

The Iterative-Hypothesis customer development method

by Jason Cohen on September 17, 2022

A simple but effective system, used to vet what is now a Unicorn, for generating insights about how your potential customers think, what they need, and what they'll buy.



"I also managed dozens of employees and oversaw international travel arrangements. Plundering was actually a very small part of the job."

This is a simple but effective process for building knowledge through interviewing (potential) customers that I've employed multiple times in the past 15 years.

This system led to rejecting some startup ideas, selecting the idea of WP Engine, and selecting the right features during the early years, which then led to hyper-growth, which led to a Unicorn company with three dozen teams who do customer development for themselves.

"Deep in the forest there's an unexpected clearing that can be reached only by someone who has lost his way."

—Tomas Tranströmer

The goal is to uncover the truth, not to sell

I've been the interviewee for many startups doing customer development, and their most common mistake is that they spend most of the time selling me on how great their idea is.

If you—a reasonably intelligent, excited, passionate person—sit down with someone who meets the criteria of a potential customer, and make an hour-long sales pitch explaining all the features and benefits, with that person wanting to be kind and supportive of this interviewer who is as passionate as they are desperate, that person will probably say something like, "yeah, that sounds pretty good."

So, what have you accomplished? Nothing. **If you don't come away knowing something new and actionable at the end of the interview, you've wasted your time and theirs.**



In the Lean Startup method they call this “validating your ideas,” so it’s tempting to spend the time convincing the person to provide you with validation. Instead, your mindset should be: “What does this person know, that invalidates something I thought was true?”

If you’ve set out to confirm your ideas, not to disconfirm, then you will easily see the confirmation and conveniently miss the disconfirmation, and you will have done worse than waste your time—you have convinced yourself to believe incorrect assumptions.

This process is a specific way to achieve that outcome: Maximizing genuine learning.

“*Listening is being able to be changed by the other person.*”

—Alan Alda

The Process

1. Goals: What you’re trying to learn

What is it that you’re trying to learn? If it’s a new product at a new company, in a B2B space, where the fundamental value-proposition is to solve some specific existing customer pain-point, the list should include everything standing between the customer’s problem and your potential product; a list might be something like:

1. What does the “perfect customer” (PC) look like?
2. What outcomes does PC need to deliver in a typical month?
(*e.g. JTBD framework*)
3. What does PC actually do in a typical day?
(*e.g. tools, workflows, things they love, things they dread*)
4. What pain points does PC experience today?
(*i.e. what actually happens, and what pain does the customer actually know about?*)

5. How does PC cope with that pain today?
(*i.e. what is your competition, including DIY?*)
6. How much would PC pay to eliminate that pain?
How is PC able to budget and execute the payment?
(*i.e. what are viable prices and terms?*)
7. What is the triggering moment? What causes PC to decide: Today’s the day I’m going to buy something?
(*because no one randomly switches vendors*)
8. What causes PC to resist or fear buying?
(*habits of the present, anxiety of change, risk or cost of implementation*)
9. Where does PC go to discover and buy products like this?
(*i.e. what are the best distribution channels?*)
10. What specific words does PC use to talk about the space; what tacit assumptions does PC have?
(*i.e. how should you talk about the product?*)
11. What ultimate, higher-level goal does PC have?
(*i.e. what outcomes are they expecting as a byproduct?*)

Decide on your list of goals first, as they will drive the content of your interviews.

Notice what is *not* here: Asking the customer what to build, or whether would buy some feature.

This has happened to me dozens of times in my 24-year career: Asked a customer “Would you buy if we build _____?” and then say “Yes” and then we build it and then they don’t buy. Every seasoned Product Manager will regale you with the same story. Here’s where we trot out the Henry Ford quote: “If I asked my customers what they wanted, they would have said ‘a faster horse.’” Except sometimes “a faster horse” is a wonderful product and a successful startup. And sometimes you should invent a car. In neither case can you find out by talking to customers.

Instead, what you *can* learn from talking to customers, is what their current life is like, which answers the many import questions listed above, leading someday to something with Product/Market Fit.

2. Hypotheses: What you currently think the answers are

It sounds funny to write down the answers to these questions ahead of time; after all, the whole point of interviewing is to empirically discover the answers, not presume you already have them! “Learning” and all that.

The first reason to do this comes from the literature on the science of predictions. Specifically, people are much better and more objective at seeking the truth when they’re forced to record their predictions—for example as formal “bets”—and observe how reality confirms or clashes with those bets. We all autonomically retcon our beliefs in the presence of new information, or just discard the new information (i.e. Confirmation Bias). Writing our predictions down helps us avoid these fallacies.

The second reason to write down hypotheses is that they will lead us to great questions for the interview.

Create hypotheses for each of your goals. At least one per goal, but more is fine. It’s also fine to have hypotheses that aren’t attached to a goal, but you’re really curious anyway.

Here’s an example list of hypotheses I had about WordPress hosting in 2009 before I started WP Engine; I’ve added a “mapping” to the goal-list above:

1. [G1,G4] Bloggers with more than 100,000 page-views per month have trouble keeping their blog fast.
2. [G1,G4] Bloggers with more than 10,000 RSS subscribers have traffic bursts that take down their site, even if the functions just fine under normal conditions.
3. [G4] All WordPress bloggers worry about getting hacked, because it’s common knowledge that blogs get hacked constantly.
4. [G3] Serious bloggers spend at least 3 hours per day inside WordPress—whether writing or answering comments.
5. [G3] Serious bloggers spend at least 2 hours per week on IT tasks related to hosting.

6. [G5,G6] Some bloggers spend thousands of dollars on consultants to make blogs fast and scalable.
7. [G5] Some bloggers just live with the problems.
8. [G1,G6] A blogger with 50,000 page-views per month will pay \$50/mo if these named problems go away.
9. [G6] Bloggers use personal credit cards to buy supporting software for their blog.
10. [G6] Bloggers need to try software before they’re comfortable buying.
11. [G7] When bloggers get hacked, it’s a traumatic moment in which they say “I never want that to happen again,” and they’re ready to switch hosting providers.
12. [G8] Scary, unclear, expensive to move all your data; what if you get there and it doesn’t work?
13. [G9] Bloggers read blogs-about-blogging for tips.
14. [G9] Bloggers trust the advice of WordPress consultants.
15. [G2] Bloggers care about driving RSS subscribers more than anything else, because those are repeat viewers.
16. [G2] Serious bloggers publish at least four times per week, to stoke page views for advertisements, reader interest, and Google search results.
17. [G10] Bloggers call themselves “bloggers,” not writers, authors, content-marketers, etc..
18. [G10] Bloggers call their website a “blog,” not a “website.”

Even if you don’t know anything about “blogging in 2009,” I’ll bet this set of theories sounds reasonable to you. So reasonable, that maybe you’d presume it’s not worth spending lots of time validating them.

But you’d be wrong. (And so was I. We’re all wrong, at the beginning.) After dozens of hours of interviews, I found that half of these hypotheses were wrong. The ones that were correct still needed to be tuned in detail. And equally valuably, I discovered additional attitudes and behaviors that weren’t in my original list.

“The greatest enemy of knowledge is not ignorance. It’s the illusion of knowledge.”

—Stephen Hawking

For example: It’s not true that most people are willing to pay extra for extra security. It turns out that selling security to bloggers is like selling backup software: If you’ve never had a hard drive failure, it’s unlikely you’ll pay \$30/mo for a backup service. Once you experience that devastating event, the first thing you do with your new laptop is sign up for any service that promises 100% full automatic backup. (Selling security to mid-sized companies is different; by then, they’re proactively executing a formal policy.)

Another example: While it is true that bloggers want to “try before you buy” for most software (whether “trial” or “freemium”), this was not the case with hosting their website. It’s such a disruption and technical ordeal to move their website to a new vendor, they think of it as permanent, not temporary. Therefore, “free trial” is not the most compelling offer, whereas “free migrations” is. By the way, we tried “free trial” anyway—how can people not love a free trial!—but the conversion rate was over 90%, and when we removed the free trial, signups didn’t decrease at all.

It’s important to list even the most obvious, mundane assumptions, because you’ll be surprised how often you’re wrong or they need adjustment, even if you’re an expert in the field.

Put these hypotheses in the first column of a spreadsheet, one per row.

3. Questions: What you ask during the interview

Generating good questions is the hard part for most people. Armed with your hypotheses, however, it becomes easy, if you adhere to the following system.

“Questions are places in your mind where answers fit. If you haven’t asked the question, the answer has nowhere to go. It hits your mind and bounces right off. You have to want to know.”

—Clayton Christensen

Questions are designed to test your hypotheses, and to suggest better ones. To achieve this, go to the second column in your spreadsheet, and write one question per hypothesis. Sometimes one question can cover a few hypotheses, if they’re closely related.

Questions must be open-ended. This is where most people go wrong. They’ll have a hypothesis like the “security” example above, so they’ll ask a question that “leads the witness,” because they’re still in “selling” mode instead of “discovery” mode:

Blogs get hacked all the time, and when they do it’s devastating, right? Would you like it if your hosting company had extra security measures to protect your blog?

Of course everyone will say “yes.” Why not? I’d sound dumb if I didn’t agree. That’s why it’s a useless question. You didn’t find out what the person actually thinks. And therefore you didn’t find out what everyone else will think when they look at your advertisement or arrive on your home page or review your pricing page. The assumption baked into your question could be wrong, and now you’ll never know.

Instead, write open-ended questions that:

1. Confirm or negate the hypothesis (*the point of the exercise*)
2. Do not hint at any one specific answer (*seek unbiased truth*)
3. Invite the generation a specific answer (*uncover the correct answer*)

4. Invite more information (*seek answers to questions you didn't know to ask*)

On “security,” for example:

Do you ever think about website security? If so, how do you think about that? Do you do anything about it today?

Here's more examples from the hypotheses above:

Hypothesis	Bad Question	Good Question
Bloggers call themselves “bloggers.”	Do you consider yourself a “blogger?”	When you meet someone new, how do you explain what you do in a few sentences?
Serious bloggers publish at least four times per week.	Do you publish often, so Google ranks you high in SEO and there's a lot of surface area for people to find you?	How often do you publish new content? Why at that rate—what led you to that decision?
Some bloggers spend thousands of dollars on consultants to make blogs fast and scalable	Would you spend \$2000 on a consultant, if it meant your blog would become much faster and more scalable, so you rank higher on Google search results and get more traffic?	How valuable is the speed of your website? Have you ever spent money to improve it? If so, how much, and what did you do? Did it work? Were you happy with that investment?

4. Iterate the hypotheses

During the interview, take notes in the spreadsheet, in a new column, next to each question (which in turn is next to each hypothesis). The conversation might deviate from the original point; that's OK, maybe straying will lead to learning new things.

“*The most exciting phrase in science isn't “eureka,” but rather, “that's funny.”*”

—Isaac Asimov

If you hear anything surprising, ask follow-up questions. **Surprise means you're learning**, and since “learning” is the whole point of the exercise, you should use “surprise” as a signal that you should dig deeper. You can use an old interviewing trick: Just say: “Tell me more about that.”

After each interview, consider what supported or contradicted your hypotheses. Should you alter some of them? Not necessarily, especially after just a few interviews, but definitely if you're seeing a pattern.

Also create new hypotheses (and associated questions) based on new learnings. The more insight you can get about your potential customers, the better. Don't worry about whether every hypothesis maps cleanly onto one of your goals; just accumulate insight.

“*There are two possible outcomes: if the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery*”

—Enrico Fermi

5. Stop when it's boring

Do you have all the right answers? Who knows. Probably not. But when this method stops producing new information, then you need a new method for making progress. That could be a proof-of-concept, a high-fidelity demo, an MVP (or actually an SLC), or something else.

When the surprises stop, that means learning has stopped, and that means you should stop the process.



"So, that thing you're here to talk about...
Talk about that a little."

A typical mistake is to do three interviews, and then stop because "I'm not learning anything." With so little input, you might not be genuinely seeking to learn. That's like a marketer saying "I tried two variants of my AdWords ad, and none of them are better than my first attempt, so I'm not going to try any more variants."

It's also possible there's nothing to learn because you're fishing in the wrong spot. Maybe there's no patterns because you haven't found a real pain-point that more than a few people have and are willing to pay for. That means you need different ideas.

Here's more detail on how to determine whether the result of customer interviews is telling you "don't pursue this idea."

Maximizing your results

So that's the whole process.

Here are more tips.

Emergent segmentation

You might notice that customers are segmented. Meaning, a certain type of customer tends towards one set of answers, while another has a different set. In the case of security, for example, when you talk to marketing departments at large companies, they do think about security, whereas when you talk to independent bloggers, they almost never do.

In this case, it's useful to make a note of the segments you think exist. First, write hypotheses and questions that you believe are the determining characteristics of the segments. You'll ask these at the top of the call. Then, keep separate spreadsheets of hypotheses, one per segment. If you're lucky, you'll end up with clarity on the types of customers, and what each are like. You might choose to target one, some, or all of these; regardless, understanding the landscape is invaluable.

Discuss price

This one is controversial; many intelligent people insist that you shouldn't discuss price in early customer interviews because it unnecessarily conflates financial considerations with the discovery of customer attitudes, behavior, and pain-points.

But in my opinion the price-tag is an essential component of the interview, because I believe the price is inextricably linked to what the product means to that person, therefore how they think about it and how it affects their life. Price also determines the business model of the company, so it should not be "figured out later."

In early interviews for WP Engine, near the end of the call I would float a price of \$50/mo for a service that made their website faster, more scalable, more secure, and came with genuinely good customer service. The responses were immediate, emotional, and vehement. One group was shocked—*shocked*—that the price tag would be so high; they would *never* pay even *close* to that amount. Another group said they would only buy the service if it were much *more* expensive, because otherwise they know it couldn't possibly fulfill its promises. (Those groups turned out to be emergent segments; see above.)

Had I not discussed price, I would never have learned about the segmentation, or the expectations behind those segments, and thus how to price correctly, and for whom.

I had another startup concept before WP Engine. Insights stemming from pricing discussions was one of the primary ways I was able to invalidate that idea, which in turn created the space for WP Engine, which is now a unicorn.

Pricing questions can be open-ended (i.e. “How much would you expect to pay for ...”), however I think quoting a specific price is an acceptable breach of protocol. Putting a specific price in front of people elicits a strong, visceral response. When someone actually visits your pricing page in future, this is also the experience they will have—reacting to a specific price. It’s smart to test what that experience will be like.

You can also tie pricing questions into your other questions, to test whether the person really values that topic. For example, another way to test the hypothesis that bloggers care about security would be:

Would you pay extra for a security package that really worked, or do you not really worry about being singled out for getting attacked by a hacker?

By asking if they’d pay extra, and by almost suggesting that they shouldn’t bother, you’re testing whether they *really* ascribe value to the concept. This worked in practice—most bloggers initially claimed security was important to them, but admitted they wouldn’t pay extra to have more of it.

Expect contradictions

You’re going to get all sorts of contradictory signals. People are different. Sometimes because they have different goals, different values, different past experiences, different roles, different projects, or for no discernible reason whatsoever. So your data is going to be noisy.

Some of your hypotheses will end up reflecting the variation. You might conclude that some number varies a lot rather than staying in a small range, or that there is no pattern in people’s opinions about some topic. That’s still learning: Knowing what patterns *don’t* exist prevents you from making false assumptions.

Real patterns will stand out from that noise; that’s your fundamental truth, that you can build products and strategies around. There might not be much of it. All the more reason to highlight it.

Ask them to explain, step-by-step, how they will use it.

The pattern: You ask a customer if they have the problem; they do. You ask whether they’d buy your product to solve it; they say yes. Then you build it, and they don’t buy. Somehow, your interrogation didn’t work.

Sales conversions are never 100%, but one technique is to ask them to describe, in painstaking detail, exactly how they will use the product in their daily life. When would they open it up, how does it fit into their workflow, which features do they invoke, how do they move the outputs into other systems?

This works because while they really do mean “yes, I think that *sounds* good,” thinking it through uncovers barriers that in fact blocks the sale; it turns out they need it to integrate better with something, or they actually need a feature you weren’t contemplating, or some other hiccup.

Everyone “knows” what everyone else wants, except they don’t

I don’t think I’ve ever conducted a customer interview where the interviewee didn’t switch into “market guru” mode. This is where the customer stops talking about her own life, her own problems, what features or price tag would be acceptable to her, and starts talking *on behalf of other people*.

“I wouldn’t pay \$50/mo, but a lot of people would.”

“I would pay \$50/mo, but most people would expect this to be free.”

“I care about security, but most people are completely clueless about that.”

“Well I use a free tool to do that, but most people don’t.”

Of course it’s coming from a good place—they want to help you, they want to explain the “state of the market,” they want to leverage their expertise. But they don’t know what everyone else wants.

Neither do you; that's why you're doing these interviews. And when you see all the crazy, different things people think, you realize that it takes tons of interviews to uncover even a modicum of truth. The person you're interviewing hasn't done that, so they don't know the truth.

One of the hallmarks of successful companies is that they found some untapped aspect of the market and owned it. That might be a feature no one had considered, or a technology that before now hadn't been accessible, or a realization that the human touch trumps everything (Zappos) or that the human touch doesn't matter at all (Geico). Even if the person you're interviewing is a market expert, you're specifically looking for interesting holes and niches that the experts haven't noticed!

So be polite, calmly tell them that's great insight, reinterpret "everyone else wants X" to either mean "I want X" or just throw out the comment completely, and redirect the conversation back to themselves and their own specific situation.

Leveraging AI... maybe

AI might help. Perhaps it could...

- Generate hypothesis from the goals, adhering to the guidelines, specifying which goal attaches to which hypothesis (*although of course you'll treat them as templates and correct them to what you actually believe*).
- Generate questions for the hypothesis, specifying which question attaches to which hypothesis, adhering to the guidelines above.
- Clean up transcripts.
- Scan conversations for common themes.
- Scan conversations for "things that contradict a specific set of hypothesis."
- Scan multiple conversations for themes.

However, **half the value of this exercise thinking through this stuff for yourself.** The "aha" moments come only when you wrestle with the details.

You find the contradictions. You discover maybe you didn't think that after all. You realize new ideas that solve for the conflicting inputs. You realize which hypothesis are right, wrong, different.

So, I advise you *not* to use AI, except to accelerate busy-work such as:

- Clean up transcripts so they occupy less space and are easier to process.
- Double-check your thinking *after* you do the thinking, maybe come up with new ideas.
- Summarize your thinking, i.e. take your thinking that you wrote out or said aloud, and make the result pithy and clear.

STFU

If you're talking, you're not learning. To maximize learning, minimize talking.

When you do talk, it should be because you're clarifying your understanding about what they just said, digging deeper on the current conversation, or opening up a new vein of conversation.

Further reading

- [12 tips for customer development interviews](#) (by [Giff Constable](#), 2012, just as relevant today as then, wholly compatible with this process.)
- [11 Customer Development Anti-Patterns](#) (also by [Giff Constable](#), 2013; sometimes it's easier to list what *not* to do.)
- [Customer Interviews: Get Actionable Insights from Every Interview](#) (Comprehensive advice from [Teresa Torres](#), also the author of the fantastic book [Continuous Discovery Habits](#) that also explains what to do with the information you get from interviews.)



"Thanks for coming in. You've given me a lot to forget about after you leave."

Printed from: *A Smart Bear*

<https://longform.asmartbear.com/customer-development/>

© 2007-2024 Jason Cohen  @asmartbear