [Secondary Market Research](#). This page is about who is in the field and where the gaps are; that one is about how big the field is.

How to

Prep

✦ AI Prompt

1. **Define the target segment first, not the competitor list.** Write a one-paragraph description of the customer you are analyzing competitors against: who they are, what job they are trying to do, what they currently use, and what they care about most. Every later judgment — a feature being a strength, a price being a problem, a channel being saturated — is judged against this paragraph. Skip this step and you produce a generic map that scores every competitor against no one in particular.
2. **Pick which competitors are worth deep-research time.** Don't try to map "everyone in the space." Aim for 8–15 names: 3–5 direct competitors, 3–5 indirect competitors solving the same job differently, and 2–3 substitutes from adjacent markets. Cross-reference at least two of these candidate-pool sources before locking the shortlist:

- **Customer discovery interview transcripts** — whom did your interviewees name as alternatives, and what did they switch from?
- **Search trend analysis** — which brands rank for the pain queries your target segment types into a search engine?
- **Keyword competition** — who is bidding on your keywords in a keyword-intelligence tool?
- **Open-ended survey responses** — answers to “what tool or method do you currently use for X?” and “who else did you consider?”
- **App-store category browsing** — which apps appear in the categories your customer browses?
- **Social listening** — which names keep coming up in community forums, chat communities, and review threads your customer reads?

If a name shows up in only one source, treat it as a weak candidate. If it shows up in two or more, it earns a slot.

3. **Decide your dimensions.** Pick the 6–10 dimensions that matter most to *your* target segment, not a generic SaaS rubric. Examples: feature coverage, pricing model, time-to-value, integrations with the buyer’s existing stack, support quality, deployment options, compliance posture, channel mix, partner ecosystem. Drop dimensions that don’t map to a real concern of your segment — “supports SOC2” is dead weight if you sell to indie creators, and “free tier” is dead weight if you sell to procurement-led enterprises.
4. **Sketch the artifacts you will produce.** Pick the visualizations that will actually drive the decision in front of you, and stub them out before you collect data. Common artifacts:

- **Feature × segment matrix** — competitors on rows, dimensions on columns, scored against the target segment.
- **2×2 positioning grid** — pick the two dimensions your segment cares about most; plot competitors; look for empty quadrants.
- **Petal diagram** — for new categories, place yourself at the center and draw petals for each adjacent market your customers will switch from.
- **Pricing-tier comparison** — list price, packaging, what’s included at each tier, common discounting patterns.
- **Channel-mix breakdown** — paid search, content, partnerships, marketplaces, outbound, community — what share each competitor leans on.

You do not need every artifact. Pick the two or three that answer the actual decision you are about to make.

5. **Decide where verified primary sourcing matters.** Some claims drive decisions (pricing, funding round size, feature presence on a paid plan, named integrations); others are background. Tag each dimension as **decision-grade** (must be verified to a primary source) or **directional** (a published roundup or AI summary is fine). Decision-grade claims will go through fact-checking later; directional claims will not. Without this triage, fact-checking either takes forever or doesn't happen.

Execution

✦ AI Prompt

The verified shortlist from Prep is now the input to deep profiling. The goal here is not to write essays per competitor; it is to fill a fixed schema so that the synthesis stage has clean inputs.

1. **Use a fixed profile schema.** Each profile records the same fields in the same order. A reasonable default:
 - Company name, URL, year founded, HQ location.
 - Funding stage and total raised (from a funding database), most recent round.
 - Customer segments served (named, not “everyone”).
 - Top 6–10 features mapped to your dimension list.
 - Pricing model and published tier prices.
 - Channel mix (paid search, content, marketplaces, partnerships, outbound, community) — qualitative description plus traffic share if traffic-analytics data is available.
 - Integrations and named partners.
 - Sentiment summary from review sites and forums (positives, complaints, common deal-breakers).
 - Any recent product or strategy moves worth noting.
 - Inline [1] , [2] citation markers for every numeric and named claim.
 - A Sources list at the end of the profile resolving every marker to a URL.
2. **Collect each field from its canonical source.**
 - Funding, stage, founding year → funding databases, SEC filings, press releases → spreadsheet column.
 - Features and positioning copy → competitor's homepage hero, product pages, pricing page → feature matrix row.
 - Channel mix and GTM motion → traffic-analytics services (traffic sources), keyword-intelligence tools (keyword bids), ad-transparency libraries

- (creative), app-store presence, the competitor's own integrations / partners page → channel-mix table.
 - Customer segment and sentiment → software review sites, community forums, and app-store reviews → segment notes + sentiment column.
 - Pricing and revenue model → published pricing page; review threads mentioning negotiated discounts → revenue model column.
3. **AI does the first pass; humans verify decision-grade claims.** A capable LLM with web access can fill most of the profile, but the remainder — and any claim you tagged as decision-grade in Prep — must be confirmed by visiting the cited source yourself or by routing through a fact-check subagent. Pricing changes often; funding round sizes are routinely overstated in press releases; “X customers” claims rarely include the basis on which a customer was counted.
 4. **Flag anything gated.** Anything that requires a login, a paid database (full paid-database reports, paid analyst reports, gated review-site detail, sales-intelligence tools) gets flagged as **MANUAL** in the profile. AI cannot fetch behind authentication, and a confidently-stated number that the agent could not actually retrieve is the most common failure mode of AI-generated profiles.

Analysis

The point of analysis is to produce decisions, not to fill templates. Every artifact below should end in a sentence that names a choice you are now better equipped to make.

1. **Build a feature × segment matrix.** Score each competitor's coverage of each dimension against your target segment, not against a generic buyer. A “Yes” cell only counts if the feature meets the segment's bar (e.g. “supports SSO” against an enterprise buyer means SAML SSO on the plan they would buy, not OAuth on the free tier). Use a three-state scale — Strong / Partial / Absent — rather than a numeric score; numeric scores invite false precision.
2. **Run focused SWOTs on the two or three competitors you actually compete with.** A SWOT scoped to your target segment is useful; a SWOT scoped to “the market” becomes a list-making exercise. For each top competitor, write 3–5 entries per quadrant, and force every entry to end in a decision you would now make differently. “Strength: incumbent brand recognition” → so we lead with social proof in our positioning. “Weakness: reviews mention slow support” → so

✦ AI Prompt

TOOLS

[Google Search](#)
[Google News Alerts](#)
[Crunchbase](#)
[Perplexity AI](#)
[Reddit](#)
[Meta Ad Library](#)
[SimilarWeb](#)
[SpyFu](#)
[SEMrush](#)
[Crayon](#)
[Klue](#)

we lead with response-time SLA in our messaging. If an entry doesn't end in a decision, drop it.

3. **Add Porter's Five Forces only if you are entering a mature market.** If you are entering an established market with multiple incumbents and clear buyer/supplier relationships, a quick Five Forces pass on competitive rivalry, threat of new entrants, threat of substitutes, buyer power, and supplier power tells you which forces are doing the most to shape margins. In a new or emerging category, skip it; you do not yet have stable forces to map.
4. **Run positioning gap analysis.** Plot competitors on the 2x2 grid you sketched in Prep. Then ask the harder question: which empty quadrants represent real demand, and which are empty for a reason? Cross-reference your discovery sources from Prep — interview transcripts, search trends, survey responses. An empty quadrant with search volume and customer mentions is an opening; an empty quadrant with neither is usually empty because no one wants what would go there.
5. **Write the strategic insight.** A short write-up — 200–400 words — answering three questions in order:
 - **Who are your customers most likely to compare you to?** Two or three names, drawn from interview mentions and search behavior, not from the full shortlist.
 - **What do those competitors do well for your segment?** Be specific, segment-relative, and source-cited.
 - **What is the gap that is actually a gap?** A claim about an empty position that has demand evidence behind it, not just an empty cell on a grid.

This is the artifact you actually act on. The matrices and SWOTs exist to make this paragraph defensible.

Biases & Tips

Confirmation bias

Founders don't want to find competitors. Make the analysis exhaustive, repeated, and externally reviewed. If your map has no direct competitors, you have not looked hard enough.

AI hallucination risk

AI competitive research is fast but routinely fabricates funding rounds, customer counts, and feature claims, at a rate high enough that no figure can be trusted on sight. Every numeric and named claim must be confirmed against a primary source before it enters your analysis. Treat unverified AI output as a draft, not a finding.

Local-optima trap

If you only look at direct competitors, you optimize for incremental differentiation. Always include 2–3 substitutes from adjacent markets. Airbnb beat hotels by studying Craigslist, not Marriott; Figma beat Adobe by studying Google Docs as much as Sketch.

Static-snapshot bias

A competitor map is stale within 90 days in fast-moving markets. Schedule a re-run quarterly, and set automated news alerts (or a continuous competitive-intelligence monitor) on the top 2–3 names so material moves don't go unnoticed.

Absolute-feature scoring

Comparing features in isolation (rather than against a named segment) produces matrices that look rigorous and mislead consistently. The same feature is a strength for one segment and irrelevant for another. Always score relationally.

Rubber-stamping AI synthesis

When an AI produces a finished SWOT, the human reader tends to accept the framing and only edit at the wording level. Force yourself to challenge at least one claim per quadrant against the underlying profile evidence; if you can't trace it, drop it.

Empty-map fallacy

A map with few or no competitors is more often a sign of insufficient research than a sign you are alone — look harder before concluding you have the space to yourself. Absence of visible competition is not confirmation of a gap; it is a prompt to expand sources.

Geographic myopia

Limiting the shortlist to your local market misses international competitors that often define the category your buyer eventually compares you to — and misses the substitute players most likely to cross into your market first. Default to a global lens unless your segment is demonstrably local-only.

Next Steps

- Validate the gap with [Discovery Interviews](#) — does the gap you found match what your target segment actually struggles with?
- Run [Competitor Usability Testing](#) on the top 2–3 competitors to surface the UX-level weaknesses your matrix will not catch.
- Test the differentiated positioning with a [Landing Page Test](#) before committing to it in launch messaging.
- For market sizing (TAM / SAM / SOM, segment growth, willingness-to-pay benchmarks), use [Secondary Market Research](#) for how big the field is.

- Set up continuous monitoring (Google Alerts at minimum, Crayon / Klue if budget allows) on the top 2–3 names, and run [Search Trend Analysis](#) on their brand and pain queries; re-run the synthesis quarterly.

Case Studies

Figma vs Adobe: Competing on usability

Figma's analysis of Adobe's design ecosystem identified weeks-long learning curves, desktop installs, and no real-time collaboration as exploitable gaps, and Figma positioned its browser-based, collaborative product directly against those weaknesses.

[Read more](#)

Airbnb vs Craigslist: Adjacent-market analysis

Airbnb studied Craigslist's short-term-rental listings, identified poor photos and impersonal interactions as weaknesses, and responded by cross-posting listings and dispatching professional photographers — a competitive-analysis win against an adjacent-market incumbent rather than a direct competitor.

[Read more](#)

Further reading

-  Steve Blank: A New Way to Look at Competitors
-  Michael Porter (1980) — Competitive Strategy: Techniques for Analyzing Industries and Competitors
-  Anthropic: How we built our multi-agent research system (2025)
-  Hallucination to Truth: A Review of Fact-Checking and Factuality Evaluation in Large Language Models (arXiv 2508.03860, 2025)
-  AI Hallucination Rate Benchmarks 2026
-  Klue: How to do competitive analysis with AI
-  Instigator Blog: Competitive Research 101 for Startups (archived)

5.4 Competitor Usability Testing

In Brief

Watch target participants complete tasks using a competitor’s product, and note where they struggle, what workarounds they invent, and which features they ignore. The output is a set of pain points, unmet needs, and unnecessary features that directly inform the design of your own product. See [Usability Testing](#) for testing your own product once you have one.

COMMON USE CASE

You are designing a new product and want to learn where people struggle with the existing alternatives before you commit to your own design. You recruit a few participants who match your target customer, watch them complete real tasks in a competitor’s product, and note every point of confusion, workaround, or complaint. The gaps you uncover tell you where your solution can differentiate.

HELPS ANSWER

- What is the minimum feature set needed to solve the problem?
- How important is design quality to users in this category?
- Where do people struggle most with existing products?
- Which competitor features are unnecessary or confusing?

Description

Competitor usability testing is a structured observation method where you watch target participants attempt real tasks in a competitor’s product, or a substitute good, and note where they get stuck. You are mining that product for ideas: watching where the existing alternative fails the people who use it tells you what your own product



AT A GLANCE

Qualitative

In-person

Remote

Observational

TIME

 ~3 days–2 weeks

COST

 \$0–\$300

OTHER NAMES

Competitor UX Test · Comparative Usability Test

should do better. The procedure is the same as testing your own product, but the purpose is generative rather than evaluative. You are not verifying that a product works and you have no product of your own to benchmark — the competitor's product is the raw material for your design.

The competitor does not have to be a direct rival. To generate ideas for a better U.S. tax experience, you could watch participants prepare taxes in Sweden or India. The results would not tell you whether the U.S. tax experience is good, but they may suggest whether to focus on the comprehensibility of the tax code, the submission process, or the rules themselves.

How to

Prep

★ AI Prompt

1. **Pick 2–3 competitor products or substitutes to test.** Choose the ones your target participants are most likely using today. Include at least one indirect substitute — a different type of product that solves the same problem in a different way. Keep the count to four or fewer; beyond that, the analysis cost grows faster than the insight does.
2. **Write 4–6 task scenarios.** Each task should represent a core workflow your participant does regularly. Use the same tasks across all competitors so you can compare. Frame each as a goal, not a click path: “You need to [accomplish goal]. Use [Competitor A] to do it.” Keep each scenario short, specific, and free of jargon the participant would not use. To find candidate usability issues worth turning into scenarios, summarize a competitor's app-store reviews, support forums, and social-media complaints — treat the output as a list of things to watch for, not a finished answer.
3. **Recruit 5 participants per competitor.** They should match your target customer profile. If a participant already uses one of the products, that is fine — you will learn where experienced users still struggle. Five per product is the point at which most usability problems on a single product surface.
4. **Decide how you will run the sessions.** Choose between moderated sessions (you watch live and ask questions) and unmoderated ones (the participant records themselves alone), and between in-person and remote, then pick the tool to match (see Tools below). For unmoderated remote tests, pre-record a short framing message; for moderated tests, write a one-paragraph intro. Keep the framing identical across competitors so participants approach each product the same way.

5. **Write your consent and recording language, and your task prompts.** Get explicit permission to record. Make clear you are testing the products, not the participant, and that they can quit a task or the whole session at any time without explanation. Decide the questions you will ask after each task and at the end of the session now, so every session runs the same way.

Execution

1. **Run the test.** For each participant and each product:
 - Give them the task. Don't explain how the product works.
 - Watch them attempt the task. Note where they hesitate, make errors, express frustration, or find a workaround.
 - Ask your prepared post-task and end-of-session questions, the same way every time.
2. **Record and compare.** For each product, track how often participants finished the task, how long it took, how many errors they made, and how satisfied they were. The gaps and frustrations you observe are your opportunity — they reveal what your product should do better.

TOOLS

In-person kit: the competitor's product on a real device, a notebook, and a screen recorder — the cheapest way to run a moderated session.

[Maze](#)

[UserTesting](#)

[Lyssna](#)

[Lookback](#)

For the full usability testing methodology, see [Usability Testing](#). The process is the same; the difference is that you are testing someone else's product to generate ideas, not evaluating your own.

Analysis

[AI Prompt](#)

1. **Treat results as generative, not evaluative.** A failed task on a competitor's product is a candidate problem to solve, not a verdict on the competitor. The ideas tend to be unstructured and piecemeal — you will need to integrate them into a coherent solution rather than ship them one-for-one.
2. **Cluster failures into opportunity areas.** Group every observed pain point, workaround, and unmet need across participants and products. The clusters with the most repeats across participants are your highest-confidence opportunities. A pain point that appeared once is an anecdote; one that appeared in four of five sessions across two competitors is a pattern.
3. **Validate before building.** Before committing engineering effort to any opportunity, test it with a generative product method such as a [Solution Interview](#) or [Concierge Test](#). Keep validating the problem for as

long as it is cheap to do so, and resist building a feature just because a competitor lacks it – confirm the problem is worth solving first.

Biases & Tips

Observer effect

Participants behave differently when they know they are being watched. Frame the session as testing the product, not the person, and resist coaching mid-task.

Confirmation bias

You may write tasks or follow-up questions in a way that confirms what you already believe about a competitor's weakness. Have a second person review your task list and questions before you run sessions.

Selection bias

Recruiting whoever is easiest to reach, or only people who already dislike the competitor, distorts what you see. Recruit against your target customer profile and include both current users of the product and newcomers to it.

Curse of knowledge

You know the problem domain deeply, so failures that feel obvious to you may be the participant's real struggle. Resist explaining how the product is "supposed" to work; the confusion is the data.

AI analysis substitution

A model-generated summary of session recordings is not a substitute for watching the sessions. Watch at least the first three sessions yourself before relying on any tooling to surface the rest.

Next Steps

- Map the most common usability failures to specific feature opportunities in your product.
- Use competitor pain points in your marketing messaging ("Unlike X, we...").
- Test your own product or prototype with the same tasks to benchmark against competitors.
- Prioritize building features that address the highest-frustration competitor workflows.
- Use a [Solution Interview](#) to propose your alternative solution and gauge whether it addresses the competitor pain points you discovered.
- Run a [Usability Testing](#) on your own prototype using the same tasks to verify your design actually outperforms the competition.

Case Studies

American Airlines: Comparative QXscore testing

American Airlines worked with UserTesting to score four customer journeys (booking, flight changes, check-in, AAdvantage) against the same flows on competing airlines, surfacing friction like below-the-fold multi-city search; redesigns produced a 37% lift in task completion and a 15% average QXscore increase.

[Read more](#)

Loop11: Ten-airline banned-baggage benchmark

Loop11 recruited 1,000 participants (100 per site) to attempt the same banned-baggage lookup task across ten airline websites; British Airways completed at 71% in 87 seconds, while Malaysia Airlines completed at 31% and Virgin Atlantic took 199 seconds — more than twice as long as BA.

[Read more](#)

Further reading

-  Jakob Nielsen, Usability Engineering. Academic Press, 1993. ISBN 978-0125184052. The foundational text for the small-sample testing convention used here.
-  “Competitive Usability Evaluations: Definition”
-  “Tools for Unmoderated Usability Testing”
-  Steve Krug, Don’t Make Me Think, Revisited (3rd ed.). New Riders, 2014. ISBN 978-0321965516. Practical guidance on writing usability test tasks that produce real signal.
-  Tomer Sharon, Validating Product Ideas: Through Lean User Research. Rosenfeld Media, 2016. ISBN 978-1933820293. Lean research framing for treating session output as hypotheses to validate.
-  Dan Olsen, The Lean Product Playbook. Wiley, 2015. ISBN 978-1118960875. Problem-space-before-solution-space discipline for opportunities surfaced from competitor sessions.
-  Daniël De Wit: Don’t feel bad, user test your competitors

5.5 Concierge Test

In Brief

In a concierge test you deliver your product's value by hand, as a personal service, to a small number of real customers. The defining move is that a person — usually you — performs every step end-to-end, solving the problem directly, and gathers feedback on what works, what does not, and which features are essential. The output is qualitative insight into the minimum feature set you need before you invest in building an automated product.

COMMON USE CASE

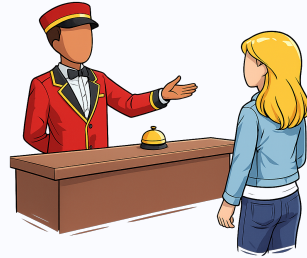
You believe your product idea will save customers time or effort, but you haven't tested the actual workflow yet. Instead of building anything, you personally deliver the service by hand for a handful of customers, learning exactly which steps matter, what they actually need, and whether they would pay — all before you write any code.

HELPS ANSWER

- Does this solution actually solve a real customer problem?
- How can the problem be solved in practice?
- What is the minimum feature set needed to deliver the solution?
- What are the biggest obstacles to delivering this solution?

Description

In a concierge test you deliver the value proposition as a service. Like a hotel concierge, the work is highly customized and customer-facing: you perform the tasks by hand, usually for only a few customers, because it is not cost-efficient to scale. The close contact gives you detailed feedback from the segment you care about, and



AT A GLANCE

Qualitative

In-person

Remote

Behavioral

Observational



TIME



~2 weeks–3 months

COST



\$0–\$300

OTHER NAMES

Concierge · Concierge MVP

because nothing is built, you can adjust the service from one customer to the next at very low cost.

When a physical or analog artifact stands in for the software instead of a person performing the service, that is an [Analog/Digital](#) test.

Start from a clear value proposition. Where a [Solution Interview](#) asks customers whether a proposed solution matches their needs, a concierge test puts the solution in their hands and watches what happens. Design the service around the steps a customer would take with your eventual product, and build it around the leap-of-faith assumptions you most need to test.

Deliver the service manually and personally, starting with a small batch so you are not overwhelmed. You do not need any code or automation at this stage. While delivering, keep collecting feedback and adjust the service as you go. Over time you learn what customers expect and what they actually value, and you can gradually automate the parts that work. Stop expanding the manual service once it stops producing new major insights — a concierge test is a phase, not a permanent way to operate.

A concierge test is distinct from a [Wizard of Oz](#) test: in a concierge test the customer knows the service is delivered by a person, whereas in Wizard of Oz the manual work is hidden behind an interface that appears automated.

How to

Prep

[AI Prompt](#)

1. **Write down the value proposition you are testing.** Be specific: “We help [customer] do [task] so they get [outcome].” This is the claim your concierge test will validate or invalidate. If you cannot finish that sentence, you are not ready to run a concierge test — you are still in problem discovery, and a [Discovery Interview](#) is the better next step.
2. **Pick the segment.** Concierge tests do not scale, so the segment you pick determines what you can learn. Choose a tightly-scoped early-adopter group: people who already feel the pain acutely, can describe their current workaround, and are reachable without a marketing budget. The wrong segment produces friendly-sounding feedback that does not generalize.
3. **Map the manual flow.** List every step the customer will go through — the same steps they would go through with your eventual product. For each step, decide how you will deliver it manually: email, spreadsheet, phone call, in-person

meeting, or a shared document. The flow you draft here is the spec for what gets automated later, so write it as if a junior teammate would have to follow it.

4. **Identify which steps to automate first.** Tag each step on the manual flow as **must stay human** (the part where the value proposition lives), **automate first** (high-volume, low-judgment, easily templated), or **automate later** (judgment-heavy but eventually scriptable). The automate-first list is what you watch the test against.
5. **Recruit 3–5 early adopters.** Find people who already have the problem and are willing to try your service. Ask for payment (even a small amount) — paying customers give more honest feedback than non-paying ones.
6. **Set expectations.** Tell customers this is an early, hands-on version of the service. You will be personally involved, things may be rough, and you want their honest feedback.

Execution

1. **Deliver the service manually.** Perform every task yourself. Track how long each step takes, where customers get confused, what they ask for that you didn't anticipate, and what they don't use.
2. **Collect feedback after each interaction.** Ask: "What was most valuable? What was frustrating? Would you pay for this again?" Record responses immediately.
3. **Adjust and repeat.** After each batch of interactions, update your service based on what you learned. Change pricing, add steps, remove steps. The concierge test is iterative — you are designing the product through direct delivery.
4. **Know when to stop.** When you stop hearing new feedback and can predict what customers will say, you've learned what you can from manual delivery. Begin automating the parts that work.

TOOLS

[Calendly](#)

[Airtable](#)

[Tally](#)

[Intercom](#)

Analysis

✦ AI Prompt

1. **Aggregate across customers.** The data is mostly qualitative. Pull together notes from every customer, grouped by each part of the service — time per step, where they got confused, what they asked for, what they ignored — so you can see patterns rather than one-off reactions.
2. **Identify the high-value and the automatable steps.** Separate the steps where the value proposition lives (the part customers are paying for) from the high-volume, repeatable steps that are candidates to automate first. Look for steps that take the most time and recur most consistently across customers.

3. **Surface obstacles.** Note where the manual delivery broke down or surprised you. These are the implementation problems your eventual product will have to solve.
4. **Feed the insights back into the service.** Adjust pricing, steps, and scope based on what you learned, then keep delivering until the feedback stops producing new major insights.

⚠ Biases & Tips

Concierge halo (false-positive bias)

The manual delivery adds value of its own: trust, responsiveness, and a personal relationship. Founders can mistake positive feedback on the hand-delivered service for validation of the product concept. When you automate, that human layer disappears and customers may reject the result. Separate feedback about the outcome from feedback about the personal touch, and treat enthusiasm that depends on you being in the loop as unproven until automation is tested.

Sampling bias (unrepresentative batch)

A small or homogeneous batch produces feedback that feels definitive but does not generalize to the broader target segment. If the first customers you can recruit share an unusual trait — same company, same network, same tolerance for rough edges — their reactions may not predict how the wider segment responds. Actively recruit from different sub-groups within your target segment to stress-test what you are learning.

Next Steps

- Iterate on the service based on direct customer feedback from concierge delivery.
- Identify which manual steps are highest-value and should be automated first.
- Run a **Pre-Sales Test** to validate willingness to pay among concierge customers.
- Once the concierge version consistently delivers value, build a **Single-Feature MVP** that automates the highest-value manual step first.
- Use a **Wizard of Oz** test to evaluate customer reactions when the manual service is hidden behind an automated-looking interface.
- Run a **Product-Market Fit Survey** once you have enough concierge customers to measure whether they would be disappointed without the service.

Case Studies

Airbnb: Air mattresses in a San Francisco loft

Brian Chesky and Joe Gebbia photographed their San Francisco apartment, posted it for design-conference attendees, and hosted three paying guests to test whether people would pay to stay in a stranger's home.

[Read more](#)

Amazing Airfare: \$8 manual deal alerts

Zachary Cohn validated his flight-deal alerts concept by collecting \$8 each from five subscribers via PayPal and emailing them deals from a spreadsheet before building any software.

[Read more](#)

GE Healthcare: Concierge-style infant monitoring trial

GE Healthcare tested a new infant monitoring concept in emerging markets by deploying trained nurses with portable tools to deliver the experience by hand, validating workflow fit and perceived value before investing in complex diagnostic equipment.

[Read more](#)

Food on the Table: Concierge meal-planning

Manuel Rosso recruited Food on the Table's first customers by accompanying Austin grocery shoppers on their trips and emailing personalized meal plans, learning the workflow by hand before any software was written.

[Read more](#)

DoorDash: PaloAltoDelivery.com

The founders launched a bare-bones site listing local restaurant menus and personally took the orders and delivered the food themselves, learning the logistics by hand before building any dispatch software or incorporating as DoorDash.

[Read more](#)

Further reading

-  Cindy Alvarez - Lean Customer Development: Building Products Your Customers Will Buy
-  I Build MVPs - The Concierge Minimum Viable Product Maximizes Customer Learning

- ☰ HBR - Building a Minimum Viable Product
- ☰ The Concierge MVP Board
- ☰ Wizard of Oz vs Concierge MVP: When to pick which

5.6 Demo Pitch

In Brief

A demo pitch is a live presentation of your product or prototype to a prospect, designed to test their willingness to buy. You walk them through a demonstration of your solution, gauge their reaction, and ask for a concrete next step in the buying process — a pre-order, a letter of intent, or an introduction to the decision-maker. The output is qualitative feedback on purchase intent, objections, and positioning: the prospect who commits is a buyer, and the one who does not tells you what is missing.

COMMON USE CASE

You have a working prototype or detailed mockup and want to test whether potential buyers will actually commit. You schedule a demo meeting with a decision-maker, walk them through the product, and gauge whether they are willing to take a concrete next step like signing a letter of intent, placing a pre-order, or introducing you to their procurement team.

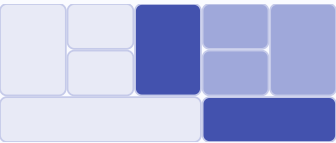
HELPS ANSWER

- Who is our early adopter or first customer?
- Who is the actual decision-maker?
- Is the product valuable enough for them to commit?
- Are we positioning the product correctly?
- Are we highlighting the most compelling features?
- How much is it worth to them?
- What does their buying process look like?
- Can we actually sell this?



AT A GLANCE

- Qualitative
- In-person
- Remote
- Attitudinal
- Transactional



TIME  ~3 days~2 weeks

COST  \$0-\$200

OTHER NAMES
Product Demo · Sales Demo

Description

A demo pitch is a live demonstration of your solution to a real prospect that ends in an explicit ask for a commitment — a deposit, a signed letter of intent, an order, or a booked meeting with the decision-maker. You put a working demonstration in front of the prospect and then ask them to do something that costs them. The answer you record is the action, not the opinion: a prospect who praises the demo and then declines the ask has told you more than one who only nods along. This is what separates it from a [Solution Interview](#), which stops at whether someone would use the product; the demo pitch asks them to act on it.

The method runs differently for B2C and B2B. For B2C, the demonstration is usually a short video or a brief in-person walkthrough delivered near the point of sale of similar products, and the commitment is light: a pre-order, a deposit, an email signup behind a real intent to buy. For B2B, the pitch is typically a video call or screen share with the decision-maker and the influencers around them, and the commitment is heavier: a signed letter of intent, a purchase order, a slot on the vendor list. You can also use it to test commitment from channel partners — whether they will enter an agreement or a trial with you — which surfaces how you should go to market.

The method works when you have something concrete enough to demonstrate and a real commitment you can ask for. It misleads you when neither is true: a demo with no offer attached collects compliments, and a beautiful demo with no working substance behind it tests your slides, not your product.

How to

Prep

✦ AI Prompt

1. Define the learning goal and the next step you will offer.

Decide what you want to learn from the demo and what concrete commitment you will ask for at the end. The next step is the test: a pre-order, a signed letter of intent, a deposit, a purchase order, or an introduction to the actual decision-maker. If you cannot name the commitment up front, you are running a generic sales meeting, not a demo pitch.

2. Select an audience of likely early adopters.

For B2C, target prospects near the point of sale of similar products or reachable through a tight social channel where you can measure how many take the next step. For B2B, identify the actual decision-maker plus the influencers and stakeholders

around the buying process, and prioritize early adopters who will not stall on “Who else has used this?”

3. Prepare demo artifacts that show the solution at its best.

Build a working prototype, interactive mockup, or short narrated walkthrough. Decide what to show live and what to describe verbally so prospects do not get distracted by half-finished surfaces. AI design and presentation tools generate polished demo materials in hours; the easier that is, the more disciplined you have to be about testing real purchase intent rather than presentation quality (see the AI-polished demo bias below).

4. Rehearse a tight pitch with one clear “wow” moment.

Structure the pitch around a hook, the problem, the solution demonstration, and a call to action. Identify the single moment that has to land; a memorable demo usually turns on one clear “wow” moment rather than even coverage of every feature.

5. Set up the commitment-test offer in writing.

Draft the next-step artifact in advance: the pre-order page, the letter-of-intent template, the order form, the calendar link to the decision-maker. The harder it is to act in the room, the easier it is for the prospect to politely say “interesting” and disappear.

Execution

1. **Deliver the rehearsed pitch.** Present to the selected audience without interruption, in person, over a video call, or via screen share.
2. **Run the demonstration to its best light.** Show the solution as planned, including the “wow” moment, and avoid the unfinished surfaces you decided to describe rather than show.
3. **Capture reactions in real time.** Note which features draw questions, which moments produce the strongest response, and where attention drops. Recording the session (with consent) lets an AI tool transcribe and tag reactions afterward.
4. **Make the next-step offer explicit before the meeting ends.** Put the commitment in front of them: a pre-order link, a letter of intent, a deposit form, or a booked follow-up with the decision-maker.

TOOLS

Scrappy kit (free): a slide deck plus a live screen share or recorded walkthrough of a working prototype — enough to run a credible demo with no dedicated tooling.

[Figma](#)

[Loom](#)

[Storylane](#)

[Navattic](#)

Analysis

✦ AI Prompt

1. **Separate polite interest from real intent.** The qualitative signal comes during and after the pitch. Distinguish a polite “that looks great” from a prospect who actually changes their behavior, and work out which moments drove the difference.
2. **Count who moved forward, not who showed up.** The decisive data is how many prospects took the concrete next step you offered. Track the share who committed against the number you pitched, and look at how it breaks down by role, company size, or vertical where you have enough sessions to compare.
3. **Cluster the recurring objections.** Group the questions and pushback into a few themes, and tag each as a positioning problem, a missing-feature problem, or a buying-process problem so you know what to change next.

⚠ Biases & Tips

Vanity-meeting bias

A high number of demos booked is not itself a sign of success. Count concrete next steps taken, not meetings held.

Social-desirability (politeness) bias

People often make positive comments to your face. Treat a verbal “this looks great” as noise until it is backed by a commitment; if prospects praise the demo but never advance the buying process, you are pitching the wrong people or the offer needs to change.

Confirmation bias

As the founder, you are primed to hear interest and discount objections. Decide before the demo what a pass and a fail look like, and have someone else review the session notes or recording so your read is not the only one.

AI-polished demo effect

AI presentation tools generate highly polished demo materials quickly, and prospects may react to presentation quality rather than the value proposition. Separate “that looks impressive” from “I want to buy this,” and the easier the demo is to produce, the more disciplined you need to be about testing real purchase intent.

Selection bias

Demoing only to friendly contacts or eager early adopters inflates your commit rate. Pitch a spread of realistic prospects so the signal reflects the market you intend to sell to, not the warmest corner of it.

Next Steps

- If demos convert to commitments, use a [Pre-Sales Smoke Test](#) to turn that interest into purchase commitments and confirm willingness to pay at scale.

- Reuse your strongest demo recording on a [Landing Page Smoke Test](#) to measure demand from a wider, colder audience.
- If prospects want to go beyond the pitch, run a [Wizard of Oz](#) test to deliver the full experience manually before you build it.
- If commitments stall on objections, return to a [Solution Interview](#) to dig into why the value proposition is not landing.

Case Studies

Dropbox: Narrated demo video

Drew Houston posted a short demo video of Dropbox where tech-savvy early adopters lived (Hacker News, Digg); when the video went viral, the waitlist exploded from 5,000 to 75,000 signups in a single day — validating demand before the product was built.

[Read more](#)

Tesla Cybertruck: Broken-window demo, 250K pre-orders

At the November 2019 Cybertruck unveiling in Los Angeles, designer Franz von Holzhausen threw a metal ball at the “Armor Glass” windows and both shattered on stage; within five days Musk announced 250,000 refundable \$100 pre-orders — showing a memorable demo plus a low-friction commitment offer can drive interest, though pre-order counts measure intent, not revenue.

[Read more](#)

Further reading

-  Steve Blank & Bob Dorf — The Startup Owner’s Manual (K&S Ranch, 2012)
-  Oren Klaff — Pitch Anything (McGraw-Hill, 2011)
-  Marty Cagan — Inspired: How to Create Tech Products Customers Love, 2nd ed. (Wiley, 2017)
-  Y Combinator — How to Pitch Your Startup
-  Eric Ries — The Lean Startup (Crown Business, 2011)

5.7 Pocket Test

In Brief

A pocket test means carrying a rough prototype and pulling it out mid-conversation with a prospect to get an unrehearsed first reaction. You watch how they try to use it and ask follow-up questions. The output is feedback on whether the concept makes sense, what features matter, and how people expect to interact with it. It fits products with a tangible form factor — physical goods, hardware, or anything someone can pick up or tap on a screen.

COMMON USE CASE

You have a rough version of what you want to build — a physical prototype, a mobile app mockup, or a clickable demo — and want honest reactions before investing further. During a casual conversation with someone in your target market, you pull it out and watch how they react to it.

HELPS ANSWER

- Am I building the right product?
- Do people understand what the product is and how it works?
- How do people expect to interact with this kind of solution?
- What limitations or concerns surface when someone holds it?

Description

A pocket test is a rough prototype you carry with you and surface in a live conversation, so a prospect reacts to it on the spot rather than to a description of it. They handle it, try to use it, and tell you what they think while you watch — the prototype does the explaining, and their reaction signals fit or rejection. It is a

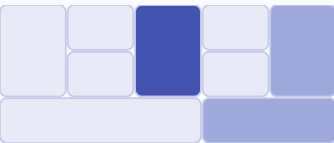


AT A GLANCE

Qualitative In-person

Observational

Attitudinal



TIME  ~3 days–2 weeks

COST  \$0–\$300

OTHER NAMES

Pocket Demo · In-Person Demo ·
Pocket Smoke Test

generative product-research method, and a natural next step after a

[Solution Interview](#)

or

[Discovery Interviews](#)

: once an interview surfaces a problem, the pocket test puts a rough solution in front of the same kind of person so you can watch how they expect to use it.

Treat the prototype as a conversation piece, not a finished product. You are testing the main case: is this something the prospect wants, and could it believably address a problem they have voiced? An ugly prototype is fine as long as it teaches you something you did not already know. Because you are showing a specific solution to a stated problem, you learn more than an interview gives you — you watch how the prospect expects to use the product instead of asking them to describe it.

The method depends on getting in front of people outside your own team, where real reactions rather than internal opinions decide what to build next. Carrying the prototype with you is the lowest-overhead way to do that.

How to

Prep

[AI Prompt](#)

1. Build the minimum viable prototype.

Only build enough to start a conversation:

- **Physical products:** Foam, cardboard, 3D-printed, or assembled from off-the-shelf parts. Ugly is fine — you're testing the concept, not the finish.
- **Software/apps:** A clickable design prototype, a few screens in a slideshow, or a live demo on your phone or laptop. It doesn't need to work end-to-end — just enough to show the core flow.
- **Services:** A one-page description, a mock invoice, or a sample deliverable.

Deliberately dial back the fidelity so the value proposition does the work, not the finish — AI tools can generate realistic-looking mockups very quickly, so resist the default and keep the prototype rough enough to signal “this is a concept.”

2. Identify 10–15 people to demo to.

They should have the problem you're solving. Sources: existing users, prospects from interviews, conference attendees, coworking space contacts. Avoid friends and family unless they are actually in your target market.

3. Prepare your observation checklist.

Decide what you're watching for before you demo:

- Do they understand what it does without you explaining?
- Do they try to interact with it in the way you intended?
- Do they ask about price, availability, or next steps? (Strong buying signal.)
- Do they compare it to something they already use? (Tells you your competition.)

Execution

1. Start a conversation about the problem, not the solution.

Don't lead with "check out what I built." Ask about their experience with the problem first. Once they've described the pain, then show the prototype: "I've been working on something — want to take a look?"

TOOLS

Figma

v0

Protolabs

2. Show the prototype and observe.

Hand it to them (physically or screen-share). Watch how they interact with it. Don't explain how it works unless they ask. Silence is data — if they stare at it confused, that's a usability signal.

3. Ask follow-up questions.

- "What do you think this does?"
- "Would this be useful to you? Why or why not?"
- "What would you expect to happen next?"
- "How does this compare to how you handle this today?"

4. Record your observations immediately after.

Note their reaction (excited, confused, polite, skeptical), key quotes, and whether they asked about price or next steps. Don't take notes during the conversation — it changes the dynamic.

Analysis

✦ AI Prompt

1. Tally reactions across demos.

After 10+ demos, categorize responses: How many understood it without explanation? How many asked about price or next steps? How many compared it to something they already use? Look for patterns, not individual reactions.

2. Identify the dominant failure mode.

- If most people don't understand what it does → the concept needs clearer framing, not a better prototype.
- If they understand it but aren't interested → the problem may not be painful enough for this audience.
- If they're interested but point out the same missing feature → you've found your build priority.

3. For small samples (under 10 demos): Focus on qualitative signals. Two people who ask “can I buy this?” is a stronger signal than eight people who say “that’s interesting.”

⚠ Biases & Tips

Confirmation bias

Don't explain, correct, or coach the prospect on how to use the prototype. If you catch yourself saying “no, you're supposed to...” you've stopped testing and started selling.

Social-desirability (politeness) bias

People will say “that's cool” to avoid awkwardness. Only concrete follow-up actions (asking about price, requesting a follow-up meeting, offering to refer someone) count as real interest.

Selection bias

Demoing only to people who are easy to reach — friends, coworkers, your existing users — skews reactions warm. Recruit from your actual target market, and discount enthusiasm from anyone who has a reason to be kind to you.

Fidelity misdirection

When a prototype looks finished, people redirect attention to design details instead of reacting to the core concept. Keep fidelity low enough that the value proposition, not the finish, carries the conversation.

Next Steps

- If most reactions are enthusiastic, scale the winning pitch to a wider audience with a [Landing Page Test](#) or pre-sales test.
- Iterate on the prototype based on the most common objections or confusion points before re-testing.
- Use a [Sales Pitch Test](#) to formalize your pocket demo into a structured pitch and measure close rates.
- Track which demo contexts (industry events, coworking spaces, customer offices) produce the strongest reactions to refine your target segment.

Case Studies

Further reading

- ☰ [Steve Blank — Customer Development Manifesto](#)
- ☰ [GLIDR Help Center — Pocket Test](#)
- ☰ [Eduardas Afanasjevas — A guide to building a hardware prototype](#)
- ☰ [Ben Einstein — The Illustrated Guide to Product Development](#)

5.8 Dogfooding

In Brief

Your team uses your own product in real daily work, just as a customer would, and records every bug, friction point, and unexpected use case they hit. There is no script: you go through the actual workflows, note where things break or feel awkward, and capture ideas for improvement. The output is a firsthand issue list that formal testing often misses.

COMMON USE CASE

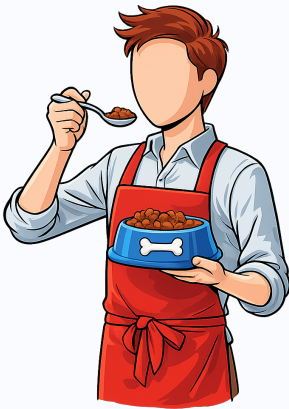
Your team has built the first version of your product and you want a quick reality check before putting it in front of customers. You and your teammates use the product in your daily work for a week, noting every moment of friction, confusion, or delight so you can fix the biggest issues before anyone else sees it.

HELPS ANSWER

- Does the product actually deliver on the value proposition?
- Is the product working correctly in real use?
- What is the minimum viable feature set?
- Where does the workflow break down or feel awkward?

Description

Using your own product is common practice among technology startups, especially when the team built the product to solve a problem they have themselves. There is no predefined script and it is not a formal quality assurance (QA) process. Its main advantage is surfacing unorthodox use cases that the requirements and QA tests never covered.



AT A GLANCE

Qualitative

In-person

Remote

Observational

TIME

 ~2 weeks–2 months

COST

 \$0

OTHER NAMES

Eat Your Own Dog Food · Internal Use Test

Treat the output as generative research, not an experiment. Your team is not a representative sample of your users, so dogfooding generates ideas and finds obvious breakage but cannot validate that a feature works for the people you are building for. Pair it with external [Usability Testing](#) before drawing conclusions about real-user behavior.

You cannot delegate dogfooding to an AI agent or an automated testing script. The value comes from a person encountering friction, confusion, or delight while trying to finish a real task, so AI helps with capture and synthesis but not with the using. If your product includes AI features, firsthand use matters more: you need direct experience of how the model behaves in real scenarios, where it fails, and where it produces unexpected results. Automated QA and AI-powered testing tools catch bugs, but they are not dogfooding.

How to

Prep

✦ AI Prompt

1. **Set a clear scope.** Pick the workflows and personas you intend to walk through (new-user signup, daily power-user task, admin path) so the team exercises the same surfaces and you can compare notes afterward.
2. **Set up a low-friction capture channel.** Have a place to take notes that is easy to reach and does not interrupt the workflow: a shared issue tracker, a single team-chat channel, or a running doc.
3. **Decide who participates.** Pull in teammates outside the build team where possible. Engineers and designers carry too much prior knowledge to notice every friction point alone.

Execution

1. **Use the product in real work.** Do not switch to a workaround when something annoys you. Record the friction and the workaround you reached for; both are data.
2. **Log wins and failures as they happen.** Take notes whenever something works surprisingly well or falls short, and record any ideas that occur while using the product.
3. **Flag every hand-off.** Note any point where the workflow is interrupted or another service is needed to finish the task. These hand-offs are the most common hidden friction.

TOOLS

[Luciq](#)

[Linear](#)

[Jam](#)

4. **Capture context, not just bugs.** Screenshot the screen, note what you were trying to do, and write down what you expected to happen.

Analysis

✦ AI Prompt

1. **Read the findings as generative, not evaluative.** Your team knows the product design too well to stand in for a first-time user. This is especially true for the new-user flow, where internal users carry far more prior knowledge and expectation about signup and onboarding than any real user would.
2. **Watch for missing edge cases.** A team that is not diverse may never exercise the product the way users with disabilities or users from other backgrounds would, and thus overlook substantial parts of the experience. This grows as the product scales beyond its initial niche audience.
3. **Surface patterns, not gripes.** When multiple team members dogfood the product, collect and sort the notes with [Card Sorting \(Features\)](#), stack ranking, or other standard UX methods rather than reacting to individual complaints.

⚠ Biases & Tips

Confirmation bias

Creators of a product can subconsciously avoid situations and use cases they know are incomplete or buggy, leaving a positive impression that the product works according to the specification even if it has serious flaws in ordinary usage.

Expert blind spot

Team members know the product too well to notice friction a first-time user would hit immediately. Recruit teammates outside the build team or pair internal testing with external usability testing.

Next Steps

- Validate the priority ordering with external [Usability Testing](#), which covers the blind spots your team misses through product familiarity, and fix the highest-severity, most-repeated issues first.
- Sort the captured notes with [Card Sorting \(Features\)](#) to turn individual gripes into a prioritized issue backlog.
- Run a [Net Promoter Score Survey](#) to check whether real customers share the satisfaction your team feels internally.

Case Studies

Microsoft, Lyft, Facebook: Dogfooding roundup

A practitioner roundup of dogfooding programs, including Microsoft's 25,000+ employee Elite program, Lyft's "four hours per quarter" driver shift requirement for executives, and Facebook's internal block on desktop access during its Android push.

[Read more](#)

Twitter: Odeo's internal SMS tool

The first Twitter prototype was an internal service for Odeo employees built by Jack Dorsey and Florian Weber; the team's enthusiasm and SMS-bill spikes were the demand signal that prompted the July 2006 public launch.

[Read more](#)

PostHog: Dogfooding drove the data warehouse and surveys

PostHog's internal use of its own product shaped major features: the data warehouse came from engineers needing PostHog to query Stripe and HubSpot, and the surveys feature began as an internal popup the product team built to streamline user-interview booking.

[Read more](#)

Bubble: Building Bubble on Bubble

Bubble built its website (everything except the editor) on its own no-code platform, turning internal developers into demanding users and creating a "forcing function" that drove the platform to support complex multi-page apps.

[Read more](#)

Microsoft: From "Eating our own Dogfood" memo to 20,000-node network

Paul Maritz's 1988 email to LAN Manager test manager Brian Valentine, titled "Eating our own Dogfood," coined the term; Windows NT later involved 200+ developers running daily builds, and by 2005 InfoWorld reported Microsoft running its 20,000+ node international network on 99% Windows technology.

[Read more](#)

Lyft: Mandatory driver shifts for corporate

Lyft requires corporate employees to drive as part of the platform, forcing leadership to experience the supply-side product and surfacing driver-facing UX issues that internal dashboards miss.

[Read more](#)

CrowdStrike: Dogfooding as post-outage remediation

Following the July 2024 worldwide outage, CrowdStrike's Adam Meyers testified to Congress that increased internal dogfooding of agent updates was part of the company's remediation plan — a rare case of dogfooding as regulatory commitment.

[Read more](#)

Frontegg: Running on its own Entitlements Engine

Frontegg uses its own Entitlements Engine to manage subscription plans and permissions for its own products, so internal subscription changes continuously validate the same engine customers depend on.

[Read more](#)

Further reading

-  [Intercom: The Danger of Dogfooding](#)
-  [PostHog: How We Do Dogfooding \(with Examples\)](#)
-  [PostHog: Why Every Team Member Should Be Dogfooding](#)
-  [Wikipedia: Eating Your Own Dog Food — History and examples of the practice](#)
-  [Eric Ries, The Lean Startup \(Crown Business, 2011, ISBN 978-0307887894\) — frames continuous internal use of a product as part of the build-measure-learn loop.](#)

5.9 Picnic in the Graveyard

In Brief

Picnic in the graveyard is the study of startups and products that already failed in your market or an adjacent one. You research what they built, interview their former founders and customers when you can reach them, and identify the specific assumptions that proved fatal. The output is a clear picture of what went wrong, what has changed since, and which risks to test first so you avoid repeating the same mistakes.

COMMON USE CASE

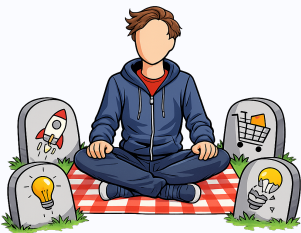
You have an idea that has been tried before by other startups that failed. Before repeating their mistakes, you want to study what went wrong, talk to their former customers and founders, and figure out what has changed that might make the idea viable now.

HELPS ANSWER

- What similar products have been tried in the past?
- What did customers like or dislike about those previous products?
- What would the founders of failed similar products do differently if they tried again?
- Which features are essential based on what previous attempts got wrong?
- How is our approach different from what has been tried before?

Description

Picnic in the Graveyard is the study of products and companies that already tried — and failed at — something close to what you are building. It pairs desk research on what they built and why they died with candid interviews of the people who were there: former founders and former customers. The goal is to build a fuller picture of



AT A GLANCE

Qualitative

Remote

Observational

Attitudinal



TIME  ~1 hour–1 day

COST  \$0–\$50

OTHER NAMES
Failed Product Postmortem

what has been attempted in your space so you can spot the landmines before you step on them. Many failed products got most of a feature set right and missed on one thing; some simply shipped too many features. If you are about to introduce something that has been tried before, you want to be clear about what is different this time and why that difference will matter to a buyer.

Studying the graveyard often generates ideas as well as warnings. The lessons may point at the product itself — which features were essential and which were noise — or at the market, suggesting a different segment to target or a change in direction.

The desk-research phase is fast, but the highest-value part of the method is the candid interview with a former founder or former customer. A former founder is essentially a domain expert on the exact failure you are trying to avoid, so treat that conversation like an [Ask an Expert](#) session. Published post-mortems tend to be sanitized, so the honest account of what really went wrong usually comes out only in conversation.

How to

Prep

✦ AI Prompt

1. **Scope the graveyard.** Define the product space or adjacent market you are studying — narrow enough that the failures you find are comparable to your idea, broad enough that more than two or three dead products show up.
2. **Identify failure databases and post-mortem sources.** Curated failure archives, post-mortem compilations, and web archives are good starting points for written post-mortems; press archives and old crowdfunding pages help fill in companies that died quietly.
3. **Set a timebox for outreach.** Decide up front how long you will spend tracking down each former founder or former customer. One day per founder is reasonable when there are only a handful of comparable startups; tighten the box if the list is long.
4. **Draft your research questions.** Write the questions you will ask former customers (what they used the product for, what they liked, what they switched to) and the separate set you will ask former founders (what assumption proved wrong, what they would do differently, what has changed in the market since).

Execution

✦ AI Prompt

TOOLS

[Crunchbase](#)

[CB Insights](#)

[Failory](#)

[Wayback Machine](#)

1. **Build the list of dead products.** Run desk research to identify previous attempts at a similar product in your market or an adjacent one. AI search and the failure databases from Prep do most of this work; for niche or older spaces, press archives and a reference librarian can surface attempts that never made it online.
2. **Find former customers.** Use public signals to locate people who used the dead products: old posts mentioning the product by name, profiles listing the company as an employer, app-store reviews, and archived community-forum threads.
3. **Interview the former customers.** Reach out and run the interviews using the questions drafted in Prep, covering what they used the product for and what they liked and disliked. Stay neutral and let them talk.
4. **Interview the former founders or product leads.** Track down the people who built each product — founders for startups, the project or product lead at a larger company — and interview them about what they assumed, what proved wrong, and what they would change.

Analysis

✦ AI Prompt

1. **Cluster failures by root cause.** Group the post-mortems and interview notes by what actually killed each product — wrong buyer, no willingness to pay, weak distribution, bad timing. The clusters show where the space is most dangerous.
2. **Find the repeated assumptions.** Note the assumptions that show up in two or more failures. A mistake several teams made independently is a pattern, not a coincidence, and should be treated as default-risky for your own idea.
3. **Check what has changed since.** For each fatal assumption, ask whether the market, technology, or buyer behavior has shifted enough that it might no longer hold. A failure caused by conditions that no longer exist is an opening, not a warning.
4. **Turn each pattern into a test.** For every assumption you cannot rule out, design a cheap, falsifiable experiment that checks whether it is still true today. These tests become your validation priorities.

⚠ Biases & Tips

Overconfidence

“We’re different, so what they did doesn’t apply” is the reflex that keeps founders from studying past failures at all. Treat those failures as evidence to disprove, not noise to ignore.

**Survivorship
bias**

Studying only the failures (or only the survivors) gives a skewed picture. Look at both, and at near-misses that pivoted, to see the full shape of the space.

**Premature
abandonment
bias**

Exposure to a graveyard of failures can tip founders into discarding a viable idea before testing it. If the evidence shows others failed, weight that as a signal to design falsifiable experiments — not as a verdict on whether your version can succeed.

Next Steps

- Run a pre-mortem listing the most likely ways your own startup could fail, drawing on the patterns you found.
- Run a [Discovery Interviews](#) to verify whether the problems that sank your predecessors still exist for today's customers.
- Run a [Competitor Analysis](#) to map the current landscape and see who else has entered the space since the failures you studied.
- Run a [Solution Interview](#) to put your differentiated approach in front of customers and check whether it actually avoids the failure patterns you found.
- Revisit the graveyard quarterly, since new failures and post-mortems appear as the market evolves.

Case Studies

Critica: First-time founder post-mortem

Jason Huertas reflects on building Critica for 18 months without validating demand and treating a v0 as if it were a mature product — the kind of candid founder retrospective a graveyard study mines for the assumption that actually proved fatal.

[Read more](#)

Quibi: .75B burned in six months

Quibi shut down six months after launch, never having validated the core assumption that consumers wanted premium short-form mobile video; the post-mortem identifies a mobile-only launch during COVID lockdowns and \$5–\$8/month pricing against free TikTok and Instagram as fatal positioning errors. A worked example of the failure-assumption mapping this method produces.

[Read more](#)

Google Glass: ,500 AR glasses to enterprise pivot

The Glass Explorer program priced consumer units at \$1,500, drew a privacy backlash that earned wearers the “glasshole” label, and was eventually pivoted to manufacturing, logistics, and healthcare — sound technology, wrong original market. The kind of “right idea, wrong segment” pattern a graveyard study is built to surface.

[Read more](#)

Further reading

-  Kromatic: Generative Research — The Picnic in the Graveyard Technique
-  SKMurphy: Pretotyping – Techniques for Building the Right Product
-  Phillip Kaplan: F’D Companies: Spectacular Dotcom flameouts
-  Eric Ries — The Lean Startup (Crown Business, 2011)
-  CB Insights — The Top 12 Reasons Startups Fail
-  Failory — Startup Graveyard

5.10 Prioritization

In Brief

Prioritization games are structured exercises that force participants to make trade-offs between features, capabilities, or product attributes. By introducing constraints — limited budgets, forced rankings, or spatial metaphors — these games bypass the tendency to call everything “important” and reveal what people value most when they cannot have it all.

COMMON USE CASE

You have a backlog of candidate features and need to decide what to build first. You convene a small group of target participants (typically 4–8) for a 60–90 minute session and run a prioritization game to force trade-offs that surveys can’t capture. The output is a ranked list plus the reasoning participants used to rank — which is the part that informs product decisions.

HELPS ANSWER

- Which features do customers value most?
- What should we build first?
- Where are the biggest pain points in our current product?
- Which capabilities do customers consider must-haves versus nice-to-haves?
- What words do customers use to describe value?
- Are there features customers would pool resources to obtain?

Description

Prioritization games work because constraints force a choice. When a participant can only pick 5 features out of 20, or has only \$100 of play money to distribute across 15 options, they must make the same kind of trade-offs they make with real purchasing decisions. This produces a stronger preference signal than asking people



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal



TIME  ~3 days–2 weeks

COST  \$0–\$400

OTHER NAMES

Innovation Games · Preference Games

to rate features on a 1-to-5 scale, where the answers tend to cluster at the high end — though it remains stated-preference data from a small group, not statistically representative measurement. The conversations that emerge during gameplay are often more useful than the final rankings, because the reasoning behind a choice is what transfers to product decisions. For a measurement that scales, use conjoint analysis (which estimates the value of each feature from choices between bundled options) or MaxDiff (a forced best/worst ranking across many items).

Three of the four games here (Buy a Feature, Product Box, Speed Boat) come from Luke Hohmann's *Innovation Games*; card sorting comes from the broader information-architecture and UX-research tradition. Like all stated-preference methods, the signal is sensitive to who is in the room and who speaks first. Facilitation matters, and each game carries its own group-dynamics risks — anchoring, dominant voices, social-desirability bias — that the individual method pages address.

These methods assume you have already validated that demand exists for your product category. They answer “what should we build?” not “should we build it?” For market-level research on which problems matter most, see [Pain Point Sorting](#). For internal team-side scoring frameworks (RICE, ICE, MoSCoW, Kano), see [Scorecards](#) — those rank options using team estimates, not customer behavior.

Choosing the Right Prioritization Game

Each game uses a different constraint to force trade-offs. Pick the one that matches the decision you need to make and the group size you can convene. Two or three sessions per game produce stronger signal than one.

Available Games

Buy a Feature

Participants spend a fixed pot of play money across priced features, forcing trade-offs and group negotiation over what to fund. Best used when you need to see what people will pay for and how they reason about cost.

Card Sorting (Features)

Participants rank a known list of feature cards to show what they value and in what order. Best used as a first session when you have a defined backlog and need a fast, low-prep ranking.

Product Box

Participants design the packaging for their ideal product, surfacing the features they wish for and the words they use to describe value. Best used when you need to discover features and messaging you might not have thought of.

Speed Boat

Participants mark pain points as “anchors” slowing a boat, a visual metaphor that makes negative feedback easy to give. Best used when you need to surface frustrations before deciding what to fix or remove.

5.10.1 Buy a Feature

In Brief

In Buy a Feature, participants receive a limited budget of play money and decide which features to purchase from a priced list. Each participant gets only enough money to buy roughly one-third to two-thirds of the available features, so the budget forces trade-offs. Pricing one or two features beyond what any single participant can afford pushes them to pool money, and the negotiation that follows surfaces their priorities and the reasoning behind them. The output is a feature list ranked by spend, plus the negotiation notes.

COMMON USE CASE

You have a backlog of fourteen to thirty candidate features and a small group of target participants you can convene for sixty to ninety minutes. You need to know which features they would actually pay for, not just rate as “important” on a survey. The output is a ranked-by-budget feature list plus the negotiation reasoning participants used to get there.



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal



TIME



~3 days–2 weeks

COST



\$0–\$400

OTHER NAMES

Buy a Feature

HELPS ANSWER

- Which features would customers actually pay for?
- How do participants make trade-offs between features?
- What is the perceived relative value of different features?
- Are there features that only specific segments care about?
- What coalitions form around which features?

Description

Buy a Feature is the [Prioritization](#) variant that uses a fixed play-money budget to rank features by what participants will spend on them. A fixed budget forces

participants to trade off features against one another, the way they trade off real money against real purchases. That produces sharper signal than a 1-to-5 rating scale, where most features cluster near the top because nothing is at stake. The conversations during the game — the debates, the compromises, the lobbying — are as informative as the final purchases: they show not just what participants buy, but why they buy it and what they give up to do so.

The game produces three layers of insight:

1. **Individual priorities.** What each participant buys with their own money reveals their personal must-haves.
2. **Negotiation behavior.** When expensive features require pooling money, participants persuade others to join them. The arguments they make and the compromises they accept reveal how strongly they want a feature.
3. **Feature abandonment.** What participants choose *not* to buy — even when they can afford it — flags features that sound appealing but are not worth the trade-off.

Buy a Feature is stated-preference data from a small group, not a representative measure of willingness to pay. Use it to compare features against each other and to hear the reasoning, not to forecast revenue. When the question is which problems matter rather than which features to fund, prioritize pains first with

 Pain Point Sorting .

How to

Prep

 AI Prompt

1. Build the feature list.

Write fourteen to thirty candidate features on cards, a poster, or a handout. Each feature gets a clear name, a one-sentence description, and a price. Keep the list in this range: long enough to force trade-offs, short enough that participants can hold the whole set in their head. Use participant-facing language, not internal codenames.

2. Set prices to provoke trade-offs, not to model engineering cost.

Pricing is the most important design choice. Price at least one or two features high enough that no single participant can buy them alone, so the group has to pool money to fund them — the negotiation that follows is where most of the insight comes from. Price to push participants into trade-offs, not to signal how much each feature costs you to build.

A workable distribution for a fourteen-to-thirty feature list:

- Five to ten low-priced features that any participant can afford.
- Five to twelve medium-priced features that require trading off other purchases.
- Four to eight high-priced features that require two to four participants to pool money.

Use round numbers so participants can do the math in their heads.

3. Distribute play money.

Each participant receives the same budget — enough to buy roughly one-third to two-thirds of the total feature value if they spent only on cheap features. Keep the budget tight enough that participants cannot have everything; the constraint is what makes their choices meaningful. Print bills, use poker chips, or hand out paper certificates — whatever holds up to being passed back and forth on a table.

4. Recruit four to seven participants per group.

Aim for four to seven participants per group: small enough that everyone has to talk, large enough that coalitions can form. Run multiple groups if you can recruit more than seven — features that win across several groups carry far more weight than a single session's result.

5. Draft the rules sheet.

Write the rules out so the facilitator can read them verbatim. The core rules:

- “You have \$X to spend on the features in front of you.”
- “You can buy any feature you can afford on your own.”
- “You can pool money with other participants to buy a feature together.”
- “You do not get change. Leftover money has no value.”
- “You have forty-five minutes to make your purchases.”

Execution

1. Open with the rules and the budget reveal.

Read the rules verbatim from the sheet you drafted in Prep. Hand out the play money in front of the group so the budget feels tangible. Take ten minutes for introductions and rules so participants understand the mechanic before any money moves.

2. Step back and let them play.

Once the rules are clear, the facilitator's job is to observe, not to guide. Participants only negotiate honestly if they believe they have to convince each other, so stay out of it. Do not validate choices. Do not explain features beyond the printed description. If a participant asks what something means, note the confusion as a finding and redirect them to the description on the card.

Forty to sixty minutes is the right window for purchasing and negotiation. Take notes on who approaches whom, what arguments get made, what deals get struck, and what features get abandoned without discussion.

3. Track purchases as they happen.

As participants finalize purchases, record:

- Which features were bought.
- Who paid for each feature, and how much each contributor put in.
- Which features were purchased through pooled money, and which participants formed the coalition.

A simple spreadsheet column for each feature works fine. If you are running the session remotely, a shared online whiteboard lets participants drag chips onto features while you log the same data.

4. Debrief for fifteen to twenty minutes.

After time is up, walk the group through the results. The high-leverage questions:

- "What was the hardest trade-off you made?"
- "What feature did you want but decided not to buy? Why?"
- "What surprised you about other people's choices?"

TOOLS

Play money — printable templates are widely available online; poker chips also work well.

Feature list poster or printed handouts — one card per feature, with name, one-line description, and price.

[Miro](#)

[FigJam](#)

The reasoning that comes out of the debrief is often more useful than the purchase data itself.

Analysis

✦ AI Prompt

Analyze results across several dimensions:

- **Most-purchased features** are clear priorities. If a feature was bought across multiple groups, it is a strong roadmap candidate.
- **Features that inspired pooling** are aspirational and high-value. These often represent capabilities that could become key differentiators — and the negotiation around them tells you which segments will fund them.
- **Features no one bought** are candidates for cutting from the roadmap, regardless of how much engineering effort has already been invested.
- **Negotiation patterns** reveal segment dynamics. If certain participants consistently lobbied for the same features, they may represent a distinct customer segment with shared priorities.
- **Leftover money** tells you about satisfaction with the available options. If participants consistently had unspent money at the end, they may prefer a simpler product or the feature list may be missing something important — go back to generative interviews or [Pain Point Sorting](#) to find it.

⚠ Biases & Tips

Dominant negotiators

Assertive participants may drive group purchasing decisions. Observe whether quieter participants were genuinely persuaded or merely yielding. Ask each participant to privately write down their top three before the game begins so you can compare individual intent against group outcome.

Price anchoring

Participants may equate higher price with higher value, rather than making independent judgments. Remind the group at the start that prices are for gameplay only, not signals of importance or quality.

Coalition pressure

Once a coalition forms around an expensive feature, participants may feel social pressure to contribute even if it is not their top priority. Watch for reluctant contributions and probe them in the debrief.

Framing bias

Feature names that read as marketing copy or internal codenames prime participants to buy based on the label, not the underlying capability. If purchasing patterns look suspiciously unanimous,

check whether the winning features had more vivid or aspirational names than the rest of the list.

Unequal
information

Some participants may understand certain features better than others, giving them outsized influence. Make sure each description is clear enough that the least-informed participant can evaluate it.

Next Steps

- Translate the most-purchased and most-pooled features into a draft roadmap, and flag the unbought features as cut candidates.
- Run [Card Sorting \(Features\)](#) to confirm the ranking with a different group and surface gaps in the feature list.
- Run [Product Box](#) to learn what messaging participants would put on the front of the box for the winning features.
- Run [Speed Boat](#) to surface the pain points behind the features participants fought hardest to fund.

Case Studies

Capital One: 00 Buy-a-Feature across two audiences

Capital One ran Buy a Feature with both front-line retail bankers and executive management on a \$100 virtual budget; the two groups split predictably (bankers wanted customer-serving features, management wanted process-efficiency features), and the high-value/low-effort intersection became the MVP scope.

[Read more](#)

City of San José: Buy-a-Feature as participatory budgeting

San José's 2016 participatory-budget event used Luke Hohmann's Budget Games adaptation to let roughly 300 residents allocate a ~\$64M maintenance-and-beautification budget across 30 projects in small facilitated groups, with senior city officials present to answer questions.

[Read more](#)

Further reading

-  Hohmann, Luke. *Innovation Games: Creating Breakthrough Products Through Collaborative Play*. Addison-Wesley, 2006. The primary source for the Buy a Feature mechanic, including the pricing rule, group-size guidance, and budget design.

- ☰ [Buy a Feature — Luke Hohmann](#)
- ☰ [Buy-a-Feature — ProductPlan](#)
- ☰ [How to play the “Buy a Feature” design game — UX for the Masses](#)
- ☰ [Innovation game — Wikipedia](#)

5.10.2 Card Sorting (Features)

In Brief

Card Sorting writes proposed features or product attributes on individual cards and asks participants to rank them in order of importance. The exercise forces explicit trade-offs and, more usefully, gets participants to articulate *why* they rank features the way they do. The output is a ranked list per session plus the reasoning that explains it.

COMMON USE CASE

You have a list of eight to twelve candidate features and you can spend an hour with each of three to five target participants. You want to know which features each ranks highest, but more importantly, you want to hear them think out loud about why one feature beats another.

HELPS ANSWER

- Which features do customers value most?
- What is the minimum feature set for launch?
- Why do customers prefer certain features over others?
- Are there segments with different feature priorities?
- What language do customers use to describe features?

Description

Card Sorting for features works by making abstract choices concrete. Instead of asking “what features do you want?” — which produces vague wish lists — you put specific options in front of participants and ask them to choose. It is the lightest-weight game in the [Prioritization](#) family, surfacing a rank order and the reasoning



AT A GLANCE

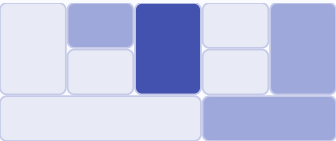
Quantitative

Qualitative

In-person

Remote

Attitudinal



TIME  ~3 days–2 weeks

COST  \$0–\$300

OTHER NAMES

Card Sorting · Feature Card

Sorting · Feature Prioritization

behind it before you invest in heavier trade-off mechanics. The physical act of arranging cards creates a tangible artifact you can discuss, point to, and rearrange.

The ranking itself matters less than the conversation it provokes. When a participant hesitates between two cards, or moves one above another and then explains why, you are hearing the decision logic behind the preference. That narration is the data the ranking alone cannot give you, so probe for it throughout the sort.

This method should not be confused with the better-known information-architecture variant of card sorting, which asks participants to group and label cards to learn how they organize content. Here you are ranking features by value, not discovering categories.

This page covers the feature variant. Its sibling, [Pain Point Sorting](#), uses the same mechanic to prioritize problems rather than features. Use the Pain Points variant when you are still exploring the problem space; use this Features variant when you have a validated problem and are deciding what to build.

How to

Prep

✦ AI Prompt

1. Create feature cards.

Write 8 to 12 proposed features or product attributes on individual cards. Use clear, jargon-free language. Each card should describe one feature in 5 to 10 words, with a brief clarifying sentence if needed. Keep the wording at the participant's level of abstraction, not your team's — no internal codenames.

2. Include blank cards.

Give participants 2 to 3 blank cards so they can write in features you did not think of. This is one of the most valuable parts of the exercise — the write-ins are where you discover the gaps in your own feature list.

3. Recruit three to five participants.

A small qualitative sample is enough to surface patterns in the reasoning. The information-architecture variant sometimes uses larger samples for quantitative cluster analysis, but that is a different method with a different output. Here you are after the verbalized decision logic, which converges quickly across a handful of representative participants. Schedule individual sessions rather than group sessions — in a group, participants hear each other's rankings and social pressure inflates

“responsible” features like security or accessibility relative to what each person would choose privately.

Execution

1. Ask the participant to rank.

Shuffle the cards and place them on a table in random order, then ask the participant to arrange them from most important to least important. Do not provide further instructions; observe how they approach the task. The goal is to see how the participant makes sense of the set, not to coach them into your preferred answer.

TOOLS

Physical kit (in-person): index cards, a Sharpie, a flat surface — ≤\$10

[Miro](#)

[FigJam](#)

[Optimal Workshop](#)

2. Probe the reasoning while they sort.

As they sort, ask open-ended questions: “Tell me why that one went to the top.” “You hesitated between these two — what were you thinking?” “What would happen if you could not have the bottom three?” The point is to capture the verbalized decision logic, not just the final order.

3. Optional: add an economic trade-off.

To capture intensity of preference rather than just order, run a short budget-allocation pass after ranking. This is the mechanic at the heart of [Buy a Feature](#) — see that page to run it in full.

4. Document results.

Photograph the final arrangement, record key quotes, and note which blank cards were added. The photograph is the artifact you will compare across sessions; the quotes are the evidence behind the ranking.

Analysis

[+ AI Prompt](#)

1. Look for convergence across sessions.

Look for features that consistently land in the top 3 to 5 across multiple sessions. Convergent picks across three to five participants are strong launch candidates. Features that appear in the top group for every participant are stronger signal than features that appear in the top group for one participant by a wide margin.

2. Investigate high-variance features.

Pay attention to features that show high variance — some participants rank them first while others rank them last. High variance often indicates different participant

segments with different needs. Pull the verbalization data for those features specifically; the explanation usually reveals the segmentation.

3. Read the blank cards.

Blank cards that participants add are especially valuable. If multiple participants independently add the same missing feature, it is a strong signal that your feature list has a gap. A single write-in is interesting; convergent write-ins across participants are an instruction.

4. Triangulate against adjacent methods.

If you have run a [Buy a Feature](#) session or customer interviews on the same feature set, compare the rankings. Features that win in card sorting but lose in Buy a Feature are features customers *say* they want but would not pay for; features that win in both are roadmap candidates.

Biases & Tips

Availability bias

Participants may prioritize features tied to a problem they hit this morning rather than their most important needs overall. Probe for frequency separately from rank (“how often do you run into this?”) so recency does not drive the order.

Coverage bias

The features you put on cards anchor the entire conversation. Categories you leave off cannot surface unless participants use blank cards, and they rarely volunteer a feature they were never asked to name. Review the card set for missing feature families before the first session, and lean on the blank cards.

Social-desirability bias

Participants tend to rank “responsible” features they think they should value — security, accessibility, compliance — higher than features they would actually choose. The optional budget-allocation pass partially counteracts this by forcing scarcity; staying neutral while probing counteracts the rest.

Anchoring and order effects

The card a participant reads first, and any card you place at the top of the deck, can shift the relative ranking of everything else. Shuffle the cards before each session and present them in random order.

Interviewer-approval bias

Nodding, smiling, or echoing a participant’s choice signals approval and anchors subsequent card placements. Maintain a neutral expression throughout the sort; reserve all reactions for the debrief.

Next Steps

- Compare the top-ranked cards across sessions and draft a minimum feature set for launch from the convergent picks.
- Use a [Buy a Feature](#) to add economic trade-offs once you know which features rank high.
- Use a [Product Box](#) to discover the aspirational features and messaging that did not appear on your card list.
- Use a [Pain Point Sorting](#) if the rankings show you have not yet validated the underlying problems.

Case Studies

Further reading

- [Rosenfeld Media, 2009](#)
- [Card Sorting: A Definitive Guide — Donna Spencer and Todd Warfel, Boxes and Arrows](#)
- [Hohmann, Luke. Innovation Games: Creating Breakthrough Products Through Collaborative Play. Addison-Wesley, 2006. Card Sorting for features is not one of Hohmann's twelve Innovation Games, but the closest siblings in the catalog are 20/20 Vision and Prune the Product Tree, both of which work the same relative-importance decision shape.](#)
- [Card Sort \(Product Feature Prioritization\) — Qualtrics Marketplace](#)
- [Card sorting — Ballpark research glossary](#)

5.10.3 Product Box

In Brief

In Product Box, participants design the retail packaging for their ideal version of your product. Using a blank cardboard box and a tableful of craft supplies — colored markers, crayons, glitter glue, plain and colored paper, stickers — they build a box they would want to buy off a shelf, complete with product name, key features, slogans, and imagery, then pitch it back to the room. Rather than reacting to your ideas, they articulate what they want in their own words and visual language.

COMMON USE CASE

You have a product concept or category, and you’ve talked to customers enough to know there is a real problem worth solving. What you do not yet know is which capabilities they would actually put on the front of the box if they were buying it — versus which ones your team is over-emphasizing internally. You have access to four to eight target customers for a couple of hours and a budget for craft supplies. The output is a set of customer-designed boxes plus the language, imagery, and feature priorities that emerge across them.

HELPS ANSWER

- What features do customers find most compelling?
- How do customers describe the product’s value in their own words?
- What aspirational outcomes do customers associate with the product?
- What messaging and positioning resonate most?
- What would customers put on the “front of the box” versus the fine print?



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal



TIME  ~3 days–2 weeks

COST  \$0–\$400

OTHER NAMES
Product Box

Description

Product Box is a structured exercise where participants design the retail packaging for their ideal version of your product, then pitch it back to the room. It is the generative, open-ended end of the [Prioritization](#) family: rather than ranking features you already have, participants invent the box from scratch, which puts them in the role of creator rather than critic. In most research methods, participants react to what you show them; in Product Box, they build what they want. What someone crafts and sketches reveals motivations they will not articulate in a direct interview, so the exercise tends to surface features, messaging, and value propositions a product team would not arrive at by surveying or interviewing alone.

The physical, hands-on format also lowers social barriers. Participants who stay quiet in a focus group often become animated when cutting out paper, picking stickers, and arguing about what goes on the front of the box, and the playful framing makes it safer to express bold or unusual ideas. The shopping metaphor — a supermarket aisle full of colorful boxes competing for attention, or a tradeshow booth — taps a mental model participants already use as shoppers.

How to

Prep

✦ AI Prompt

1. Recruit four to eight target participants.

Recruit from the segment you are actually trying to serve. Mixing segments produces noise: participants from different segments want different things on the front of the box, and you will not be able to tell whether a recurring feature reflects a shared priority or one dominant voice. If you have access to more than eight participants, run two sessions in parallel rather than one large one.

2. Gather the materials.

Provide a blank white cardboard product box per participant or small group, colored markers, crayons, colored pencils and pens, glitter glue, plain and colored paper, and stickers — including word stickers like “New” or “Exciting” alongside stars and decorative images. Set out two or three sample product boxes per table for reference so participants have a baseline mental model of what a finished box looks like, and cover each table with butcher paper so the table itself becomes part of the working surface.

3. Set up the room.

Arrange tables so participants can work individually or in groups of two to three, with enough room to lay supplies out and walk around the box as they build. Place the sample boxes and craft supplies within reach. If you need participants to work without copying each other during construction, position the tables so participants cannot easily see neighboring boxes — but keep the room open enough that the sell-it-back presentations can happen with everyone watching.

4. Decide on the framing you will use.

You will introduce the exercise with a metaphor. Two work well: the supermarket shelf (“imagine this product on the shelf at your favorite store”) and the tradeshow, retail outlet, or public market (“imagine this product at a tradeshow booth”). Pick whichever fits your category better and stick to it.

Execution

1. Set the context with the metaphor, not the feature list.

Open the session with the metaphor you chose in Prep — the supermarket shelf or the tradeshow booth. Briefly describe the product concept or category. Do not describe specific features. The goal is to learn what participants want, not to validate your feature list. Say something like: “Imagine this product is on a store shelf next to the others you already buy. Design the box that would make you want to pick it up and take it home.”

2. Let them build.

Allow about thirty to forty-five minutes for participants to design their boxes. Circulate and observe but do not guide or suggest. If asked what to include, say “whatever would make you want to buy it.” Take notes on which supplies participants reach for, who works individually versus collaboratively, and what they pause on.

3. Run the sell-it-back, participant standing.

Have each participant or small group present their box to the room, allowing about five to ten minutes per box. This is where prioritization happens: pitching the box forces participants to decide which features lead and earn a spot on the front, which are mentioned in passing or relegated to the back, and which are dropped entirely. The staging is part of the method: make certain the

TOOLS

Blank white cardboard product boxes — one per participant or small group.

Craft supplies — colored markers, crayons, pens, glitter glue, plain and colored paper, stickers (including word stickers like “New” or “Exciting”), two or three sample boxes per table for reference, and butcher paper to cover each table.

[Miro](#)

[FigJam](#)

participant is standing and that you, the facilitator, are sitting while they present. The participant is selling the imagined product to you, so they take the presenter's posture and you take the audience's, rather than answering questions from an authority. Encourage others to ask questions as buyers, not as critics — the audience's reactions add another layer of insight.

4. Facilitate a closing discussion.

After all presentations, lead a short group discussion about common themes, surprising ideas, and disagreements. Useful prompts:

- "What did you see on other boxes that you wish you had included on yours?"
- "What was the hardest decision about what to put on the front?"
- "What did you decide to leave off entirely, and why?"

5. Document everything.

Photograph every box from all sides. Record the presentations with consent. Note key quotes, recurring themes, and any features or claims that appeared on multiple boxes.

Analysis

✦ AI Prompt

Analyze the boxes across multiple dimensions:

- **Frequency.** Which features or messages appeared on the most boxes? These are likely core value propositions, especially when they show up across multiple participants who did not see each other's work.
- **Placement.** Features on the front of the box are what participants consider the primary selling point. Back-of-box features are supporting. Side-of-box and fine-print details are nice-to-haves.
- **Aspirational language.** The slogans and imagery participants choose reveal the outcomes they care about, which may differ from the features they list. A box that puts "weekends back" on the front is selling a different value proposition than one that puts "20 integrations" on the front, even if the feature set is identical.
- **Pricing signals.** If participants include pricing on their boxes, note the range and how they frame it (one-time purchase, subscription, premium versus budget positioning).
- **Sticker patterns.** Where participants used pre-made word stickers like "New" or "Exciting," note which features earned the sticker. The unprompted decision to mark a capability as "new" or "exciting" is a signal independent of the descriptive copy.

- **Surprises.** Features or messages that you did not anticipate are often the most valuable findings. These can represent unmet needs or positioning angles your team has not explored. Compare them against your current roadmap and marketing copy – gaps are where to dig next.

Biases & Tips

Craft skill variance

Some participants are more comfortable with arts and crafts than others. Emphasize at the start that artistic quality does not matter – it is the ideas that count. Provide pre-made word and image stickers so less-confident drawers can still express specific features.

Selection bias

The exercise appeals to creative, expressive participants. Quieter or more analytical participants may be underrepresented. Pair them with a more outgoing partner in small-group formats, and weight the analysis by what shows up across multiple boxes rather than what the most expressive participant emphasized.

Facilitator-stance bias

If the facilitator stands during the sell-it-back, the participant tends to shift into “answering an authority” mode rather than “selling to a buyer” mode, which flattens the signal. Keep the participant standing and the facilitator seated.

Next Steps

- Turn the front-of-box features and slogans into draft positioning copy, then run a [Value Proposition Test](#) with a different group of customers to validate which messaging resonates.
- Use a [Buy a Feature](#) to see whether customers will spend scarce play money on the features they put on the box.
- Use a [Card Sorting \(Features\)](#) to rank the new features customers invented during the box exercise.
- Use a [Speed Boat](#) to surface the pain points that the aspirational box messaging implicitly promises to solve.

Case Studies

Qualcomm: Product Box for trucking-fleet customers

Luke Hohmann ran Product Box (and Speedboat) workshops with Qualcomm's trucking-fleet customers around 2003; customer box designs informed the feature set and positioning of the DataVisor business-intelligence platform, and Qualcomm executive Joan Waltman coined the term "Innovation Games" for the technique. (corroborated by the DataVisor launch announcement, Commercial Carrier Journal, 2006)

[Read more](#)

Further reading

-  Hohmann, Luke. Innovation Games: Creating Breakthrough Products Through Collaborative Play. Addison-Wesley, 2006. The primary source for Product Box, including the sell-it-back step, the supermarket-shelf framing, and the role-reversal staging during presentations. One of Hohmann's twelve Innovation Games.
-  Product Box — Luke Hohmann
-  Innovation game — Wikipedia

5.10.4 Speed Boat

In Brief

Speed Boat surfaces and prioritizes pain points by having participants hang “anchors” below the waterline of an imagined boat. A boat is drawn on a whiteboard or shared digital canvas; the boat is the product or service, and the anchors are the things slowing it down. Participants write each frustration, complaint, or obstacle on a sticky note and place it as an anchor at a depth that represents how badly it slows the boat down — the deeper the anchor, the more debilitating the pain. The result is a depth-ranked map of pain points, ranked by severity and the language participants use.

COMMON USE CASE

You have an existing product and a small group of target customers you can convene for about an hour. You suspect customers are sitting on frustrations they have not voiced through support tickets, surveys, or interviews — and you need to know which of those frustrations are debilitating enough to justify product investment before any new feature work. The output is a depth-ranked map of customer pain points, with relative severity, clusters of shared pain, and the customer language describing each one.

HELPS ANSWER

- What are the biggest pain points in our current product or service?
- Which frustrations are most debilitating to customers?
- Are there pain points shared across customers, or are issues segmented?
- What is slowing adoption or retention?
- What should we fix before adding new features?



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal

TIME

 ~3 days–2 weeks

COST

 \$0–\$400

OTHER NAMES

Speed Boat · Speedboat

Description

Speed Boat combines two mechanisms: a metaphor that makes negative feedback feel safe, and a spatial layout that forces participants to weigh the relative severity of their complaints. It is the [Prioritization](#) game that ranks pains rather than features. The metaphor lowers the social cost of criticism — instead of attacking a product or team, participants are adding anchors to a boat, which shifts the conversation from blame to mapping. The layout then forces a comparison most feedback channels never ask for. In surveys, support tickets, and interviews, people report issues one at a time without weighing their relative impact. When a participant places an anchor at medium depth and then sees a peer place a different issue deeper, they must decide whether their issue is really less important or should move down. That negotiation produces a more calibrated picture of frustration than any single channel collects.

Speed Boat surfaces pain points, not features. Once you know which anchors are deepest, [Buy a Feature](#) and [Card Sorting \(Features\)](#) tell you which features participants would actually fund to remove them.

The boat-and-anchors metaphor is also used in team retrospectives, where it is usually called Sailboat. This page covers pain-point research with the people you serve — not the retrospective variant.

For a more complete picture, you can optionally add wind or engines above the waterline to represent positive drivers — features and experiences that propel the product forward. The balanced view prevents the exercise from becoming purely negative and highlights strengths worth protecting. If your primary goal is to surface pain points, keep the focus on anchors and save the positive framing for a separate exercise.

How to

Prep

[AI Prompt](#)

1. Recruit the participant group.

Recruit from the segment whose pain you are trying to map. Mixing segments produces noise: different segments hit different anchors, and you will not be able to tell whether a recurring pain reflects a shared issue or one dominant participant. If you have access to more people than one small group can hold, run multiple sessions in parallel rather than one large one.

2. Gather the materials.

You need a whiteboard or large sheet of paper (or a shared online whiteboard for remote sessions), a stack of sticky notes or index cards, markers for every participant, and tape if you are working on paper. For digital sessions, prepare a Speed Boat template with the boat drawn, the waterline, and a chain hanging below.

3. Draw the boat and the waterline.

On the whiteboard, draw a simple boat sitting on the water. Draw a clear horizontal waterline. Below the waterline, draw an anchor chain hanging down with empty hooks or pegs at varying depths. Label the area below the waterline “anchors.” If you plan to run the optional engines/wind variant, also label the area above the waterline “engines” or “wind.” Keep the drawing simple — a child’s-book sailboat is enough; the diagram is a placeholder for sticky notes, not the artifact.

4. Decide on the framing you will use.

You will introduce the exercise with one short metaphor speech: the boat is your product or service, and the things slowing it down are its anchors. Write your version of that speech down so the facilitator delivers it the same way each session. Keep it neutral and do not seed specific pain points.

Execution

1. Open with the metaphor, not a feature list.

Read the framing speech you drafted in Prep. Keep it short. The boat is the product or service, the anchors are the things slowing it down, and deeper anchors are more debilitating. Do not list specific pain points the team has already discussed; the goal is to learn what participants want to surface, not to validate your existing list.

TOOLS

Whiteboard or large paper, sticky notes, markers — the analog kit is enough.

[Miro](#)

[FigJam](#)

2. Time-box silent anchor writing.

Hand each participant sticky notes and a marker. Ask them to write one pain point per note — a brief phrase or sentence — and to write as many as they can think of. Time-box this to a few minutes of silent writing before any anchors are placed. Silent writing first is the single most effective protection against groupthink: once a participant sees others’ anchors on the board, recall narrows toward what is already there.

3. Place the anchors with depth as severity.

One at a time, each participant reads their anchor aloud, explains it briefly, and places it below the waterline. Deeper placement means more debilitating. Encourage participants to place their anchors relative to others already on the board — if the new anchor is more painful than the current deepest one, it goes deeper; if less painful, it goes higher. The act of comparing and placing is where the calibration happens.

Optionally, ask participants to estimate how much faster the boat would go if a given anchor were cut, as a percentage, a multiplier, or a qualitative phrase, and write that estimate on the card. These speed estimates read as a pain proxy in Analysis.

4. Observe; do not validate.

The facilitator's job during placement is to observe and take notes, not to react. Do not validate or explain anchors. Do not push back on placements. Do not explain features that are mentioned. If a participant asks whether something "counts" as an anchor, the answer is: if it slows you down, place it. Take notes on who places first, who hesitates, what arguments come up about depth, and what gets clustered together.

5. Optional: add engines or wind.

If you decided in Prep to include the positive variant, repeat the process above the waterline once anchors are placed. Ask: "What is propelling this boat forward? What do you love?" The same silent-writing-then-placement sequence applies. Keep the time-box tight — engines are a balance, not the focus.

6. Photograph and document.

Take a photo of the final board. Transcribe every anchor with its relative depth, any speed estimate written on the card, and which participant placed it. Record key quotes from the discussion. The board is the artifact, but the verbal context is where the explanation lives.

Analysis

✦ AI Prompt

The spatial layout provides immediate visual prioritization. Read the board across several dimensions:

- **Deep anchors** represent the most debilitating issues. These are the candidates for urgent attention and the strongest signal that a fix would meaningfully change the participant's experience.
- **Clusters of anchors** at any depth indicate widely shared pain points. Even moderate-depth anchors that appear on multiple participants' notes may warrant priority — convergence across participants matters.

- **Isolated deep anchors** may represent severe issues for a specific segment rather than the whole group. Note the participant and explore in follow-up whether they represent a high-value segment with shared but unspoken pain.
- **Speed estimates** (if collected) work as a pain proxy, not an engineering estimate. A “10x faster” anchor next to a “10% faster” anchor tells you which one participants feel is debilitating versus annoying, regardless of how much engineering each fix actually requires.
- **Coalition patterns.** Pay attention to which anchors several participants nodded along with, asked about, or moved deeper after a peer placed theirs. Anchors that triggered visible recognition across the room are the ones likely to convert into product investment.
- **The gap between anchors and engines** (if used) shows the overall balance of frustration versus satisfaction. A board heavy on anchors with few engines suggests a product at risk of churn.

Biases & Tips

Groupthink

Once the first few anchors are placed, later participants may cluster around similar themes and miss their own original pains. Mitigate by having every participant write their anchors silently before any are placed on the board.

Depth calibration

Different participants use depth differently. Some place everything deep; others stay conservative. The relative positioning within each participant’s set matters more than absolute depth across participants. Note who tends to go deep and weight accordingly.

Recency bias

Participants tend to report frustrations from their most recent experiences. Ask explicitly: “Beyond what happened this week, what has been frustrating you for months?”

Dominant participants

A confident or senior participant can set the tone for what counts as a “real” anchor and how deep is acceptable. Silent writing first protects against this; in placement, watch for participants who soften or move their anchors after a dominant peer speaks.

Emotional saturation bias

A session designed to surface complaints can tip into venting fatigue, suppressing honest depth-calibration as participants disengage. Watch for signs the room has locked into frustration rather than mapping it — quality of new anchors drops and participants stop comparing depths — and pivot to the engines exercise or close with a forward-looking question if this sets in.

Facilitator
validation

Nodding, agreeing, or restating an anchor in better language signals which pains the team already cares about, and participants will tilt toward those. Take notes silently and resist the urge to coach.

Next Steps

- Run [Discovery Interviews](#) on the deepest anchors to investigate root causes and assign an owner before any feature work.
- Use a [Pain Point Sorting](#) to validate that the deepest anchors are widespread pain across more customers, not just this room.
- Use a [Buy a Feature](#) to test whether customers will spend scarce budget on features that remove the deepest anchors.
- Use a [Product Box](#) to see whether customers would put the relief from these pain points on the front of the box.

Case Studies

Qualcomm: Speedboat for telematics-data pain

Luke Hohmann facilitated Speedboat workshops with Qualcomm's trucking-fleet customers around 2003 to surface anchors around interpreting and managing the flood of telematics data coming off SensorTRACS and OmniTRACS; the resulting pain points drove the requirements for Qualcomm's DataVisor business-intelligence platform, launched in 2006. (corroborating product launch: Commercial Carrier Journal, 2006)

[Read more](#)

Further reading

-  Hohmann, Luke. Innovation Games: Creating Breakthrough Products Through Collaborative Play. Addison-Wesley, 2006. The primary source for Speed Boat, including the boat-and-anchors metaphor and the use of anchor depth as a measure of pain. One of Hohmann's twelve Innovation Games.
-  Speed Boat — Luke Hohmann
-  Innovation Games (2006) — Luke Hohmann
-  Sailboat — Retrospective Wiki
-  Innovation game — Wikipedia

5.11 Solution Interview

In Brief

You walk a prospect through a proposed solution — a prototype, mockup, or demo — and listen for what works, what is missing, and what they would pay. The output is qualitative feedback that defines the minimum feature set, tests whether the approach actually solves the problem, and informs pricing.

COMMON USE CASE

You have identified a clear customer problem through earlier research and now have a prototype or mockup of your solution. You want to sit down with potential customers, walk them through it, and learn whether this solution actually addresses their needs, what features are essential, and what they would pay for it.

HELPS ANSWER

- What is the minimum feature set needed to launch?
- Which features are must-haves versus nice-to-haves?
- Does the solution actually work for the target user?
- What are the strengths and weaknesses of this approach?
- Is the messaging and presentation clear?
- Is the price right compared to alternatives?
- How does this fit into the customer’s existing tools and workflow?

Description

A solution interview puts a proposed solution in front of the people who have the problem and watches how they respond. You walk a prospect through a proxy for the product, ask open-ended questions, and listen for whether the approach actually



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal



TIME  ~3 days–2 weeks

COST  \$0–\$300

OTHER NAMES

Product Interview · Solution
Validation Interview

solves their problem, which features are essential, and what they would pay. It is the step you run once you have identified a real problem — usually through

[Discovery Interviews](#)

— and want to know whether a specific solution addresses it for your early adopters.

The interview runs against a proxy for the product: screenshots, a clickable prototype, a storyboard, or a hand-drawn mockup. The proxy gives the participant something concrete to react to instead of a verbal pitch, which keeps the conversation grounded in the solution rather than in your description of it. Two outputs matter most: the minimum set of features needed to solve the problem, and a price the prospect signals they would pay.

The fidelity of the proxy should match the question — rough enough that the conversation stays on whether the solution works, not on visual polish. For a smaller-dollar product, a [Landing Page Test](#) may be a faster way to gauge demand; a solution interview is useful for preparing that test or for understanding why it underperformed. For a larger-dollar product the interview can feel like a consultative sale, but its job is research — adjusting the solution to real needs, not closing a deal.

Run solution interviews in pairs. One person asks the questions and builds rapport; the other takes notes and watches for reactions the speaker will miss.

AI design tools generate interactive prototypes from a text description, and for software products you can stand up a working slice that simulates the core value proposition, giving the participant something tangible to react to. The interview itself stays a human conversation. Do not hand it to an AI chatbot: the signal you are after — hesitation, enthusiasm, confusion, the unprompted follow-up question — needs an interviewer who can read the room and follow an unexpected thread.

How to

Prep

[AI Prompt](#)

1. **Define what you want to learn.** Pick a single, focused learning goal — for example, which of three features participants consider essential, or whether a proposed price is in the right range. If you can't state the goal in one sentence, you're not ready to interview.
2. **Formulate a research hypothesis.** Write down what you currently believe is true (for example, that mid-market finance teams will pay a set monthly price per seat for automated month-end close) so the interview can falsify it, not just confirm it.

3. **Recruit participants who fit the segment.** Re-screen for the actual problem before booking time — feedback from someone who does not have the problem generalizes to nothing.
4. **Prepare the proxy.** Use whatever fidelity is just sufficient to provoke a real reaction: a paper sketch, a clickable mockup built in a prototyping tool, or a working slice for a software product. Keep it rough enough that the conversation stays on whether the solution addresses the problem, not on visual polish.
5. **Write the interview script.** Decide the story you will use to set the problem context, the prompts for walking through the solution, and the open-ended questions you will ask afterward. Plan the script to test the concept without selling it, and decide who asks and who takes notes.

Execution

1. **Greet and screen.** Welcome the participant and confirm they fit your criteria and have the problem you want to solve.
2. **Set and confirm the problem.** Use the problem-context story from your script, then reflect the problem back and have the participant confirm or correct it before you show anything.
3. **Walk through the solution.** Present or demo the proxy and let the participant react. Do not explain features unprompted — ask what they make of it.
4. **Probe pricing.** Where relevant, discuss price, particularly relative to the alternatives the participant already uses or pays for.
5. **Ask about next action.** Ask what they would need to do to make progress, which surfaces real intent better than asking whether they like it.
6. **Close and follow up.** Get permission to follow up and ask for referrals to others who have the problem.
7. **Capture results.** Record reactions, quotes, and pricing signals immediately — a small pocket notepad works if you are interviewing on the street.

TOOLS

Paper and a pen — the fastest proxy for an early concept; sketch the screen or flow and react to it live, no tooling required.

[Figma](#)

[User Interviews](#)

[Dovetail](#)

[Looppanel](#)

Analysis

✦ AI Prompt

1. **Separate must-haves from nice-to-haves.** Across sessions, mark which features participants leaned in on, asked about unprompted, or named as the reason they would adopt. Those are must-haves; features that drew mild approval but did not drive the adoption decision can be deferred.

2. **Flag the dead reactions.** Note features that drew confusion, indifference, or “why would I need that?” Decide whether to redesign, remove, or test them differently next round.
3. **Read the pricing signal.** Pull both direct price comments and indirect cues — comparisons to alternatives, cost-saving framing, hesitation — into a range to test next, rather than a single number.
4. **Collect the quotes.** Save the strongest verbatim quotes for each must-have feature and for the pricing theme; they carry more weight with your team than your summary will.
5. **Treat the data as directional.** With a small sample you are looking for patterns that repeat across participants, not statistical proof. Decide which findings are strong enough to act on now and which need more interviews before you commit.

Biases & Tips

Leading-question bias

Closed or leading questions prime participants to agree rather than reveal how they behave. “Would you use a dashboard like this?” invites a polite yes; “Walk me through how you’d use this” surfaces a real reaction. Watch for questions that contain the answer you hope to hear.

Social-desirability bias

Participants soften criticism to be polite, especially face to face with the person who built the thing. Make it easy to be negative (“what would stop you from using this?”), and weight what they do — the unprompted question, the wince — over what they say.

Confirmation bias

You will hear the comments that support your idea more loudly than the ones that undercut it. Write the falsifiable hypothesis before the round, have the note-taker log disconfirming reactions verbatim, and review the dead and confused reactions before the enthusiastic ones.

Pitch creep

Once you start selling, you stop listening. The moment you catch yourself explaining a feature unprompted, you have shifted from researcher to salesperson and the participant’s answers shift with you. Stop and ask a question instead.

Imagined-future bias

Asking how someone would feel about a hypothetical feature produces unreliable answers. Give them something concrete to react to, and watch what they actually do with it.

Selection bias

Interviewing whoever is easy to reach — friends, your existing network, the most enthusiastic responders — produces feedback that does not generalize. Re-screen every participant for the actual problem and the target segment before booking time.

Next Steps

- If the solution resonates, build a minimum viable product or run a [Pre-Sales Test](#).
- If reactions are lukewarm, iterate on the solution and re-interview using [Discovery Interviews](#).
- Run a [Card Sorting \(Features\)](#) to separate must-have from nice-to-have features from interview feedback.
- Use solution interview insights to write your [Value Proposition Test](#) and landing page copy.
- Use a [Demo Pitch](#) to test whether the solution generates real purchase commitment, not just positive interview feedback.
- Run a [Landing Page Test](#) to evaluate whether the value proposition language from interviews converts strangers who have never spoken with you.

Case Studies

ProductPlan: 30 interviews before writing code

ProductPlan ran 30 customer-validation interviews with product managers across company sizes before writing any code, and recommends a baseline of at least 10 problem-discovery and 20 product-validation interviews.

[Read more](#)

Further reading

-  Steve Blank — The Four Steps to the Epiphany, K&S Ranch, 2013 (5th edition, ISBN 978-0989200509). Origin of the customer-development model and the framing of solution interviews as evidence-gathering, not selling.
-  Steve Blank & Bob Dorf — The Startup Owner's Manual, K&S Ranch, 2012 (ISBN 978-0984999309). Step-by-step expansion of customer development including solution-interview structure and pass/fail criteria.
-  Rob Fitzpatrick — The Mom Test, CreateSpace, 2013 (first printing, ISBN 978-1492180746). Practical playbook for keeping interview conversations grounded in real behavior rather than hypothetical praise.

- ☰ Cindy Alvarez — Lean Customer Development, O'Reilly, 2014 (ISBN 978-1449356354). Field guide for running solution interviews, recruiting interviewees, and synthesizing findings into product decisions.
- ☰ Dan Olsen — The Lean Product Playbook, Wiley, 2015 (ISBN 978-1118960875). Maps solution interviews onto the product-market-fit pyramid and ties feedback directly to MVP feature-set decisions.
- ☰ How to Generate Valuable Insights From a Solution Interview
- ☰ The Sales Pitch is Dead. Long Live Solution Interviews!
- ☰ LeanSteps: Solution Interviews
- ☰ How Superhuman Built an Engine to Find Product-Market Fit — First Round Review
- ☰ Idea-Validation Framework, Backed by 100+ Founder Interviews — Consumer Startups

5.12 Storyboard the Solution

In Brief

Storyboard the Solution is a comic-strip narrative — typically 4 to 8 panels — that draws a proposed solution into the context of a customer’s real life. Unlike wireframes or prototypes that show the product in isolation, a storyboard shows the entire journey: the trigger event, the decision to use the product, the experience of using it, and the outcome. You show this narrative to people in your target segment and listen for confusion, skepticism, excitement, and corrections that refine how the solution should work in context — not a yes/no verdict, but the reactions that reshape your design.

COMMON USE CASE

You have validated that a problem is real and you have a proposed solution in mind, but you have not yet built wireframes or a prototype. You want to test whether the end-to-end journey — trigger, decision, action, value, outcome — makes sense to customers before committing engineering effort. The output is a corrected storyboard plus a list of participant objections, missing panels, and reframings that should change your design.

HELPS ANSWER

- Does the proposed user experience make sense to customers?
- Is the trigger event realistic — would customers actually seek this solution at that moment?
- Are there steps in the journey that feel confusing, unrealistic, or unnecessary?
- What do participants expect to happen that the storyboard does not show?
- Does the outcome depicted match what the customer actually wants?



AT A GLANCE

Qualitative

In-person

Remote

Attitudinal

Observational



TIME  ~3 days–2 weeks

COST  \$0–\$300

OTHER NAMES

Solution Storyboard · Storyboard

Description

Storyboard the Solution is a sequence of rough drawn panels showing your proposed solution in use, from the moment that triggers it through to the outcome it produces. Drawing it forces you to think through the entire experience, not just the product interface. The payoff starts before anyone else reviews it: many product concepts fall apart when you try to draw the panel before the participant opens the app and the trigger event turns out to be implausible, or the panel after they finish and the outcome turns out to be underwhelming. A sequence of rough panels exposes narrative leaps that prose, slide decks, and feature lists hide.

Showing the storyboard moves the conversation beyond “would you use this?” (which usually gets a polite yes) to the specific experience depicted. Participants react to the narrative naturally: “That’s not how it would happen for me,” “I wouldn’t do it at that point, I’d do it when...,” “What happens after this panel? Does it also...?” These reactions are the data you are looking for.

 **Storyboard the Pain** is the companion technique: it visualizes the problem before any solution exists. Run it first to confirm the problem, then run Storyboard the Solution once you have a concept worth testing.

How to

Prep

 AI Prompt

1. Define the customer journey you are depicting.

Pick one specific customer segment and one specific scenario. Storyboards fail when they try to show the general case — the panels become abstract and participants cannot react to them. Map the journey as five beats before you start drawing: **trigger** (the moment they notice the problem), **decision** (the moment they reach for a solution), **action** (the core interaction with your product), **value** (the moment they get what they came for), and **outcome** (what life looks like after). These five beats become the spine of your 4-to-8 panel storyboard.

2. Draft the first and last panels first.

The first panel is the trigger; the last panel is the outcome. Get these two right and the middle panels follow naturally. An implausible first panel undermines the whole journey, and an underwhelming last panel cannot be fixed by adding features.

3. Fill in the middle panels.

Use 2 to 6 middle panels to show how the customer moves from problem to outcome: how they discover the solution, their first interaction with it, the key moment of value delivery, and the transition to the outcome. Each panel should advance the story by one meaningful step.

4. Show the person, not just the product.

Every panel should include the user and their context — at a desk, on the bus, in a meeting. The product is a supporting character, not the star. A storyboard that shows only screens is a wireframe with extra steps; it loses the narrative leverage that makes the method work.

5. Keep the artwork rough and the text sparse.

Stick figures are perfectly effective. Polished illustrations distract participants into commenting on the art rather than the experience. A short caption or thought bubble per panel is enough — the storyboard should be immediately understandable without reading paragraphs of text. AI image generators can produce panels in minutes, but the polish often hurts the conversation; if you use them, ask for sketch-style output and resist the urge to refine.

6. Write the two open questions you will ask per panel.

The session runs on two questions you ask after the participant narrates each panel: one that tests whether the panel is realistic (“Does this feel realistic? What would actually happen differently for you?”) and one that tests what the participant expects next (“What would you expect to see next?”). Phrase them as open, non-leading questions and keep them fixed across sessions so reactions are comparable.

7. Review internally before you run a session.

Show the storyboard to a team member who was not involved in creating it. If they cannot follow the story without explanation, revise for clarity. The storyboard has to stand on its own — in the feedback session, you will not be allowed to narrate it for the participant.

Execution

1. Recruit 5 to 10 participants from the target segment.

You are looking for the same kind of participant you would recruit for a [Solution Interview](#): people who have the problem, in the context the storyboard depicts. Five sessions is the minimum for any pattern to emerge; ten is the point at which corrections start to repeat.

TOOLS

Pen and paper
(simplest and most effective)

[Miro](#)

[Storyboard That](#)

[Figma](#)

2. Walk through the storyboard one panel at a time.

Show one panel at a time, in order. Ask the participant to narrate what they see — do not explain it yourself first. Their narration is data: where they hesitate, what they assume, what they ignore. After they have described the panel in their own words, ask the two questions you prepared in Prep. Avoid validating their interpretation. Your job is to surface their mental model of the journey, not to defend yours.

3. Observe corrections, extensions, and objections.

Note four classes of reaction in real time:

- **Confusion** — “What’s happening here?” The panel does not communicate; rewrite or replace it.
- **Skepticism** — “I wouldn’t do it that way.” The journey beat does not match the participant’s mental model; investigate.
- **Excitement** — “Oh, that would be great.” The panel resonates; treat as a candidate for the value moment in your messaging.
- **Correction** — “Actually, the problem for me starts earlier/later.” The participant is rewriting your storyboard for you; capture the rewrite verbatim.

4. Iterate between sessions when corrections converge.

If three participants in a row correct the same panel the same way, revise the storyboard before continuing. The point of running 5 to 10 sessions is to converge on a journey that resonates, not to collect uniform feedback on a fixed artifact. Two to three rounds of iteration typically produce a storyboard that stops generating surprises.

Analysis

[★ AI Prompt](#)

The storyboard is generating useful data when participants engage with the narrative — arguing with it, correcting it, extending it. If they simply nod and say “looks good,”

either the storyboard is too vague to react to or the participant is being polite. Push for specifics: “Walk me through your version of panel 3.”

Categorize each correction into one of four buckets, then read the pattern across participants:

- **Broken trigger.** Participants tell you the trigger event is unrealistic, mistimed, or attached to the wrong emotion. If multiple participants correct the same trigger panel, your go-to-market assumptions may be wrong even if the product is right. Fix the trigger before anything else — every later panel depends on it.
- **Missing decision logic.** Participants describe a step between the trigger and the action that is not on the storyboard (“first I would check whether...”). Your journey is skipping the part where the participant talks themselves into using the product. Add the missing panel; that is often where adoption actually breaks.
- **Feature-set mismatch.** Participants want to skip a panel (“I would not need this step”) or ask what happens behind a panel (“does it also...”). The product scope you are depicting is wrong — too much in some places, not enough in others. Tighten the action panels to what the product will actually do at launch.
- **Outcome inflation.** Participants want the outcome panel to show more than your solution delivers. There is a gap between expectation and product scope, and the storyboard is leaking promises the product cannot keep. Either pull the outcome panel back to what is real, or expand the solution.

Convergent corrections across all four buckets tell you the concept is salvageable but the journey needs work. Corrections that cluster in only one bucket — usually trigger or outcome — tell you to fix that one thing and re-test before changing anything else. No corrections at all tell you the panels are too abstract, the artwork is too polished, or the participants are being polite; tighten the panels and recruit a new round.

The highest-leverage solution-design changes are the ones that resolve corrections in the trigger and decision buckets, because errors there propagate through every later panel. Action and outcome corrections are usually local — they change one or two panels, not the whole journey.

Biases & Tips

Narrative bias

A well-told story is persuasive regardless of whether the underlying product works. Participants may respond to the quality of the storytelling rather than the feasibility of the experience. Use rough sketches to keep the focus on substance.

Anchoring

The storyboard frames the conversation around your proposed journey. Participants may accept the framing rather than imagining alternatives. Ask explicitly: “How would you do this differently?”

Selection bias

Participants who agree to look at storyboards may be more imaginative and open to new products than your general target market. Pair the storyboard sessions with a recruiting screener that does not advertise the topic.

Overgeneralization from a single scenario

One storyboard covers one journey for one persona. Corrections from that session describe that scenario, not the whole segment. When corrections cluster around segment or trigger differences, create a second storyboard for the other case before drawing product conclusions.

Next Steps

- Revise the storyboard based on participant corrections and run a second round with new participants until the narrative stops generating surprises.
- Use a [Storyboard the Pain](#) to confirm the trigger panel reflects a real, validated problem before investing further.
- Use a [Solution Interview](#) to dig into the specific objections and corrections participants raised about the journey.
- Use a [Paper Prototyping](#) session to put the key panel of value delivery in participants’ hands once the story holds together.
- Use a [Demo Pitch](#) to test whether the storyboard’s outcome panel translates into a pitch customers would say yes to.

Case Studies

Savioke: Storyboarding a hotel-delivery robot

Ahead of a hotel pilot, Savioke ran a five-day Google Ventures design sprint that storyboarded the end-to-end guest-meets-robot delivery journey across lobby, elevator, and hallway, then tested the prototype with real guests at a Cupertino hotel — validating personality and interaction design before scaling.

[Read more](#)

Further reading

 [Storyboard — IDEO Design Kit](#)

- ☰ [Storyboard — Wikipedia](#)
- ☰ [Storyboards Help Visualize UX Ideas — Nielsen Norman Group](#)
- ☰ Knapp, Jake, John Zeratsky, and Braden Kowitz. *Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days*. Simon & Schuster, 2016. Day 3 of the design sprint is “Storyboard”; the book’s framing of the storyboard as the bridge between sketching and prototyping informs the trigger-to-outcome structure used here.

PART 6

Evaluative Product Experiments



“You’ve got to start with the customer experience and work back toward the technology – not the other way around.”

—Steve Jobs

CONTENTS

■ Dashboards

■ Net Promoter Score (NPS) Survey

■ Product-Market Fit Survey

■ Prototyping

■ Usability Testing

■ Wizard of Oz

6.1 Dashboards

In Brief

A dashboard is a visual display that brings key product and business metrics into a single view. You select the metrics that matter most — such as active users, conversion rates, revenue, or retention — and present them in a format the whole team can check at a glance. The output is a real-time picture of product health that surfaces trends and anomalies as they happen.

COMMON USE CASE

Your product is live and you need a single place where the whole team can see how things are going. You want to track key metrics like active users, conversion rates, and revenue in real time so you can spot problems early and make decisions based on data rather than gut feeling.

HELPS ANSWER

- What is happening in the business right now?
- How are key metrics changing over time?
- Do we have any major blind spots in our data?
- Does the team have access to the right metrics to track progress?
- Are we making the right decisions based on current data?
- What should our current priorities be?

Description

A dashboard is a single screen that brings a product’s or business’s most important metrics together so the whole team can read the current state at a glance. Instead of planning around forecasts, you watch what is actually happening now — active users, conversion, revenue, retention — and react when something moves.



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~2 weeks–3 months

COST  \$0–\$200

OTHER NAMES
Metrics Dashboard · KPI
Dashboard

A dashboard makes the relationships between parts of a business visible. A \$10k investment in a channel may seem like a lot of money until you put it next to last year's revenue of \$250k on the same screen. Dashboards also create focus: choosing one or a handful of metrics for the whole organization tells everyone what matters, and a shared screen everyone checks keeps that focus from drifting.

A dashboard can serve any level — the company as a whole, a department, a role (such as the VP of Marketing's dashboard), or an individual contributor. It can be anything from a manually updated spreadsheet to a custom system wired into your other business tools for a live view. Either way, the goal is the same: a reader should grasp the state of the business in seconds. The hard part is deciding what to leave off.

Choosing what goes on the dashboard is what most teams get wrong. "Vanity metrics" — total registered users, raw page views, cumulative downloads — climb steadily regardless of whether the product is improving, and they crowd out metrics the team could actually act on. Favor metrics that are actionable (a change tells you what to do), accessible (the whole team can understand them), and auditable (you can trust how they are calculated). Treat the dashboard as a prompt for investigation, not the answer.

How to

Prep

✦ AI Prompt

1. **Decide what the dashboard is for.** Write one sentence naming the decision the dashboard exists to support (e.g., "should we keep investing in this acquisition channel?"). If you can't name the decision, the dashboard will collect metrics no one acts on.
2. **Choose 5–7 candidate metrics that pass the actionability test.** A metric earns its place only if a movement changes what the team will do next week. For each candidate, write the action you'd take if it moved 20% up or down; if you can't name one, cut it.
3. **Confirm the data source is ready.** Check that the underlying events, metric definitions, and time-zone handling are stable enough to trust before you build on them.
4. **Decide audience, cadence, and access policy.** Whole-company, departmental, or role-specific? Reviewed in standup, weekly, or monthly? Public or restricted? Each combination changes what belongs on the screen.

5. **Pilot one metric before scaling.** Wire a single metric end-to-end (data → display → review ritual) and run it for a week. You'll surface the data-quality and visualization gotchas before they're embedded across seven panels.

Execution

✦ AI Prompt

1. **Lay out the metrics you chose in Prep.** Take the 5–7 metrics from your Prep shortlist — the ones that passed the actionability test — and organize them so the panel tells a coherent story. A useful starting framework for a SaaS or digital product is AARRR (pirate metrics): Acquisition (where do users come from?), Activation (do they complete onboarding?), Retention (do they come back?), Revenue (do they pay?), Referral (do they invite others?). Map your shortlisted metrics onto the stages they cover, aiming for the 1–2 most important per stage, and note any stage with no coverage as a gap to revisit.
2. **Choose your tool based on your stage.**
 - *Pre-revenue or very early:* a spreadsheet updated manually once a week. Free, flexible, forces you to touch the data.
 - *Product live with users:* a product analytics platform on its free tier, connected to a dashboard tool.
 - *SaaS with paying customers:* a subscription-metrics dashboard tracking recurring revenue, the rate at which customers cancel (churn), and the total revenue an average customer brings over their lifetime. Combine with product analytics for the full picture.
3. **Build the first version in under a day.** Don't over-engineer. A dashboard that exists and gets checked beats a perfect dashboard that takes weeks to build. Start with the most important 3–5 metrics and add more later.
4. **Make it visible.** Put the dashboard somewhere the whole team sees it daily — a shared screen, a chat channel with automated updates, or a pinned browser tab. A dashboard nobody checks is worse than no dashboard at all.
5. **Set review cadence.** Check the dashboard in your weekly team meeting. If a metric moves more than 20% week-over-week, investigate. Revisit which metrics are on the dashboard every quarter — remove what nobody acts on, add what's missing.

TOOLS

Mixpanel

Metabase

Amplitude

Google Analytics 4

Analysis

✦ AI Prompt

1. **Read the layout before the numbers.** Color, shape (a traffic-light red/amber/green), and status icons let you interpret a panel at a glance, and the

size of each panel should match how important the metric is — the most important numbers get the most space.

2. **Strip the chart down to the data.** Remove decoration that carries no information: 3D effects, drop shadows, and redundant gridlines. Every mark on the chart should encode a number, not dress it up.
3. **Check that the picture matches the data.** The size of an effect on the chart should match its size in the underlying numbers. A truncated axis that makes a 2% change look like a doubling is misleading; use zero-based axes (start the scale at zero) for value comparisons.
4. **Use a compact format for goal-tracking panels.** A bullet graph — a single horizontal strip showing the current value, the target, and a qualitative range (poor / fair / good) — packs a metric and its goal into one line and is a good default for the panels you check most.
5. **Investigate movement; don't just report it.** A metric that crosses its action threshold is the start of analysis, not the conclusion. Before acting, name the most plausible causes (including data-quality ones) and run one check that could disprove your leading explanation.

Biases & Tips

Confirmation bias in metric choice

Teams choose metrics that show their work in a flattering light and avoid metrics that would surface inconvenient truths. The dashboard then becomes a self-congratulation tool. Counter by including at least one metric whose movement could legitimately argue against the current strategy (e.g., qualified retention alongside signups, gross margin alongside revenue).

Automation bias

AI-generated summaries and green-status indicators create the illusion that the dashboard explains cause, not just correlates movement. The dashboard is the prompt for investigation; a metric that moves is the start of qualitative or causal analysis, not a conclusion to share.

Outdated metric definitions

Definitions drift silently. “Active user” may have meant 7-day login a year ago and now means any API call. When the underlying calculation changes without explicit migration, comparisons across time become invalid even though the chart looks continuous. Version your metric definitions and date the changes; treat any unannounced calculation change as a bug.

Dashboard-as-
decision-
substitute

A dashboard that exists and gets checked is not the same as a team that makes decisions from it. Status meetings can devolve into reading the dashboard aloud rather than acting on it. The bias is treating the act of monitoring as equivalent to the act of operating. Pair every recurring dashboard review with a decision template: what did the data tell us, what will we change, by when.

AI-amplified
garbage-in-
garbage-out

AI-powered analytics platforms surface insights from whatever data you feed them and present them with authoritative tone, including from broken pipelines and miscoded events. Automated anomaly detection generates alert fatigue when thresholds are wrong, and natural-language summaries can paper over fundamental data-quality problems. Audit the underlying data quality before trusting AI-surfaced patterns; treat AI insight quality as upper-bounded by data hygiene.

Next Steps

- If metrics reveal unexpected patterns, investigate with [Discovery Interviews](#) or other qualitative research.
- Use [A/B Testing](#) to run controlled experiments on the features or flows your dashboard highlights as underperforming.
- Run a [Product-Market Fit Survey](#) to complement quantitative dashboard data with a direct measure of customer attachment.

Case Studies

Geckoboard: AARRR pirate metrics example

Worked dashboard tracking acquisition, activation, retention, referral, and revenue in a single view, aimed at founders, marketing, and product teams reviewing daily.

[Read more](#)

VentureBeat: Six dashboards for startup CEOs

A startup CEO walks through six dashboards he reviews daily to operate the business.

[Read more](#)

Cal.com: Open startup dashboard

Cal.com publishes Metabase-powered public dashboards covering weekly merged pull requests, monthly active users, and team morale as part of its build-in-public commitment.

[Read more](#)

Spotify: Dashboard Quality Framework

Spotify's internal Dashboard Portal aggregates Tableau and Looker Studio dashboards and rates them Low, High, or Golden using automated Vital Signs checks plus a manual design checklist.

[Read more](#)

Instagram: Burbn analytics pivot

Burbn's usage analytics reportedly showed users gravitating to the photo-sharing flow, prompting the team to strip the app down to filtered photo posts and relaunch as Instagram.

[Read more](#)

Further reading

-  Eric Ries — The Lean Startup
-  Vanity Metrics vs. Actionable Metrics — Eric Ries, 2009
-  Stephen Few — Information Dashboard Design (2nd ed., Analytics Press)
-  Edward Tufte — The Visual Display of Quantitative Information
-  Avinash Kaushik — Web Analytics 2.0 (Sybex/Wiley, 2009)
-  Spotify's Dashboard Portal — InfoQ Sept 2024
-  Statistics How To: Misleading Graphs: Real Life Examples

6.2 Net Promoter Score (NPS) Survey

In Brief

A Net Promoter Score (NPS) survey is a single-question loyalty measurement that asks customers to rate, on a scale from 0 to 10, how likely they are to recommend your product or company to a friend. Respondents are grouped into Promoters (9-10), Passives (7-8), and Detractors (0-6), and the final NPS is calculated by subtracting the percentage of Detractors from the percentage of Promoters. The output is a trackable score you can repeat over time to measure whether satisfaction is improving, plus optional follow-up answers that explain the reasons behind each rating.

COMMON USE CASE

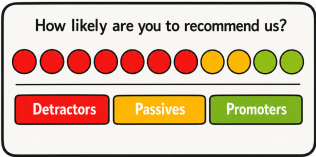
You launched a new feature three months ago and want to know whether your customers are happy enough to tell their friends about it. You need a simple, repeatable way to measure loyalty over time and figure out which customers are your biggest fans and which ones might leave.

HELPS ANSWER

- How loyal are our customers to this product?
- Which customers would recommend us to a friend?
- How do different customer groups feel about our product?
- Are customers becoming more or less satisfied over time?

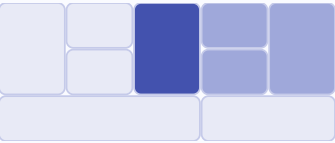
Description

A Net Promoter Score (NPS) survey is a single-question loyalty survey that asks people who have used your product one fixed question — how likely they are to



AT A GLANCE

- Quantitative
- In-person
- Remote
- Attitudinal



TIME ~1–3 weeks

COST \$ \$0–\$120

OTHER NAMES
NPS Survey · Net Promoter Score
· NPS

recommend it — on a 0-to-10 scale, then converts the answers into one score you can track over time. It is meant to measure loyalty and identify the customers most likely to recommend you to others.

The question wording does not change: ask “*How likely is it that you would recommend [company / product] to a friend or colleague?*” on a 0-to-10 scale. Sort the answers into three groups — Promoters (9-10, your most enthusiastic customers), Passives (7-8, satisfied but unexcited), and Detractors (0-6, unhappy). The score is the percentage of Promoters minus the percentage of Detractors, which lands somewhere between -100 and +100. Keeping the wording and the brackets fixed is what lets you compare one round against the next.

The number itself is only an anchor; the value comes from the open-ended “why” question you ask alongside it. Those written answers are where the useful themes live — what promoters praise, and what detractors want fixed.

The score tells you about past satisfaction, not future behavior, so pair it with what customers actually do — how much they use the product, who is at risk of leaving, who refers others — rather than reading it in isolation. Teams also tend to game NPS when they tie bonuses to it, chasing the number instead of acting on the feedback; treat the score as a prompt to do the follow-up work, not a target to hit.

How to

Prep

★ AI Prompt

1. **Write down why you’re sending this NPS.** “Track quarterly trend,” “find detractors before they churn,” and “benchmark against competitors” require different cadences and follow-up workflows. If you can’t name the decision the score will inform, you’re not ready to field it.
2. **Decide who to survey.** Pull the list from your CRM or product analytics, filtering for users past onboarding and active recently enough to have a real opinion. Gather enough qualified responses for the score to be meaningful — sampling noise stays large until you clear roughly 30 replies (see Analysis) — and if your user base is small, survey everyone who qualifies.
3. **Decide cadence.** Pick one and commit. *Relationship NPS* surveys the same cohort every 90 or 180 days and tracks the trend of overall loyalty. *Transactional NPS* is triggered by a specific event (purchase, support ticket close, feature first-use) and diagnoses which moments produce promoters and which produce detractors. Running both at once without segmentation muddies the signal.
4. **Write the survey.** The core question is standardized — don’t change it:

- “How likely are you to recommend [product] to a friend or colleague?” (0 = Not at all likely, 10 = Extremely likely)
- Add one open-ended follow-up: “What is the primary reason for your score?”
- Optionally: “What could we do to improve your experience?”
- Keep the NPS question first. If you embed NPS in a longer survey, the score is not comparable to standalone NPS.

5. **Plan the close-the-loop workflow before you send.** Decide who will contact every detractor and within what window (Bain’s standard is 24–48 hours). If nobody is on the hook to respond, don’t run the survey — collecting feedback you won’t act on burns goodwill with customers most likely to be saved.

Execution

✦ AI Prompt

1. Pick your distribution channel.

- **Email** (most common): Short message, 2–3 sentences, subject line like “Quick question about [product].”
- **In-app** (higher response rate): Trigger after the user completes a key action. In-app survey tools make this easy.
- **Post-interaction**: Send immediately after a support ticket is resolved or a key milestone is reached.

TOOLS

Sprig
Retently
Hotjar
Typeform

2. **Send and wait.** Give respondents 5-7 days to reply. Send one reminder after 3 days. Expect 10-30% of emailed customers to respond (the response rate), and higher for in-app or post-interaction triggers.
3. **Close the loop on detractors within 24–48 hours.** A real human contacts every detractor (0–6) to understand their specific issues. AI-powered platforms can automate the routing (detractors to support, passives to re-engagement, promoters to referral programs), but the human conversation with the detractor is non-negotiable.
4. **Don’t argue with the score.** When a detractor explains their reason, listen and capture; don’t defend the product. The verbatim is the data — debating it teaches the customer to stop responding.

Analysis

✦ AI Prompt

The original NPS calculation, per Reichheld and Bain, subtracts the percentage of Detractors from the percentage of Promoters and ignores Passives:

$$\text{NPS} = \% \text{ of Promoters} - \% \text{ of Detractors}$$

Before you read too much into the number, remember it is an estimate from a sample, not a census, so it carries a margin of error that shrinks as responses grow. At around 30 responses, sampling noise alone can swing the score by 10 points or more, so treat early readings as directional. A wave-over-wave change is only statistically meaningful when it is larger than that margin — with the small samples typical of early-stage startups, that usually means a move of several points on a few hundred responses, not a one- or two-point wobble on thirty.

A single score doesn't mean much on its own. Anchor it three ways:

- **Against your prior wave.** A 5-point movement across consecutive quarterly waves with the same cohort and same instrument is real signal. Smaller moves sit inside the noise floor.
- **Against industry benchmarks.** Published SaaS benchmarks put the median NPS around +36, with the strongest performers well above. Treat these vendor-published numbers as rough reference points, not targets — they are self-reported and inconsistently measured.
- **Against behavior.** Compare each NPS bracket against retention, account expansion, and referral activity in your own data. If your promoters do not actually retain or refer at higher rates than your passives, the score is not telling you much about loyalty.

The open-ended “why” is where the useful insight lives. Group the written answers by score bracket and look for repeated language: what promoters praise tells you what is working; what detractors complain about repeatedly tells you what to fix first.

Segment before you act on the headline. A single company-wide number blurs distinct cohorts. If you have meaningful groups — plan tier, acquisition channel, tenure, or region (kept within consistent geographic groupings, per the cultural-scoring bias below) — recompute the score within each. A +20 overall can hide a +50 enterprise cohort and a -10 free-tier cohort, and those two readings point at opposite actions. Only cut where you have enough responses per group (aim for 30 or more) to trust it.

Mine the Passives, not just the Detractors. Closing the loop on detractors is the canonical NPS move, but Passives (7-8) are usually the cheapest points to gain: they already like the product and sit one or two notches below recommending it. Their open-ended answers tell you the specific gap between satisfied and enthusiastic — often a single missing capability or unresolved friction. Investigating why a 7-8 isn't a 9-10 gives you a clearer, lower-effort path to lifting the score than winning back detractors who may never have been a fit.

AI analysis tools sort hundreds of written answers into themes and surface the most common reason behind each bracket in minutes. Spend your own attention on the outliers and the ambiguous answers the AI flags rather than reading every response one by one. Then have AI line the scores up against behavior — how often customers use the product, who is at risk of leaving, who refers others — to check that your promoters are actually driving growth.

Biases & Tips

Selection-timing bias

Sending the NPS survey shortly after a positive event (a successful onboarding, a closed support ticket, a new release) inflates the score; sending after a negative event deflates it. Pick a consistent trigger (e.g., 30 days after first paid month) and stick to it so scores are comparable over time.

Sample composition bias

The customers who do not respond are not represented in the score. The ones most likely to answer are usually your strongest promoters and your most frustrated detractors; passives tend to skip surveys, so the headline score over-represents the two extremes. Track the response rate alongside the score, and check for systematic gaps (for example, 50% response from your enterprise tier but 5% from your free tier).

Cultural-scoring bias

Cultures vary in their willingness to use the high end of the 0–10 scale. Western European and Japanese respondents systematically rate lower than US respondents on identical experiences; some Asian markets cluster around the middle. Compare segments only within consistent geographic groupings, and use change-over-time within a region rather than absolute scores across regions.

Detractor self-selection

A vocal minority of detractors will respond, write extensively, and dominate the verbatim analysis even when they represent a small share of the customer base. Weight detractor commentary by their actual proportion of the response set, and contrast against silent-majority behavior (usage, retention) before reorganizing the roadmap around their complaints.

Survey-position fatigue bias

Placing the NPS item after a long battery of questions lowers scores by 5–10 points because tired respondents rate less generously — making results incomparable to standalone NPS.

Next Steps

- Run [Discovery Interviews](#) with detractors to understand the specific issues behind their low scores.
- Interview your Passives (7-8) as well — they are the group most easily tipped into promoters. A few [Discovery Interviews](#) usually surface the one gap keeping them from a 9 or 10.
- If distinct cohorts show up, recompute the score per segment and track each on a [Dashboard](#); a healthy segment can hide inside a mediocre average.
- Use a [Product-Market Fit Survey](#) alongside NPS to learn whether detractors lack product-market fit or just have fixable complaints.
- Track NPS on a regular cadence and watch the trend on a [Dashboard](#) to measure how product changes move the score over time.
- Use the three brackets (promoters, passives, detractors) to tailor how you communicate with each group, and run [Discovery Interviews](#) with each to learn what drives their rating.

Case Studies

Apple Retail: Closing the loop on detractors

Reichheld and Markey's The Ultimate Question 2.0 (Harvard Business Review Press, 2011) cite Apple store managers calling detractors within 24 hours, with the closed-loop program credited for an estimated \$25M in additional annual sales.

[Read more](#)

Charles Schwab: 24-hour detractor follow-up

Bain cites Schwab alongside Apple Retail as leading NPS organizations that contact every customer who submits a negative rating within 24 hours, with frontline staff briefed to resolve issues.

[Read more](#)

Taylor & Hart: NPS as the one metric

The London bespoke jeweler tracked separate service and product NPS scores above 80, used the gap to overhaul manufacturing and logistics, and grew to roughly €4.5M annual revenue with a 70% revenue lift.

[Read more](#)

Manheim: Operationalized closed loop

The auto-auction operator collected over 10,000 transactional and relationship survey responses in 18 months, identified 110 improvement initiatives, and contacted detractors within 24 hours.

[Read more](#)

Groove: SaaS NPS primer

Groove's customer-support team published a general NPS playbook, a useful small-team reference for setting up a first survey; the survey tool the playbook describes (Delighted) is being discontinued, and the post itself is preserved on the Internet Archive.

[Read more](#)

CustomerGauge: SaaS NPS benchmarks

Self-published benchmark sets the SaaS median at +36, with Nutanix (92), Zoom (72), and Google (58) as top performers — a directional anchor for interpreting a first reading.

[Read more](#)

Further reading

-  The One Number You Need to Grow — Reichheld, HBR Dec 2003
-  The Ultimate Question 2.0 — Reichheld & Markey, HBR Press 2011
-  Introducing the Net Promoter System — Bain & Co.
-  Closing the Loop — Bain & Co.
-  Net Promoter Score — Wikipedia

6.3 Prototyping

In Brief

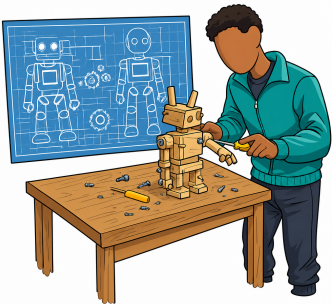
Product prototyping is a family of techniques for building representations of a product at varying levels of fidelity, from quick paper sketches to full-scale functional models. The purpose of every prototype is the same: to answer a specific question about your product with the smallest investment that gives you a real answer. Start at the lowest fidelity that can answer your current question, and increase fidelity only after the lower level confirms you are heading in the right direction.

COMMON USE CASE

When you have a candidate solution and you want to test whether it works before committing engineering or manufacturing budget. You want answers about flow, form factor, or core-feature pull, and you want them at the lowest fidelity that can give you a real signal — paper before clickable, clickable before single-feature MVP.

HELPS ANSWER

- Can users accomplish key tasks with this design?
- Does the physical form factor work for the intended use?
- Is a single core feature valuable enough to retain users?
- Does the full experience hold together when assembled from existing tools?
- Are there usability problems that need to be resolved before launch?
- Does the product work under realistic conditions?



AT A GLANCE

Quantitative

Qualitative

In-person

Remote

Observational



TIME  ~1 week–3 months

COST  \$0–\$50K

OTHER NAMES

Prototyping

Description

Product prototyping is a family of techniques for building a representation of a product so you can test it before committing to a full build. The representations range in fidelity from hand-drawn paper screens through to working full-scale models, and you choose the level that matches the question you need to answer. Each prototype tests whether a specific solution works: whether users can navigate it, whether the form factor feels right, or whether a core feature delivers enough value to retain users.

A common mistake is starting at too high a fidelity level: jumping to a clickable prototype or a functional MVP before confirming that the basic concept, flow, and value proposition work. This spends weeks or months on polished execution of a flawed concept, when a paper prototype shown to five users in an afternoon would have revealed the core problem.

The rule that prevents this: do not increase fidelity until the current level has confirmed your direction. If users cannot follow the basic flow on paper, building a clickable prototype will not fix the problem — it will only make the problem more expensive to discover.

Choosing the Right Prototype

Pick the lowest fidelity that can answer your current question. The seven methods below are grouped by fidelity from low to high. Move up the ladder only after the lower rung confirms your direction.

Low Fidelity

Paper Prototype

Hand-drawn screens or physical mockups that users walk through with a facilitator acting as the “computer.” Best used as the first round when the basic flow is still uncertain and you want answers in an afternoon.

Medium Fidelity

3D Print

A physical model customers can hold and examine to evaluate form factor and ergonomics. Best used when the open question is whether the physical form feels right before any manufacturing investment.

☒ Clickable Prototype

Interactive digital screens connected by click targets, with no real backend. Best used when users need to navigate detailed interactions and you want to run usability testing without writing code.

High Fidelity

☒ Single Feature MVP

A limited product version that delivers one core feature exceptionally well. Best used when the question is whether that feature alone provides enough value to retain users.

☒ Vibe-Coded Disposable MVP

Real working software generated by an AI coding assistant and deleted after the test. Best used when you need behavioral evidence from a live software product fast and have pre-committed to throwing the code away.

☒ Mash-up

A functional product assembled from existing third-party tools and services, capable of serving real customers without custom development. Best used when you need to test the full end-to-end experience before investing in a custom build.

☒ Life-Sized Prototype

A full-scale prototype with working mechanisms and real materials. Best used when full real-world performance is the only remaining unknown — and only after lower-fidelity tests have confirmed direction.

Cost-Reduction Techniques

Prototyping costs can escalate quickly at higher fidelity levels. Several strategies cut cost without cutting learning:

- **Pretend to own.** Instead of buying equipment, materials, or space, borrow or rent. Use a coworking space's 3D printer, rent a commercial kitchen for a food product, borrow specialized tools from a makerspace. This can cut cost substantially at every fidelity level.
- **Prototype only the critical path.** Do not build every screen, feature, or component. Focus on the 2 to 3 tasks that test your riskiest assumption.
- **Use existing assets.** Stock photos, open-source UI kits, template services, and pre-built components dramatically reduce time and cost at medium fidelity.
- **Time-box ruthlessly.** Fix the time, stop when it expires. A prototype 80 percent finished in 2 days teaches nearly as much as one 100 percent finished in 2

weeks.

6.3.1 Paper Prototype

In Brief

Paper prototyping is a low-fidelity usability test where you draw screens or interface elements on paper and have a real user interact with them as if they were a working product. One person acts as the user while another swaps paper screens in response to each action, simulating the product’s behavior. The output is direct observation of where users get confused, stuck, or surprised — giving you design feedback in minutes without writing any code.

COMMON USE CASE

You have multiple design directions for a key part of your product and you are not sure which one makes sense to users. Before investing in a digital prototype, you want fast, cheap feedback from real people. You put rough paper sketches in front of them and watch where they tap, hesitate, or get confused.

HELPS ANSWER

- What basic shape might our solution take?
- Where do users get stuck or confused in our design?
- Is our product intuitive enough to use without instructions?
- Are there situations or errors we did not think of?
- What information do users need to complete a task?

Description

Paper prototyping is a low-fidelity usability test where hand-drawn screens stand in for a working interface. A real user attempts a real task, tapping or pointing at the paper. A second person — the “computer” — swaps screens or moves cutouts in



AT A GLANCE

Qualitative

In-person

Remote

Observational



TIME  ~2 days–2 weeks

COST  \$0–\$30

OTHER NAMES

Paper Prototype · Low-Fidelity Prototype · Paper Mockup · Lo-Fi Prototype

response to each action, simulating how the product would behave. A facilitator coaches the user to think aloud; one or more observers take notes. It is the lowest-fidelity variant in [Prototyping](#), the first round you run before building a [clickable prototype](#).

Use paper prototyping when the questions you have are about flow and structure: does the user understand what to do next, do they look in the right place for a function, do they recognize the labels you chose. Low-fidelity paper sessions surface substantially the same usability defects as code-based prototypes, at a fraction of the cost.

Paper is the wrong tool when the question is about visual placement, scroll behavior, animation, micro-interactions, or anything keyboard- and pointer-specific. For those, you need a clickable prototype or the real thing. Keep the fidelity deliberately rough: a sketch is meant to invite change, while a finished mockup invites the user to defend or critique the visual treatment instead of the flow you came to test. If you would rather not draw by hand, an AI wireframe tool can draft the screens from a text description in minutes — but its default output looks polished enough to push users toward design feedback, so strip the fidelity back: turn off color, use placeholder copy, and tell the user up front that the visual treatment is not what you are testing.

How to

Prep

★ AI Prompt

1. **Define the task.** Write the user task you are testing in one sentence — for example, “A first-time user signs up and creates their first project.” If you cannot state it in one sentence, you are testing too many things at once.
2. **Sketch the happy path, then the branches.** Draw the screens a user sees when they complete the task without trouble. Then add screens for the most likely error or branch states (validation failure, empty state, “not found”). Keep the fidelity deliberately rough; polished sketches invite the wrong kind of feedback.
3. **Recruit five users who match your real target user.** Five users per round is the working number — beyond that, new defects arrive slowly. The catch is selection bias: five people who already understand your domain will not surface the issues a real beginner would. Recruit against the actual target user, not your network.
4. **Assign roles before the session.** Decide in advance who plays the “computer” (swaps screens, never speaks), who facilitates (prompts the user to think aloud, stays neutral on the design), and who observes (takes notes, watches body

language). Mixing these roles mid-session is how facilitators accidentally lead the user.

5. **Pilot once internally.** Run the full script on a teammate who has not seen the screens. You are looking for missing screens, ambiguous prompts, and any place the “computer” has to improvise — those are the screens you forgot to draw.
6. **Decide how you will capture data.** Pick one of: a written defect log per session, a photo of every screen state the user reached, or a recorded screen of a tappable digital version. You only need one — pick before you start.

Execution

1. **Lay out the screens.** For each screen or interface state in the task, lay out the paper mockup illustrating what the user would see. For quick internal ideation, paper is faster than any digital tool — you can sketch, rearrange, and discard in real time with no overhead. For external sessions (especially remote ones), an AI-generated low-fidelity wireframe is easier to share and modify between sessions. In either case, keep fidelity deliberately rough: if users start commenting on visual polish rather than flow and structure, the prototype is over-engineered.
2. **Assign the user.** One person (or a pair) plays the user attempting the task.
3. **Assign the “computer.”** Another person (or several) plays the “computer” — the software simulator that swaps screens in response to each action.
4. **Run the task.** The user interacts with the paper prototype as if it were a real application, physically tapping or pointing at the interface. Encourage them to explain their thinking aloud as much as possible.
5. **Swap screens silently.** For each action the user takes, the computer moves or swaps the paper to reflect the new state. As a general rule, the computer does not talk. Breaking that rule is the most common way a paper-prototype session collapses into a tutorial.
6. **Observe and facilitate.** For more formal sessions, observers watch for additional insights and capture notes. A facilitator may also be useful, prompting the user to think aloud without coaching them through the flow.
7. **Iterate on the spot.** If a layout or interaction clearly is not working, mock up an alternate flow on the spot and re-run the same task. Paper is cheap, so iterating mid-session costs almost nothing.

TOOLS

[Figma](#)

[Balsamiq](#)

[Marvel](#)

[Sneakpeekit](#)

Paper, Sharpies, and index cards — the canonical low-fidelity prototyping kit for the first round

Analysis

✦ AI Prompt

You are looking for places where the user got stuck, could not find what they were looking for, or accidentally went down the wrong path. Anything misleading, confusing, or hard to find is noteworthy. For each, dig into what information the user was missing that led to the confusion and how that information could be provided — or how the need for it could be eliminated. Watch for situations where the response by the “computer” was not defined, or where an action was possible that you did not want to be possible. Also revisit assumptions about the user’s motivations and the knowledge or experience base they bring; the test will often expose assumptions that did not hold and suggest ways to improve the solution.

Cluster the friction observations across all five sessions before deciding what to change. A single confused user might be noise; the same hesitation in three out of five is a defect. Trust what you saw on paper, and do not wait for a polished prototype to “really” test it.

⚠ Biases & Tips

Confirmation bias

A facilitator or observer who knows the design too well can drop “hints” that lead the user through the flow. Brief facilitators to stay neutral, read prompts as written, and respond to questions with “What would you expect?” rather than an answer.

Social desirability bias

Users may hide feedback when they are worried about appearing incompetent or offending the team. Reassure them up front that you are testing the design, not the person, and that confusion is data you need.

Observer effect

A user who knows they are being watched behaves more carefully and deliberately than they would in real use, masking the careless mistakes a real user would make. Keep the framing casual, ask them to work at normal speed, and treat unusually flawless runs with suspicion.

Next Steps

- Iterate the paper prototype based on test findings before going digital.
- If the core flow works, create a [Clickable Prototype](#) for wider testing.
- Share paper-prototype photos with the team to align on the design direction.
- Use [Usability Testing](#) with a higher-fidelity prototype to confirm the flows that worked on paper still work on screen.

- Run a [Solution Interview](#) to confirm that the design direction addresses the underlying customer problem before investing in development.

Case Studies

Palm: PalmPilot wooden mock

The Computer History Museum documents Jeff Hawkins's circa-1995 wooden block prototype with a chopstick stylus, used to test PalmPilot interactions by taking pretend notes in meetings before the device shipped in 1996.

[Read more](#)

TenTune: Pencil-iterated music app

A small music-product team used paper sketches and pencil iteration to settle on a core interaction model, discarding two early designs that tested poorly before writing any code.

[Read more](#)

IBM: Enterprise paper-prototype adoption

Among the first large firms to standardize paper prototyping in the mid-1990s; Carolyn Snyder's book documents IBM-team work showing paper finds substantially the same usability problems as higher-fidelity prototypes.

[Read more](#)

Further reading

-  Paper Prototyping — Carolyn Snyder, Morgan Kaufmann 2003
-  Prototyping for Tiny Fingers — Marc Rettig, CACM 1994
-  Sketching User Experiences — Bill Buxton, Morgan Kaufmann 2007
-  The Design of Everyday Things — Don Norman, MIT Press 2013 rev. ed.
-  PalmPilot wooden model — Computer History Museum
-  NN/g: Paper Prototyping

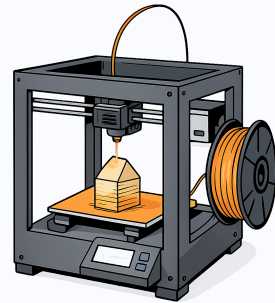
6.3.2 3D Print

In Brief

A 3D printed prototype is a physical model of a product concept that a participant can hold, examine, and handle. You print the model to evaluate form factor, ergonomics, size, weight, and spatial relationships before committing to tooling or manufacturing. A 3D print is typically not a functional prototype — no electronics, no moving parts, no working mechanisms. Its value is making the physical design tangible, so a participant reacts to something real rather than imagining a product from a screen or drawing.

COMMON USE CASE

You have a physical product concept where dimensions, shape, or ergonomics are the open question, and you want a participant to react to a real object rather than a drawing or rendering. You have already validated the underlying problem and the rough product idea through interviews or paper prototypes, and you can spend \$5 to \$50 on filament and a day at a makerspace, library, or on-demand 3D-printing service to print and test two or three variations.



AT A GLANCE

Qualitative

In-person

Observational

Behavioral



TIME  ~1–3 weeks

COST  \$80–\$1.2K

OTHER NAMES

3D Print · 3D Printed Prototype

HELPS ANSWER

- Does the physical form factor feel right in a participant's hands?
- Is the product the right size and weight?
- Are buttons, grips, or interaction points in ergonomic locations?
- Does the product look like what participants expect?
- Are there physical design issues that are not apparent on screen?

Description

A 3D printed prototype is a physical model of a product concept that you print, hand to a participant, and watch them handle, so you can evaluate form factor, ergonomics, and spatial relationships before committing to tooling. It is a

Prototyping

fidelity variant that sits between paper and life-sized prototypes. It is physical enough to test things a screen cannot, but cheap enough to iterate quickly: materials cost \$5 to \$50 per print and printing takes 1 to 24 hours depending on size and complexity, so a product team might print three or four variations of a handle design in an afternoon and test all of them the next day.

The key limitation is that 3D prints are typically non-functional. They have no working electronics, no moving mechanical parts, and not the materials or finish of a final product. For early-stage testing that is an advantage: the participant responds to the physical form without being distracted by whether the technology inside works, so the test isolates physical design from everything else.

A 3D print is most valuable for products where physical form matters — consumer electronics, medical devices, kitchen tools, wearables, packaging, toys, furniture — and where physical dimensions, shape, or ergonomics are the critical open question. Reach for it when you need to test multiple form factor variations quickly, want a participant to react to a physical object rather than a drawing or rendering, or are deciding between physical design alternatives before investing in tooling. It is less useful for purely digital products or services, though even software companies sometimes 3D print physical accessories, packaging, or hardware companions.

How to

Prep

★ AI Prompt

1. **Define exactly what you are testing.** Be specific: overall size? Grip ergonomics? Button placement? Visual appeal? The answer determines how much detail your print needs and how to evaluate the result.
2. **Set pass/fail criteria before you print.** Ergonomic thresholds, dimensional tolerances, “fits in the target use environment” checks. Without predefined criteria, the temptation is to rationalize a borderline print as good enough once you have spent the time and filament.
3. **Build the 3D model.** Use a free, beginner-friendly 3D modeling tool, or a more capable parametric CAD tool if you need precision. Model only what matters for the test — if you are testing grip, the handle needs to be accurate but internal

details do not. AI text-to-CAD tools can produce a first-draft mesh from a description; refine before printing.

4. **Choose the print process.** FDM (filament) printers are sufficient for most prototyping and the most accessible. Use SLA resin printers if you need smoother surfaces or finer detail. If you do not own either, route to a makerspace, a library, or an on-demand 3D-printing service.
5. **Print two or three variations.** A single print is a single point of data. Print at least two variants — different sizes, different grip angles, different button placements — so participants can compare and reveal preference rather than just react.
6. **Post-process if needed.** Sand rough edges, paint if color or finish matters to the test, add weight if the final product will be heavier than the plastic print. Surface finish should match the variable you are testing — if grip is the question, sand it; if visual presence is the question, paint it.
7. **Recruit participants.** Recruit 5 to 8 participants from the target segment, with diversity in hand size, grip strength, and physical ability. Ergonomics vary across body types, so an all-developer test pool will not represent the people who will actually use the product.

Execution

✦ AI Prompt

1. **Present in context.** Place the prototype in the environment where the final product will be used. Kitchen tool? Test it in a kitchen. Wearable? Have them wear it. Hand tool? Test it where the hand task happens. The realistic-context test only works if the context is actually realistic.
2. **Hand it over without a script.** Give the participant the printed object and stay quiet. Watch how they pick it up, how they grip it, whether they rotate it, whether they look confused about orientation. Their unprompted physical interaction is the data; your narration about what the object does will bias the reaction.
3. **Use a think-aloud protocol after the first physical interaction.** Ask the participant to narrate what they noticed while they were holding it. “How does that feel in your hand? Is it the right size? Where would you expect the button to be?” Record verbatim — language about size and weight (“bigger than I expected,” “lighter than it looks”) is signal.
4. **Test all variants with each participant.** Hand them variant A, then B, then C. Ask which felt better and why. Counterbalance the order across participants — half see A first, half see B first — so order does not bias preference.

TOOLS

[Tinkercad](#)

[Fusion 360](#)

[Prusa](#)

[Formlabs](#)

[Hackerspaces.org
directory](#)

5. **Note the comparison products.** Participants will naturally compare the prototype to products they already own. “Reminds me of a [X]” or “smaller than my [Y]” tells you the competitive physical-design landscape and the mental category they put the product in.
6. **Capture the failure modes.** When a participant adjusts their grip after picking it up, when they cannot tell which side faces forward, when they ask “is this real?” – those are the failures. Write them down precisely; the reason for each failure is the redesign target.

Analysis

✦ AI Prompt

1. **Score each variant against the pass/fail criteria separately.** Do not aggregate into “best variant.” A variant can pass on grip and fail on visual presence, and the two failures route to different fixes.
2. **Cluster the unprompted physical interactions.** When most participants adjust their grip after picking it up, the form factor is the problem. When most participants rotate the object to find the orientation, the affordances are the problem. The unprompted behavior is more reliable than the verbal report – the verbal report often rationalizes the physical reaction.
3. **Map the verbal feedback to specific physical features.** Group quotes by feature (handle, button, weight, surface). Quotes that cluster on one feature mean that feature is the problem; quotes spread across many features mean the overall concept is the problem.
4. **Reconcile with the comparison products participants named.** If participants consistently compared the prototype to a product in a different category than you intended, the physical design has placed it in the wrong mental category. That is a positioning problem the prototype surfaces, not a manufacturing problem.
5. **Decide the next move.** One of three things should happen: a variant clearly passes on the criteria, so lock the form factor and move to a [Single Feature MVP](#) or [Life-Sized Prototype](#) that adds function; one variant is close but fails on specific criteria, so iterate on those features and reprint only the affected geometry; all variants fail across criteria, so return to the underlying concept and reassess whether the physical form is the right approach to the problem.

⚠ Biases & Tips

Material mismatch

3D printed plastic feels different from production materials such as metal, rubber, or glass. A participant may react negatively to plastic when the final product will use a different material. Acknowledge

this upfront and ask them to evaluate shape and ergonomics, not material feel.

Novelty effect

A participant may be excited by the novelty of holding a 3D printed prototype regardless of whether the design is good. Discount initial enthusiasm and focus on specific behavioral observations and on the comparison between variants.

Sampling bias

With only 5 to 8 participants, a pool skewed toward one hand size, grip strength, or use context will report ergonomics that do not hold for the real range of users. Recruit for spread across the body types and use contexts that matter, and treat a finding from a narrow pool as a hypothesis, not a conclusion.

Sunk-print fallacy

Hours of design and printing time create pressure to declare the print “good enough.” Honor the pass/fail criteria you set in Prep; a failed print is data, not waste.

Next Steps

- For a locked form factor that needs functional validation, advance to a [Single Feature MVP](#) to validate that the core functional benefit retains users once the physical design is fixed.
- For a form factor that needs full-scale and production-grade-material confirmation, advance to a [Life-Sized Prototype](#) before committing to tooling.
- For any companion app or UI before adding electronics, run a [Paper Prototyping](#) test on the digital side.
- For broad failures across variants, drop back a step and re-test the form with a cheaper, faster [Paper Prototyping](#) round on the physical concept before reprinting — if even sketches do not land, the problem is the concept, not the fidelity.

Case Studies

Adidas: Futurecraft 4D midsole

Adidas partnered with Carbon to 3D-print Digital Light Synthesis midsole lattices for the Futurecraft 4D, iterating on lattice geometry across printed variants before locking the production design.

[Read more](#)

Further reading

- ☰ Hod Lipson and Melba Kurman — Fabricated: The New World of 3D Printing (Wiley, 2013)
- ☰ Karl T. Ulrich and Steven D. Eppinger — Product Design and Development (McGraw-Hill, 2015)
- ☰ Bill Buxton — Sketching User Experiences (Morgan Kaufmann, 2007)
- ☰ Stephanie Houde and Charles Hill — What Do Prototypes Prototype? (Handbook of Human-Computer Interaction, 1997)
- ☰ Jake Knapp, John Zeratsky, and Braden Kowitz — Sprint (Simon & Schuster, 2016)
- ☰ Carbon — adidas 4D case study (Digital Light Synthesis)

6.3.3 Clickable Prototype

In Brief

A clickable prototype is an interactive digital representation of a product where screens are interconnected through click targets — buttons, links, and hotspots — so users navigate through tasks as if using a real application. No code is written; the interactions are simulated by linking static screens. Despite the simplicity, many users cannot tell a well-made clickable prototype from a real product, which makes it a strong tool for usability testing, stakeholder alignment, and investor demos.

COMMON USE CASE

You have a candidate UX flow that already passed at low fidelity — paper prototype, storyboard, or solution interview — and now the question is whether the actual interaction design holds up under unsupervised use. You want measurable task-completion data and observable confusion points before any engineering investment, and you can spend a few days in a design tool plus 5 to 8 user sessions to get the read.

HELPS ANSWER

- Can users complete key tasks without guidance?
- How long does it take users to accomplish core tasks?
- Where do users get confused or take wrong paths?
- Is the navigation structure intuitive?
- Does the visual hierarchy guide attention correctly?
- Do users understand what the product does from the first screen?



AT A GLANCE

Quantitative

Qualitative

In-person

Remote

Observational



TIME  ~1–2.5 weeks

COST  \$80–\$1.8K

OTHER NAMES

Clickable Prototype · Digital Prototype · Clickthrough Prototype

Description

A clickable prototype is an interactive digital mockup in which static screens are linked through click targets — buttons, links, and hotspots — so a participant can move through a task as if using the real product. It is the highest-fidelity

 **Prototyping** variant you can build without code: the interactions are simulated by wiring screens together in a design tool. The fidelity is high enough to support real usability testing, which makes it the primary way to check that a product's interaction design works before engineering starts.

On the fidelity ladder, it sits one rung above a  **paper prototype**, which needs a human facilitator to simulate the interface, and one rung below a

 **single feature MVP** or  **mash-up**, which run on real backends with real data. That position is what lets you observe whether users complete tasks, measure how long they take, and identify where they get lost — without building anything real.

A clickable prototype should not be the first thing you build. If the basic concept and flow have not been validated through paper prototypes, storyboards, or solution interviews, you risk spending days polishing an interaction design for a concept that does not work. A clickable prototype answers “can users use this?” — not “should we build this?” It is not a working product and not a code prototype: there is no backend, database, or real data, and nothing is programmed. A good one covers 1 to 3 core tasks rather than every possible interaction, with unconnected areas showing a “this area is not part of the prototype” message.

Once the concept is validated, reach for one to test detailed interaction design with measurable task-completion metrics, to demonstrate the product to stakeholders, investors, or partners who need to “see it working,” or to test specific interaction patterns such as navigation, onboarding flows, or checkout before engineering invests in building them.

How to

Prep

 AI Prompt

1. **Define the tasks to test.** Choose 1 to 3 core tasks that represent the product's primary value. Each task needs a clear start point and end point — for example, “find a product, add it to cart, and check out” or “create an account and set up a project.”
2. **Map the screens.** For each task, list every screen the user will see, including error states, empty states, and confirmation screens. A typical task takes 5 to 10

screens.

3. **Design the screens.** Build each screen in your design tool. Start with the happy path, then add error states and edge cases. Use realistic content — placeholder text like “Lorem ipsum” confuses users and invalidates results. An AI UI generator can produce a first pass from a text description in minutes; treat the output as a draft to refine, not as final.
4. **Connect the screens.** Add click targets (hotspots) that link screens. Every clickable element either leads somewhere or shows a “not available in this prototype” indicator. Dead clicks frustrate users and corrupt your test data.
5. **Pilot the prototype yourself.** Walk through every task path twice. Check for dead ends, missing screens, and broken links. Have a colleague test it cold to catch assumptions you missed.
6. **Recruit and schedule participants.** Recruit 5 to 8 users from the target segment. Match technical comfort level to the real audience — tech-savvy participants navigate prototypes more easily than typical users and will give you a falsely optimistic read.
7. **Write the moderator script.** Standardize the task introduction, the think-aloud prompt, and your responses to user questions. Decide in advance what you will and will not help with — typically nothing.

Execution

1. **Run the sessions one at a time.** With 5 to 8 participants from the target segment, give each person the task instructions and observe them using the prototype. Do not help, hint, or react. Where a participant struggles is the data you came for.
2. **Use a think-aloud protocol.** Ask participants to narrate what they are looking at, what they expect to happen, and what they are about to click. Record verbal hesitation (“hmm, I think I’d click...”) because it surfaces uncertainty that successful clicks hide.
3. **Capture the click path and the failure points.** Most prototyping tools record click data automatically. Note where users hesitate, where they backtrack, and where they click outside the connected hotspots — those are the design defects.
4. **Read the moderator script verbatim.** A consistent task introduction across sessions removes facilitator-induced variance. Do not improvise.
5. **Watch for misidentification.** If a participant describes the product as something other than what it is, the value proposition has not landed. Note the words they used.

TOOLS

[Figma](#)

[ProtoPie](#)

[Maze](#)

[Framer](#)

[v0](#)

Analysis

↗ AI Prompt

1. **Calculate task-completion rate per task.** Percentage of users who complete the task without help. Below 80% indicates a real usability problem; below 50% indicates a flow-level problem that paper would have caught.
2. **Plot time on task and first-click accuracy.** Large variance between users may indicate inconsistent mental models. First-click accuracy above 70% correlates strongly with overall task success – below that, the call-to-action is wrong, not the task.
3. **Cluster the path deviations.** When 3 of 5 users click the same wrong button, that is a design defect, not a user defect. Patterns matter more than counts at this sample size – for a sense of the uncertainty in a small-n usability metric like completion rate, ask an AI assistant to estimate a plausible range for the true rate rather than running a pass/fail significance test the sample is too small to support.
4. **Synthesize the verbal feedback.** Group hesitation moments by screen, by widget, and by task phase. Hesitation that clusters on a screen means the screen is the problem. Hesitation that clusters on a phase (“checkout in general feels off”) means the flow is the problem.
5. **Decide the next move.** One of three things should happen: tasks pass cleanly → polish the flow and run quantitative validation in [Usability Testing](#); tasks fail at specific decision points → redesign those screens and re-test the affected paths; tasks fail across the board → return to [Paper Prototyping](#) and reassess the concept itself.

⚠ Biases & Tips

Visual fidelity distraction

A polished prototype invites comments on aesthetics rather than usability; a rough prototype gets dismissed as unfinished. Match visual fidelity to the question being tested – wireframe-quality for flow, hi-fi for visual hierarchy.

Confirmation bias

Teams read click patterns to support a decision already made. Set pass/fail thresholds before the test, not after.

Small-sample overconfidence

Five-user tests are directional, not conclusive. Frame conclusions as “early indication” until a second round confirms.

Facilitator influence

A nod, a hint, or a reworded task steers the participant and contaminates the result. Read the moderator script verbatim and stay silent while they work, even when they get stuck.

Sampling bias

Tech-savvy participants navigate prototypes more easily than the real audience and give a falsely optimistic read. Recruit for the technical comfort level of the actual target segment, not for convenience.

Next Steps

- For a passing flow, run a quantitative **Usability Testing** round with 15 to 30 participants to confirm the small-sample read.
- For specific failed decision points, redesign the affected screens and re-test only those paths with a fresh group of users in a focused **Usability Testing** round.
- For a flow that broke down across tasks, return to **Paper Prototyping** — the cheapest place to fix a bad concept is at the lowest fidelity.
- For a validated flow ready for real users, use a **Vibe-Coded Disposable MVP** to turn the flow into a working build.
- For backend-heavy interactions where the click flow worked but the data is faked, use a **Wizard of Oz** before committing engineering to custom development.

Case Studies

BBC News: Mobile-app prototype programme

BBC R&D built 35 mobile prototypes over a year to test new storytelling formats with readers before iOS and Android engineering work began.

[Read more](#)

Further reading

-  Steve Krug — Don't Make Me Think, Revisited (New Riders, 2014)
-  Jakob Nielsen — Why You Only Need to Test with 5 Users (Nielsen Norman Group, 2000)
-  Jeff Sauro and James R. Lewis — Quantifying the User Experience (Morgan Kaufmann, 2016)
-  Jake Knapp, John Zeratsky, and Braden Kowitz — Sprint (Simon & Schuster, 2016)
-  Tomer Sharon — Validating Product Ideas Through Lean User Research (Rosenfeld Media, 2016)
-  Nielsen Norman Group — How to Conduct a Usability Test on a Mobile Device

6.3.4 Life-Sized Prototype

In Brief

A life-sized prototype is a full-scale model built with working mechanisms and actual (or near-actual) materials, tested with participants under realistic conditions. It is the highest-fidelity prototyping method, testing form, fit, and function before committing to manufacturing tooling, construction, or full-scale production. It is also the most expensive, so use it only after cheaper, lower-fidelity tests have confirmed the product direction and narrowed the remaining questions to ones that can only be answered at full scale.

COMMON USE CASE

You have a physical or hybrid product that has already passed lower-fidelity tests — paper, clickable, 3D print, or single feature MVP — and the only remaining unknowns are about how the product behaves at full scale: ergonomics, structural performance, materials over time, assembly, regulatory fit, or the in-context user experience. You are willing to spend weeks to months and a four-to-five-figure budget to answer them, because the next step after this prototype is committing to tooling, construction, or production.

HELPS ANSWER

- Does the product work at full scale under realistic conditions?
- Do the chosen materials perform as expected?
- Are there assembly, manufacturing, or installation issues?
- Does the full-scale user experience match expectations from smaller prototypes?
- What are the actual production costs?



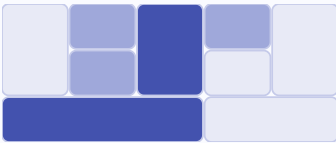
AT A GLANCE

Quantitative

Qualitative

In-person

Observational



TIME  ~4 weeks–5 months

COST  \$3K–\$60K

OTHER NAMES

Life-Sized Prototype · Full-Scale Prototype

- Are there safety, durability, or regulatory concerns?

Description

A life-sized prototype is a full-scale, working model built with actual or near-actual materials and tested with participants in the setting where the product will be used. It is the highest-fidelity [Prototyping](#) variant — the last rung above paper, 3D-printed, and clickable prototypes. It is the final validation gate before committing to production, and it exists to answer questions that cannot be answered at lower fidelity: whether the product works under real-world conditions, structural integrity under load, material performance over time, assembly sequence feasibility, manufacturing and installation issues that only emerge at scale, and whether the full-scale experience matches what smaller models suggested.

Because of the cost and time it requires, confirm the following before you commit to building one:

- The core concept is validated (through earlier research and lower-fidelity prototypes).
- The target customer segment is confirmed.
- The business model can support the production costs.
- The remaining unknowns are specifically about full-scale performance and cannot be tested any other way.

This method belongs in the validation phase, not the discovery phase: it is for confirming that a product people have already said they want can actually be built and delivered as envisioned, not for exploring whether demand exists. If you are still testing demand, use cheaper methods.

How to

Prep

✦ AI Prompt

1. **Define exactly what you are testing.** Write down the specific questions the life-sized prototype must answer. Every element of the prototype should serve at least one of these questions. Anything that does not serve a question is unnecessary scope.
2. **Set pass/fail criteria before you build.** Tolerances, performance benchmarks, ergonomic standards, regulatory thresholds. Without predefined criteria, there is

a strong temptation to rationalize disappointing results once the prototype exists and money has been spent.

3. **Source materials and fabrication.** Identify the materials, tools, and fabrication methods you will need. Apply the Pretend-to-Own principle: rent, borrow, or partner before purchasing. Get quotes from fabricators, machine shops, or contract manufacturers.
4. **Plan the test environment.** Decide where the prototype will live during testing — a real home, a real retail space, a real workshop floor — and what adverse conditions you will deliberately introduce (heat, cold, heavy use, inexperienced users). The realistic-conditions test only works if the conditions are actually realistic.
5. **Schedule the independent reviewer.** Identify someone outside the build team — an experienced practitioner, a domain expert, or a prospective user — who will run the evaluation against the pass/fail criteria. Get them on the calendar before you start building, so the review is insulated from the build team's investment in the result.

Execution

1. **Build in phases.** Construct the prototype in stages, testing as you go. Do not build the entire thing before testing anything. If the first component fails, you have saved the cost of completing the rest.
2. **Test under realistic conditions.** Place the prototype in the environment where the final product will be used. If it is a piece of furniture, put it in a home. If it is a retail concept, set it up in a real retail space. If it is a device, have real users operate it in real conditions. Deliberately introduce adverse conditions: extreme temperatures, heavy use, inexperienced users.
3. **Run the independent evaluation.** Bring in the reviewer you scheduled in Prep. Give them the pass/fail criteria and the prototype, and step out of the room. Their evaluation is the test, not your team's interpretation of their evaluation.
4. **Document everything.** Photograph and video the build process, testing conditions, and results. Record exact materials, dimensions, costs, and time expenditures. This documentation becomes the basis for production planning if the prototype passes.
5. **Capture failure precisely.** When a component fails, record the conditions, the failure mode, and the time to failure. The point of a life-sized prototype is to find

TOOLS

[Fictiv](#)

[Xometry](#)

[Protolabs](#)

Local makerspaces
— shared equipment,
fabrication space,
and hands-on help
for full-scale builds

these failures here, not in production, so a failure found in test is a useful result, not a wasted build.

Analysis

✦ AI Prompt

1. **Score each pass/fail criterion separately.** Do not aggregate into an overall pass/fail. A prototype can pass on structural performance and fail on ergonomics, and the two failures route to completely different fixes.
2. **Read the override log from the independent reviewer.** Where did they disagree with the build team's assessment? Each disagreement is a signal that builder's pride is shaping interpretation. Resolve these before drawing conclusions.
3. **Map failure modes to redesign cost.** For each failure, estimate what it would cost to fix: a new material, a new mechanism, new tooling, a complete redesign. Multiply by the criticality of the criterion. Failures on critical criteria with expensive fixes are the engineering risk worth surfacing to the rest of the team.
4. **Reconcile with smaller-prototype predictions.** Where did the full-scale experience match the lower-fidelity tests, and where did it diverge? Divergences are often the most valuable finding, because they show where the lower-fidelity methods misled you and which questions to test at full scale next time.
5. **Decide the next move.** Choose one of three outcomes: passes all criteria → produce the production-readiness checklist and advance to manufacturing planning; fails on specific criteria with feasible fixes → redesign the failed components and rebuild only those, not the whole prototype; fails broadly across criteria → return to a lower-fidelity prototype and reassess the concept itself rather than commit to a second life-sized build.

⚠ Biases & Tips

Sunk cost fallacy

The significant investment in a life-sized prototype creates strong pressure to declare it a success regardless of results. Commit to pass/fail criteria before building and honor them.

Builder's pride

The team that built the prototype may be emotionally invested in its success. Have an independent party run the evaluation and treat their read as the test result.

Sample size of one

A single prototype cannot reveal manufacturing variability. Results tell you what is possible, not what is consistent. Plan for production variability in your evaluation.

Lower-fidelity overconfidence

Smaller prototypes that passed cleanly bias the team toward expecting the full-scale build to pass. Treat divergences from the

lower-fidelity prediction as the most informative finding, not the most disappointing one.

Next Steps

- Convert the prototype's documented build process and pass/fail outcomes into a production-readiness checklist.
- Use a  **Single Feature MVP** on any companion software so the digital experience is validated alongside the physical product.
- Use a  **3D Print** to iterate cheaply on any specific component that failed at full scale.
- Use a  **Paper Prototyping** to revisit any user-facing flow that did not work as expected at full scale.
- Revisit  **Prototyping** to confirm whether the remaining unknowns warrant another life-sized build or a cheaper test.

Case Studies

Dyson: DC01 cyclonic vacuum

James Dyson is widely reported to have built more than 5,000 full-scale prototypes of the bagless cyclonic vacuum between 1979 and 1984, iterating cyclone geometry under real dust loads before licensing the design.

[Read more](#)

Tesla: Roadster "Mule" platform

Tesla's first life-sized Roadster prototype was the "Mule," a Lotus Elise re-platformed with the early electric drivetrain to test thermal performance, range, and assembly sequence before custom tooling.

[Read more](#)

IDEO: Nightline shopping cart

ABC Nightline's 1999 "Deep Dive" segment followed an IDEO team building a full-scale shopping-cart prototype in five days, with real shoppers using it in a grocery context to drive the redesign.

[Read more](#)

Further reading

- ☰ Karl T. Ulrich and Steven D. Eppinger — Product Design and Development (McGraw-Hill, 2015)
- ☰ Stephanie Houde and Charles Hill — What Do Prototypes Prototype? (Handbook of Human-Computer Interaction, 1997)
- ☰ Tom Kelley — The Art of Innovation (Currency Doubleday, 2001)
- ☰ Donald Schön — The Reflective Practitioner (Basic Books, 1983)
- ☰ Bill Buxton — Sketching User Experiences (Morgan Kaufmann, 2007)
- ☰ IDEO — The Deep Dive: Redesigning the Shopping Cart (ABC Nightline)

6.3.5 Mash-up

In Brief

A mash-up is a functional product experience assembled by connecting existing third-party tools, platforms, and services through APIs, automation platforms, or manual integrations, rather than building custom software from scratch. It actually works: customers can use it, pay for it, and receive real value.

COMMON USE CASE

You have a clear hypothesis about the end-to-end experience but the cost of building any of it custom is too high to justify before you have revenue. You believe most of the value comes from how a handful of commodity capabilities — payments, email, scheduling, storage, communication — are orchestrated together. You want a real, working product in front of paying customers in days or weeks, and you accept that the seams will sometimes show.

HELPS ANSWER

- Does the full product experience work end-to-end?
- Will customers pay for this product?
- Which parts of the experience are most important to get right?
- What breaks when the product is used by real customers?
- Is custom development necessary, or can existing tools deliver the value?

Description

A mash-up is a ■ Prototyping fidelity variant in which you assemble a functional product experience by connecting existing third-party tools and services, rather than



AT A GLANCE

Quantitative

Qualitative

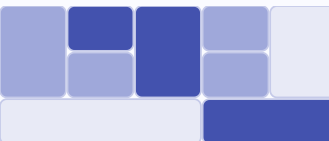
In-person

Remote

Behavioral

Observational

Transactional



TIME  ~1–3 weeks

COST  \$50–\$600

OTHER NAMES

Mash-up · Mashup · Piecemeal MVP

writing custom code, then put it in front of paying customers to test whether the experience holds together. Because it actually works, a mash-up blurs the line between prototype and product: a well-constructed one can serve a real customer base, generate revenue, and operate for months without a single line of custom code. That makes it both a validation tool and a potential first version of the product.

The key insight behind a mash-up is that most product value comes from the orchestration of capabilities, not from the capabilities themselves. Payment processing, email delivery, form collection, database storage, scheduling, and file management are all commoditized. What is unique about your product is how these capabilities are combined to serve a specific customer need. A mash-up lets you test that orchestration directly.

Because it works without significant custom development, a mash-up tests whether the full product experience holds together, from acquisition through delivery. That makes the willingness-to-pay read decisive: if customers will pay for a mash-up version, they will usually pay for a properly built version, and if they will not, custom engineering rarely fixes the problem. Unlike a [Wizard of Oz](#), where a human secretly does the work, or a clickable prototype, where nothing actually works, in a mash-up the connected tools genuinely deliver the value.

A mash-up is most valuable when the product's value comes from orchestrating existing capabilities in a new way and you want to test the full experience — and earn real revenue from real customers — before investing in custom development. It also tells you which parts of the experience need custom building and which can stay third-party.

How to

Prep

✦ AI Prompt

1. **Map the customer journey.** List every step from the customer's first touch to final delivery of value. For each step, identify the capability needed: payment, content delivery, scheduling, communication, data storage, etc. Note which steps are unique to your product and which are pure commodity.
2. **Select tools for each capability.** For each capability, choose an existing tool or platform. Prefer tools with APIs or automation-platform integrations. Prefer tools with free tiers for initial testing. Note the seam between each pair — that is where the mash-up will leak.

3. **Design the integrations.** Map how data flows between tools. What triggers what? When a customer pays, what happens next? When a form is submitted, where does the data go? Draw this as a simple flowchart and review it for handoffs that depend on polling delays.
4. **Define the willingness-to-pay test.** Decide before you build whether the read is “did anyone pay?” or “did this conversion rate clear a threshold?” — and which threshold. Picking afterward is how teams rationalize disappointing results.
5. **Set the operational budget.** Decide how many hours per week you will spend monitoring and patching the mash-up. Set a hard kill-date for the mash-up phase and treat any request to extend it as a scope change requiring re-justification. When you exceed the hour budget for two weeks running, that itself is a finding: the mash-up is telling you which component to rebuild custom first.

Execution

✦ AI Prompt

1. **Build the connections.** Use an automation platform to connect the tools. Start with the core happy path — the most common customer journey — and add edge cases later. Resist the temptation to build for hypothetical edge cases the test will not see.
2. **Test the full flow yourself.** Go through the entire customer experience as if you were a customer. Pay real money (refund yourself later). Check that every automation fires, every email sends, and every piece of data lands where it should. Record the seam-by-seam latency.
3. **Launch to a small group.** Release the mash-up to 10 to 50 early customers. Use the acquisition channel you would use at launch — not friends and family, who will pay out of loyalty and break the willingness-to-pay read.
4. **Monitor live for the first week.** Sit with the queue daily. Automation failures, edge cases, and unexpected user behavior will surface in the first week and they will distort the read if you do not catch them. Log every manual intervention as a build-versus-buy data point. Have someone outside the team walk the full purchase flow at least once during this window and report every place the experience felt stitched together — seams you have stopped noticing.
5. **Iterate based on breakage.** When something breaks (and it will), fix it and note whether the fix requires a better automation, a different tool, or eventually custom development. Do not pre-emptively rebuild components that have not yet failed.

TOOLS

[Zapier](#)

[Make](#)

[Airtable](#)

[Stripe](#)

[Typeform](#)

Analysis

✦ AI Prompt

1. **Read willingness-to-pay first.** Did customers pay, and at what conversion rate? Compare to the threshold you committed to in Prep. Surface-level satisfaction without a payment is the same outcome as a Wizard of Oz with high satisfaction and zero return visits — the magic was the human, not the value proposition.
2. **Quantify the seam quality.** For every handoff between tools, log the failure rate and the median manual-intervention time. The seams where customers visibly noticed the seam (delays, inconsistent emails, lost data) are the components that have to be custom-built first.
3. **Cluster customer feedback by mash-up tell vs. value gap.** “Why does it take 15 minutes to get my report?” is a tool limitation (automation polling delay). “I don’t understand why I would pay for this” is a value-proposition gap. The two require very different responses; conflating them is the most common analysis mistake with this method.
4. **Compute the operational cost trajectory.** Hours per week spent monitoring and patching, divided by paying customers. If the slope is flat or down as you scale, the mash-up can ship as the first version of the product. If the slope is up, model when the cost line crosses the cost of a custom rebuild.
5. **Decide the next move.** One of three things should happen: revenue and retention clear thresholds AND operational cost is sustainable → ship the mash-up as v1 and rebuild components only as breakage forces it; revenue lands but operational cost is unsustainable → start the custom rebuild on the highest-friction component; revenue does not land → the value proposition is not strong enough, and no amount of custom build will fix it.

⚠ Biases & Tips

Tool
limitations
masquerading
as product
problems

If a mash-up is slow because the automation platform has a 15-minute polling delay, customers may perceive the product as unresponsive even though a custom-built version would be instant. Distinguish between tool limitations and product problems before you act on the feedback.

Feature-creep
bias

No-code tooling makes adding components feel nearly free, so teams keep extending the mash-up instead of calling a verdict. The cognitive trap is that each individual addition seems low-cost; cumulatively they defer the build-or-stop decision indefinitely and erode the validity of the test.

Platform
dependency

A mash-up built on third-party tools is subject to those tools’ pricing changes, API deprecations, and outages. This is acceptable for

validation but is a risk for long-term operation.

**Insider
habituation**

Founders who live inside the mash-up adapt to its seams through repeated exposure — a well-documented habituation effect — and stop perceiving the jarring transitions that first-time customers notice immediately. This means qualitative feedback from the team systematically underestimates coherence problems.

Next Steps

- Track operational cost per customer for two weeks to identify which integrations break first as volume grows, then revisit the **Prototyping** fidelity ladder to choose what to build next.
- Use a **Vibe-Coded Disposable MVP** to replace the highest-friction component with custom code once breakage points are clear.
- Use a **Single Feature MVP** to rebuild the most-used capability as polished native functionality.
- Use a **Concierge Test** on any step where automation keeps failing to learn what customers actually need before re-automating.
- Use a **Wizard of Oz** for any planned new feature so you avoid adding new third-party tools before the demand is validated.

Case Studies

Groupon: WordPress and AppleScript

Andrew Mason's team published daily deals on a WordPress site, sent them via Apple Mail, and generated PDF coupons with an AppleScript — no deal engine or merchant dashboard — to validate demand before any custom build.

[Read more](#)

Makerpad: No-code stack as product

Ben Tossell built Makerpad on Webflow, Stripe, Zapier, Airtable, and Memberstack, growing a paying member base before Zapier acquired the company in 2021.

[Read more](#)

Buffer: Two-page validation MVP

Joel Gascoigne validated demand with a two-page mash-up — a landing page plus a pricing page that broke when clicked — before building the first functioning version of the scheduler over seven weeks.

[Read more](#)

Product Hunt: Linkydink mailing list

Ryan Hoover launched the original Product Hunt as a daily Linkydink email curated by a small group, validating engagement for a community-curated product feed before building any app.

[Read more](#)

Further reading

-  Eric Ries — The Lean Startup (Crown Business, 2011)
-  Steve Blank — The Four Steps to the Epiphany (K&S Ranch, 2013)
-  Ben Tossell — How Makerpad Got Acquired (Zapier blog, 2021)
-  Joel Gascoigne — Idea to Paying Customers in 7 Weeks (Buffer blog)
-  Indie Hackers — No-Code Stack Interviews
-  Nielsen Norman Group — Service Design 101

6.3.6 Single Feature MVP

In Brief

A single feature MVP is a minimum viable product (MVP) that delivers exactly one core feature — polished, functional, and complete — while deliberately excluding everything else. It tests one hypothesis: does this one feature provide enough value on its own to attract and retain users? If users come back for a product that does only one thing, that one thing is valuable on its own. If they do not return, adding more features is unlikely to save it. See also the [Vibe-Coded Disposable MVP](#).

COMMON USE CASE

You have already validated the problem and a candidate solution through lower-fidelity work — interviews, paper prototypes, or a clickable prototype — and the team is ready to build code. You want a clean read on whether the core value exchange is strong enough to retain users on its own, before you spend months adding the surrounding product. You will accept that the result is a real product in front of real users, with all the operational weight that implies.

HELPS ANSWER

- Is the core feature valuable enough to attract users on its own?
- Will users return to a product with only one feature?
- Which user segments find the core feature most valuable?
- What is the natural retention curve for the core value proposition?
- What complementary features do users request most urgently?



AT A GLANCE

Quantitative

Qualitative

In-person

Remote

Behavioral



TIME  ~3–6 weeks

COST  \$20–\$300

OTHER NAMES
Single Feature MVP · Minimum
Feature Product

Description

A single feature MVP is a working product built around exactly one core feature, shipped to real users with everything else deliberately left out, so you can measure whether that one feature retains people on its own. It is the highest-fidelity digital variant of [Prototyping](#): a real product in front of real users rather than a mock-up. It tests whether the core value proposition is strong enough to stand alone. Many products fail not because they lack features but because their core feature is not compelling enough; stripping away everything except the core gives you a clear signal about whether you have found value.

This is not a rough prototype or a half-built product. The single feature should work reliably and provide a good user experience; the constraint is scope, not quality. The deliberate exclusion of secondary features is part of the method, not a limitation: everything outside the core feature is intentionally absent to isolate the variable being tested. When users have only one thing to do, their behavior around that one thing becomes clear. You see how often they use it, how long they spend, whether they return, and what they try to do next, which tells you what to build second.

The feature you build is the one that represents the core value exchange — the one thing that, if it works, makes the product worth using even without anything else. In practice that is the feature that directly addresses the validated customer problem, provides value the first time it is used (not after extensive setup or data accumulation), can be explained in one sentence, and is clearly differentiated from existing alternatives. Everything else is deliberately left out: user profiles and social features (unless that is the core feature), settings and customization, onboarding flows beyond the absolute minimum, analytics dashboards, secondary use cases even when they seem easy to add, and administrative tools, which you handle manually instead of building.

How to

Prep

★ AI Prompt

1. **Identify the core feature.** Review your solution interviews, prioritization exercises, and prototype tests. Which single feature generated the strongest positive signal? If there is disagreement on the team, that is a sign you need more research, not a compromise feature.
2. **Define the scope ruthlessly.** Write a one-sentence description of what the MVP does. If the sentence contains “and,” you have too much scope. Cut until it is a single capability.

3. **Pick the retention window before you build.** Decide up front whether your product is daily-, weekly-, or monthly-use, and which of D1 / D7 / D30 retention is the primary success metric — the share of users still returning 1, 7, or 30 days after first use. Picking afterward is how teams rationalize disappointing results.
4. **Define the acquisition channel.** Choose one channel — direct outreach to research participants, a landing page, a targeted community post — and write the one-sentence pitch. The channel and pitch shape who shows up and what they expect; mismatches show up later as low retention.
5. **Instrument before you ship.** Wire up retention analytics before launch. Capture every session, every action on the core feature, and a single open-ended “what did you wish this could do?” prompt for retained users.

Execution

🌟 AI Prompt

1. **Build for quality within the scope.** The single feature should work reliably, load quickly, and provide a good experience. Invest the time you saved by cutting scope into polishing the core feature.
2. **Ship to real users.** Release the single feature MVP to a small group of target users (50 to 200 is sufficient for initial retention data). Use the simple acquisition channel you defined in Prep.
3. **Watch the first week live.** Sit with the data daily for the first week. Bugs in the core feature will surface fast and they will distort retention if you do not catch them. After the first week, drop to a weekly cadence and let the retention curve develop.
4. **Listen for feature requests but do not act.** Track what users ask for most urgently and most frequently. Do not ship any of it during the test window. The whole point of the method is to measure retention against the bare core; adding features mid-test destroys the read.
5. **Capture departure signal.** When a user stops returning, send a one-question email: “What were you hoping to do that you couldn’t?” Track why users leave, not just why they stay.

TOOLS

Mixpanel

PostHog

Hotjar

Vercel

Supabase

Analysis

🌟 AI Prompt

1. **Read the retention curve first.** Plot D1, D7, and D30 retention. The shape matters more than any single number: a curve that flattens at a low percentage points to real value with a small audience; a curve that goes to zero means the feature is failing regardless of how high D1 was.

2. **Compare against the metric you picked in Prep.** D1 above 40%, D7 above 20%, D30 above 10% are usable thresholds for daily-use products; weekly- and monthly-use products will look different and need to be judged against their own use frequency. Honor the metric you committed to before you saw the data.
3. **Segment retained vs. churned users.** Compare demographics, acquisition source, and first-session behavior. Patterns in the retained cohort tell you who the real customer is — which may not be who you targeted.
4. **Cluster the feature requests by retained users only.** Requests from users who are still using the product carry far more signal than requests from users who never returned. The top 2 or 3 clusters from retained users are the candidate second feature.
5. **Decide the next move.** One of three things should happen: retention curve flattens at a usable level → keep building, the second feature is the top retained-user request; retention curve is borderline → narrow the audience and re-run before adding scope; retention curve goes to zero → do not add features, revisit the core value proposition.

⚠ Biases & Tips

Survivorship bias

Users who stick with a single-feature product are self-selected enthusiasts. Their behavior may not represent the broader market. Track not just retained users but also why departing users left.

Novelty effect

Initial usage may be driven by curiosity rather than genuine value. Wait for the retention curve to stabilize (usually 2 to 4 weeks) before drawing conclusions.

Expectation mismatch

If the acquisition channel or pitch sets up the wrong expectations, low retention reflects a messaging problem, not a feature problem. Confirm that your channel accurately describes what the product does before concluding the feature has failed.

Vanity-metric substitution

Signups, satisfaction scores, and session counts are easier to look at than retention and usually look better. Pick the retention metric in Prep and refuse to substitute later.

Next Steps

- Run [Discovery Interviews](#) with retained and churned users separately to learn what kept the keepers and what lost the leavers.
- Use a [Concierge Test](#) on the most-requested second feature to validate it before committing engineering time.

- Use a  **Wizard of Oz** to fake any complex backend addition before building it for real.
- Use a  **Mash-up** to assemble adjacent capabilities from off-the-shelf tools and learn which ones move retention.
- Use a  **Vibe-Coded Disposable MVP** to spin up cheap, throwaway variants of the next feature and ship them to a sub-segment.

Case Studies

Foursquare: Check-in only launch

Launched at SXSW 2009 with one feature — broadcasting “I am here” to friends — and used the resulting feature requests (tips, friend activity) to shape the post-MVP roadmap.

[Read more](#)

Instagram: Filtered photo as MVP

The team killed the bloated Burbn check-in app and shipped a single-feature replacement — post a square photo with a filter — using retention on that one flow to fund comments, search, and the rest of the product.

[Read more](#)

Superhuman: Speed-first email

Rahul Vohra gated the app behind one promise — the fastest email experience — and ran Sean Ellis “very disappointed” surveys as the retention proxy that drove the roadmap from a 22% to a 58% PMF score.

[Read more](#)

Dropbox: Sync-folder MVP

Drew Houston shipped a single feature — a folder on the desktop that synced — before any sharing or collaboration features, using retention on that core action to fund the rest of the product.

[Read more](#)

Further reading

-  Eric Ries — The Lean Startup (Crown Business, 2011)
-  Dan Olsen — The Lean Product Playbook (Wiley, 2015)

- ☰ Rahul Vohra — How Superhuman Built an Engine to Find Product/Market Fit (First Round Review)
- ☰ Sean Ellis — The Product/Market Fit Survey
- ☰ Andrew Chen — The Law of Shitty Clickthroughs

6.3.7 Vibe-Coded Disposable MVP

In Brief

A vibe-coded disposable MVP is a working prototype built in hours by describing the product to an AI coding assistant in plain language and letting it generate the code (“vibe coding”), with an upfront commitment to throw all the code away when the experiment ends. The MVP — minimum viable product — is the smallest working version that can test your idea. You deploy it to a live URL and put it in front of real users to test whether the solution delivers value. The output is behavioral evidence: did users complete the core task, come back, or ask to keep using it? The discipline that defines the method is pre-committing to disposal — the code tests a hypothesis, it is not a foundation for production. This method is for software products; for physical products, test form factor with a [3D Print](#) or a [Life-Sized Prototype](#) instead.

COMMON USE CASE

You have a product hypothesis you want to test with a working prototype as fast as possible, even if the code is throwaway. You describe the idea to an AI coding assistant, deploy a functional version in hours, and put it in front of real users to see whether they engage with it before you invest in production-quality development.

HELPS ANSWER

- Does our solution concept actually solve the customer’s problem?
- Will users engage with this type of product experience?
- Which features matter most to early users?
- Is this problem worth solving with software at all?



AT A GLANCE

Quantitative

Qualitative

Remote

Behavioral



TIME  ~5 days–3 weeks

COST  \$30–\$700

OTHER NAMES

Disposable MVP · Throwaway MVP · AI-Coded Prototype

Description

A vibe-coded disposable MVP is working software you build by describing it to an AI coding assistant in plain language (“vibe coding”) and commit to deleting before you write the first line. You set a deletion date before you start building, deploy the result to a live URL, and use it to test one hypothesis with real users. The “disposable” qualifier is the whole point: the code is a single-use test artifact, not the first version of a product.

Because the user interacts with real software that actually does the thing, not a simulation, you get behavioral evidence about whether the solution works, not just whether the interaction flow makes sense. That distinction matters when the hypothesis is about value delivery rather than usability, and it is what sets the method apart from a paper or [clickable prototype](#).

The pre-commitment to disposal is what separates this from a durable [single-feature MVP](#). Vibe-coded software tends to accumulate technical debt quickly, because the model optimizes for making the thing work right now rather than for the structure a production system needs. Setting a deletion date up front guards against sunk-cost reasoning, which turns a throwaway prototype into the hidden foundation of a company. The discipline of treating the cheap test as a separate artifact from the eventual production build comes from prototyping (see References).

How to

Prep

✦ AI Prompt

1. **Write a clear hypothesis.** Articulate exactly what you’re testing in the form: “We believe [target customer] will [specific measurable behavior] when given [solution concept] because [reason].” If you can’t write this sentence, you’re not ready to build anything. The MVP exists to test a specific hypothesis, not to be a small version of the product you wish existed.
2. **Define the minimum feature set.** List only the features required to test your hypothesis. If a feature doesn’t directly relate to the hypothesis, cut it. You’re building a disposable prototype to test one critical user behavior, not a product.
3. **Pre-commit to disposability.** Before writing any code, set a deletion date and write it in the README. Tell your team out loud: “We will delete this on [date], regardless of what happens.” Set a calendar reminder. Keeping the test artifact separate from any future production build is what protects the experiment from sunk-cost reasoning later.

4. **Choose your AI coding tool.** Pick one AI-native editor or AI app builder and commit to it for the build. Choose based on what you can deploy fastest, not what's most powerful. The right tool is the one you can ship from in an afternoon.
5. **Recruit 5–15 real users from your target segment.** Line them up before the build, not after. The window between “the prototype works” and “I should keep iterating instead of testing” is short, and recruiting after the fact is how that window closes. Make sure these are people from your actual target persona, not friends or fellow founders who will be charitable about a rough demo.

Execution

✦ AI Prompt

1. **Vibe-code the prototype.** Use your chosen AI coding assistant to build the MVP:
 - Describe the user flow in plain language.
 - Let the AI generate the code.
 - Don't refactor. Don't write tests. Don't set up CI/CD.
 - Focus on making the user-facing experience feel real enough to test.
 - Hardcode things that would normally be configurable.
 - Use placeholder data where it doesn't affect the test.
2. **Deploy to a live URL.** Put it somewhere real users can access it — any instant-hosting platform, whatever is fastest. The deployment doesn't need to be production-grade. It needs to stay up for the duration of the test.
3. **Run the experiment.** Put the prototype in front of the 5–15 users you recruited in Prep. Observe their behavior. Measure the specific outcome defined in your hypothesis. Conduct brief follow-up interviews — but lead with what they did, not what they thought.
4. **Record everything.** Session recordings, click paths, drop-off points. The whole point of building real software (rather than a clickable prototype) is that you get behavioral data — capture it.
5. **Delete the code.** This is the hard part. Do it on the date you committed to in Prep, regardless of how the experiment went. If the hypothesis was validated, build the production version from scratch with proper architecture, informed by everything you learned. The knowledge transfers; the code doesn't.

TOOLS

Cursor

Bolt

Lovable

v0

Hotjar

Analysis

✦ AI Prompt

The signal you're looking for is behavioral, not verbal. Watch what users did with the prototype, not just what they said about it.

Strong positive signals:

- Users completed the core task without prompting.
- Users asked “when will this be ready?” or “can I keep using this?”
- Users tried to use it for things you didn’t build (indicates high engagement).
- Users shared it with colleagues unprompted.

Weak or misleading signals:

- “This is cool” (politeness, not validation).
- High page views with no task completion (curiosity, not value).
- Users praise the design or technology (they’re evaluating the demo, not the solution).

Red flags:

- Users needed extensive explanation to understand the purpose.
- Users completed the task but showed no interest in continuing.
- Users immediately suggested fundamental changes to the concept (not the execution).

Biases & Tips

Sunk cost fallacy

The dominant failure mode of this method. Once the prototype works, founders are reluctant to discard it: “It’s already built, why throw it away?” Because the code was built to test a hypothesis, not to run a product. Vibe-coded prototypes tend to accumulate technical debt quickly, so keeping the code usually means building your company on a foundation you don’t understand. The mitigation is the pre-committed deletion date set in Prep — honor it regardless of the result.

Feature accretion bias

AI coding tools generate additional features on request with no friction cost, making it easy to add features that dilute the hypothesis. Each addition makes behavioral results harder to attribute to the core concept being tested.

Prototype-as-validation bias

A working demo creates a sense of progress that can overwhelm analytical judgment. Functionality proves the software runs, not that users want the solution. Those are different questions; a prototype that runs is not evidence that the hypothesis is confirmed. Judge the hypothesis on what users did, not on the fact that you shipped something.

**Selection bias
in testers**

Because the prototype is rough, you may unconsciously recruit participants who are more forgiving — friends, fellow founders, early-adopter types. Make sure your testers match your actual target persona, or the engagement you observe will overstate real demand.

**Anchoring on
implementation**

Once you've built a solution one way, it's psychologically harder to imagine fundamentally different approaches. The prototype anchors your thinking to a specific implementation even after you delete the code.

Next Steps

- If the solution resonates but you're unsure which value is driving demand, isolate it with a [Value Proposition Test](#) before committing to a rebuild.
- If validated, plan a proper rebuild as a [Single-Feature MVP](#) with production architecture.
- If invalidated, discard the MVP (it was meant to be throwaway) and test a different hypothesis.
- Write up what you learned before you delete the code, so the findings survive the prototype.
- Use [Usability Testing](#) with your disposable MVP to observe where real users get stuck before deciding what to rebuild properly.
- Run a [Product-Market Fit Survey](#) if enough users engage with the MVP to quantify whether you have product-market fit.

Case Studies

Further reading

- ☰ The Lean Startup — Eric Ries, Crown Business 2011
- ☰ The Right It — Alberto Savoia, HarperOne 2019
- ☰ Vibe coding — Andrej Karpathy on X, Feb 2 2025
- ☰ YC W25 batch — Garry Tan on X
- ☰ Lovable launches its vibe-coding app on iOS and Android — TechCrunch, Apr 28 2026

6.4 Product-Market Fit Survey

In Brief

The Product-Market Fit Survey is a single multiple-choice question that asks existing users, “How would you feel if you could no longer use this product?” with four answer options ranging from “very disappointed” to “no longer use it.” If 40 percent or more of qualified respondents pick “very disappointed,” that is a strong quantitative signal that the product has found product-market fit for that segment. The output is a percentage score you can track over time, plus follow-up questions that reveal what users value and where the product falls short.

COMMON USE CASE

You have a few hundred active users and your investors are asking whether it is time to spend heavily on marketing. Before committing budget, you want a quick quantitative check on whether your core users truly depend on the product or are casually trying it out.

HELPS ANSWER

- Are enough customers deeply attached to this product to justify scaling?
- Which customer group would miss this product the most?
- What is the single benefit users would not want to live without?
- What is missing for the people who are only somewhat disappointed?

Description

The Product-Market Fit Survey is a single multiple-choice question, sent to people who already use your product, that asks how they would feel if they could no longer use it. The answer options run from “very disappointed” to “I no longer use it.” The



AT A GLANCE

Quantitative

In-person

Remote

Attitudinal



TIME  ~1–3 weeks

COST  \$0–\$60

OTHER NAMES

Sean Ellis Test · PMF Survey ·
40% Test · Product-Market Fit
Test

headline result is the share of qualified respondents who pick “very disappointed”: if 40 percent or more do, that is a strong signal the product has found product-market fit for the segment you surveyed. A short battery of open-ended follow-ups runs alongside the multiple-choice question so the score comes with the reasons behind it.

The survey turns product-market fit — the otherwise fuzzy sense that you are in a good market with a product that satisfies it — into a number you can track. A high “very disappointed” share is a proxy for the value hypothesis: the assumption that customers will keep using the product because it solves a real problem for them. Below the 40 percent threshold, spending heavily on acquisition tends to be inefficient, because most users would not miss the product if it disappeared; above it, growth efforts are more likely to compound.

The 40 percent figure is a heuristic, not a hard cutoff. Treat the score as a starting point, not a verdict. The real value is in the follow-ups: comparing what “very disappointed” respondents say they value against what “somewhat disappointed” respondents say is missing tells you which segment to build for and what to build next. The “very disappointed” group also defines your high-expectation customer — the segment whose standards you design toward, because meeting them tends to pull the rest of the audience along.

The survey is a leading indicator, not a finish line. Run it on people who have genuinely used the product, re-run it as the product and audience change, and read it alongside retention and engagement data rather than on its own. This page covers the survey itself: how to qualify respondents, run the question, and read the score.

How to

Prep

✦ AI Prompt

1. **Define the user qualification filter.** Sean Ellis’s original guidance is to send the survey only to users who have (a) experienced the core of the product, (b) used it at least twice, and (c) used it in the last two weeks. Surveying casual or lapsed users contaminates the score with people who never had a chance to depend on the product. If your analytics platform doesn’t track these conditions, define them as best you can — first-week churn is normal and you don’t want to count it as “not disappointed.”
2. **Decide standalone or embedded, then pick the channel and tool.** A standalone single-question survey (in-app intercept, email link, or a dedicated form) keeps

the PMF question uncontaminated by surrounding questions. Embedding it in a longer customer survey is fine if you place the PMF question early. Common channels: in-app prompt, email to active users, in-product chat. Common tools: a free form builder, an in-app survey platform, or a dedicated PMF survey template (see Tools).

3. **Set a target sample size.** Aim for at least 30 qualified responses before treating the score as anything more than directional, and 100+ before treating it as a real signal. Below 30, a handful of answers swings the percentage by 10+ points.
4. **Draft the follow-up questions.** Three follow-ups turn the score from a number into a roadmap: *What type of person do you think would benefit most from this product?* (a segmentation cue that surfaces your high-expectation customer), *What is the main benefit you receive from this product?* (the value proposition in your users' words), and *How can we improve this product for you?* (the missing-feature gap, especially valuable from "somewhat disappointed" respondents). Add a fourth — *Why did you choose that answer?* — so you can read the qualitative pattern under the quantitative score.
5. **Pre-commit to segmentation analysis.** Decide before launch which segments you'll cut by (acquisition channel, plan tier, role, company size, signup cohort). The headline percentage often masks one segment well above 40 percent and another well below — the high-expectation customer segment is typically the one to build for. Choosing the cuts upfront prevents post-hoc data dredging.

Execution

✦ AI Prompt

1. **Filter the respondent list.** Apply the qualification criteria from prep — you want users who have actually used the core product at least twice in the last two weeks. AI-powered customer analytics tools can help identify which users meet the criteria when your database doesn't have clean behavioral filters.
2. **Send the survey.** The single quantitative question is:

How would you feel if you could no longer use this product?

- Very disappointed
- Somewhat disappointed
- Not disappointed (it isn't really that useful)
- N/A — I no longer use the product

Followed by the prep-drafted follow-ups (main benefit, type of person who would benefit most, what's missing, why this answer).

TOOLS

PMF Survey by Sean Ellis and GoPractice

Sprig

Typeform

Google Forms

Send via any of:

- Email link to qualified users
 - In-app intercept (survey platform or custom)
 - A form-builder link
 - A dedicated PMF survey template (see Tools)
3. **Run a tight collection window.** Two weeks is usually enough — open windows drag and let new behavioral signals contaminate the cohort. Send one reminder; do not send two.
4. **Do not coach respondents.** If a user replies asking what “very disappointed” means or which option to pick, do not steer the answer. The point is their unprompted reaction. Note the confusion as a signal that the question itself may need a small rewording for your audience, but do not change it mid-collection.

Analysis

★ AI Prompt

If 40 percent or more of qualified respondents picked “very disappointed,” that is the threshold that tends to correlate with a startup able to scale acquisition and grow. Below 40 percent, the product is not yet a “must have” for enough of the surveyed segment, and pouring marketing budget on top is likely to be inefficient.

That percentage is an estimate from a sample, so it carries a margin of error: at around 30 responses the true figure can sit roughly 15 points either side of what you measured, which is why 30 is only directional and 100+ is defensible. Read 38 and 42 percent as the same number, and treat a change between waves as real only when it is larger than that margin rather than ordinary sampling noise.

The headline number is the start of the analysis, not the end:

- **Read the qualitative follow-ups by quantitative bucket.** Compare the language “very disappointed” users use to describe the main benefit, against the language “somewhat disappointed” users use. The very-disappointed cohort tells you what the product really is in users’ minds; the somewhat-disappointed cohort tells you what’s missing for them to cross over.
- **Segment.** Cut the score by the segments you pre-committed to in prep. A 28 percent overall score that hides a 55 percent segment is a different problem than a uniform 28 percent — the first calls for narrowing the target customer; the second calls for fundamental product change.
- **Identify the high-expectation customer.** The segment that scores highest on “very disappointed” is the segment whose expectations you should design for. Building for the median user dilutes the product; building for the high-expectation segment tends to lift the median anyway.

- **Plan the next roadmap.** A roughly 50/50 split works well: half the roadmap doubling down on the benefits the very-disappointed cohort named, half closing the gaps the somewhat-disappointed cohort named.
- **Track over time.** Re-run quarterly. Product-market fit is not a single moment but a series of measurements that should hold, and ideally rise, as the product and audience evolve.

Paste the raw responses into an LLM to auto-categorize the open-ended follow-ups and surface the language patterns that distinguish “very disappointed” respondents from “somewhat disappointed” ones. That language analysis directly informs your positioning and marketing copy.

Biases & Tips

Order effect

If the PMF question sits inside a longer survey, earlier questions can prime the answer. Place this question early or run it as a standalone.

Threshold worship

Forty percent is a heuristic, not a physical constant. Treat 38 percent and 42 percent as the same number; treat the trend across quarters and the segment-level scores as more informative than the headline.

Selection bias in respondents

Users who bother to complete a survey skew positive. If your response rate is under 10 percent, your “very disappointed” share is probably overstated.

Conversational-AI administration bias

Administering the PMF question via a chatbot or conversational AI breaks the standardized format the 40% benchmark was calibrated against; AI-generated preamble or dynamic follow-up questioning shifts how respondents interpret the question, making the resulting score incomparable to the published threshold.

Next Steps

- If below 40 percent, focus the roadmap on the highest-scoring segment rather than the average user.
- Run [Discovery Interviews](#) with both cohorts: ask “very disappointed” respondents what they value most, and “somewhat disappointed” respondents what would move them to “very disappointed.”
- Build the features “somewhat disappointed” users say would make the product painful to lose.
- Set up [Dashboards](#) to track behavioral metrics alongside your score and check whether survey sentiment matches actual usage.

→ Re-run the survey quarterly to track progress over time.

Case Studies

Superhuman: PMF engine

Rahul Vohra's First Round account lifts the Sean Ellis "very disappointed" score from 22% to 33% to 58% via a four-step loop — segment, analyze, split roadmap 50/50, repeat — aimed at a high-expectation-customer persona named "Nicole."

[Read more](#)

Further reading

-  The Pmarca Guide to Startups, part 4 — Marc Andreessen, 2007
-  The Startup Pyramid — Sean Ellis (archived)
-  How Superhuman Built an Engine to Find Product/Market Fit — Rahul Vohra, First Round Review 2018
-  The Never Ending Road To Product Market Fit — Brian Balfour
-  The Lean Product Playbook — Dan Olsen, Wiley 2015

6.5 Usability Testing

In Brief

Watch real users attempt specific tasks with your product and record what happens. The product can be anything from a paper sketch to a fully built application. You give each participant a task, observe whether they can complete it, and note where they hesitate, get lost, or express frustration. The output is a prioritized list of usability problems — the points in your design where real people struggle — along with insight into why those problems occur.

COMMON USE CASE

You have built or redesigned a key flow in your product and want to see whether real users can complete it without getting lost or giving up. You sit with a handful of participants, give them a task, and watch where they stumble so you can fix the friction before you ship.

HELPS ANSWER

- How do people actually use this product or feature?
- Can users complete the key tasks without help?
- Where do users get confused or frustrated?
- What do users experience at each step of the process?

Description

Usability testing is a structured observation method where you put a representative user in front of your product, give them a realistic goal to accomplish, and watch where their behavior diverges from what the design intended. You record whether they finish the task, where they hesitate, what they misread, and how they feel along



AT A GLANCE

Qualitative

In-person

Remote

Observational



TIME  ~5 days–3 weeks

COST  \$0–\$400

OTHER NAMES

User Testing · UX Testing

the way. The aim is to find the points where the design confuses real people before you ship it to all of them.

The method works at any fidelity. A paper sketch tested with three people can expose a confusing label as effectively as a fully built product tested with twenty. Small, frequent tests run by the team that built the product close the loop faster than commissioning occasional large studies, and you do not need a big sample: five carefully chosen testers, run in iterative rounds, will surface most of the dominant problems in a flow. Usability testing tells you *what* breaks and *why*, pair it with methods that tell you *how often*, such as analytics or [A/B Testing](#).

Where remote platforms are available, AI is an optional accelerant for the slow parts: it can generate session transcripts, flag moments of hesitation or frustration from screen and audio signals, cluster patterns across sessions, and draft summaries, keeping analysis to minutes and making it practical to run more tests with more users. It accelerates synthesis but does not replace observation — a flagged frustration moment still needs a human to decide whether the design is wrong or the task itself was unfair, and watching a user struggle live builds an understanding that a transcript summary cannot reproduce.

How to

Prep

[AI Prompt](#)

1. **Define the task scenarios.** Write 3–5 realistic tasks that cover the critical paths in your product. Each task is a goal, not a click-by-click instruction. Example: “You just signed up. Find the feature that lets you invite a teammate.” Avoid leading the user toward the answer. Set a reasonable time limit per task (2–5 minutes).
2. **Recruit 5 testers who match your target user.** Five is the rule of thumb — past five, you mostly re-confirm problems you have already found. Recruit from existing users, social channels, or online research panels. For B2B, ask users directly; most will say yes if you make it short.
3. **Decide moderated vs unmoderated, then pick the tool.** Moderated sessions (video call or in-person) let you ask “what made you click that?” in real time but cost more researcher hours. Unmoderated sessions on remote testing platforms scale cheaply, and AI-assisted platforms can analyze think-aloud audio and behavioral signals across many sessions without a researcher watching each one in real time. Mix both when budget allows: a few moderated sessions to build empathy, a larger unmoderated batch to confirm patterns.

4. **Write a short intro script.** Two or three sentences that frame the session. The standard reassurance: “We’re testing the product, not you. If you get stuck, that’s exactly the feedback we need.” Pre-writing the script keeps facilitators consistent across sessions.
5. **Pilot once internally before the first real session.** Run the full protocol with a teammate. You’re checking whether tasks are intelligible, whether the recording setup works, and whether the timing fits. Half the issues you find in piloting would have wasted a real participant slot.
6. **Decide who moderates and who takes notes.** One person facilitates and asks the questions; a second person observes and captures notes. Trying to do both at once degrades both. For unmoderated tests, the “observer” role becomes whoever reviews the AI-generated session summaries and watches the flagged moments.

Execution

1. **Frame the session.** The facilitator reads the intro script written in Prep: explain the purpose, reassure the participant that any difficulty is feedback about the product (not the user), and confirm consent to record.
2. **Explain the first task.** Read the task scenario aloud and hand control to the user. Give context, not instructions. Do not tell them where to click or what to look for.
3. **Ask the user to think aloud.** Have the participant narrate impressions, intentions, and expectations out loud as they work. Hearing their reasoning is what makes the session diagnostic rather than a pass/fail score — it surfaces the *why* behind the behavior. Do not explain, coach, or interpret. Interject only to ask “what are you thinking right now?” when the user goes silent at a moment of hesitation.
4. **Capture the session.** Record audio, screen, and (where available) video. For unmoderated remote sessions, AI-assisted platforms auto-flag moments of hesitation, backtracking, repeated clicks, and emotional cues — useful triage when you cannot watch every session in real time.
5. **Repeat for each task.** Move through the rest of the task scenarios. Watch for fatigue; if the participant is flagging, cut a task rather than push through with degraded data.

TOOLS

Scrappy kit (free): a laptop or phone with built-in screen recording, a paper note template, and a quiet room — enough for moderated tests with people you can sit beside.

[Maze](#)

[UserTesting](#)

[Lyssna](#)

[Lookback](#)

[Hotjar](#)

6. **Run a short exit interview.** Thank the user and ask 2–3 open-ended follow-up questions to clarify their experience: “What was the most frustrating part?” “If you could change one thing, what would it be?” These prompts often surface issues the user did not articulate during the task itself.

Analysis

✦ AI Prompt

1. **Synthesize observation notes across sessions.** The facilitator and any observers compare notes, focusing on moments where users hesitated, backtracked, or expressed frustration. Even usability experts sometimes disagree on interpretation, so multiple observers reduce the chance that a single experimenter’s bias filters the findings. Identify the problems that affected most or all participants — those are the ones worth fixing first.
2. **Prioritize by frequency and severity.** Given the small sample size, treat consistent problems (3+ of 5 users hit the same wall) as confirmed defects. One-off issues might be real or might be participant-specific — note them but do not over-weight them. Severity is independent of frequency: a problem that happens once but blocks the user from completing the task scores higher than a cosmetic issue that everyone hit.
3. **Use AI tools to triage what to watch closely.** AI session-analysis tools auto-flag moments of confusion, generate heatmaps (a color overlay showing where users clicked, tapped, or looked most), report how many participants finished each task and how long it took them, and produce session summaries. Treat these as a way to triage which recordings to watch closely, not as a substitute for watching sessions yourself. Automated checks for contrast, touch-target size, and navigation depth can also catch surface-level issues before you spend a real participant slot on them.
4. **Distinguish usable from desirable.** If all users complete the tasks, that tells you the product is usable. It does not tell you whether the value proposition lands or whether anyone would actually pay for it. Usability testing is a necessary but insufficient validation step — pair it with desirability and value-proposition tests before you assume you are clear to ship.

⚠ Biases & Tips

Observer effect

Users behave differently when they know they’re being watched. Frame the session as testing the product, not the user, and use unmoderated remote tests for a less observed signal where appropriate.

**Social
desirability
bias**

Users may try to complete tasks (or answer questions) in a way that makes them look good to the experimenter. Reinforce that struggling is the desired output, not a failure.

**Confirmation
bias**

Experimenters can frame tasks or questions in ways that confirm their preconceptions. Have someone outside the design team review the task scenarios before you run sessions.

Selection bias

Testing usability only with existing power users hides the issues new users hit. Recruit a mix that matches the segment you're actually trying to serve.

**Automation
bias**

Over-trusting AI-flagged frustration moments can lead you to misattribute the problem: the design looks guilty, but the task scenario was the real cause. Treat automated signals as triage cues that direct your attention, then watch the flagged moment yourself to confirm whether the design or the prompt caused the breakdown.

Next Steps

- Fix the top issues and re-test with new participants to verify the fixes worked.
- Run [Competitor Usability Testing](#) to see how the same tasks play out on a rival product and find gaps you can own.
- Use [A/B Testing](#) to confirm at scale whether a usability fix actually improves the behavior you care about.
- Run a [Net Promoter Score Survey](#) after shipping improvements to track whether satisfaction rises over time.

Case Studies

Fidelity: Remote unmoderated testing

A Fidelity researcher reports running 40+ unmoderated remote studies, including a 117-participant comparison where Wikipedia users completed 71% of Apollo-program tasks correctly against NASA's 58%.

[Read more](#)

USF: Virtual library interface study

Maryellen Allen's academic study tested the University of South Florida's virtual library interface with real students performing realistic search tasks.

[Read more](#)

Nielsen Norman Group: Iterative homepage redesign

NN/g documents three rounds of task-based testing on interactive homepage prototypes with new and returning visitors across desktop and mobile, sorting feedback into copy, architecture, layout, and visual style.

[Read more](#)

Further reading

-  Usability Engineering — Jakob Nielsen, Academic Press 1993
-  Don't Make Me Think, Revisited — Steve Krug, New Riders 3rd ed.
-  Rocket Surgery Made Easy — Steve Krug, New Riders 2009
-  The Design of Everyday Things — Don Norman, MIT Press 2013 rev. ed.
-  ISO 9241-11:2018 — Usability: Definitions and concepts
-  Tools for Unmoderated Usability Testing (Nielsen Norman Group)
-  Usability Testing (Nielsen Norman Group, n.d.)

6.6 Wizard of Oz

In Brief

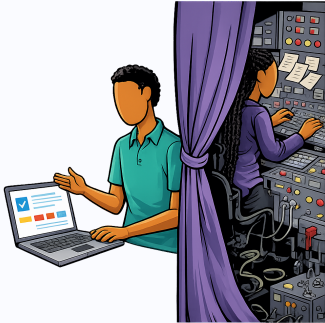
A Wizard of Oz test is a product experience that looks fully built to the user, but the response on the other end is generated by an operator behind the curtain — usually an LLM under human approval today, sometimes a person typing manually. The user does not know. You collect real engagement data and qualitative feedback on a finished-feeling experience without committing to the underlying system. The output is two things at once: signal on whether the value proposition lands, and a transcript of what the “wizard” actually had to do, which becomes your spec for the real build.

COMMON USE CASE

You have a clear hypothesis about what the product should do, but the system that would deliver it is expensive or risky to build. Instead, you stand up a polished front-end and let an operator — often an LLM under human review — generate the responses for the first 10–30 users. You learn whether the experience changes their behavior before you commit to the engineering.

HELPS ANSWER

- Will users come back, pay, or refer once they actually use this?
- What does the system actually need to do — and how often does each request type come up?
- Where does the experience break down: the inputs, the outputs, or the latency?
- Is this a real product, or is the magic actually the human in the loop?



AT A GLANCE

Qualitative

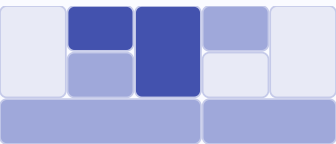
Quantitative

In-person

Remote

Observational

Behavioral



TIME  ~1–3 weeks

COST  \$40–\$400

OTHER NAMES

WoZ Test · Manual Backend MVP
· Hybrid Wizard of Oz

Description

A Wizard of Oz test is a product experiment in which a human manually performs the work behind an interface the user believes is automated. The user interacts with what looks like a finished product; an operator hidden behind the interface produces the responses. You measure how the user behaves on a finished-feeling experience without building the system that would deliver it.

The common form today is a hybrid. The user sees a real-looking interface. Behind it, an LLM generates the candidate response, and a human operator approves it, edits it, or rewrites it from scratch before it goes out. The operator is the wizard, even when they are no longer typing every word.

This matters because it changes what you're testing. A pure-AI prototype tests the model. A pure-human prototype tests the value proposition. A hybrid Wizard of Oz tests the value proposition *and* shows you exactly where the model would have failed if you had shipped the AI-only version — every override the operator makes is a labeled defect in the eventual automated build.

When to reach for Wizard of Oz vs. just shipping a real LLM-backed prototype:

- **Use Wizard of Oz** when getting a wrong answer to a real user would be expensive or harmful, when the failure modes of the underlying system are unknown, when you want to count override rates as a quality bar, or when the system you eventually want is materially harder than what an off-the-shelf LLM can do.
- **Use a [Vibe-Coded Disposable MVP](#)** when an off-the-shelf LLM with a tight prompt is good enough that you don't need a human in the loop, and you'd rather watch real users in a real product than read transcripts.
- **Use a [Concierge Test](#)** when you don't yet know the right shape of the product. In a Concierge Test the user knows a human is delivering the service, so you can ask "is this even useful?" before you hide the work behind a fake automation.

The pure-human wizard — no LLM involved — still fits two situations: the system you're testing is fundamentally not a language task (logistics, physical fulfillment, hardware control) and the user-visible "magic" is a non-AI service; or you want to evaluate a target capability that current models cannot reach, and you need a human ceiling to define what success would look like.

Ethics — The user believes they are using an automated product when a human is doing the work, and if you charge them, they are paying real money on that belief. Keep the deception proportionate: do not let the wizard give advice a real person

would act on in high-stakes domains (medical, legal, financial), refund anyone who paid for a capability you have not built, and never collect or retain data you would not be allowed to collect in the shipped product.

How to

Prep

✦ AI Prompt

1. State the value proposition you are testing.

Write it as a single sentence. “We help [user] do [task] so they get [outcome] without [the part that’s expensive to build].” If you cannot finish that sentence, you are not ready to run a Wizard of Oz — you are still in problem discovery, and a

 **Concierge Test** is the better method.

2. Decide what’s faked and what’s real.

The wizard’s job is the one capability that would be expensive or risky to build. Everything else can be real or stitched together with no-code tools. List the parts of the experience and mark each: **real**, **faked by wizard**, or **stubbed (placeholder)**. The fewer pieces the wizard is responsible for, the cleaner your signal.

3. Pick the wizard mode.

Choose one of three:

- **LLM-assisted (default).** An LLM drafts every response. The human operator reviews, edits, or rewrites before sending. Best for language-heavy products (advisors, summarizers, schedulers, agents).
- **LLM-free.** The operator types every response. Use this when the task is not a language task, when you want a clean human-ceiling baseline, or when LLM responses would be obviously wrong in this domain.
- **Tool-augmented LLM.** The LLM plus tool calls (search, calendar, database) drafts a response; the operator reviews. Use this when the eventual product needs to take actions, not just generate text.

4. Build the front-end.

Create whatever the user will interact with: a web form, a chat interface, an app screen, a chat-workspace bot, an email address. It should look like a finished product. Vibe-code it, use no-code form and app builders, or wire up a bot in your team chat tool. The user should not be able to tell that there’s a person in the loop — that includes latency cues. If your real product would respond instantly and your wizard

takes three minutes, build in an explicit “thinking...” or “results within 5 minutes” affordance so the delay reads as design, not as a tell.

5. Build the operator console.

The operator needs three things on one screen: the incoming request, the LLM-drafted response (if applicable), and a single Approve / Edit / Reject control. Log every approval, every edit (with the diff), and every rejection. The edit log is the most valuable artifact this test produces — it’s a labeled dataset of where the eventual automation will fail.

6. Write the wizard playbook.

A two-page document the operator can read in five minutes. It defines:

- The persona the wizard is responding as (tone, voice, length, what they will and will not do).
- A response template for each request type you expect to see.
- An escalation rule: when does the wizard say “I don’t know” or “let me get back to you” rather than make something up?
- A safety rule: what categories of request must be rejected outright (medical advice, legal advice, anything outside the tested value proposition)?

If two operators will share shifts, the playbook is what keeps their behavior consistent.

Execution

1. Recruit 10–30 real users.

The same audience you would target with the real product. Free users will give you fluffy data; people who pay (or who give you something costly — time, data, a referral) will give you signal. Set the test window: one to three weeks is typical.

2. Run the test live.

The operator works requests as they come in — drafting (or reviewing the LLM draft), editing, sending. Cover the actual hours your users are active. Do not batch responses overnight unless the real product would also respond overnight; latency is part of the value proposition you are testing.

3. Log everything.

TOOLS

[Retool](#)

[Airtable](#)

[Zapier](#)

[OpenAI](#)

[LangSmith](#)

For each interaction, capture: incoming request, LLM draft (if any), final response, edit diff, response latency, operator notes (one line: “had to look up the answer,” “model hallucinated, rewrote from scratch,” “out of scope, escalated”). This log is the test.

4. Watch for drift.

When two operators share shifts, the wizard’s voice will drift. When an LLM generates the draft, the operator can become a rubber stamp (“looks fine, send”). Both kill the data. Spot-check one in five responses against the playbook — if the operator is approving without reading or editing without restraint, retrain mid-test.

5. Run the same diagnostic the real product would run.

If the eventual product is supposed to drive a behavior — a return visit, a payment, a referral — measure that behavior, not just whether the user enjoyed the response. A Wizard of Oz that produces five-star reviews and zero return visits is telling you the magic was the human, not the value proposition.

Analysis

✦ AI Prompt

1. Read the override log first.

Every edit and rejection is a labeled defect. Cluster them: which request types did the LLM consistently get wrong? Which did the human have to write from scratch? Which would have been actively harmful if the user had received the LLM’s first draft? This is the gap between “we have a working prototype” and “we have a real product.”

2. Compute the override rate.

Approve / Edit / Reject as a percentage of total responses. A 90% approve rate means an off-the-shelf LLM is probably enough — drop the wizard and run a

 **Vibe-Coded Disposable MVP** instead. A 30% approve rate means the human is doing most of the work — the eventual product is a much harder build than you thought, and you need to decide if the value proposition is strong enough to justify it.

3. Measure the real behavior.

Did users come back? Did they pay? Did they refer? Compare to the assumption you wrote down in Prep step 1. Surface-level satisfaction does not validate the value proposition; behavioral commitment does.

4. Estimate the cost of automation.

For each cluster of requests in the override log, estimate what it would take to build that capability: a better prompt, a fine-tune, a tool call, a custom model, or “we cannot

build this in the next 12 months.” Multiply by frequency. The clusters that are common AND expensive to automate are where your engineering risk lives.

5. Decide the next move.

One of three things should happen:

- The override rate is low, users came back, the real behavior is there → ship a real build, no wizard.
- The override rate is high but the value proposition lands → keep the wizard, scale operators, treat the human-in-the-loop as a feature for now.
- The override rate is irrelevant because the real behavior didn't show up → the magic isn't real. Stop. Go back to a [Concierge Test](#) and figure out what users actually want.

 Biases & Tips

Observer effect	Users behave differently when they suspect they are being watched or that the system is not what it appears. Make sure the front-end and the latency are good enough that the user is not actively wondering whether a human is on the other end.
Rubber-stamp bias (LLM-assisted mode)	When the LLM drafts every response, the operator stops reviewing carefully. Flag a random 10% of responses for blind double-review by a second person, or the override-rate metric becomes meaningless.
Anchoring on the first generation	Operators tend to keep most of what the LLM produced even when a fresh response would be better. Build “regenerate from scratch” into the console as a one-click option, and track how often it gets used.
Automation bias	Operators under time pressure defer to plausible-sounding LLM outputs rather than evaluating them critically; a confidently wrong answer that reaches the user contaminates both the response data and the behavioral outcome you are trying to measure.
Human-attention false positive	Users respond warmly to anything that feels personally attended to. A high satisfaction score with no return visits usually means you measured the human, not the product. Pair satisfaction with a behavioral metric.
Deception debt	A Wizard of Oz test deceives users into believing an automated system is real, so plan to debrief participants afterward where feasible, never ship anything that could harm them on the strength of

that belief, and honor any commitment the wizard made during the test.

Next Steps

- If override rate is low, ship a **Vibe-Coded Disposable MVP** without the human in the loop and see if the behavioral metrics hold.
- If override rate is high but the value proposition landed, productize the wizard: hire or contract operators, formalize the playbook, treat the human-in-the-loop as a feature, and automate the largest override clusters first — each cluster becoming a **Single-Feature MVP**.
- If the value proposition did not land, drop the front-end and run a **Concierge Test** to re-discover what users actually need.
- Run a **Product-Market Fit Survey** with the Wizard of Oz cohort once you have enough returning users to measure whether they would be disappointed without the product.

Case Studies

Strella: Camera-off AI moderator

According to Bessemer's account, co-founder Priya Krishnan joined Zoom interviews with her camera off and a deliberately robotic voice, pausing after each response to simulate AI latency, validating demand before any model was wired up.

[Read more](#)

Aardvark: Human-routed social search

The social-search service routed questions to friends and friends-of-friends in the asker's network who answered them live, validating a knowledge-market interaction pattern before Google acquired it, reportedly for around \$50M, in early 2010.

[Read more](#)

Zappos: Manually-fulfilled shoe orders

Nick Swinmurn photographed shoes in local stores and posted them online, buying and shipping each pair himself when an order came in — a non-language Wizard of Oz validating an inventory-and-fulfillment system that did not exist; Amazon later acquired the company in a deal reportedly valued at around \$1.2B in 2009.

[Read more](#)

Further reading

- ☰ J.F. Kelley — An Iterative Design Methodology for User-Friendly Natural Language Office Information Applications (ACM TOIS, 1984)
- ☰ Maulsby, Greenberg, Mander — Prototyping an Intelligent Agent through Wizard of Oz (CHI '93)
- ☰ On LLM Wizards: Identifying Large Language Models' Behaviors for Wizard of Oz Experiments (ACM IVA 2024)
- ☰ Prototyping Multimodal GenAI Real-Time Agents with Counterfactual Replays and Hybrid Wizard-of-Oz (arXiv 2510.06872, 2025)
- ☰ The AI of Oz: A Conceptual Framework for Democratizing Generative AI in Live-Prototyping User Studies (Applied Sciences, 2025)
- ☰ Concierge vs. Wizard of Oz Prototyping (Kromatic)
- ☰ Pay No Attention to the Man Behind the Curtain (Kromatic)
- ☰ The Wizard of Oz Method in UX (Nielsen Norman Group)

PART 7

Out of the Box



“Constraint inspires creativity.”

—Biz Stone

CONTENTS

[A/B Testing](#)

[Off-Brand Testing](#)

[Scorecards](#)

7.1 A/B Testing

In Brief

A/B testing randomly splits your participants into two groups, shows each group a different version of a product, feature, or design element, and measures which version performs better against a metric you choose in advance — signups, purchases, or the share of participants who use a feature. Both versions run at the same time under the same conditions, so the only difference between the two groups is the change you are testing.

COMMON USE CASE

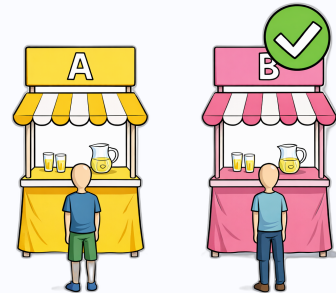
Your landing page converts visitors at 2% and you believe a different headline could do better. You want to show half your visitors the current headline and the other half a new one, then measure which version gets more signups over two weeks.

HELPS ANSWER

- Which version of this page or feature converts more participants?
- What design changes increase sales or signups?
- Does this change actually improve the metric we care about?
- How much of a difference does a specific change make?

Description

A/B testing shows two versions of the same element to your participants and lets their behavior decide which one wins. As the name suggests, you compare two versions (A and B) that are identical except for one change that might affect what participants do. Version A is usually the current version — the **control**, the baseline you measure against. Version B changes one thing — the **variant**. You split incoming



AT A GLANCE

Quantitative

Remote

Behavioral



TIME  ~1–4 weeks

COST  \$0–\$1.4K

OTHER NAMES

Split Test · Split Testing

participants randomly between the two and compare a single number for each: the share who take the action you care about, such as the percentage who sign up or buy.

The method is most often used on web pages, where the goal is to find changes that increase an outcome you care about: revenue, signups, downloads, leads, donations. Testing copy, layout, images, or colors can move those numbers, and even a small improvement on a high-traffic page can add up to a meaningful gain.

A purchase flow — the sequence of pages a visitor moves through from product page to completed order — is a common place to test, because reducing the share of people who drop out at each step turns directly into more sales. Whatever the metric, A/B testing measures whether a specific change actually moved it rather than leaving you to guess.

A/B testing changes one thing at a time. Testing several changes together (for example a new headline *and* a new button at once) is **multivariate testing**, a related but more involved method that needs more traffic to untangle which change did what. A/B testing also assumes you control the conditions and the random split; it does not apply to data you merely observe after the fact, such as survey responses or historical records, where you cannot assign people to versions.

Imagine a meal-kit service, Greencrate, whose growth depends on turning first-time visitors into trial subscribers. The signup page is where most of that conversion happens or fails, so it is the natural place to look for a win.

The current signup page leads with the headline “Sign up for Greencrate.” The team suspects the headline does no persuasive work — it names the action without giving a reason to take it. They write a variant, “Dinner solved — fresh ingredients and recipes at your door,” and leave everything else on the page identical. An A/B testing tool serves the two pages and records which produces more completed signups.

While the test runs, every visitor is randomly assigned to one page or the other, split evenly, and sees only their assigned version. Nothing else about the page differs between the two groups.

After enough visitors have passed through both versions, the team reads the result: the benefit-led headline converts at 6.2 percent against 4.8 percent for the original — a gap wide enough, on the traffic collected, to act on. Greencrate adopts the new headline and carries forward the lesson that a headline naming the customer’s outcome tends to beat one naming the company’s action.

You can also test more than two versions of the same one thing — say three possible headlines. Splitting your participants (or, for an email, your recipients) into three

groups instead of two is still the same method, and it is more efficient than running three separate two-way tests. Give a three-way test an extra couple of days so each group still gathers enough results to draw a conclusion.

Some AI experimentation platforms take this further, generating the versions themselves — copy, layout, even imagery — and automatically steering traffic toward whichever is winning instead of waiting for you to read a dashboard and pick. That suits high-volume optimization of a fixed goal and audience (pricing-page versions for a known segment, checkout tweaks, rotating ad creative), but it is the wrong tool for discovery: it optimizes hard against whatever metric you give it without questioning whether the metric or audience is right (see the local-optima and wrong-target pitfalls in Biases & Tips).

How to

Prep

★ AI Prompt

Decide what you are testing, the metric that defines the win, and whether you have enough traffic before writing any variant code. AI tools can do the sample-size math and flag confounders for you, but the decisions are yours.

1. **Confirm you have enough traffic.** A/B testing only works once enough participants pass through each version that the result is unlikely to be chance — what statisticians call *statistical significance*. That takes volume. Use a sample-size calculator (see References) to check. As a rough guide, with under 1,000 unique visitors per week to the page, you likely cannot finish a meaningful test without waiting months. If that is you, switch to a method that learns from a handful of people — [Usability Testing](#) or [Discovery Interviews](#) — or test bolder changes that produce bigger, easier-to-detect differences. For B2B software teams, the best opportunity is usually inside the product (onboarding, feature adoption, the pricing page), not the marketing site: a feature flag (a switch that turns a change on for some users and off for others) lets you test against existing users where you control who sees what.
2. **Define one hypothesis.** State what you are testing, what metric will change, and by how much. Example: “Changing the onboarding flow from 5 steps to 3 will increase the activation rate by at least 5 percentage points.” One change at a time — testing several at once is multivariate testing, not A/B.
3. **Pick the primary metric and the smallest difference worth detecting.** Choose the single metric that matters most for this test — signups, activation, purchases, retention. Then decide the smallest improvement that would actually

justify shipping the change, and write it down before you look at any data. This number (often called the minimum detectable effect) sets how big a sample you need and stops you from chasing a 0.1% win that isn't worth the work.

4. **Calculate the required sample size and duration.** Feed your current baseline rate, the smallest difference worth detecting, and your daily traffic into a sample-size calculator. It tells you how many participants each version needs and how many days that will take. Decide now to run that long — do not stop early because the numbers “look significant” partway through (see Peeking bias in Biases & Tips).
5. **Design the control and variant.** Version A is the current experience (the control). Version B changes one thing only (the variant). Everything else stays identical. Write down what “shipping the change” means so the engineering work is scoped before the test, not after. Decide how many variants to run, too: two is the default, but a high-traffic site optimizing a fixed goal can use an AI-managed multi-variant platform to run many at once and auto-allocate traffic toward the winner.
6. **Pilot the setup.** Run the test on 1% of traffic for a day to confirm the tracking fires correctly, the split is even, and the metric records as expected. A broken tracking tag discovered on day 14 is a wasted test.

Execution

★ AI Prompt

1. **Set up the test in your tool.** Use a feature flag or A/B testing platform to randomly split participants between the control and the variant. Verify the split is even within the first hour of traffic. Confirm both versions run at the same time under identical conditions — same time of day, same channels, same audience.
2. **Run for the full planned duration.** Do not check partway through and stop the moment results “look significant.” Run until you reach the sample size from Prep step 4. Some tools offer always-valid testing, a built-in method that lets you look at results as they come in without inflating the odds of a false result — if yours does, follow its stopping rule rather than improvising.
3. **Watch for breakage, not winners.** While the test runs, check for instrumentation drift, pipeline outages, or external events (a viral post, an outage, a concurrent campaign) that could contaminate results. Pause and restart if contamination is severe — do not try to “correct” the data after the fact.
4. **End the test cleanly.** When the duration completes, freeze the data, snapshot the assignment logs, and remove the test infrastructure from the live experience as soon as the decision is made. Lingering test code is a future bug.

TOOLS

[LaunchDarkly](#)

[Statsig](#)

[GrowthBook](#)

[Optimizely](#)

Analysis

✦ AI Prompt

1. **Decide how sure you need to be — and don't over-demand it.** Statistical significance just measures how likely your result is real rather than a fluke of chance. Tools usually report it as a *p-value*: 0.05 means roughly a 5% chance you'd see a gap this big even if the two versions were truly identical (often stated as "95% confidence"). To check whether your numbers clear that bar — or how many more participants you'd need — run them through an experiment calculator, or hand the counts to a statistician or an AI to do the math. But 0.05 is a convention, not a requirement: an early-stage startup rarely has the traffic to reach it, and rarely needs to. Deciding whether to keep iterating on a headline is not the bet a public company makes when it changes checkout for millions of users. If the change is cheap and reversible and the early numbers point clearly one way, act on the weaker evidence and keep moving — holding out for a *p-value* your sample size will never reach is its own mistake. Match the bar to the cost of being wrong.
2. **Check whether the result is worth acting on.** A small *p-value* only tells you the difference is probably not zero — not that it is big enough to matter. A 0.1% improvement that is statistically solid on millions of participants may not be worth the work to build and maintain. Weigh the size of the change against what shipping it costs.
3. **Segment the results.** Aggregate results can hide that one segment loved the variant and another hated it. Slice by acquisition channel, device, new versus returning, and any segment where you expect different behavior. Treat segment effects as hypotheses for the next test, not conclusions.
4. **Decide: ship, kill, or iterate.** If the variant wins on both bars (statistical and practical), ship it and document the learning. If it loses, kill it and document why you thought it would work. If results are inconclusive, decide whether the next move is a bolder change, more traffic, or a different method — not a longer run of the same test.

⚠ Biases & Tips

Peeking bias

Checking results before the planned sample size is reached and stopping the moment they "look significant" sharply raises the odds of a false win: the more often you look, the more likely you are to catch a random swing and mistake it for a real effect. Pre-commit to the sample size from Prep, or use a tool with always-valid testing built to handle repeated looks.

Novelty effect

A new variant often outperforms the control simply because it is new and users notice it. The lift fades after 2-4 weeks. For changes that will live in production long-term, run the test long enough to see the novelty wear off, or follow up with a holdout cohort.

Randomization imbalance

If the random split is uneven across important segments (gender, plan tier, device, geography), the comparison is contaminated and the measured lift cannot be trusted.

Multiple comparisons

Run 20 tests at once and one will likely “win” at the usual cutoff by luck alone, the same way one of 20 coin flips comes up heads several times in a row. When you run many tests, ask your tool to tighten the cutoff to account for the number of tests (a standard correction it can apply), or focus on one test at a time.

Survivorship bias in results

The participants who reach the end of a flow are not a random sample of everyone who entered it, and where people drop out can itself differ between the control and the variant. Measure every step of the flow, not just the final action, so you see where each version gains or loses people.

Confirmation bias in analysis

If you want the variant to win, you will find a segment where it does. Pre-register the primary metric and the segments you will analyze before the test ends. Treat post-hoc segments as hypotheses, not findings.

Method-fit failure

A/B testing tends to mislead on brand-new features no one has seen before (the novelty effect dominates), on changes too large to pin the result on a single variable, and on tests where the variant adds something the control simply lacks. For brand-new or sweeping changes, where novelty dominates or you cannot isolate one variable, the result is usually too tangled to interpret — reach for a qualitative method instead.

Local-optima trap (AI-managed multi-variant)

Auto-convergent platforms optimize hard against the metric you give them. They do not propose bolder reframings or cross-segment alternatives. After a few rounds of converging variants, you can plateau on a winner that beats the control but sits well below what a different framing or a different segment would reach. Periodically rerun against a deliberately distant variant to check whether you have flatlined.

Wrong-target
optimization
(AI during
discovery)

AI-managed multi-variant tools faithfully optimize for whichever signal converts now. They do not know whether the signups, clicks, or conversions are coming from the audience you actually want to serve, or whether the pain point your variants happen to address is the one you intended to validate. Run during discovery — when the value proposition or target customer is not yet locked in — and the platform will happily converge on a value proposition the founder is not actually interested in. Lock the segment and value proposition first (via [Discovery Interviews](#) and [Landing Page Test](#)) before letting AI take the wheel.

Next Steps

- If you cannot reach significance with the traffic you have, test a **bolder** change instead of a subtle one. A dramatic difference — a whole new page, not a new button color — produces a larger effect that shows up clearly in a far smaller sample, so an early-stage startup learns faster from big swings than from fractional tweaks.
- If a result was inconclusive because traffic was thin, validate the idea with a smaller-sample method first — a [Landing Page Test](#) or [Usability Testing](#) tells you whether a change is worth a full A/B test before you spend the run time.
- If thin traffic is the recurring blocker, drive more participants to the next test with an [Online Ad Test](#) so it can reach a trustworthy result faster.
- Set up [Dashboards](#) to monitor the long-term impact of winning variants and catch any regression over time.

Case Studies

Majestic Wine: Wedding-page redesign

A redesign of the UK wine retailer's wedding-services category page — clearer headings, prominent CTAs, less clutter — lifted enquiry-form submissions by 201% at 99.99% statistical significance. A follow-up that removed a competing PDF download button added another 36.9%.

[Read more](#)

Six Pack Abs: Pricing test

Carl Juneau tested \$19.95 against \$29.95 on a workout product checkout page. Conversion rates were statistically equivalent (1.1% vs 1.0% across roughly 1,200–1,400 participants per arm), and the higher price generated 61.67% more revenue.

[Read more](#)

Airbnb: Three experiment pitfalls

The engineering team warns against stopping early on a p-value that crosses 0.05 in week one, judging on a single aggregate metric (a search redesign looked neutral overall but was broken in Internet Explorer, a 2% loss), and trusting the assignment system without running A/A dummy tests.

[Read more](#)

Grab: Chat to reduce booking cancellations

Grab tested in-app chat between drivers and riders as a way to drive down booking cancellations, finding that prompting an early automated GrabChat message cut post-allocation cancellations by up to 2 percentage points.

[Read more](#)

Coinbase: Restoring trust in experimentation

Coinbase's in-house platform (CIFER) lacked sequential confidence intervals and CUPED, and a recommendation system it had marked a success turned out to have no real impact when re-tested. After moving to Eppo (now Datadog Experiments), the team reported 40% faster experiment analysis.

[Read more](#)

Notion: Scaling experiments with Statsig

Notion moved from single-digit experiments per quarter on an in-house tool to 300+ per quarter on Statsig, released 600+ features behind flags, and reported a 6% lift in activation rate.

[Read more](#)

Statsig: Experimentation in the age of AI

Statsig argues that as AI cheapens the build phase, measurement matters more: offline evals feed production experiments, AI-generated code needs metric-linked logging, and every engineer becomes a growth engineer testing ~10x more concepts. Customers cited include OpenAI, Figma, Notion, Grammarly, Atlassian, and Brex.

[Read more](#)

Booking.com: Homepage-redesign experiment

In December 2017 Booking.com's design director proposed testing an entirely new homepage layout that stripped the interface down to destination, dates, party size, and three options (accommodations, flights, rental cars) — discarding years of optimized content as a test of the company's experimentation culture.

[Read more](#)

Booking.com: 25,000 tests per year

Booking.com runs more than 25,000 A/B tests annually (about 70 per day) and has grown roughly twice as fast as the S&P 500 over more than a decade, the canonical case for experimentation as a competitive moat.

[Read more](#)

Further reading

-  [A/B Testing](#)
-  [Online A/B Testing Course](#)
-  [A/B Testing Sample Size Calculator Tool](#)
-  [Experiment Calculator](#)
-  [How Long Should You Run Your A/B Test?](#)

7.2 Off-Brand Testing

In Brief

Off-brand testing presents a product or concept under a neutral, fictitious, or unbranded identity so that customer responses reflect interest in the idea rather than loyalty to (or bias against) an existing brand. You run a small-scale test — typically a landing page, ad campaign, or product trial — stripped of recognizable logos, colors, and messaging. The output is a conversion rate or engagement signal showing whether the concept has demand on its own merits, giving you data to justify or kill a larger investment.

COMMON USE CASE

Your company has strong brand recognition, and you want to test whether a new product idea has genuine demand or whether people are just responding to your logo. You set up a simple landing page under a neutral brand name and measure how many strangers sign up based on the idea alone.

HELPS ANSWER

- Is there genuine interest in this idea, separate from our brand reputation?
- Would people want this product even if they did not know who made it?
- Are there customer groups outside our current base who would be interested?

Description

Off-brand testing is an experiment that presents a product or concept under a neutral, fictitious, or unbranded identity, so that the responses you measure reflect interest in the idea itself rather than loyalty to (or bias against) an existing brand. You run a small-scale test — usually a landing page, ad campaign, or product trial — stripped of



AT A GLANCE

Quantitative

In-person

Remote

Behavioral

Transactional



TIME  ~1–4 weeks

COST  \$250–\$2.5K

OTHER NAMES

Off-Brand Test · White-Label Test

recognizable logos, colors, and messaging, and read the conversion rate: of the people who see the offer, what percent take the action you asked for.

Brand associations shape how people perceive a product before they evaluate the offer itself. The mental shortcut a person attaches to a brand can dominate their reaction to anything that brand presents, so by the time a prospect reaches your branded landing page, the brand is already doing part of the persuading. If those visitors have been repeatedly exposed to the brand, especially with positive associations, the conversion rate you observe will likely run higher than if the same offer appeared under a neutral identity. That can produce a false positive: you walk away thinking you have a strong product idea when the result really reflects trust in the brand. Off-brand testing strips the brand away so you can read raw demand for the idea.

Getting this read right matters because a strong brand can prop up a concept that has no demand of its own. Knowing which concepts have pull on their own merits — versus which only convert because a strong brand carries them — is what keeps a portfolio from funding products that cannot stand alone.

The method extends the smoke-test approach — a minimum-viable test of demand — by stripping out the brand so the signal reflects interest in the concept itself rather than recognition of who made it.

The technique also works in reverse. If a company or brand has a poor reputation in a particular segment, an off-brand identity lets it evaluate products or features without that baggage dragging down the signal.

A common application is launching an app in a similar but smaller market before the final target market — sometimes called “test in Canada” by practitioners. The geography is incidental; the principle is the same as off-brand testing in general: get a clean read in a context where the brand isn’t doing the work for you.

How to

Prep

✦ AI Prompt

1. **Define the hypothesis.** Write one sentence: “We believe [target customer] will [specific action] when presented with [concept] — even without knowing who made it.” If you cannot state it in one sentence, you are not ready to design the test.

2. **Choose the off-brand identity and a defensible reason for the choice.** Decide on a name, visual direction, and tone. Use an AI tool to generate options if helpful, but pick one you can defend: explain to a reviewer why this identity is a clean read rather than a thinly disguised corporate test page. The identity should look professional but unremarkable — you want people to evaluate the idea, not the brand.
3. **Choose the audience and channel.** Pick a target customer profile and a channel that will reach those prospects without leaking the connection to your real brand. Paid ads, targeted communities, or a related-but-smaller geography (the “test in Canada” pattern) are common. Posting from your real company’s social accounts is not.
4. **Pre-register what counts as success.** Before you spend on traffic, write down the conversion threshold (e.g. $\geq 3\%$ sign-up rate), the sample-size minimum (e.g. 500 unique visitors), and any subgroup cuts you intend to look at. Pre-registration prevents post-hoc rationalization of a noisy result.
5. **Set a timebox.** Pick a stop date and a stop budget. Off-brand tests that run “until we feel sure” tend to keep running until the team is sick of them and the data is contaminated by drift. Two to four weeks is typical for a landing-page-style off-brand test.

Execution

✦ AI Prompt

1. **Build a minimal test property.** Stand up a landing page (a simple site builder or vibe-coded) under the off-brand identity. Present the concept clearly: what it does, who it’s for, and one call to action (signup, waitlist, pre-order). Avoid persuasion tactics, scarcity messaging, and heavy CRO — you are measuring raw interest, not marketing skill.
2. **Drive traffic from your target audience.** Use the channel chosen in Prep. The traffic source must not be connected to your real brand. AI ad platforms can handle targeting and optimization with minimal manual effort, and AI-powered localization tools can quickly adapt the test for different geographies and languages if you are running the “test in Canada” approach across multiple markets.
3. **Run the test against the pre-registered plan.** Hold to the timebox and budget you set in Prep. If conversion is wildly higher or lower than expected, do not pull the test early — early stopping is one of the easiest ways to fool yourself. Note anomalies in a log, but let the planned sample accumulate.

TOOLS

[Carrd](#)

[Framer](#)

[v0](#)

[Unbounce](#)

[Webflow](#)

4. **Gather raw conversion data.** Track unique visitors, conversions, and any subgroup cuts you pre-registered (channel, geography, audience segment). Export the data rather than relying on the dashboard view, so you can re-cut it during Analysis.

Analysis

✦ AI Prompt

1. **Compute the conversion rate against the pre-registered threshold.** Did the off-brand test clear the bar you set in Prep? If yes, the concept has demand on its own merits. If no, the demand you previously saw under your brand was likely brand loyalty, not product-market fit for this specific idea.
2. **Compare to a branded baseline if you have one.** If you also ran (or have data from) a branded version of the same concept, the difference between the two conversion rates is the brand-lift delta — how much of the demand is the brand doing versus the concept. A small delta tells you the concept stands on its own; a large delta tells you the brand is doing the heavy lifting.
3. **Decide on next steps.** If the off-brand test cleared the bar and the brand-lift delta is small, you can bring the concept under your main brand with confidence — or even drop the off-brand identity going forward, because the brand isn't adding the lift you assumed it was. If the test missed the bar, the idea isn't strong enough to carry on its own and needs more generative work before another evaluative round.

This technique is similar to single-blind studies in scientific research, where the experimenter knows the condition but the subject does not, removing perceptual bias. It is also related to blind taste tests, which strip brand from a more mature product category to compare offerings on the product itself.

Because off-brand testing is a bias-reduction technique, there are no major *additional* biases it introduces. It is still possible to go overboard with selling instead of validating: heavy scarcity tactics will increase “sales” but will not tell you about underlying demand, and offering too many options will depress response. Keep the test page clean.

⚠ Biases & Tips

False positive
from brand
halo

If any thread of your real brand leaks into the test (a familiar visual cue, a domain that resolves back to your company, a shared tracking pixel), you reintroduce the very bias you set out to remove. Audit the off-brand property and traffic sources for leaks before launch.

**Identity
perfectionism**

Teams over-invest in making the off-brand identity polished, burning time on a deliverable that has no effect on conversion signal. The identity needs to be defensible, not beautiful.

Ethics

If the call to action collects real money (a pre-order or deposit) under a fictitious identity, a real person believes they are buying from a company that does not exist. Fulfil or refund every commitment promptly, disclose what is appropriate, and never keep money for a product you have no intention of shipping.

Next Steps

- Compare off-brand test results with branded equivalents to isolate the brand effect, where you have both.
- If the concept succeeds off-brand, decide whether to launch it as a sub-brand or integrate it into the main brand.
- Use the off-brand test to explore market segments your main brand cannot reach.
- Archive the test brand materials — you may need them for future category experiments.
- Use a [Landing Page Test](#) under the off-brand identity to measure raw conversion without your existing brand's influence.
- Run [A/B Testing](#) to compare the off-brand version against your branded version and quantify the exact brand effect on conversion.

Case Studies

Dell: Off-brand variants inside an A/B program

Siroker and Koomen's A/B Testing describes Dell using off-brand variants in marketing experiments to read demand for ideas the main brand was not yet ready to carry, treating off-brand identity as one more axis to vary inside a standard A/B program.

[Read more](#)

Target: Smartly owned brand

Target launched Smartly in October 2018 with 70+ everyday essentials priced from \$0.59 to \$11.99, most under \$2 and roughly 70% less than comparable national brands. The separate brand identity let Target reach a price-first segment its main brand could not credibly occupy.

[Read more](#)

Further reading

-  Nirmalya Kumar — “Kill a Brand, Keep a Customer,” Harvard Business Review, December 2003
-  Al Ries & Jack Trout — Positioning: The Battle for Your Mind, 20th Anniversary Edition, McGraw-Hill, 2001 (ISBN 978-0-07-135916-0)
-  Eric Ries — The Lean Startup, Crown Business, 2011 (ISBN 978-0-307-88789-4) — frames the smoke-test approach off-brand testing extends.
-  Dan Siroker & Pete Koomen — A/B Testing: The Most Powerful Way to Turn Clicks Into Customers, Wiley, 2013 (ISBN 978-1-118-53609-4)
-  Target Corporate — “Smartly: Target’s new owned brand of essentials” (2018-10-14)

7.3 Scorecards

In Brief

A scorecard is a weighted ranking tool, usually a spreadsheet, that lists your options as rows and your decision criteria as columns, scores each option against each criterion, then weights and sums the scores to surface one clear top priority. The output is a ranked list that replaces circular debate with a transparent, repeatable comparison, so the team can commit to one option and move.

COMMON USE CASE

Your team has brainstormed eight possible features to build next and everyone has a different favorite. You need a quick, structured way to compare them against three or four criteria so you can pick one and move forward instead of debating endlessly.

HELPS ANSWER

- Which option should we prioritize among several choices?
- What criteria actually matter for this decision?
- How do our options compare when we score them the same way?

Description

A scorecard lists your options as rows, the criteria that matter as columns, weights each criterion, scores every option against each criterion, and sums (or multiplies) the values into one number per option that ranks the whole list.

Unlike the research and experiment methods in this book, a scorecard generates no new evidence about your market or product. It is a decision-making tool, not a

	0.30	0.25	0.20	0.25	
1	6	7	5	6	5.95
2	7	6	6	5	6.10
3	5	6	4	7	5.45
4	8	8	7	8	7.80
5	6	5	6	6	5.95
6	4	5	5	6	4.90
7	5	4	5	5	4.75
8	3	4	4	4	3.70

AT A GLANCE

Qualitative

Quantitative

In-person

Remote

Attitudinal



TIME ⌚ ~2–4 hours

COST 💰 \$0

OTHER NAMES

Weighted Scorecard ·

Prioritization Scorecard

research method: it organizes judgment you already hold and makes visible the option that judgment favors. Its inputs are usually the *generative output of other methods* — the segments surfaced by [Discovery Interviews](#), the features ranked in a [prioritization game](#), the risks named in an assumptions map. The scorecard's job is to *choose among* what those methods surfaced, not to discover anything new. Treat its output as structured judgment made visible, never as data.

The named variants are the same shape with different columns: RICE (Reach, Impact, Confidence, Effort) for roadmap prioritization; ICE (Impact, Confidence, Ease) for ranking growth experiments; and WSJF (Cost of Delay ÷ Job Size) for sequencing enterprise backlogs. The same template absorbs at least four common decisions: ranking segments (size, willingness to pay, accessibility), features or backlog items (impact, effort, confidence), risks (impact and likelihood), and channels (reach, fit, cost-to-test). That cross-cutting reach is why a scorecard is a general decision tool rather than a method tied to one stage of validation.

Because the inputs are subjective estimates, the output looks quantitative but is not evidence. Its primary value is preventing analysis paralysis: it produces one clear top priority so the team can act, then learn, rather than debate further. If you act quickly you can change your mind later; if you never act, you generate no result to learn from.

How to

Prep

★ AI Prompt

Most of the work happens before any cell gets a number. What's on the list, what counts, how much each thing counts, what scale you score on, and who scores — these decisions determine whether the result is useful or theatre.

1. List the options as rows. Keep the set tight — five to ten candidates is the sweet spot. If you have thirty, make a fast cut first (a 2×2 impact/effort pass, or a “least important first” sweep) before reaching for a scorecard.
2. Choose three to five criteria as columns. Borrow from the published frameworks as priors — RICE uses Reach, Impact, Confidence, and Effort; ICE uses Impact, Confidence, and Ease; WSJF weighs Cost of Delay against Job Size — or name your own concrete, scoreable criteria. AI can propose a candidate criteria set from a description of your decision; treat it as a first draft to edit, not the answer.
3. Decide the scoring rubric and use it for every option: 1–5 (forces fewer ties), 1–10 (more headroom, more noise), or a fixed scale with named anchors (massive

/ high / medium / low). Express confidence as a percentage to keep the team honest about uncertainty.

4. Weight the criteria. Either count every column equally or assign integer multipliers (Impact $\times 3$, Effort $\times 2$). The pitfall is changing weights *after* you see the scores — that reverse-engineers the answer you wanted, so lock the weights now.
5. Pick the scorers. Score with at least two people if the decision matters; one scorer alone amplifies the halo effect. Decide upfront whether you'll average, take the median, or discuss outliers — disagreement on a row is signal, not noise.

Execution

With the rubric, weights, and scorers set, the scoring itself is fast.

1. Score every option against every criterion using the rubric. Move quickly — your instinct is most of what the scorecard captures. If you stall on a cell, mark it and come back; don't let one cell anchor the rest of the row.
2. If you're scoring as a team, score independently first, then compare. Rows where scorers agree stand. Rows where two scorers diverge by three or more points get a 60-second conversation before you reconcile them — that gap is where the real judgment lives.
3. Apply your formula (sum, or weighted multiply) to produce one total per option.
4. Sort the totals from highest to lowest. The top row is your candidate priority.
5. If the top option feels obviously wrong, don't override it yet — take it to Analysis and stress-test the weights first.

TOOLS

[Google Sheets](#)

[Notion](#)

[Airtable](#)

[Aircfocus](#)

[Productboard](#)

Analysis

★ AI Prompt

A scorecard earns its keep when the result is counterintuitive. If the top row is what you'd have picked anyway, it has surfaced and confirmed your judgment — act on it. If the top row surprises you, that's the most useful possible outcome: either the scorecard is wrong (test that now) or your intuition was. Run two stress-tests before you commit:

- **Sensitivity to weights.** Halve the weight on your highest-weighted criterion. If the top row still wins, the ranking is robust. If a different row jumps to the top, the scorecard is really a one-criterion decision dressed up as a multi-criterion one —

name that, and decide whether you trust that single criterion. AI can run this sweep across every criterion in seconds.

- **What's missing.** List what the rubric does *not* measure — strategic fit, optionality, irreversibility, second-order effects. These get dropped because they're hard to score, not because they don't matter. If the top row is weak on something the rubric ignores, that's a flag, not a tiebreaker.

Biases & Tips

False precision

A scorecard looks quantitative, but the numbers are subjective estimates. Don't mistake a total for a data-backed result; it is judgment made legible, not evidence.

Anchoring

The first option you score sets a reference point that distorts the rest. Score every option against the same rubric, then revisit your top and bottom rows to sanity-check.

Halo effect

A favored option tends to score well on every criterion regardless of fit. Score each criterion independently, ideally with a second scorer, before you compute any totals.

Delegating judgment to AI

AI can brainstorm criteria, suggest weights, and run sensitivity sweeps, but the value judgments about what matters most must stay yours. Hand the weights to an AI and you have outsourced the decision while dressing it as objective. Use AI to challenge your assumptions; own the weights and the go/no-go call.

Inside-the-building bias

A scorecard ranks options using the knowledge already in the room, which lets you decide without ever testing reality. Pair the ranking with at least one external data point or validating conversation before committing real resources.

Diluted-priority bias

There is only one top priority at a time. Treating the top two or three rows as equally urgent guarantees none of them gets the focus a single priority would; pick the top row and sequence the rest behind it.

Next Steps

- Treat the top-ranked option as a hypothesis, not a verdict — run the experiment that tests it before you commit fully (a [Landing Page Test](#) for a feature's demand, a [Concierge Test](#) for a service).

- If the scores leaned on guesses, replace them with evidence: run a [Closed-Ended Survey](#) to put real numbers behind the criteria you estimated.
- If you're unsure the criteria reflect what customers value, validate them with [Customer Discovery Interviews](#) before you re-score.

Case Studies

Intercom — RICE to end roadmap debates

Facing loudest-voice-wins roadmap arguments, Intercom's product team scored every candidate feature on Reach, Impact, Confidence, and Effort so ideas were compared on one consistent formula instead of by intuition; they published the RICE scorecard in 2018.

[Read more](#)

Sean Ellis — ICE to sequence growth experiments

To decide which growth bets to run first at high tempo, Sean Ellis scored each idea 0–10 on Impact, Confidence, and Ease and ran them in ranked order; the payoff comes from applying the scorecard consistently, wave after wave.

[Read more](#)

Scaled Agile (WSJF) — sequencing enterprise backlogs

Teams using SAFe decide build order by scoring each job's Cost of Delay against its size (WSJF), so the highest-value, lowest-effort work ships first rather than whatever was loudest in planning.

[Read more](#)

Further reading

-  [Tristan Kromer — Completing a 2x2 Risk Prioritization Matrix](#)
-  [Dan Toma — Five Steps to Lean Startup Prioritization That Actually Work](#)
-  [Don Reinertsen — The Principles of Product Development Flow](#)

Appendix



CONTENTS

■ Tags and Other Frameworks

■ Version History

Content Principles

Tone of Voice

Editorial Style Guide

Tags and Other Frameworks

Other Frameworks

There are many great methods, books, and frameworks out there on how to identify and prioritize risky assumptions, hypotheses, and questions.

All the methods in this book are tagged so they are easily searchable depending on any other frameworks we might use. This includes simple tags such as *qualitative* or *quantitative*, which are used to denote the type of information the method produces.

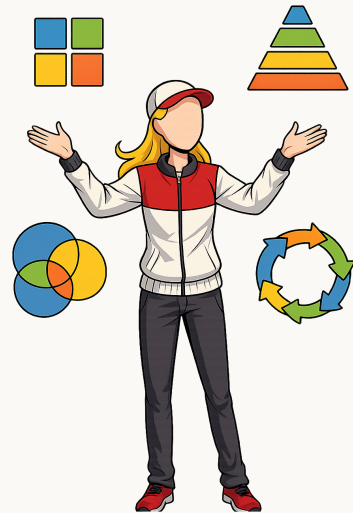
It also includes tags related to the type of business model, such as:

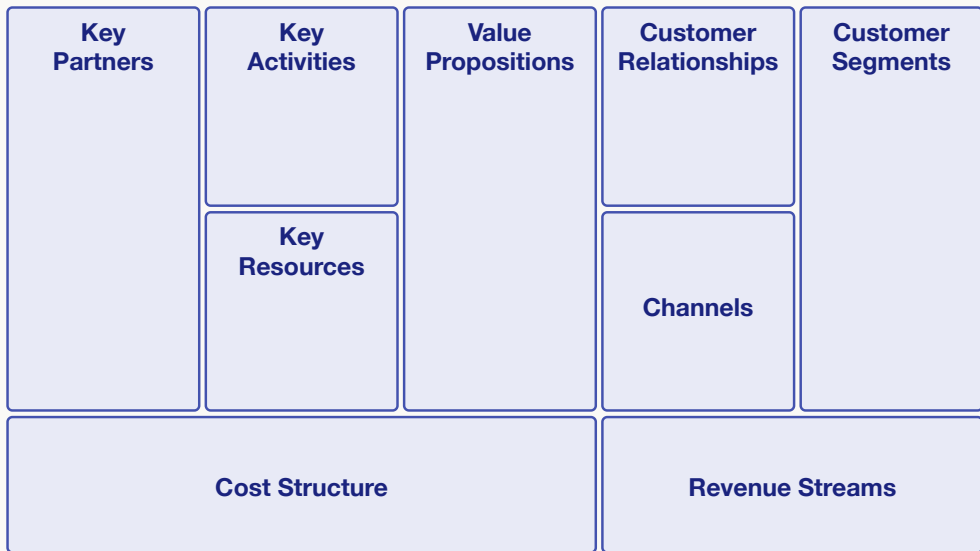
- *B2B* for business-to-business
- *B2C* for business-to-consumer
- *B2G* for business-to-government
- *Two-Sided Market* for a business with buyers and sellers.

These are included for convenience and additional context. Only using those tags to navigate may result in a large selection of methods not entirely suited to the learning goal.

Using the Business Model Canvas

The [Business Model Canvas](#) is a popular framework that identifies the nine basic building blocks of any business model and asks us to make assumptions as to what our business will be. Those blocks are:





Based on our completed canvas, we choose the area of greatest risk to our success. Sometimes this is the customer segment, but in the case of an existing market it may be the value proposition, channels, or even key partners.

Each method in this book is tagged with these blocks. If we can identify the greatest risk to our business model via the Business Model Canvas, we can search the tags for a complete list of experimental methods relating to that building block.

For example, if the customer is the biggest risk to our customer segment, then we are asking “Who is our customer?” or “Is this our correct customer segment?” Based on that, there are several tools available to learn more about our customer, including:

- Discovery interviews
- Contextual Inquiry
- Data mining
- Surveys (closed-ended)
- Focus groups

This search won’t differentiate between generative research and evaluative experiments, so you’ll still need to take that extra step.

Tags

These tags are auto-compiled from all method and parent pages.

Business model tags

- [B2B](#) (3)
- [B2C](#) (4)
- [B2G](#) (1)
- [C2C](#) (1)
- [DTC](#) (3)
- [Two-Sided Market](#) (1)

Research type

- [Qualitative](#) (44)
- [Quantitative](#) (39)

All tags

- [Attitudinal](#) (24)
- [B2B](#) (3)
- [B2C](#) (4)
- [B2G](#) (1)
- [Behavioral](#) (25)
- [C2C](#) (1)
- [DTC](#) (3)
- [In-person](#) (40)
- [Observational](#) (21)
- [Qualitative](#) (44)
- [Quantitative](#) (39)
- [Remote](#) (62)
- [Transactional](#) (10)
- [Two-Sided Market](#) (1)

Version History

- **Version 2.0 (2026-06-20)** — The AI-era second edition. Rewritten for building in the age of AI: an AI prompt on every method, new AI-forward methods (*Synthetic Persona Screening*, *AI-Moderated Interview at Scale*, *Vibe-Coded Disposable MVP*), expanded method detail, and the Index-forward structure. Free online edition published June 20, 2026; Kindle and print editions published July 6, 2026.
- **Version 1.8** - Added *A/B Testing* AI-managed multi-variant section with local-optima and wrong-target pitfalls; added tooling cross-links across *Customer Discovery Interviews*, *Concierge*, and *Landing Page*
- **Version 1.7** - Rewrote 18 method pages (incl. *Comprehension Test*, *Net Promoter Score Survey*, *Paper Prototyping*, *Usability Testing*, *Vibe-Coded Disposable MVP*); Stage 2 remediations across 14 additional method pages; substantial AI-era rewrites of *Secondary Market Research* and *Competitive Analysis*
- **Version 1.6** - Added 25 new methodology pages with new parent groupings *Prioritization Games* (*Buy a Feature*, *Card Sorting Features*, *Product Box*, *Speed Boat*) and *Product Prototyping* (*3D Print*, *Clickable Prototype*, *Life-Sized Prototype*, *Mash-up*, *Single Feature MVP*); added *Card Sorting Pain Points*, *Customer Support Analysis*, *Search Trend Analysis*, *Social Media Campaign*, *Storyboard the Pain*, *Web Traffic Analysis*, *Storyboard the Solution*, and seven new VPTs (*Boomerang*, *Crowdfunding*, *Extended Landing Page*, *Mock Sale*, *Online Ad*, *Pop-Up Store*, *Referral Program*)
- **Version 1.5** - Generalized *Common Use Case* sections across 10 method pages; revised 26 method pages for actionability across sections 3-7; updated appendix sections for clarity and inclusivity
- **Version 1.4** - Added researched case studies and tools across all 39 method pages, with fact-check corrections applied to 84 case studies in 31 pages
- **Version 1.3** - Added AI prompts and next steps to all 39 method pages
- **Version 1.2** - Rewrote *In Brief* paragraphs for all 39 method pages
- **Version 1.1** - Refreshed all book illustrations (43 chapter + 40 method-page) to Kromatic style with alt text
- **Version 1.0** - Published as Kindle Edition on Amazon (originally tagged v0.5)
- **Version 0.5** - Minor updates to visual styling, general chapter cleanups, added direct download functionality with www.realstartupbook.com
- **Version 0.4** - Moved project to Gitbook, general chapter cleanups
- **Version 0.3** - Updated book's formatting, added *Customer Discovery Interviews* and merged with *Customer Discovery*, added *Secondary Market Research*, added *Concierge Test*, added *Net Promoter Score*, added *Appendices* and *Biases*

- **Version 0.2** - Updated *Generative vs Evaluative* (1.3), added *Generative Market Research* sections 1-3, added *Evaluative Market Experiments* sections 1-4, added *Generative Product Research* sections 1-3, added *Evaluative Product Experiments* sections 1-2, added *Out of the Box* sections 1-2
- **Version 0.1** - Added *Preface* sections 1-3, added *The Index* sections 1-6

Contents

Dedication	iii
Contributors	iv
Foreword to the Second Edition	v
Preface to the Second Edition	viii

Part 1: The Index

1.1	What Are You Trying to Learn?	2
1.2	Index of Questions	5
1.3	Index of Methods	6
1.4	Reading the Methods	7

Part 2: Foundations

2.1	Asking the Right Question	9
2.2	From Question to Method	13
2.3	Market vs. Product	15
2.4	Generative vs Evaluative Research: When to Use Each	17
2.5	Hypothesis Checklist	18
2.6	AI Pitfalls	23

Part 3: Generative Market Research

3.1	AI-Moderated Interview at Scale	27
3.2	Pain Point Sorting	33
3.3	Contextual Inquiry	39
3.4	Discovery Interviews	46
3.5	Data Mining	52
3.5.1	Customer Support Analysis	54
3.5.2	Search Trend Analysis	60
3.5.3	Web Traffic Analysis	67

3.6	Experience Sampling	73
3.7	Focus Groups	79
3.8	Open-Ended Survey	85
3.9	Social Media Campaign	92
3.10	Storyboard the Pain	98
3.11	Synthetic Persona Screening	104

Part 4: Evaluative Market Experiments

4.1	Closed-Ended Survey	110
4.2	Comprehension Test	120
4.3	Secondary Market Research	127
4.4	Value Proposition Test	137
4.4.1	Relabel Test	141
4.4.2	Broken Promise Test	146
4.4.3	Crowdfunding Test	151
4.4.4	Event Test	158
4.4.5	Fake Door Test	164
4.4.6	Flyer Test	171
4.4.7	High Bar Test	177
4.4.8	Landing Page Test	182
4.4.9	Landing Page (Extended) Test	190
4.4.10	Mock Sale Test	196
4.4.11	Online Ad Test	202
4.4.12	Pop-Up Store Test	208
4.4.13	Pre-Sales Test	214
4.4.14	Referral Program Test	221
4.4.15	Sales Pitch Test	229
4.4.16	Video Test	237

Part 5: Generative Product Research

5.1	Analog/Digital	245
5.2	Ask an Expert	251
5.3	Competitive Analysis	256
5.4	Competitor Usability Testing	264
5.5	Concierge Test	269
5.6	Demo Pitch	275
5.7	Pocket Test	280

5.8	Dogfooding	285
5.9	Picnic in the Graveyard	290
5.10	Prioritization	295
5.10.1	Buy a Feature	298
5.10.2	Card Sorting (Features)	305
5.10.3	Product Box	310
5.10.4	Speed Boat	316
5.11	Solution Interview	322
5.12	Storyboard the Solution	328

Part 6: Evaluative Product Experiments

6.1	Dashboards	336
6.2	Net Promoter Score (NPS) Survey	342
6.3	Prototyping	349
6.3.1	Paper Prototype	353
6.3.2	3D Print	358
6.3.3	Clickable Prototype	364
6.3.4	Life-Sized Prototype	369
6.3.5	Mash-up	375
6.3.6	Single Feature MVP	381
6.3.7	Vibe-Coded Disposable MVP	387
6.4	Product-Market Fit Survey	392
6.5	Usability Testing	398
6.6	Wizard of Oz	404

Part 7: Out of the Box

7.1	A/B Testing	413
7.2	Off-Brand Testing	422
7.3	Scorecards	428
	Appendix	433
	Tags and Other Frameworks	434
	Version History	437