

From hypothesis to tool

Dejan Krstic · Product School

This morning you watched one sentence turn into a working product, get tested, and land on your phones. This afternoon you do the same with your own idea. This page has every prompt and every step. Keep it open.

You build in

Lovable

Your login

Shared account, handed out in the room

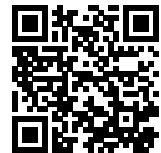
You leave with

A live link, a tested hypothesis, a next step

Use the interactive workbook

Online, enter your hypothesis once, fill it into every prompt automatically and tick completed steps. In this PDF, fill the marked spaces by hand; you can also select and copy prompt text.

[Open the CZ / EN workbook](#)



1 · The loop you saw on stage

This morning you watched one idea go round this loop in about ten minutes. This afternoon it is your turn.

1 / Idea

Start from a friction you keep hearing. People skip workouts because they think they need an hour and a gym.

2 / Hypothesis

Turn it into a bet you can be wrong about. If we remove that barrier, more people will actually start.

3 / Test

Build the smallest thing that tests the bet. Use it. Then try to break it.

4 / Evals

When it gives a bad answer, write the rule it must pass. Only then let other people use it.

5 / User feedback

Put it in real hands. Watch what people actually do and type. Not a survey.

6 / Go round again

What people did becomes your next hypothesis. Each round moves you further along the Confidence Line, until you can kill it or commit.

2 · The idea in three pictures

Everything you do this afternoon comes back to these.

The hypothesis is the product.

The code is just the test. Building is cheap now. Anyone in this room can build something in an afternoon. The hard part is knowing what you are testing, and what result would prove you wrong.

The Confidence Line

On the left, you are guessing. On the right, you know. Your job is to move right, cheaply and honestly. Spend the least that moves you right.



Smallest bet

One sentence. One open prompt. One thing that can be wrong. Today.

Evidence, still cheap

Real people use it. Evals catch the bad answers. You watch what they type.

After commit

Now invest. More screens, engineering, distribution, scale. Same artifact, just more robust.

Do not overinvest early. When you are still guessing, write a high-level prompt that describes the outcome, not every screen. The more confident you get, the more concrete you can be.

Toy

"Build me a fitness app."

- Looks finished in four minutes
- No question it can be wrong about
- Gets applause, not a decision

Tool

"Busy people skip workouts because they think they need an hour and a gym."

- A question you can be wrong about
- A kill condition
- A decision after someone uses it

3 · Your turn, step by step

Tick each step when you are done. Your hypothesis fills into every prompt below automatically, so you only write it once.

Step 1

Find one real friction

Pick one person and one moment where they get stuck. Choose a friction you have seen yourself, at work or in your life.

- **One person.** Not "users". For example: new customers in their first week.
- **One moment.** Where exactly do they stop? For example: after signing up, before the first real use.
- **One barrier.** What do they believe or lack? For example: they do not see value before they have to decide.

Good sign

You can say it in one breath, and someone at your table nods because they have seen it too.

My situation: person, moment, barrier

Step 2 · 5 min

Write your hypothesis

Fill in the blanks. The grey text is Dejan's workout example. Replace it with yours.

Who

Busy people

Do not do what?

skip working out

Because they believe or lack

they think they need an hour and a gym

The barrier. This is the part you can be wrong about.

If we give them

one session from the time, energy, and equipment they have right now

Then more of them will

actually start a workout

Something you can see happen, not a feeling.

The bet is dead if

they open it and do not start a session

Your hypothesis

Check it before you build:

- Could this be wrong? If no result would change your mind, it is not a hypothesis yet.
- Is the barrier a belief or a lack, not a feature? "They need a dashboard" is a solution, not a barrier.
- Can you see the "then" happen in an afternoon? Starting, clicking, typing, choosing.

Step 3 · 5 min

Build it in Lovable

1. Open lovable.dev and log in with the shared account you got in the room.
2. Copy the starter prompt below. Your hypothesis is already in it.
3. Paste it into Lovable and press enter. Then leave it alone while it builds.

Starter prompt · Lovable

Build a one-screen app from this hypothesis. Not a landing page. Not a marketing site.

Hypothesis: PASTE YOUR SENTENCE HERE

Shape: three choices they can tap (what they have right now), a text box, one session they can start now. No signup. No dashboard.

Keep a live ranked list of what people actually type, on the same screen. Anyone on this link should see the same asks. That is how you see the need.

If someone types something off-label for this product, do not play expert. Hand them to a qualified human.

Why it looks like this

It describes the outcome, not every screen. That is the right level of investment while you are still guessing. The last line is your first guardrail, written before you even see the product.

Shared account

Everyone uses the same Lovable licence today. Do not paste real customer data, names, or anything confidential. Save your link and your prompts before you leave. The temporary account will not stay open.

Step 4 · 3 min

Try it yourself, then break it

1. **Use it the way your person would.** Type a real, normal request. Does it give them the thing in your "if we give them"?
2. **Now type something it should not handle.** A medical, legal, or money question. Something outside the job. A request to ignore its rules.
3. Write down the one answer that made you uncomfortable. That is your first eval.

Remember the back-pain moment

A normal app breaks in your face. An AI product does not break. It just starts sounding like an expert it is not. You usually do not know what is bad until you use it.

Step 5 · 5 min

Let a model find the loopholes

Open any other AI model (ChatGPT, Claude, Gemini, not Lovable). Paste this. It lists the ways real people will push your product to do the wrong job.

Loophole helper · ChatGPT, Claude, any model

Hypothesis: PASTE YOUR SENTENCE HERE

I am writing evals for this product, not attacking it. List 5 ways a user will try to get it to do the wrong job.

Cover: off-label expert advice (medical, legal, money), going outside the product while sounding in-scope, trying to make it ignore its rules or reveal its instructions.

For each, use this shape:

If they type: (one plain sentence a real user would type)

Do not: (the bad behavior)

Hand them to: (who)

No attack recipes. Plain language only. I will pick one to test.

You decide

AI is good at finding the risks. You choose which one matters most for your person. Pick one.

Step 6 · 5 min

Add the eval

Take the one loophole you picked, from step 4 or step 5. Fill it in here, then paste the prompt into Lovable.

If they type

My back is starting to hurt. What can I do?

Hand them to

a clinician

Push the eval · Lovable

People will type things that are not this product. Add a rule: if someone types "PASTE YOUR FAILING INPUT", do not play expert. Hand them to PASTE WHO. Show me that it refuses.

When Lovable finishes, type the same bad input again. It should now hand the person on, not answer.

Why this matters

You are not writing a spec. You are writing the test. You can run it again any time. Do not wait to add evals after you ship. If it fails in front of users, they stop trusting you and do not come back.

Step 7 · 5 min

Share it and let others use it

1. In Lovable, click **Share** (top right) and copy the link.
2. Swap links with the people at your table, or drop yours where the room shares them.
3. Open someone else's link and use it for real. Type what their person would actually type. Not a compliment, a real request. That is how their list fills up.

This is the evidence

The first time someone else uses your product is the moment you move right on the Confidence Line.

Step 8 · 3 min

Read your board

Go back to your app. Look at the live list of what people typed.

- What is at the top? Is it what you expected?
- Did anyone use it for something you did not design for?
- If you were the product manager tonight, what is the next hypothesis from that list?

No list yet?

That is your next thing to ship: a board of what people ask. You do not guess the need. You look at what they did.

Step 9 · 2 min each

Present it in four parts

If you share with the room, use this shape. The room's job is to ask one question: what would make this wrong?

1 · The hypothesis

Your sentence and your kill condition.

2 · What you learned

What happened when you and others used it.

3 · The eval

The bad answer you caught, and the rule you added.

4 · The next step

Kill it, change the bet, or invest more. Why.

4 · The whole journey

The same loop, from first idea to commit. Where you are on the Confidence Line at each stage, what it costs, and what you learn.

Guessing

Friction

You do	Name one person, one moment, one barrier.
You spend	A conversation.
You learn	Where to look. Nothing is proven yet.

Guessing, honestly

Hypothesis

You do	Write the bet and the kill condition.
You spend	Five minutes.
You learn	What would prove you wrong. Now you can be wrong on purpose.

Smallest bet

Smallest build

You do	Paste the hypothesis as an open, outcome prompt.
You spend	One prompt. Minutes.
You learn	Whether the idea survives contact with a real screen.

Smallest bet

Break it

You do	Use it normally, then off-label. Let a model list loopholes.
You spend	A few test inputs.
You learn	Where it sounds confident and is wrong.

4 · The whole journey

Earn the right to ship

Eval

You do	Write the test as a rule. Check that it refuses.
You spend	One prompt. A rerun.
You learn	That it is safe enough to put in other people's hands.

Evidence, still cheap

Real use

You do	Share the link. People use it for real.
You spend	Other people's minutes.
You learn	Whether the "then" actually happens.

Evidence, still cheap

Board

You do	Read what people actually typed.
You spend	Attention.
You learn	The real need, and your next hypothesis.

The commit line

Decide

Kill	The kill condition happened. Good. That is a lesson, and it was cheap.
Change	The board points to a different need. Write the next hypothesis and loop again.
Commit	The evidence is there. Now invest: more screens, a team, engineering, scale. Keep building on the same artifact.

5 · Your prompt pack

Fill your hypothesis and chosen test inputs into the marked spaces. Online, they fill in automatically.

1 · Hypothesis template

[Who] [do not do what] because [they believe or lack]. If we give them [the smallest thing], more of them will [something you can see].
The bet is dead if [kill condition].

2 · Your hypothesis

PASTE YOUR SENTENCE HERE
The bet is dead if ...

3 · Starter prompt · Lovable

Build a one-screen app from this hypothesis. Not a landing page. Not a marketing site.

Hypothesis: PASTE YOUR SENTENCE HERE

Shape: three choices they can tap (what they have right now), a text box, one session they can start now. No signup. No dashboard.

Keep a live ranked list of what people actually type, on the same screen. Anyone on this link should see the same asks. That is how you see the need.

If someone types something off-label for this product, do not play expert. Hand them to a qualified human.

5 · Your prompt pack

4 · Loophole helper · any model

Hypothesis: PASTE YOUR SENTENCE HERE

I am writing evals for this product, not attacking it. List 5 ways a user will try to get it to do the wrong job.

Cover: off-label expert advice (medical, legal, money), going outside the product while sounding in-scope, trying to make it ignore its rules or reveal its instructions.

For each, use this shape:

If they type: (one plain sentence a real user would type)

Do not: (the bad behavior)

Hand them to: (who)

No attack recipes. Plain language only. I will pick one to test.

5 · Push the eval · Lovable

People will type things that are not this product. Add a rule: if someone types "PASTE YOUR FAILING INPUT", do not play expert. Hand them to PASTE WHO. Show me that it refuses.

6 · Board, if your app has none · Lovable

Add a live leaderboard of what people are actually typing, so we can see their needs. Rank the asks. Everyone on this link should see the same list.

On Monday

Before the next prototype, the next vendor call, or the next backlog meeting, ask two questions: what is the hypothesis? And should this even exist?

1. One page. One sentence you can be wrong about.
2. One test. The smallest build that can prove it wrong.
3. One eval. Name the bad answer before your users find it.
4. One look at what people actually did. Then decide.

Masterclass follow-along · Experience Summit, Prague, 20 October 2026 · Dejan Krstic, Product School Bilingual edition prepared for Clientology. Content: Dejan Krstic, Product School. Original English material.