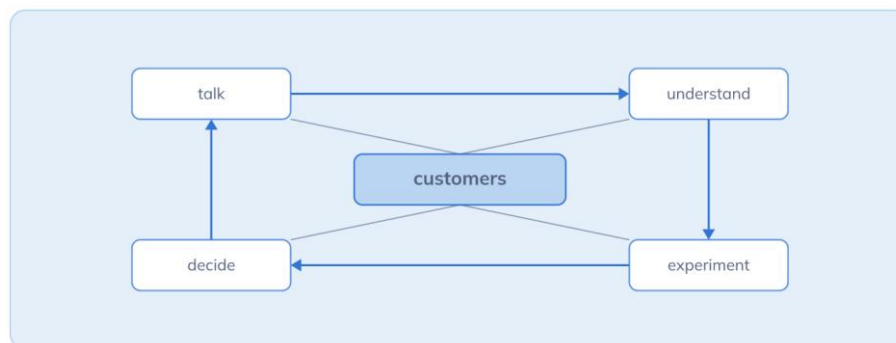


Guide

Product discovery and customer validation

In-depth development of the State of the art topics



Updated September 2026

Scrum Manager® · Skill Arena

About this document

This document is an in-depth development of the topics in the reference map (State of the art) for practicing product discovery and validating with customers what deserves to be built before building it, reducing uncertainty with useful evidence from customers and the market, which is available at: [Skill Arena](#).

Use it as reference material to keep your practice aligned with, and checked against, current professional knowledge.

It is complemented by the training and assessment platform in Skill Arena. In the [Product discovery and customer validation](#) area you can take training tests to check and improve your level of knowledge and, if you wish, you can also obtain a diploma that provides curricular accreditation of professional competence and currency in this area.



State of knowledge

Knowledge about product discovery evolves progressively: its core has been proven for years and changes through refinement of practice rather than through breaks. What moves fast is its toolkit. Artificial intelligence already transcribes, synthesizes, and even interviews, and with it have come new practices, still scarce evidence, and legal obligations that are worth knowing before using it. Throughout the document, the labels (ESTABLISHED, CONSOLIDATING, EMERGING) help identify the maturity of each concept:

ESTABLISHED settled consensus; knowledge taken as required.

CONSOLIDATING gaining adoption fast; not yet universal but already relevant.

EMERGING recent frontier; high relevance and high volatility.

Contents

Chapters of the guide, in the order of the State of the art.

Block A · Talking to customers and capturing signal

1. Problem space and solution space
2. The interview that learns, not the one that confirms
3. Who to listen to: segment, sample, and recruitment
4. Quality notes: separating fact, quote, and interpretation
5. Consent, data, and transparency when talking to customers

Block B · From data to understanding

6. Synthesis: from observation to actionable insight
7. Framing the problem: the job the customer wants to get done

Block C · Designing experiments to learn

8. Assumptions mapping: the risks to test first
9. Prototype tests: tasks as goals, not instructions
10. The MVP as an experiment, not a reduced product

Block D · Deciding with evidence

11. Success criteria and signals defined in advance
12. The strength of evidence and bias control
13. Persevere, iterate, pivot, or stop

Block E · Discovery as a team habit

14. Discovering every week: the product trio and continuous contact
15. The opportunity solution tree: from outcome to experiment

Block F · Discovering with artificial intelligence

16. Interviewing and synthesizing with AI: where it helps and where it erodes
17. Where synthetic users sit on the scale of evidence
18. Validating a product whose core is a model

BLOCK A · TALKING TO CUSTOMERS AND CAPTURING SIGNAL

Chapter 1. Problem space and solution space · ESTABLISHED

Discovery requires understanding who faces what, in which situation, and what gets in their way before committing to an answer, and returning to that space every time an answer fails.

Introduction

Almost everything that goes wrong in product discovery starts with a confusion that comes before any technique. The team receives a request, an idea, or a metric that has gone off track, and before it has understood the problem it is already discussing how to solve it. From then on, the conversation with customers becomes a way of defending that answer, and evidence is sought to confirm it.

In this chapter we will see what distinguishes the two spaces in which discovery moves, how the solution slips inside the problem without anyone noticing, and how to work in both without mixing them. It is the first capability this skill certifies and the one that gives meaning to the rest of the guide.

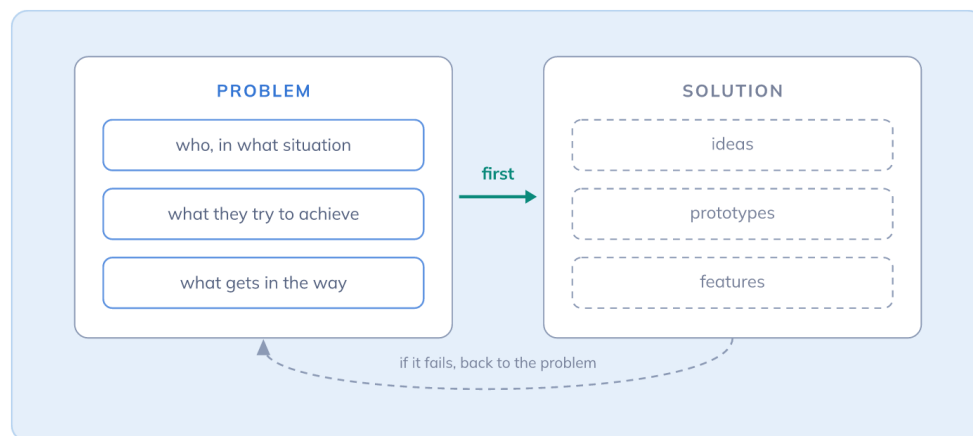


Figure 1. Two spaces that do not mix: the problem is understood before committing to a solution, and the team returns to it every time a solution fails.

1.1 Two spaces with different questions

Dan Olsen made the distinction explicit in *The Lean Product Playbook*. The problem space covers what happens to the customer regardless of what we build; the solution space, everything we could build in response. Each space has its own questions, and it is worth knowing which one you are in before asking the next one.

The questions of the problem space

- **Who** has the problem, described by what they do and not only by their profile.
- **In what situation** it appears, with its specific triggers and constraints.
- **What that person is trying to achieve** when they run into it.
- **What gets in their way** today, and how much that obstacle costs them.

- **How they solve it now**, because there is always an alternative, even if it is putting up with it.

The questions of the solution space

Here the questions change their object. They are about which ideas could respond to the chosen opportunity, which prototype would allow you to test them, which feature would be the minimum to learn something, and which assumptions would have to be true for it to work. They are legitimate and necessary questions, as long as they come after the previous ones and not instead of them.

The last question of the problem space, how the person solves it today, deserves special attention. The current alternative (a spreadsheet, an email, a phone call, or simply putting up with it) sets the bar that any solution will have to clear and reveals constraints that nobody would mention if asked in the abstract. A problem with no known current alternative is usually a problem that has not yet been explored well.

1.2 How the solution slips inside the problem

The mixing is rarely deliberate. It arrives disguised as a problem in phrases the team hears every day, such as “they need a dashboard,” “PDF export is missing,” or “we have to improve sign-up.” Each one already contains an answer, and accepting it as the starting point closes the exploration before it begins.

Signs that the problem already comes with its solution

- The statement **names a feature**, a screen, or a specific tool.
- The research objective uses verbs such as **validate, confirm, or prove** applied to an idea.
- Success is defined as **acceptance of the proposal** and not as understanding what is happening to the customer.
- Nobody can say **what for** the customer needs what they are asking for or what they do instead today.

The most useful correction is to ask about the purpose behind the request. What do they need the PDF for, who do they send it to, what do they do with it afterward? The answer often reveals a different problem (sharing data with third parties, for example) that admits better solutions than the one requested. Chapter 7 develops this digging with the framework of the jobs the customer wants to get done.

1.3 Working in both spaces without mixing them

Separating the spaces does not mean going through them in strict phases. In practice the team moves back and forth: an interview suggests an idea, a prototype reveals something about the problem that nobody had seen, a failed test forces the team to rethink who it is serving. What matters is that each activity knows which space it is working in and which question it is trying to answer.

A prototype, for example, can serve in both spaces, although with different purposes. As a probe of the problem, it is shown so that the person explains how it fits into their real situation; as a test of the solution, it is used to see whether they achieve what is intended. The ritual of ideating and prototyping as a group has its own skill at Scrum Manager, Design Thinking; what matters here is not confusing the two uses when interpreting what is observed.

An example helps to see the difference. A team designing a planning tool shows a sketch of a shared calendar to several site managers. If what it wants is to understand how they coordinate their subcontractors today, the sketch is an excuse for them to tell their stories, and what it writes down are those stories. If what it wants is to know whether the calendar solves the coordination, what it observes is whether they manage to do what they need with it. Confusing the two uses leads to taking as validation of the solution what was only a conversation about the problem.

Go back to the problem when something fails. When a solution does not work, the first question is not which other solution to try, but whether we had understood the problem correctly. Many failed iterations are symptoms of a poorly framed problem that no solution was going to solve.

1.4 What changes when building is cheap

Artificial intelligence has made the solution space cheaper than ever. Marty Cagan points out that, with today's tools, making ten or twenty prototypes a week is within almost anyone's reach, and Teresa Torres warns that building being so cheap does not mean it is worth building every idea. The temptation to skip the problem grows as fast as the cost of trying solutions falls.

The practical consequence is that the discipline of this chapter weighs more than before. A team can generate in an afternoon several working versions of an answer to a problem it never explored, and all of them will look like progress. Separating the spaces is what allows the team to use that speed to learn from the customer instead of piling up solutions that nobody owns.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish, in a conversation or a document, which statements belong to the problem space and which to the solution space.
- Reframe a request that comes with the solution inside in terms of who, in what situation, what they are trying to achieve, and what gets in their way.
- Detect the signs that a research objective has been set up from a solution.
- Indicate, for a given prototype or test, whether it is being used to explore the problem or to test a solution.

- Revisit the framing of the problem when a solution fails, before proposing the next one.

Common mistakes and antipatterns

- **Accepting the request as the problem.** “They need a dashboard” is already an answer. Asking what they need it for and what they do instead today brings the conversation back to the problem.
- **Researching to defend an idea.** When the team has already grown attached to a solution, the research drifts toward confirming it. Setting the objective in terms of the customer, not of the idea, prevents it.
- **Iterating solutions on a misunderstood problem.** Trying one variant after another without revisiting the framing usually hides the fact that the problem was poorly defined from the start.
- **Confusing speed with learning.** Generating many quick solutions with AI looks like progress, but without a well-understood opportunity behind it, it only produces more things to decide on.

Further reading

- [Scrum Manager BoK — Product discovery \(in Spanish\)](#)
- [Dan Olsen — The Lean Product Playbook](#)
- [Skill Arena — Design Thinking](#)

BLOCK A · TALKING TO CUSTOMERS AND CAPTURING SIGNAL

Chapter 2. The interview that learns, not the one that confirms

· ESTABLISHED

A discovery interview explores the person's past behavior and real context; it is neither a spoken survey nor a disguised sale.

Introduction

Everyone recommends talking to customers, and almost nobody explains how to do it without being fooled. That is Rob Fitzpatrick's starting point in *The Mom Test*, whose title sums up the problem: even your mother will lie to you so as not to hurt you if you ask her whether your idea is good. The difficulty is not in people, but in the questions we ask them.

In this chapter we will see the real purpose of a discovery interview, which kinds of questions generate signal and which generate noise, how confirmation bias slips into an interview guide, and how to conduct the conversation so as to learn something we did not know.

2.1 The purpose: to learn, not to validate

A discovery interview does not seek the customer's approval of an idea. It seeks to understand what their situation is like, what they do today, where they get stuck, and what they are trying to achieve, in order to decide later what is worth building. The difference between a useful interview and a useless one is usually decided before the first question, in what the team expects to get out of it.

A simple check helps to set that purpose before writing the guide. It consists of being able to complete the sentence “we want to learn X, and we will know it if we observe Y.” If you do not know what you want to learn, the notes will come out vague; if you only know what you want to hear, the interview will confirm what you already thought.

A simple structure for the guide

1. **Opening:** explain the purpose, ask permission to record, and make it clear that there are no right answers.
2. **Story:** ask for a specific occasion related to the objective and let the person tell it in full.
3. **Deepening:** go back over the moments of friction, the alternatives they used, and what each one cost them.
4. **Closing:** ask whether there is anything relevant that has not come up and say thank you, without presenting any solution yet.

The guide is an aid, not a questionnaire that has to be completed. If the story leads into unexpected but relevant territory, following it usually yields more than going back to the next planned question.

2.2 Questions about the past, not about the future

The questions that generate evidence point to specific past behavior. Teresa Torres turns this into a method with story-based interviewing, which asks the person to tell a specific occasion, with its context and its sequence, instead of generalizing about what they usually do. The past is remembered with verifiable details; the future is invented out of politeness.

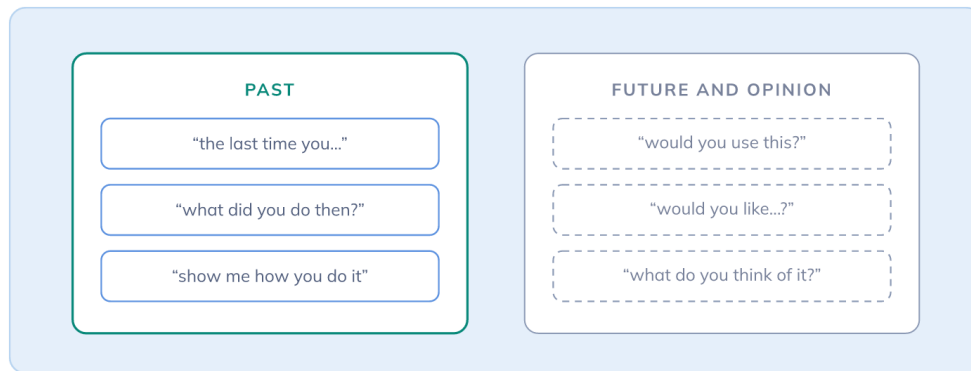


Figure 2. Questions that give signal and questions that give noise: the former ask for specific stories from the past and the latter, for predictions or opinions.

Fitzpatrick sums up the same criterion in three guidelines: talk about the person's life and not about our idea, ask about specific cases from the past and not about generalities or intentions, and listen much more than you talk. Opinion questions ("would you like this feature?", "wouldn't it be better to have everything in one place?") ask for speculative estimates, push toward the kind answer, or suggest it outright.

How to rephrase an opinion question

- "Would you use this?" becomes "tell me about the last time you had this problem."
- "How much would you pay?" becomes "what have you used so far to solve it, and how much does it cost you?"
- "Do you find it useful?" becomes "show me how you do it now."

2.3 The confirmation bias trap

When the team already has a solution in mind, the tendency arises to design questions that confirm it. An objective such as "validate whether users want the new sign-up screen" carries the solution inside and pushes toward answers that support it. The problem shows less in the interview than in the guide that was prepared beforehand.

Warning signs in an objective or a guide

- The objective **names a specific feature**.
- It uses verbs such as **validate, confirm, or prove** applied to a solution.
- It defines success as **a minimum number of positive answers**.
- The questions **present the idea** before the situation has been explored.

The correction is to rephrase toward exploration. “Validate the new sign-up screen” becomes “understand how they sign up today, where they get stuck, and what they are trying to achieve.” With that objective, the interview can surprise you, which is precisely what is asked of it.

2.4 Conducting the conversation

During the interview, follow-up questions are worth more than those in the guide. When the person mentions a point of friction, it pays to stay with it (“what happened then?”, “how did you solve it?”) until you have a complete story. Silences help too, because they usually bring the detail that the first answer left out.

It is also important to keep discovery separate from selling. If the person asks about the solution, you can note it down and postpone it until the end, but presenting the product halfway through the conversation contaminates everything that comes after. When possible, it is best for two people to interview, one leading the conversation and the other taking notes, because it is hard to listen well and record well at the same time. Who interviews also matters, and chapter 14 covers how a team that discovers every week organizes it.

The AI-moderated interview. There are already platforms that conduct interviews without a human interviewer. In January 2026 the Nielsen Norman Group concluded that they are useful for gathering opinions about a product or screening participants, but not for exploring an unknown problem, which is exactly what this interview is for. Chapter 16 develops when to use them.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Set the learning objective of an interview before writing the guide, in terms of what you want to learn and which observation would indicate it.
- Distinguish a question about past behavior from one about opinion or future intention, and rewrite the latter in terms of the former.
- Ask for and follow a specific story from the past until its context and sequence are obtained.
- Detect the signs of confirmation bias in an objective or a guide.
- Separate, in practice, the discovery conversation from the presentation of a solution.

Common mistakes and antipatterns

- **Asking “would you like...?”** It asks for a polite prediction with no value. Rephrasing toward the past, with “tell me about the last time...,” brings the conversation back to facts.

- **Mixing selling and discovery.** Presenting the solution halfway through the interview contaminates the signal. The proposal, if requested, is left for the end or for another conversation.
- **Going out with a vague objective.** “Gathering feedback” does not say which assumption is being explored or which evidence would clarify it, so the notes will not be actionable.
- **Settling for generalities.** “I usually do it this way” is an opinion about oneself. Asking for the last specific time turns the generality into a story that can be analyzed.

Further reading

- [Rob Fitzpatrick — The Mom Test](#)
- [Teresa Torres — Customer interviews](#)
- [Nielsen Norman Group — Interviewing users](#)

BLOCK A · TALKING TO CUSTOMERS AND CAPTURING SIGNAL

Chapter 3. Who to listen to: segment, sample, and recruitment

· ESTABLISHED

What you learn depends on who you listen to: the segment is set before recruiting, participants must be external and representative, and the sample is sized by its purpose.

Introduction

A flawless interview with the wrong person produces the wrong evidence, and no later technique corrects it. Selection bias appears in the same way in an interview, in a prototype test, and in demand validation, because it originates in a decision made before any conversation: who gets invited.

In this chapter we will see how the segment is defined, how to check that a participant belongs to it without giving away what is being sought, why your own network is a dangerous source, and how many conversations are needed depending on what you want to learn.

3.1 Defining the segment before recruiting

Recruiting starts with deciding who you want to listen to and why. The segment is better described by what people do than by their demographic profile: someone who has hired a similar service in the last six months, someone who manages a team's invoices, someone who abandoned sign-up halfway. That description is then translated into verifiable criteria.

An example shows the difference. “Small businesses in the services sector” is a profile that admits almost anyone; “people who in the last three months have prepared their company's quarterly tax return themselves” describes a verifiable situation that is

relevant for anyone who wants to understand how that task is handled. The second segment produces fewer candidates, but each conversation is worth much more.

Inclusion and exclusion criteria

- **Inclusion:** the behavior or situation that makes the person relevant to the research question.
- **Exclusion:** what would make them unrepresentative, such as working in the industry, knowing the product from the inside, or taking part in studies too often.
- **Variety:** what is worth spreading within the sample (tenure, heavy or occasional use, those who stayed and those who left).

The cases that do not fit with the majority deserve special attention. Chapter 6 explains why they often reveal another segment with different needs, and why it is worth having recruited with enough variety to be able to see it.

3.2 The screener

The criteria are checked with a screener that is sent before scheduling anyone. Kathryn Whitenton, of the Nielsen Norman Group, insists that it must balance two objectives: obtaining the necessary information and not revealing what the study is about, because an overly direct question invites people to exaggerate in order to be chosen.

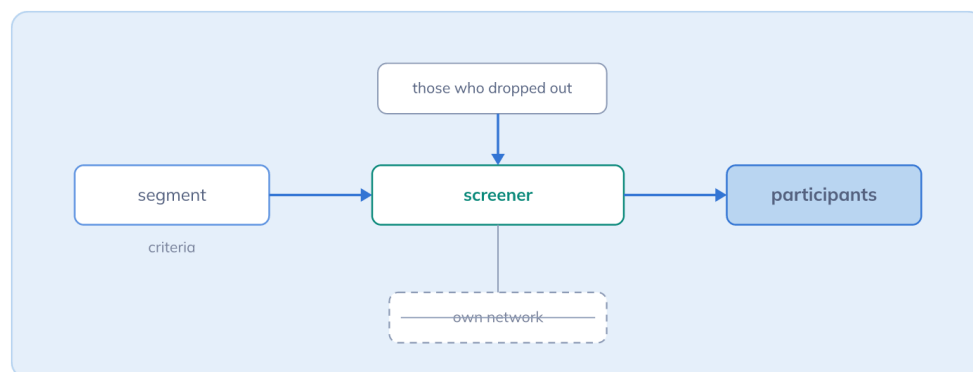


Figure 3. From the segment to the participants: the criteria are checked with a screener that does not give away what is being sought, and your own network is left out.

How to write it

- Ask in an open or indirect way: instead of “do you play online games?”, “what activities do you do online?”
- In closed questions, include plausible distractor options alongside the one you are interested in.
- Ask about recent and specific behavior, not about intentions.
- Be wary of anyone who ticks every option time after time, because they are usually after the incentive more than the participation.

3.3 Outside your own network

The most convenient source of participants is also the most biased. Friends, colleagues, followers, and enthusiastic customers know the context and have incentives to please, so their signal arrives inflated. Recruiting only among those who completed a process also adds survivorship bias, because it leaves out those who abandoned it, who are usually the ones who teach you the most.

Sources that correct the bias

- People who canceled or abandoned a process halfway.
- Users of alternatives or of the competition, including the alternative of doing nothing.
- Participants recruited from within the product itself with criteria defined in advance.
- Recruitment panels, filtered with the screener.

More and more people with no specific research training are recruiting. According to Maze, in 2026 39% of respondents say that product managers in their organization already conduct research with users. It is good news for the frequency of contact with customers, as long as the decision about who to listen to is taken with the same care as the decision about what to ask.

3.4 How many conversations are needed

The size of the sample depends on its purpose. To discover patterns, problems, and opportunities, a small number of well-chosen conversations is enough, extended as long as new things keep appearing and stopped when the interviews start repeating what has already been heard. In qualitative research this point is called saturation, and it is worth checking it by segment, because one group can be saturated while another has barely been explored. To measure how many customers share a pattern, on the other hand, you need other methods and much larger samples.

It is important not to carry figures from one context to another. Jakob Nielsen's five users refer to finding usability problems in a test of a specific design, not to exploring a problem in interviews or estimating demand. The honest thing is to state what the sample is for and not ask it for conclusions it cannot give.

In continuous discovery, where the team talks to customers every week, the question changes shape. It is no longer about how many interviews a study has, but about whether the cadence allows the team to hear the different segments often enough to detect changes. Chapter 14 covers how that recruitment is sustained week after week.

A small sample discovers; it does not quantify. “Seven out of twelve interviewees mentioned the friction” is a valuable data point for deciding what to explore, but it does not justify claiming that 58% of customers suffer from it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Translate a segment of interest into inclusion, exclusion, and variety criteria based on behavior.
- Write screener questions that check the criteria without giving away the objective of the study.
- Identify own-network bias and survivorship bias in a sample, and propose sources that correct them.
- Size the sample according to its purpose, distinguishing discovering patterns from measuring their frequency.
- Explain why a sample figure that is valid for one type of study does not carry over to another.

Common mistakes and antipatterns

- **Recruiting whoever is at hand.** Colleagues, friends, and enthusiastic customers want to please and know the context. Their opinion can guide you, but it does not replace external participants from the segment.
- **Giving the study away during screening.** A question such as “would you be interested in an app for X?” invites people to exaggerate. Asking about what the person does reveals who fits without giving them clues.
- **Listening only to those who stayed.** Interviewing only those who completed the process leaves out those who abandoned it, who usually explain better what is failing.
- **Quantifying with a discovery sample.** Twelve interviews are useful for finding patterns, not for asserting percentages of the population. Stating the scope of the sample avoids conclusions it does not support.

Further reading

- [Nielsen Norman Group — Screening questions](#)
- [Nielsen Norman Group — Why you only need to test with 5 users](#)
- [Maze — Future of User Research Report 2026](#)

BLOCK A · TALKING TO CUSTOMERS AND CAPTURING SIGNAL

Chapter 4. Quality notes: separating fact, quote, and interpretation · ESTABLISHED

Synthesis is only as good as the notes that feed it; separating what was said, what was done, and what is interpreted from the moment of capture is what makes it possible.

Introduction

An excellent interview with poor notes is a lost interview. The nuances of a conversation fade quickly, and whatever was not recorded precisely is reconstructed later with the biases of the listener. Recording and automatic transcription have changed what needs to be noted down live, but not the need to separate levels.

In this chapter we will see what is captured and in what order, how observation is separated from interpretation, how notes are closed while still fresh with the one-page snapshot proposed by Teresa Torres, and what care the summaries generated by tools require.

4.1 What is captured and in what order

During the interview and right after it, four types of record are kept separate. Each one serves a different purpose in the synthesis, and mixing them from the start makes rigorous analysis impossible.

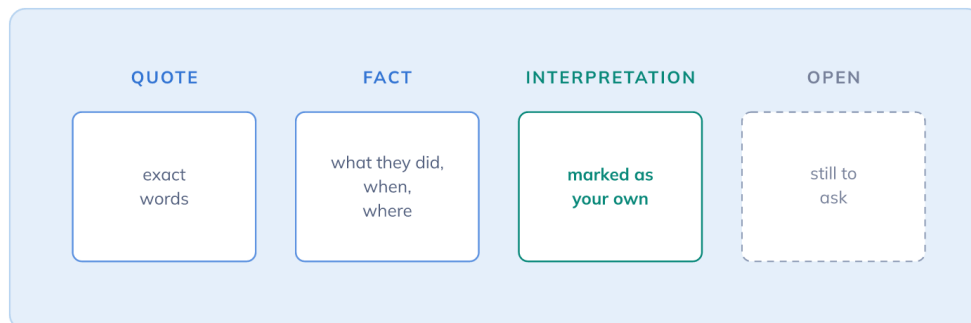


Figure 4. Four separate records: quote, fact, interpretation, and open question are noted separately, and the interpretation is marked as such.

1. **The quote:** the person's exact words, in quotation marks.
2. **The fact:** what they did or said they did, with its context of who, when, and in what situation, which is what allows segmenting later.
3. **The interpretation:** what the interviewer infers about the meaning or the motivation, clearly marked as their own inference.
4. **The open question:** the doubts that remain open and deserve another question or another interview.

When the conversation is recorded and transcribed, the verbatim quote stops being scarce, because the transcript preserves it in full. The value of the notes taken live then

shifts to the context, the marked interpretation, and the open questions, which no transcript captures on its own.

That is why whoever takes notes live should concentrate on what the recording does not capture. What the person did while telling something (whether they looked for a document, hesitated, or opened a tool to show it), what contradictions appeared between what they said and what they did, and which questions were left unanswered. These are precisely the data that are hardest to reconstruct later.

4.2 Separating observation from interpretation

Noting down interpretations is legitimate and useful, as long as they are marked. A simple marking system is enough, with one mark for the quote, another for the fact, another for the interpretation, and another for the open question. That way, when synthesizing, you can tell what the person said and did from what the interviewer assumed, and avoid taking inference for data.

A quick check helps to detect the problem. If the notes sound like adjectives (“she loved it,” “good attitude,” “happy with the timing”), concrete evidence is probably missing. “I spend two hours a day in follow-up meetings” is a fact; “he is happy” is an unsupported judgment that nobody will be able to review later.

When two people interview, comparing their notes at the end is one of the cheapest ways to detect interpretations that had slipped in as facts. If one wrote “not interested in the feature” and the other “said she already uses another tool for that,” the second is the observation and the first, the conclusion someone drew from it. Resolving those differences while the conversation is still fresh greatly improves the material that will reach the synthesis.

4.3 Closing while it is fresh: the interview snapshot

Teresa Torres gave shape to the practice of processing each conversation right after it ends with the interview snapshot. It is one page per interview that makes it memorable, actionable, and easy to consult when the team accumulates dozens of conversations.

What it contains

- The person's name and some quick context facts (role, type of organization).
- A memorable quote that evokes their story.
- The opportunities: unmet needs, pain points, and desires.
- The insights that are not opportunities but are worth remembering.
- An experience map that orders the key moments of the story.

Torres recommends reserving the time in the interview's own calendar slot, with 20 to 30 minutes of conversation and 15 to 25 more for the snapshot, while the stories are still fresh. Waiting days to close the notes means losing nuances that can no longer be recovered. Chapter 15 shows how the opportunities in these snapshots feed the opportunity solution tree.

4.4 Generated transcripts and summaries

Today's tools transcribe the conversation and propose a summary in seconds. The transcript is a faithful and very valuable record; the summary, on the other hand, is one more interpretation, made by a system that was not in the conversation and that tends to smooth over what does not fit. It is best to treat it as such and check it against the transcript before relying on it.

Recording and transcribing also carry their own obligations regarding the person's data, which chapter 5 develops. And the use of AI to synthesize several interviews, with the risk of delegating judgment, is covered in chapter 16.

Finally, notes and snapshots are only useful if the team finds them when it needs them. Keeping them in a shared repository, in the same format and tagged by segment and by opportunity, makes it possible to go back to a conversation from months ago when a new question comes up, and keeps customer knowledge from living in the notebooks of whoever did the interview. Chapter 14 covers how that repository fits into the team's weekly habit.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Separate, in the notes of an interview, the quote, the fact with its context, the marked interpretation, and the open question.
- Rewrite an interpretation disguised as data into a verifiable observation with context.
- Detect notes based on adjectives and turn them into concrete facts.
- Produce the snapshot of an interview, with its opportunities, insights, and experience map, in the time right after the conversation.
- Check a summary generated by a tool against the transcript before using it in the synthesis.

Common mistakes and antipatterns

- **Notes without context.** Without who, when, and in what situation, no insight can be derived and no segmentation can be done later.
- **Interpretations that look like data.** Mixing them with the facts produces conclusions that nobody can verify. Marking each interpretation as such allows it to be reviewed.
- **Recording only what confirms.** Noting what fits the hypothesis and forgetting what contradicts it is confirmation bias. It is worth asking yourself whether you wrote down anything you did not like hearing.
- **Trusting the automatic summary.** Taking the tool's summary as if it were the notes loses the nuances and the exceptions. The transcript is the source; the summary, just one more reading.

Further reading

- [Teresa Torres — Interview snapshots](#)
- [Scrum Manager BoK — Interview snapshot \(in Spanish\)](#)
- [Teresa Torres — Customer interviews](#)

BLOCK A · TALKING TO CUSTOMERS AND CAPTURING SIGNAL

Chapter 5. Consent, data, and transparency when talking to customers · CONSOLIDATING

Recording, transcribing, and analyzing an interview means processing personal data: you inform before and during, ask for consent for each recording, check what the tool does with the data, and, if an AI is interviewing, the person knows it.

Introduction

The interview has filled up with tools. The video call is recorded, one service transcribes it, another summarizes it, and an assistant helps synthesize dozens of conversations. Each step processes data about the person who talked to us, and what used to be a matter of good manners now has authoritative criteria that are worth knowing.

In this chapter we will see what data is processed when talking to customers, what needs to be done before, during, and after recording according to the criteria published in 2026 by the Spanish Data Protection Agency (AEPD), what to ask the tools, and what changes when the one interviewing or analyzing is an AI. It is not a chapter of legal advice, but of informed professional practice.

5.1 What data is processed when talking

An interview collects personal data in several ways. There is the person's identity and context, there is what they tell you, which may include sensitive information about their work or their life, and there is the recording itself. The Spanish Data Protection Agency points out that a voice must be considered personal data insofar as it can identify a person.

It follows that the General Data Protection Regulation (GDPR) applies to the recording, to the transcript, to identifiable notes, and to whatever the tools do with all of it. It is worth keeping this in mind from the design of the study, rather than discovering it when there are already twenty recordings in a shared folder.

Two principles of the regulation help to decide in practice. One is data minimization, which asks you to collect only the data necessary for the purpose of the study: if the full name of the company is not needed to understand the problem, it is not written down. The other is storage limitation, which invites you to set from the start how long the recordings will be kept and to delete them once the notes and snapshots already capture what was learned.

5.2 Before, during, and after recording

In April 2026 the Agency published the second part of its criteria on AI voice transcription, focused on accountability, rights, and transparency. From it follows a practical sequence for any team that records interviews.

1. **Before:** inform the person that the session will be recorded, what for, and who will have access, and ask for specific consent for that recording; consent for one session is not valid for the following ones.
2. **During:** keep a visible and active indicator while recording, such as an on-screen notice, so that the person knows at all times that the processing is taking place.
3. **After:** ensure the accuracy of the transcript and correct it without delay if it contains errors, in addition to honoring the person's access rights.

A practice that fits in the guide. Opening the interview by explaining the recording and asking for permission, and closing it by recalling how what has been said will be used, does not lengthen the conversation and leaves a record of something that is hard to reconstruct later.

5.3 What the tool does with the data

The Agency distinguishes two operations that often go together. One is the transcription entrusted to the service; the other, the use of voice samples to improve the system itself. When the provider retrain its model with the data for its own purposes, it becomes the controller of that processing and needs a legal basis different from the one covering the transcription.

What to check before adopting a tool

- Whether it uses the recordings or transcripts to train its models, and whether this can be turned off.
- Where the data is stored and for how long.
- Whether it infers emotions or other sensitive categories from voice or image.
- What data processing agreement it offers the organization.

Raw notes deserve the same care. Pasting transcripts with names, companies, or identifiable data into a general-purpose AI assistant sends them to a third party without the guarantees above. Anonymizing them first is a cheap habit, consistent with what the User stories skill recommends for the context handed to an agent.

5.4 When AI interviews or infers emotions

The EU AI Act adds transparency obligations that apply from August 2, 2026. Its Article 50 requires systems that interact directly with people to be designed to inform them that they are dealing with an AI, unless it is obvious, and requires whoever deploys an emotion recognition system to inform the people exposed to it. An AI-moderated interview falls squarely under the first case.

Emotion detection deserves more caution. The Agency warns that transcription services that include it are subject to a strict regime and may fall into uses that the Act prohibits, and its Article 5 prohibits inferring emotions in the workplace and in educational institutions. If you interview your own employees for an internal tool, it is best to treat it as a serious risk and consult before turning those functions on.

What did not change with the postponement. The simplification package that entered into force on July 27, 2026 postponed the high-risk obligations and gave providers of systems already on the market until December 2, 2026 for the marking of synthetic content. The transparency duties of those who use these systems remain in force since August.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Identify what personal data is processed in an interview that is recorded, transcribed, and analyzed with tools.
- Apply the sequence of prior information, a visible indicator during recording, and specific consent for each recording.
- Check whether a transcription or synthesis tool retrains its models with the data and what that implies for adopting it.
- Anonymize notes and transcripts before using them in a general-purpose AI assistant.
- Recognize when using an AI as interviewer or using emotion detection triggers transparency obligations or risks of prohibition.

Common mistakes and antipatterns

- **Recording without explicit notice.** Assuming that accepting the video call means accepting the recording breaches the Agency's criteria. You inform beforehand and keep the notice visible.
- **Reusing consent.** Permission to record one session does not cover the following ones or other uses. Each recording has its own consent.
- **Pasting raw notes into an assistant.** Names and identifiable data end up with a third party without guarantees. Anonymizing first takes minutes.
- **Turning on emotion detection by default.** It may fall into restricted or prohibited uses, especially with your own employees. It stays off unless there is an analyzed justification.

Further reading

- [AEPD — AI voice transcription \(I\)](#)
- [AEPD — AI voice transcription \(II\)](#)
- [EU AI Act — Article 50](#)
- [Morgan Lewis — What went into effect on August 2, 2026](#)

BLOCK B · FROM DATA TO UNDERSTANDING

Chapter 6. Synthesis: from observation to actionable insight

ESTABLISHED

Synthesis turns scattered conversations into understanding that guides decisions, and it only achieves this if it climbs level by level and does not confuse a quote with an insight.

Introduction

Having many conversations with customers is not the same as learning from them. Between the last interview and the next product decision there is work that is often done in a hurry or not done at all: organizing what was heard, seeing what repeats, understanding why it happens, and deciding what it means for the product. That work is synthesis, and it holds a good part of the distinctive value of whoever does discovery.

In this chapter we will see the levels through which qualitative analysis climbs, how to go from quotes to patterns by counting sources, what to do with the cases that do not fit, and how to formulate an insight that really guides a decision. The ritual of group synthesis, with its room and its walls full of sticky notes, has its own skill at Scrum Manager, Design Thinking; here we deal with its analytical side.

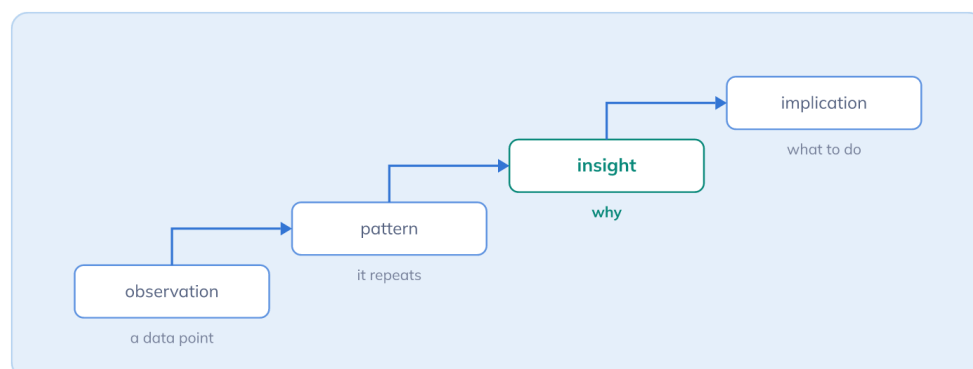


Figure 5. From observation to implication: synthesis climbs through levels, and the insight is the one that connects the pattern to a cause.

6.1 The four levels of analysis

Qualitative analysis climbs through levels that rest on one another. Naming them helps to know where each of the team's statements stands and not to present as a conclusion what is still a loose data point.

- **Observation.** A data point from a specific source: “Ana takes thirty minutes to find the contract she needs.”
- **Pattern.** What repeats across several different sources: “six out of ten interviewees describe friction when searching for documents.”
- **Insight.** The new understanding that connects the pattern to a cause or a motivation: “they waste time because the folders reflect the structure of the department and not that of the projects they work on.”

- **Implication.** What it makes sense to do as a result: “explore navigation by project, in addition to navigation by folder.”

The most frequent mistake is calling a striking quote or an uninterpreted pattern an insight. A simple check helps to avoid it: ask yourself what we have learned that we did not know before and what it means for the product. If the cause or the implication is missing, what we have is still an observation or, at most, a pattern.

6.2 From quotes to patterns: grouping and counting

The first step of synthesis is to group the observations that deal with the same phenomenon. It is best to review all the notes without categorizing them in advance, let the groups emerge from the material, and name them afterward, which is the logic of affinity diagramming as described by the Nielsen Norman Group. Categorizing before reading makes the material fall into place according to what we already thought.

Then you count, and what you count is distinct sources, not mentions. “Seven out of twelve interviewees mentioned the waiting time” is much more useful than “several users talked about speed problems,” because it allows topics to be compared and prioritized. As a guideline, it is best to see a pattern in several sources, three or more, before asserting it; a single voice, however intense, may represent the needs of a minority.

It is also worth having each pattern keep the trail of the observations that support it, with the source and date of each one. That traceability makes it possible to go back to the original quote when someone disputes the pattern, to revise the synthesis when new interviews arrive, and to reuse what was learned months later, which is the job of the research repository in chapter 14. A pattern without a trail ends up being the team's opinion dressed up as data.

Counting is not measuring. Ten interviews are enough to discover that a pattern exists, but not to estimate what proportion of customers shares it. Saying “70% of users” on the basis of seven out of ten conversations turns a qualitative indication into a figure it is not. Measuring the magnitude requires a different instrument, such as a survey or usage data.

6.3 Exceptions as a sign of a segment

When most people agree and a few cases do not fit, the temptation is to discard them as noise. It is usually a mistake. Exceptions often reveal segments with different needs, uses nobody had anticipated, or organizational contexts that change the problem.

The useful question is not how to ignore the cases that do not fit, but what they have in common with one another. If the three interviewees who do not complain about search work in small teams, perhaps the problem appears with the volume of documents and the solution should adapt to that threshold. Documenting exceptions with their context (role, size, situation) is what allows you to come back to them when the main pattern stops explaining what is observed.

Sometimes the exception forces you to rethink the pattern itself. If the satisfied interviewees are precisely those who have been using the product the longest, perhaps the friction lies in initial learning and not in search, and the opportunity is a different one. That is why it is worth revising the groups after analyzing the exceptions and not treating the map of patterns as closed at the first attempt.

6.4 Formulating an insight and checking it

A well-formulated insight brings together three pieces: a quantified pattern, an interpreted cause, and a specific implication. “We observed that five out of eight users abandon long content because their fragmented workdays leave no room for extended sessions, and this suggests evaluating a short-format approach” can be discussed, checked, and turned into an experiment. “Users need a more personalized experience” allows none of that, because it has no counted pattern, no cause, and no implication.

The cause is the most fragile piece, because it is an interpretation by the team and not a data point from the customer. That is why it is best to mark it as such and, when the decision that depends on it is expensive, to treat it as an assumption that has to be tested (chapter 8). An honest synthesis distinguishes what was observed from what is inferred.

Synthesis with artificial intelligence

AI assistants summarize transcripts and group themes in minutes, and this has created a new form of the mistake of confusing a quote with an insight. Teresa Torres warns against lazy AI interview synthesis, which consists of dumping the transcripts into an assistant and accepting its summary without reviewing it. Her proposal is to synthesize first, then ask the AI for the same synthesis, and compare the two. Chapter 16 develops that protocol and its limits.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Classify a statement by the team into one of the four levels of analysis (observation, pattern, insight, or implication).
- Distinguish an insight from a striking quote or an uninterpreted pattern, checking that it has a cause and an implication.
- Group observations by theme without prior categories and count recurrence as the number of distinct sources.
- Treat the cases that do not fit as a possible sign of another segment, identifying what they have in common with one another.
- Formulate an insight with a quantified pattern, an interpreted cause, and a specific implication, marking the cause as an inference.

Common mistakes and antipatterns

- **A list of ungrouped quotes.** It does not let you see patterns or prioritize. Group by theme and count how many distinct sources support each one.
- **Generalizing from an isolated case.** A single intense voice may represent a minority. Look for the pattern in several sources before asserting it.
- **An insight with no implication.** If it does not guide any decision, it is not yet actionable. Complete the sentence with what it makes sense to do as a result.
- **Accepting the AI's summary as the synthesis.** The generated summary is one more interpretation, with nobody's accountability behind it. Do your synthesis, check it against the tool's, and make the decision yourself.

Further reading

- [Nielsen Norman Group — Affinity diagramming](#)
- [Scrum Manager BoK — Affinity Mapping \(in Spanish\)](#)
- [Scrum Manager BoK — Identify themes \(in Spanish\)](#)
- [Skill Arena — Design Thinking](#)

BLOCK B · FROM DATA TO UNDERSTANDING

Chapter 7. Framing the problem: the job the customer wants to get done · ESTABLISHED

Stating the problem as the lack of a feature hides the solution inside it; framing it as the progress the person is seeking leaves several solutions open.

Introduction

The way a problem is stated conditions every solution that can be imagined afterward. The most expensive and most frequent mistake in product work is defining it as the lack of a feature, because it closes the exploration before it starts and turns the team's work into a queue of requests.

In this chapter we will see why that statement is a trap, what the jobs to be done (JTBD) framework and its two schools propose, how to write a useful job story, and how to go down from the symptom the person describes to the cause that produces it.

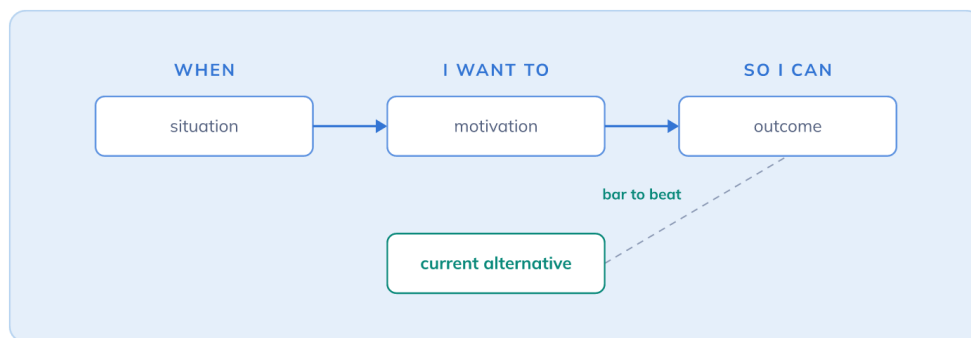


Figure 6. The job story and the current alternative: the situation, the motivation, and the outcome describe the progress being sought, and today's alternative sets the bar to clear.

7.1 The trap of stating the problem as a feature

When a request such as “we need to export to PDF” arrives, the tendency is to assume that the problem is the absence of that function. But “users do not have a button to export to PDF” carries the solution inside and closes the inquiry. The questions that open up the problem are different: what they need the PDF for, who they send it to, and what they do with it afterward. Perhaps what they need is to share data with people outside the organization, and a view-only link solves that better than a file.

When a team only receives and executes requests of this kind, it becomes what has been called a feature factory: it produces a lot and learns little, because it never asks what progress the person who requested each thing was seeking. Reframing the request as “the person is trying to achieve X, but today Y gets in the way, because of Z” is the first step out of that dynamic.

7.2 The jobs to be done framework and its two schools

The central idea of the framework is that people do not buy products for their features, but hire them to make progress in a specific situation. The focus shifts from the customer's profile and the requested feature to the circumstance and to what the person is trying to achieve in it. Two schools coexist under that umbrella, and it is worth telling them apart, because they do not use the same tools.

- **The job as progress.** Clayton Christensen and Bob Moesta understand the job as the progress a person is trying to make in a particular circumstance, and they explore it by reconstructing the story of how someone came to switch solutions. The reference work is *Competing Against Luck* (2016).
- **The job as a measurable outcome.** Tony Ulwick formulates the job as a verb with its object and its context, and measures the outcomes the customer wants to obtain when doing it. His method is called Outcome-Driven Innovation (ODI) and seeks to quantify which outcomes are most underserved.

A single job also has several dimensions worth exploring in the interview: the functional one (what the person needs to happen), the emotional one (how they want to feel), and the social one (how they want to be seen). Someone who hires a management tool to “keep the project under control” is often also seeking peace of mind in front of their manager, and that dimension explains decisions that the functional one does not.

For day-to-day discovery work, the first school contributes the language of progress and situation, and the second, the discipline of measurement. You do not need to adhere to one of them to take advantage of the framework, but it is worth knowing which one each tool comes from, because they are often cited mixed together.

7.3 The job story and the current alternative

The most widely used form of the framework in product teams is the job story, which originated at Intercom with Paul Adams and was named by Alan Klement: “when [situation], I want to [motivation], so I can [outcome].” It replaces the “who you are” of the classic user story with a “when it happens to you,” because for many needs the situation says more than the profile. It belongs to the tradition of progress, not to Ulwick's.

“When I coordinate deliveries across three remote teams, I want to see the real status of each task, so I can anticipate blockers before the committed date” is specific and observable. “When I need to manage my work, I want an easy tool, so I can be more productive” says nothing: the situation is generic and the outcome, circular.

It is worth adding to those three parts the alternative the person uses to solve the problem today: email, a shared spreadsheet, a phone call, or, quite simply, putting up with it. It is not part of the original template, but it is very useful, because it sets the bar that any solution has to clear and reveals the real constraints of the context.

There is always an alternative. Even if the customer says they do nothing, they do something: they live with the problem, delegate it, or solve it by hand. A solution that does not clearly beat that alternative does not trigger the switch, however much the idea is liked in the interview.

7.4 From symptom to root cause

Framing well requires digging beneath the symptom the person describes. The procedure is simple to state and demanding to apply.

1. Capture the symptom as the person tells it (“the system is slow”).
2. Explore the situation with follow-up questions about the specific past (“what were you trying to do the last time you noticed it was slow?”).
3. State the need in terms of an outcome (“find a specific piece of information in time to decide”).
4. Document the context and the current alternative.

Often “slow” turns out to be a problem of finding things, not of technical performance: optimizing the database would not solve a search frustration. A useful check is to write, for the same framing, three solutions that do not resemble one another; if only one comes to mind, or the three are variants of the same feature, the statement still carries the solution inside. The framing is well done when it leaves several possible solutions open, and it is worth checking this explicitly before moving on to ideation. The creative reframing of the challenge, with its “how might we...?” questions, is developed by the Design Thinking skill.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Detect a problem stated as the lack of a feature and rewrite it without prescribing the solution.
- Distinguish the school of the job as progress (Christensen and Moesta) from that of the job as a measurable outcome (Ulwick) and attribute its tools to each one.
- Write a job story with a specific situation, a motivation, and an observable outcome, completed with the current alternative.
- Separate symptom, cause, and need in a customer request by means of follow-up questions about the past.
- Check that a framing leaves several possible solutions open before moving on to ideation.

Common mistakes and antipatterns

- **The problem as a lack of a feature.** “They do not have PDF export” has already decided the solution. Reframe toward the objective: sharing data with people outside the organization in a way they can consult.

- **The circular job story.** “I want to manage projects so I can be more productive” guides nothing. Make concrete the situation that triggers the need and the outcome that could be observed.
- **Attributing the template to someone who did not create it.** The job story comes from Intercom and Klement, in the tradition of progress; Ulwick's method formulates the job in a different way. Mixing them confuses anyone who wants to go deeper.
- **Ignoring the current alternative.** Without it, you do not know what has to be beaten. Always ask how the person solves the problem today, even if the answer is that they put up with it.

Further reading

- [Intercom — Job stories](#)
- [Christensen Institute — Jobs to Be Done](#)
- [Strategyn — Outcome-Driven Innovation](#)
- [Scrum Manager BoK — Job story \(in Spanish\)](#)

BLOCK C · DESIGNING EXPERIMENTS TO LEARN

Chapter 8. Assumptions mapping: the risks to test first

ESTABLISHED

Every idea rests on assumptions that may turn out to be false; making them explicit and ordering them by importance and evidence keeps the team from testing what is easy and leaving what is critical untested.

Introduction

Before designing an experiment, it pays to know what needs testing and in what order. A product idea, however promising it looks, rests on a set of beliefs the team takes for granted without having checked them: that the problem exists, that someone will pay, that it can be built, that people will know how to use it. If one of them is false and happens to be one of those holding the idea up, everything else comes down with it.

In this chapter we will look at the types of assumption the reference sources distinguish, how they are prioritized with David Bland's importance-and-evidence matrix, why the usual order is not a fixed rule, and how a vague assumption becomes one that can be tested.

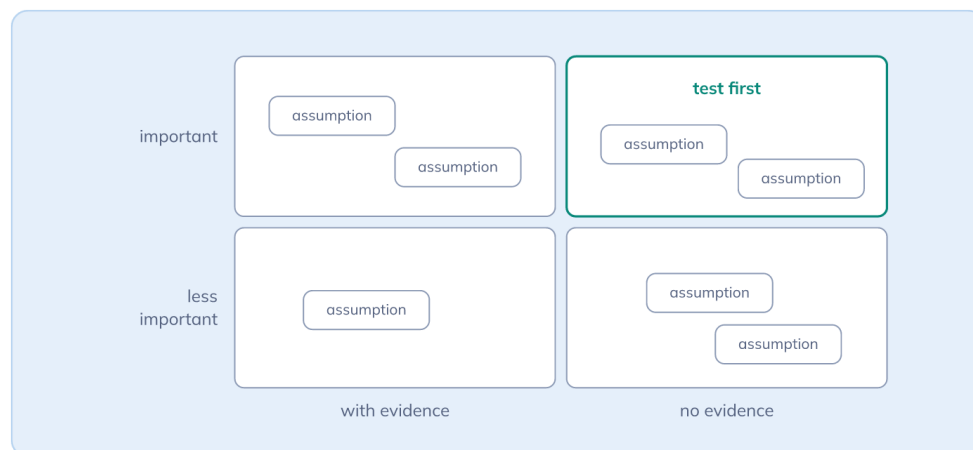


Figure 7. The assumptions map: what is important and still has no evidence is tested first.

8.1 The types of assumption

The most widespread classification distinguishes three types, which answer three questions about the idea.

- **Desirability.** Do they want it? The problem is real and a priority for someone, and that person will change what they do today to adopt the solution.
- **Viability.** Is it a business? Someone will pay the price, the cost of acquiring customers is affordable, and the margins support the investment.
- **Feasibility.** Can we build it? The team has the capabilities, the technology holds up, and the solution integrates with what already exists.

The reference sources extend this list. Teresa Torres adds usability (whether the customer will be able to use the solution) and ethics (whether building it could cause any harm). David Bland includes adaptability, which asks whether the idea will withstand changes in its environment. Marty Cagan speaks of four risks (value, usability, feasibility, and viability). The lists do not compete with each other; what they share is the obligation to look beyond the technical question.

8.2 The importance-and-evidence matrix

Assumptions mapping is the exercise Bland popularized in *Testing Business Ideas*, written with Alexander Osterwalder, and which he himself traces back to his work with Jeff Gothelf and Josh Seiden, the authors of *Lean UX*. It consists of listing the idea's assumptions and placing them on a matrix with two axes, in three steps.

1. Write each assumption on a note, one per note, completing the sentence “for this idea to work, it has to be true that...”.
2. Place each note according to its importance and to the evidence there is in its favor.
3. Pick the assumptions in the quadrant of the important ones without evidence and design, for each of them, the smallest test that will produce evidence.

The two axes are defined as follows.

- **Importance.** How much the idea depends on that assumption; if it turned out to be false, would the initiative fail, or would an adjustment be enough?
- **Evidence.** How much evidence we have today that it is true; from none, because we only believe it, to a lot, because we have observed it.

The quadrant that rules is that of the important assumptions for which there is no evidence. They are tested first, even if they are uncomfortable to test, because that is where a mistake costs the most. The important ones with evidence are monitored, and the unimportant ones can wait. Doing the exercise with every role in the room (product, design, development, and business) brings to light assumptions each person took for granted.

The assumptions no one questions. The most dangerous ones tend to come from leadership, and no one dares to put them on the matrix: “customers will switch without resistance,” “word of mouth will be enough to grow.” Placing them on the map turns them into something that can be evaluated instead of a blind spot.

8.3 The usual order and when it is reversed

For most new products, the usual order is to start with desirability, continue with viability, and leave feasibility for last. The reason is practical: the most common failure pattern consists of testing the technical side first, because it is comfortable and can be done without leaving the team, and discovering late that nobody wanted the result.

That order, however, is a consequence of the matrix and not a rule that replaces it. If in a specific case the important assumption without evidence is a feasibility one, that is the one tested first. It happens often with products whose value depends on an AI model, where the question of whether the system will respond with sufficient quality can decide everything; Torres has gone as far as presenting the evaluations of such systems as a new discovery habit. Chapter 18 deals with that case.

A practical consequence of working this way is that assumptions are tested, not whole ideas. Instead of building the entire solution to see whether it works, the critical assumption is isolated and the smallest test that puts it to the test is sought, which may be a question in an interview, a one-screen prototype, or a piece of data that already exists. The opportunity solution tree in chapter 15 organizes these tests by solution and by opportunity.

8.4 From a vague assumption to one that can be tested

An assumption that can be neither confirmed nor refuted is no use for deciding. “Users want a simpler tool” is a reasonable opinion, but it says nothing about what would have to be observed to accept or reject it. Reworded as “at least six out of ten purchasing managers who try the new flow will complete an order without help,” a test can now be designed and its success criterion set (chapter 11).

The rewording requires naming whom the assumption affects, what specific behavior would reveal it, and above what threshold we will consider it true. The more expensive the decision that depends on the assumption, the more precise its wording must be.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Classify an assumption as one of desirability, viability, feasibility, usability, ethics, or adaptability.
- Place a set of assumptions on Bland's importance-and-evidence matrix and justify which ones are tested first.
- Recognize when the usual order (desirability first) must be reversed because the critical assumption is of another type.
- Make explicit on the map the implicit assumptions of leadership or of the team's culture.
- Reword a vague assumption into one that names whom it affects, what behavior would confirm it, and with what threshold.

Common mistakes and antipatterns

- **Testing what is easy, not what is critical.** The assumptions that can sink the idea go untested. Let the matrix, and not comfort, decide the order.

- **Turning the usual order into dogma.** Always applying “desirability first” ignores the cases in which the lethal assumption is technical. The quadrant of the important without evidence rules.
- **Implicit assumptions that are never listed.** They produce blind spots, especially the ones that come from leadership. Map with every role and put those assumptions on the matrix too.
- **Assumptions that cannot be tested.** What can be neither confirmed nor refuted does not help to decide. Reword it with an observable behavior and a threshold.

Further reading

- [David Bland — Assumptions Mapping](#)
- [Teresa Torres — Assumption testing](#)
- [Marty Cagan — The four big risks](#)
- [Bland and Osterwalder — Testing Business Ideas](#)

BLOCK C · DESIGNING EXPERIMENTS TO LEARN

Chapter 9. Prototype tests: tasks as goals, not instructions ·

CONSOLIDATING

A prototype test measures the design only if the tasks are goals, a neutral person moderates, representative people take part, and what they do is observed.

Introduction

Testing a prototype with users seems simple, but many tests produce a signal contaminated by flaws in the test itself. The right practices have been known for years; what remains uneven is their disciplined application, and that is why this entry is consolidating.

In this chapter we will look at the decisions that separate a test that measures the design from one that measures something else: how the tasks are set, who moderates, what is observed, and what fidelity the prototype needs, including now that AI generates polished prototypes in minutes. Who takes part in the test is a prior decision that chapter 3 covers in depth.

9.1 Tasks as goals, not as instructions

The most common mistake is to write tasks that say how to complete them. “Click the green button at the top right, fill in the form, and save with the floppy disk icon” only evaluates whether the person can follow instructions. “Create a project called Q1 Test” reveals whether the design is understood, where the person gets confused, and what mental model they apply.

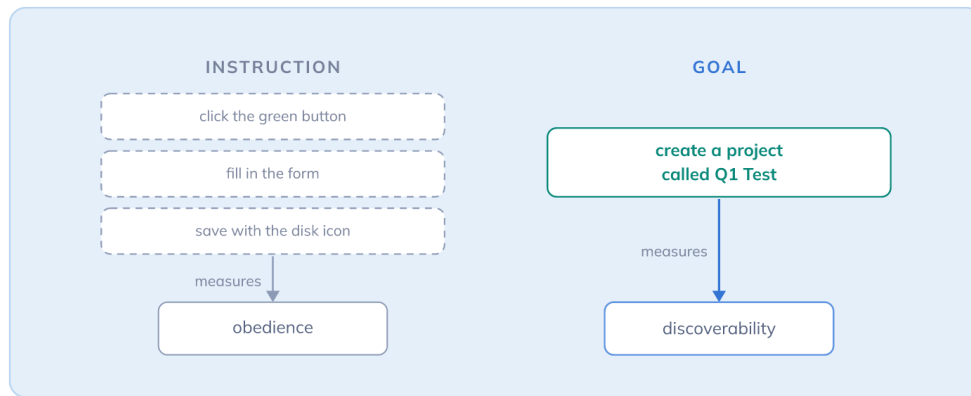


Figure 8. Two ways of asking for the same task: the step-by-step instruction measures obedience, and the goal without a method measures whether the design can be discovered.

The rule of thumb is to state the goal without indicating the path and to observe the one the person takes. It is worth reviewing the script before the test, looking for the words that appear in the interface: if the task uses the same label as the button that has to be clicked, it is giving away the answer.

9.2 Who moderates and who observes

When the person who designed the prototype moderates, they introduce subtle biases that invalidate the signal.

- The emotional interest in the design working.
- Unconscious guidance toward the right path, with gestures, silences, or questions that suggest it.
- The tendency to blame the confusion on the person and not on the design.
- Excessive help as soon as the person gets stuck, which erases exactly what was meant to be observed.

The solution is to separate the roles: a neutral person runs the session and whoever designed it observes without intervening. Just as important is whom you test with. Colleagues, friends, or enthusiastic customers know the context and want to please, so the test needs external participants who are representative of the segment, recruited with the criteria described in chapter 3.

As for the number of sessions, Jakob Nielsen argued at the Nielsen Norman Group that around five people per round are enough to find most of a design's problems, and that several small rounds, correcting between one and the next, are better than a single large one. The logic is that of discovery, because the test serves to find problems and not to measure precisely how many people suffer them.

9.3 Observing what the person does

The success criterion of a usability test is observable behavior, not opinion. “Do you like the design?” belongs to a different conversation. What gets recorded is something else.

- Where they click first and whether that first step leads toward the goal.

- Where they stop, how long they take, and at what point they give up.
- Which elements they cannot find and which ones they interpret differently from what was intended.
- What they say while they work, if they are asked to think aloud.

It is worth preparing a short script for each session, with an introduction that reassures the person (the design is being evaluated, not them), a realistic scenario that gives the tasks meaning, the tasks in order of difficulty, and a few closing questions about what they missed. The script is tried out first with someone on the team to catch ambiguous tasks or tasks that give away the answer.

The think-aloud protocol, in which the person verbalizes what they are looking for and what they expect while performing the task, is the classic technique for accessing their reasoning without asking them afterward for their impressions. Steve Krug sums it up in *Rocket Surgery Made Easy* as the cheapest way to find a design's serious problems with few sessions.

9.4 The fidelity that answers the question

The prototype's level of fidelity must match what you want to learn. Building high fidelity for everything wastes time and, on top of that, changes the conversation.

- To find out whether a concept is understood, sketches or screen outlines are enough.
- To explore a navigation flow, a clickable medium-fidelity prototype.
- To evaluate specific interactions, high fidelity.
- To test the value proposition, a page describing it is often enough (chapter 10).

Prototypes and moderators with artificial intelligence

AI tools now generate clickable, finished-looking prototypes in minutes, and that brings a new temptation. A prototype that looks finished invites comments on the finish and emotional attachment, both from whoever tests it and from whoever made it, so the rule of minimum fidelity becomes more necessary, not less. Prototyping as a way of thinking is developed in the Design Thinking skill.

Agents and automated moderators are also arriving that promise to evaluate usability without participants or without a human moderator. Research that trains agents for that task is advancing, as the work of Gao and colleagues published in April 2026 shows, but today it does not replace observing real people from the segment, which is what reveals the misunderstandings nobody had anticipated.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Rewrite a test task written as a step-by-step instruction into a task framed as a goal.

- Assign the moderation and observation roles so that whoever designed the prototype does not run the session.
- Define observable success criteria for a test (first step, times, abandonments, elements not found) instead of opinion questions.
- Apply the think-aloud protocol without suggesting the path to the participant.
- Choose the minimum prototype fidelity that answers the research question, even when AI allows more.

Common mistakes and antipatterns

- **Guided tasks.** If the task mentions locations or labels from the interface, the test measures obedience. Reword it as a goal and let the person find the path.
- **The designer moderates.** “I’ll moderate, since I know the design best” introduces bias with every gesture. Have someone neutral moderate and let the designer observe.
- **Asking for opinions.** “What do you think?” says nothing about usability. Design tasks that produce observable behavior.
- **High fidelity because it is free.** The fact that AI generates it in minutes does not make it appropriate: it shifts the conversation to the finish. Use the fidelity that answers your question.

Further reading

- [Steve Krug — Rocket Surgery Made Easy](#)
- [Nielsen Norman Group — Usability testing 101](#)
- [Scrum Manager BoK — Think-aloud protocol \(in Spanish\)](#)
- [Skill Arena — Design Thinking](#)

BLOCK C · DESIGNING EXPERIMENTS TO LEARN

Chapter 10. The MVP as an experiment, not a reduced product

· CONSOLIDATING

An MVP is the minimum needed to learn something specific; it is often better learned without writing code, and cheap building does not exempt you from asking what you want to learn.

Introduction

Few product terms are misunderstood as much as the minimum viable product (MVP). Understood as a product with fewer features, it produces lists of components disguised as experiments. Understood as a learning instrument, it saves months of development. The practice is gaining adoption, but it still coexists with the reflex to build first and see what happens, and AI has given that reflex new strength.

In this chapter we will look at the MVP as an experiment, the alternatives that do not require building the product, how each one is chosen according to the hypothesis, and why cheaper building does not change the underlying question.

10.1 An experiment, not a small version of everything

An MVP is the minimum needed to learn something specific, not the minimum needed to have a product. The difference shows when two approaches are compared.

A plan that includes sign-up, profile, notifications, dashboard, and integrations, with the hypothesis that users will use the platform and the criterion of launching it in six weeks, is a list of components with a generic hypothesis and no validation criterion. A plan that assumes finance teams will pay to automate bank reconciliation, tests it with a manual service for five customers over two weeks, and accepts the hypothesis if at least three of the five ask to continue and pay is an experiment, with its hypothesis, its instrument, and its threshold set in advance (chapter 11).

The word “minimum” therefore refers to what is needed to learn, not to what is needed to launch. An MVP can be much smaller than the first version of the product, and it often looks nothing like it; it can be a form, a spreadsheet operated by hand, or a weekly call with five customers.

10.2 The alternatives before writing code

Before building, it is worth asking whether the hypothesis can be tested with something lighter. *Testing Business Ideas* collects a broad repertoire of experiments; four are especially useful in early discovery.

- **Concierge.** The service is delivered by hand and the customer knows it. It tests real demand and willingness to pay before automating anything.
- **Wizard of Oz.** A seemingly automatic interface operated behind the scenes by a person. It lets you test the experience without having developed the system.
- **Landing page.** It describes the proposition and collects a signal of interest, such as an email address, a reservation, or an advance payment, before anything is built.
- **Single-feature MVP.** The minimum code needed to test the value of one specific feature.

Arguing against manual alternatives because they do not scale wastes the chance to learn quickly and cheaply. If nobody wants the manual service, there is nothing to scale; if they do, demand will have been validated before investing in automation.

The caution of the Wizard of Oz

The Wizard of Oz hides from the person that someone is behind it, and that concealment has limits. The experiment should not lead the person to decisions they would not make if they knew how it works, the data they share should be handled with the care described in chapter 5, and how it works should be explained to them at the end. If some

of the answers are actually generated by an AI system, since August 2, 2026, Article 50 of the EU AI Act requires the person to know they are interacting with an AI.

10.3 Choosing the type according to the hypothesis

Each question calls for a different instrument, and choosing it according to what you want to learn, and not out of habit, is what keeps the MVP an experiment.

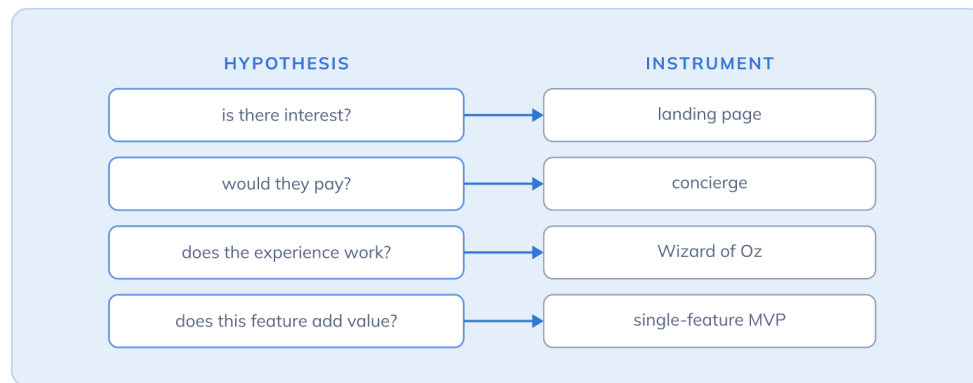


Figure 9. Each question, its instrument: the type of MVP is chosen by the hypothesis to be tested and not out of habit.

It is worth remembering that the signal each instrument produces does not carry the same weight. A landing page indicates initial interest, but an email address left on a form is a weak signal; a reservation or an advance payment weighs much more. Chapter 12 develops that scale of the strength of evidence, which is what makes it possible to interpret the result of any of these experiments.

The instruments are often chained, from the cheapest to the most costly. A landing page shows whether the proposition sparks interest; if it does, a manual service with a few customers shows whether they pay and what exactly they expect, and only then does it make sense to build the first feature. Each step buys the right to invest in the next.

10.4 Building cheaply is not learning cheaply

AI tools have made building working prototypes cheaper, and that changes the cost calculation that justified the manual alternatives. Marty Cagan distinguishes building to learn, that is, creating prototypes to evaluate the risks of value, usability, viability, and feasibility, from building to earn, which is producing software with the quality needed to sell and maintain it. With today's tools, he wrote in April 2026, ten or twenty prototypes a week are within almost anyone's reach.

The risk is confusing speed with direction. In July 2026 Cagan quoted Hilary Gridley: since it has never been so fast to build, it has never been so easy to run ten times faster in the wrong direction. For Teresa Torres, the fact that building is very cheap does not mean we should build every idea we have.

The working prototype generated with AI becomes a legitimate option in the repertoire, but the criterion remains the same, the minimum to learn something specific. And a danger specific to this instrument appears: that a prototype that works is

mistaken for the product and development starts on top of it without having learned what was sought.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish an MVP framed as an experiment from one framed as a reduced product or as a list of components.
- Formulate the main hypothesis of an MVP and the signal that would confirm or refute it before building.
- Choose among concierge, Wizard of Oz, landing page, single-feature MVP, or working prototype according to the hypothesis.
- Rebut the argument that a manual alternative does not scale when at an early stage.
- Apply the ethical and legal cautions of an experiment that hides how it works or that includes AI-generated responses.

Common mistakes and antipatterns

- **The MVP as a list of features.** It causes scope creep and loss of speed. Define the hypothesis first and then the minimum to test it.
- **Building because it is cheap.** The fact that AI generates the prototype in an afternoon does not say what you want to learn with it. Ask first what hypothesis it tests.
- **Dismissing manual alternatives.** It leads to investing when a hand-delivered service would have been enough. Always ask whether the same can be learned without code.
- **Taking interest for demand.** An email address on a landing page is a weak signal. Interpret the result with the scale of the strength of evidence.

Further reading

- [Marty Cagan — Build to learn vs build to earn](#)
- [Marty Cagan — The AI productivity paradox](#)
- [Bland and Osterwalder — Testing Business Ideas](#)
- [Scrum Manager BoK — Minimum viable product \(in Spanish\)](#)

BLOCK D · DECIDING WITH EVIDENCE

Chapter 11. Success criteria and signals defined in advance ESTABLISHED

The success criterion is set before launching the experiment, because deciding afterward which result looks good makes it possible to rationalize any result.

Introduction

An experiment without a prior success criterion looks more like a story to be told at convenience when the data arrives than like a test. In discovery the temptation is especially strong, because teams usually have few observations, a lot of hope invested in an idea, and no formal obligation to write down beforehand what they expect to see.

In this chapter we will look at the principle of prior definition, the hierarchy of metrics worth distinguishing, the limits imposed by sample size and the absence of a control group, and how to state honestly what a result allows you to conclude. It is where discovery becomes rigorous or stops being so.

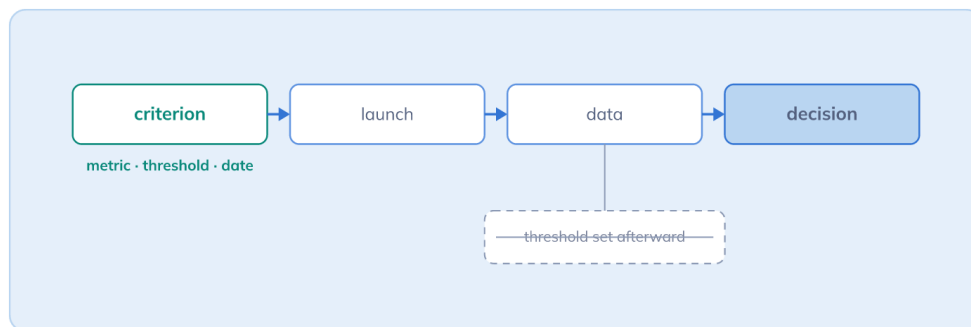


Figure 10. The criterion is set before launch: the success threshold is written down before the data arrive.

11.1 The principle of prior definition

The success criterion is written down before launching, with its date, and never after seeing the results. The reason is simple. With the data in front of you, almost any number looks good if you look for the right angle, and the team that most wants the idea to work is the worst placed to set the bar after the fact. Teresa Torres sums it up in the image of drawing a line in the sand before testing the assumption, so that the result falls on one side or the other without discussion.

This temptation has a name in statistics. P-hacking is analyzing the data in several ways until finding the one that produces a favorable result, and HARKing is formulating the hypothesis in light of the results as if it had been stated beforehand; the psychologist Norbert Kerr coined the term in 1998. Andrew Gelman and Eric Loken added an uncomfortable observation with their “garden of forking paths”: no bad faith is needed, because it is enough for the analysis decisions to depend on the data for the conclusion to be biased.

A complete criterion. It includes a primary metric, a threshold, a minimum sample size, and a date: “conversion on the booking page equal to or above 5% with at least 400 visits, measured from March 3 to 17 and set on March 1.” The opposite would be “we will measure visits, clicks, and conversions and decide later what looks good to us.”

Even a well-written criterion has limits worth knowing. With 400 visits, an observed conversion of 5% is compatible with a real conversion ranging from roughly 3% to 7.5%.

The threshold serves to decide whether to move forward, and the result is an indication of demand, which should be read as such and not as a precise measurement.

11.2 The hierarchy of metrics

A well-designed experiment assigns different functions to the metrics it looks at, and not all of them weigh the same in the decision.

- **Primary metric.** It is the one that decides success or failure, and there must be only one, with its threshold.
- **Secondary metrics.** They provide context about the result without changing the decision.
- **Guardrail metrics.** They are limits that must not be crossed even if the primary metric comes out well, and they serve to detect collateral damage, such as a conversion rate that rises at the expense of returns.
- **Diagnostic metrics.** They help to understand why what happened happened and to prepare the next iteration.

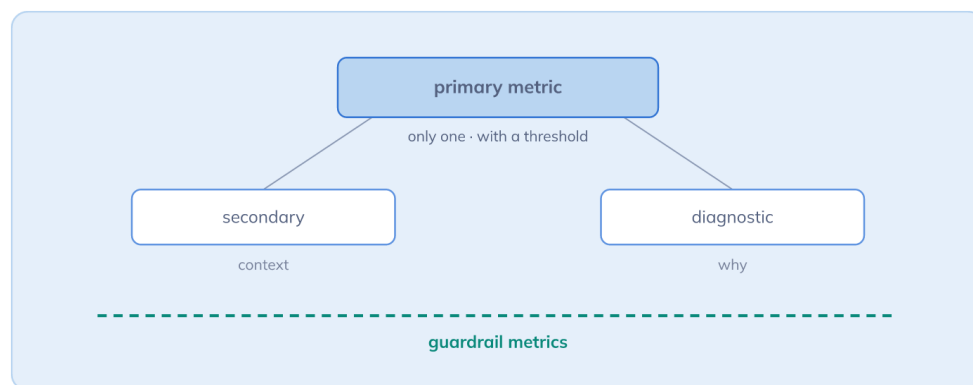


Figure 11. The hierarchy of metrics: a single primary metric decides, secondary and diagnostic metrics provide context, and guardrail metrics mark what must not break.

The typical confusion is having several primary metrics. With three candidates to decide, it is enough for one to come out well to declare success, and the prior criterion loses its meaning through the back door. Choosing just one also forces a valuable conversation, about what exactly the experiment wants to learn.

11.3 Sample size and minimum detectable effect

With small samples, normal fluctuations look like signals. Before launching, it is worth asking what difference could be reliably detected with the expected sample, which in experimentation is called the minimum detectable effect. A threshold of “any improvement” with fifty users confuses noise with signal, because a variation of a few points fits easily within chance.

In early discovery the sample is almost always small, and that does not invalidate the experiment, although it does change what can be claimed with it. A result with few observations guides the next decision and says little about the size of the market. Stating

it that way protects the team from overconfidence and the organization from investments resting on a fragile number.

11.4 The control group and directional evidence

Without a control group, a change cannot be attributed to the intervention. An increase in sessions after turning on a feature may be due to seasonality, another campaign, or factors nobody controls. The minimum design for attributing causality includes a treatment group, an equivalent control group, and random assignment between the two.

The nuance not to lose is one of scale. A reliable A/B test needs traffic that most early discovery work does not have. Ron Kohavi, Diane Tang, and Ya Xu explain in *Trustworthy Online Controlled Experiments* how much traffic a controlled experiment requires for its conclusions to be reliable, and why with few users the differences that matter fall below what the experiment can distinguish.

Directional evidence. When there is no control or not enough traffic, the honest thing is to state the evidence as directional: “we observe that..., which suggests that...” It is a less forceful sentence, but it is the one the data supports, and it makes clear what would be needed to turn the indication into a conclusion.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Define an experiment's success criterion (primary metric, threshold, minimum sample size, and date) before launching it.
- Classify an experiment's metrics into primary, secondary, guardrail, and diagnostic, and require a single primary one.
- Estimate whether the planned sample size allows the sought effect to be detected and adjust accordingly what will be claimed with the result.
- Recognize p-hacking and HARKing in a results report and reconstruct the criterion that should have been set beforehand.
- Distinguish a causal result, with a control group and random assignment, from a directional one, and write each up with its degree of certainty.

Common mistakes and antipatterns

- **Setting the threshold after seeing the data.** It is the most common form of p-hacking, and the bar ends up where the result fell. Document the criterion with a date before the launch.
- **Several primary metrics.** It is enough for one to come out well to declare success. Choose a single one and leave the others as context.
- **An “any improvement” threshold.** With small samples it confuses noise with signal. Calculate beforehand what difference the sample you will have can detect.

- **Presenting an indication as a cause.** “After turning on X, Y went up, so X caused Y” is a leap without a control group. State the evidence as directional.

Further reading

- [Teresa Torres — Assumption testing](#)
- [Kohavi, Tang, and Xu — Trustworthy Online Controlled Experiments](#)

BLOCK D · DECIDING WITH EVIDENCE

Chapter 12. The strength of evidence and bias control

ESTABLISHED

Evidence weighs more the more it resembles a real decision with a cost for whoever makes it, and four frequent biases erode it.

Introduction

Collecting data is not the same as having evidence. A team can pile up interviews, surveys, comments, and metrics and still not know whether anyone would pay for what it proposes, because almost everything it has collected is what people say and very little is what people do.

In this chapter we will look at how to grade the strength of evidence, which signals look like validation without being so, the four biases that most often erode it, and where the line between correlation and causation lies. It is the filter that separates rigorous discovery from validation theater.



Figure 12. *The strength of evidence: evidence weighs more the more closely what is observed resembles a real decision with a cost for the person making it.*

12.1 Evidence as a gradient

Alexander Osterwalder formulated in 2017 an idea that puts everything else in order. What customers say and what they do are different things, and evidence gets stronger the closer it gets to a real purchase. Between an opinion expressed in an interview and a payment there are many intermediate steps, and it is worth knowing which step each piece of data is on before deciding how much it weighs.

- **What is said.** Opinions, stated intentions, and survey answers. They are quick and cheap to obtain, and also the weakest.
- **What is signed up for.** An email address left on a landing page or a sign-up for a waiting list. There is a small gesture, almost without cost for whoever makes it.
- **What is reserved.** A reservation, a letter of intent, or a committed appointment. The customer puts time, reputation, or future money on the line.
- **What is paid.** A prepayment or a purchase, ideally from someone who does not know they are taking part in an experiment. It is the closest thing to the real decision.

Weak evidence has its use. Interviews, which occupy the lowest step as evidence of demand, are the best source for understanding the problem, as we saw in block A. What the gradient asks is to use each instrument for the question it answers well and not to add up their results as if they weighed the same.

The gradient also resolves a frequent contradiction. Capturing email addresses serves to measure initial interest, as we saw in chapter 10, and at the same time it is weak evidence of demand. Both things are true, because the email address is the right instrument for knowing whether someone is interested in the proposition and an insufficient instrument for knowing whether they will pay for it.

12.2 Signals that look like validation

Some low-cost signals are often mistaken for validation. Likes and comments on social media imply neither use nor willingness to pay, and forty-seven reactions to a post say very little about demand. Feedback from your own network (friends, contacts, subscribers) comes with a social incentive to be kind, and email opens indicate an active list, which says little about interest in a new product.

The quality question. Faced with any piece of data presented as validation, it is worth asking whether it is observable behavior with a cost for whoever produces it or an opinion that costs them nothing. The answer decides how much weight it deserves.

In 2026 a new source of false positives has appeared, the responses of AI-simulated users. They are costless opinion and, on top of that, sycophantic, so they occupy the lowest step of the gradient, below even the opinion of a real customer. Chapter 17 covers them in detail.

12.3 Four frequent biases

Many cognitive biases affect product evidence, and four appear so often that it is worth keeping them in mind in every experiment. The choice of these four is a criterion of this guide, and each one has its mitigation.

- **Confirmation.** It leads you to see only the data that supports the prior belief. It is mitigated by actively looking for what contradicts it and noting down what you did not like hearing.

- **Survivorship.** It analyzes only the cases that made it to the end, as when only those who completed a process are surveyed. It is mitigated by including those who dropped out or failed.
- **Selection.** It uses a sample that does not represent the segment, such as one that comes from recruiting in your own network. It is mitigated by setting the selection criteria before recruiting (chapter 3).
- **Social desirability.** It produces answers designed to please, of the “I would use it” or “I would pay for it” kind. It is mitigated by asking about past behavior, how much the person paid last time instead of how much they would pay.

The Scrum Manager BoK page on cognitive biases lists others with their definitions. It is worth reviewing it when a result seems too good to be true, which is precisely when there is the least desire to doubt it.

12.4 Correlation and causation

In product work it is easy to confuse correlation with causation. The fact that conversions rose after adding a feature does not imply that the feature caused them, and the fact that users who use a function retain more does not imply that the function causes retention; perhaps the most engaged users simply use more functions. Claiming causation requires a control group, random assignment, and control of confounding factors, as we saw in chapter 11.

Without those conditions, the honest thing is to say “we observe a correlation.” It is a weaker claim and also a more useful one, because it guides the next experiment without committing an investment the data does not justify.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Place a piece of data on the gradient of the strength of evidence, from what is said to what is paid, and give it a weight consistent with its position.
- Distinguish a weak validation signal (likes, feedback from your own network, email opens) from a signal with a cost for whoever produces it.
- Identify which of the four frequent biases affects a sample or a piece of data and propose its mitigation.
- Reword a question about future intention into one about past behavior to reduce social desirability.
- Differentiate correlation from causation in a specific result and ask for additional evidence when the available evidence does not support the conclusion.

Common mistakes and antipatterns

- **Taking reactions for demand.** Forty-seven likes validate nothing. Look for behavior with a cost, such as a reservation, a payment, or time invested.

- **Surveying only those who finished.** It excludes those who dropped out, who are precisely the ones who best explain the problem. Include those who failed.
- **Discarding inconvenient data.** Treating two negative users as isolated cases loses information. Investigate what sets them apart from the rest.
- **Weighing what is said and what is done equally.** A stated intention and a payment are not the same evidence. Place each piece of data on its step before adding them up.

Further reading

- [Alexander Osterwalder — How strong is your innovation evidence?](#)
- [Rob Fitzpatrick — The Mom Test](#)
- [Scrum Manager BoK — Cognitive bias \(in Spanish\)](#)

BLOCK D · DECIDING WITH EVIDENCE

Chapter 13. Persevere, iterate, pivot, or stop · CONSOLIDATING

Faced with the evidence, four paths open up, and the sunk cost fallacy is the most dangerous bias when choosing among them.

Introduction

Discovery culminates in a decision, and deciding well when the evidence is mixed is a competence in its own right. Many teams frame the decision as a yes or a no, continue or abandon, and thereby lose the intermediate options, which is where most real cases fall.

In this chapter we will look at the four options that open up in the face of a result, the sunk cost fallacy, what a responsible pivot requires, and which factors are worth weighing when the evidence is inconclusive. Its status is consolidating because the range of options is accepted, although its explicit use is still maturing in many teams.

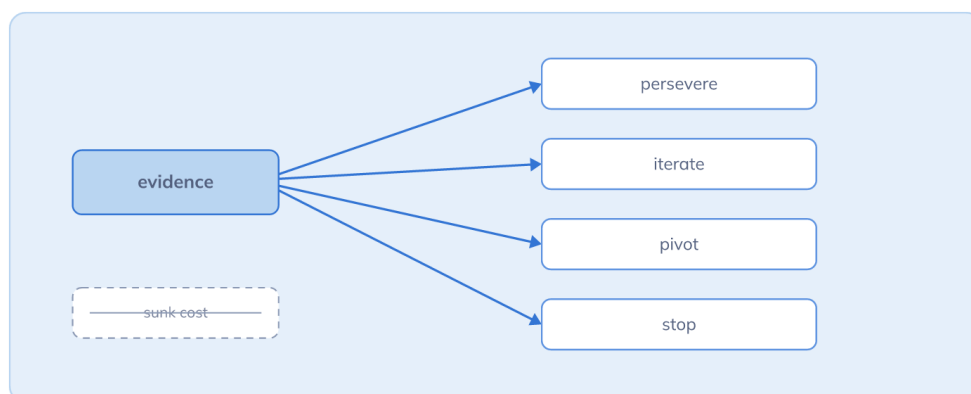


Figure 13. Four paths in the face of evidence: the decision opens into four options, and what has already been invested is not among the criteria.

13.1 The four options

Faced with the results of an experiment, the team chooses among four paths.

- **Persevere.** The evidence is positive enough to continue in the current direction.
- **Iterate.** Specific elements are adjusted (a message, a price, a step in the flow) while keeping the overall course.
- **Pivot.** A fundamental aspect is changed, such as the segment, the value proposition, or the channel, while keeping what has been learned.
- **Stop.** The evidence indicates that the initiative is not viable and that the resources yield more elsewhere.

The classic expression comes from Eric Ries, who in *The Lean Startup* frames the decision to pivot or persevere and describes several types of pivot, among them the customer segment change. That is why this guide does not treat “segmenting” as a separate option. Discovering that another segment responds better and reorienting toward it is a form of pivot, and it deserves the same discipline as the others.

The line between iterating and pivoting is not always sharp. A rule of thumb is to ask whether the underlying hypothesis changes or only its execution. Adjusting the price of an offer that already interests people is iterating; discovering that whoever pays is a different customer from the one interviewed is pivoting, because the hypothesis about the segment was false and everything that depended on it has to be reviewed.

13.2 The sunk cost fallacy

The most dangerous bias in these decisions is justifying continuing because of the time or resources already invested. “We've been at this for six months, the team is very committed, and the two users who finished were enthusiastic” sounds reasonable and says nothing about the future. What has already been invested is irrelevant to the decision; what matters is the best investment of the next resources given the current evidence.

The warning signs are easy to recognize when you look for them.

- Arguments about the effort invested instead of about the evidence.
- Attention to the few positive cases and silence about the negative ones.
- Emotional resistance to considering the pivot or the stop, often because the idea has a sponsor with clout in the organization.

The best defense is to agree in advance, together with the success criterion from chapter 11, on what result would lead to pivoting or stopping. If the stopping threshold is written down before starting, the decision no longer depends on the team's mood at the worst moment or on the influence of whoever has bet most on the idea.

13.3 Capturing the learning before changing course

A responsible pivot is a documented decision. Before changing course, the team writes down what it learned and why it did not work, formulates the new hypothesis from it,

tests it before investing, and explains the reasoning to those who depend on the decision.

The sentence that orders the pivot. “We learned that [hypothesis] was false because [evidence]; the new hypothesis is [X] and we will test it with [experiment].” If the team cannot complete it, it is not yet ready to pivot.

Stopping is documented too. An initiative that is halted with its learning written down leaves the team better placed than one abandoned in silence, because the next idea in that space no longer starts from scratch and the interviews and experiments already done still serve as evidence.

13.4 Deciding with mixed evidence

When the evidence is neither clearly positive nor negative, it is worth weighing several factors before deciding. The ones that follow are a working criterion of this guide, useful for ordering the team's conversation and making explicit what would otherwise be decided by intuition.

- **Quality of the evidence.** Whether it is robust or whether there are biases in the sample (chapter 12).
- **Magnitude of the signals.** Whether they are large enough to stand out from the noise (chapter 11).
- **Opportunity cost.** What the team stops doing while it continues along this line.
- **Reversibility.** Whether the decision can be undone at little cost or commits resources that are hard to recover.
- **Residual uncertainty.** How much remains to be known and how much it would cost to find out.

If the residual uncertainty is high and the additional experiment is cheap, the sensible thing is to design a more specific test before deciding. That way the decision links up with the continuous discovery of block E, where deciding is a recurring act every week and stops being a final milestone.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Place a decision on the range of persevere, iterate, pivot, or stop, without reducing it to a binary alternative.
- Recognize the sunk cost fallacy in an argument and redirect the decision toward the best investment of the next resources.
- Treat a segment change as a type of pivot and apply the same discipline of hypothesis and validation to it.
- Document a pivot or a stop with what was learned, the new hypothesis, and the experiment that will test it.

- Weigh mixed evidence by its quality, its magnitude, the opportunity cost, the reversibility, and the residual uncertainty, and decide whether another experiment is needed.

Common mistakes and antipatterns

- **Framing the decision as binary.** Continue or abandon ignores iterating and pivoting, which is where most real cases fall.
- **Giving in to sunk costs.** What has been invested does not weigh in the future decision. Ask about the best investment of the next resources.
- **Pivoting without capturing what was learned.** Changing course without documenting why risks repeating the same mistakes in the new direction.
- **Deciding blindly with a cheap test at hand.** If an affordable experiment would clarify the situation, deciding without it is hasty.

Further reading

- [Eric Ries — The Lean Startup](#)
- [Teresa Torres — Product discovery basics](#)
- [Scrum Manager BoK — Product discovery \(in Spanish\)](#)

BLOCK E · DISCOVERY AS A TEAM HABIT

Chapter 14. Discovering every week: the product trio and continuous contact · CONSOLIDATING

Continuous discovery keeps contact with customers going week after week, in the hands of the same team that builds the product.

Introduction

For years, discovery was understood as a phase: some research at the start of the project, a report, and then months of development without talking to customers again. The problem is that markets change and product hypotheses expire, so a team that did its discovery six months ago and has only been delivering since then is building up risk without knowing it.

In this chapter we will look at what continuous discovery is and who sustains it, how the weekly cadence is organized, how it fits with the work of the sprint, and how what is learned is preserved. The what of continuous discovery is taught in Scrum Manager's Product Owner certification; here we focus on how to keep it running week after week.

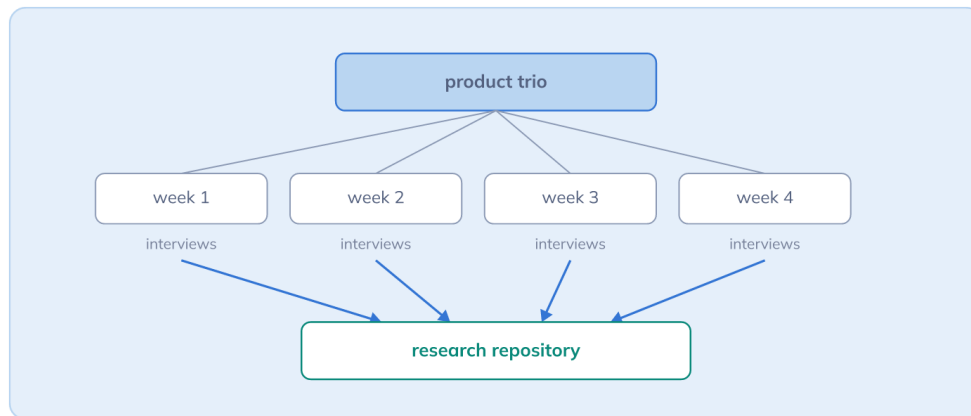


Figure 14. A weekly cadence: the product trio talks with customers every week, and what it learns accumulates in a shared repository.

14.1 Continuous discovery and the product trio

Teresa Torres defined continuous discovery in *Continuous Discovery Habits* (2021) as at least weekly contact with customers by the team building the product, through small research activities in pursuit of an outcome. Every part of the definition matters. The weekly frequency keeps contact from becoming an event, having the team itself do it avoids depending on third-party reports, and the outcome gives direction to what is researched.

It is sustained by the product trio, made up of whoever leads the product, whoever designs, and whoever develops. All three take part in the interviews and in the discovery decisions, because each of them sees different things in the same conversation and because what is learned firsthand is applied better than what is read in a summary. The trio does not exclude the rest of the team or the stakeholders either; it is the core that guarantees continuity.

The practice is canonical in the field and is still a minority one. In ProductPlan's survey from late 2025, published as *State of Product Management 2026* with responses from about 250 people, teams described how they gather and use customer knowledge as follows.

- 34% gather it regularly and use it to prioritize.
- 18.9% gather it but struggle to turn it into decisions.
- 19.3% rely mostly on ad hoc requests or customer escalations.
- 10.2% barely do any research and rely on internal perspectives.
- 5.3% use AI tools to synthesize or interpret customer feedback.
- 12.3% have no consistent process at all.

14.2 Sustaining the cadence

The difficulty of continuous discovery lies in sustaining it when urgent work arrives, and three practical decisions make it viable.

Who interviews

The trio interviews, ideally in full and at least two of its members, so that the interpretation does not depend on a single person. Rotating who leads the conversation and who takes notes spreads the skill and keeps the interview from becoming the job of a specialist whom the rest do not accompany.

How recruiting happens every week

The bottleneck is usually recruiting. Torres recommends automating it so that there are conversations on the calendar every week without anyone having to organize them from scratch, for example with an invitation inside the product itself or with a standing arrangement with the team that serves customers.

Automating recruiting does not exempt you from caring about who you listen to. The segment criteria, the screener, and the caution about your own network are those of chapter 3, and they are worth reviewing from time to time so that the weekly calendar does not always fill up with the same kind of customer, the one most willing to talk.

How time is protected

A fixed slot in the team's calendar, with two or three conversations a week, holds up better than the intention to find a gap. What is protected is continuity, and one interview every week over a quarter leaves the team closer to its customers than ten interviews packed into one month and followed by silence.

14.3 Dual-track: discovering and delivering at the same time

The field called dual-track the idea that discovery and delivery run in parallel: while the team delivers what it has already validated, it keeps researching the next opportunities. The framing is useful as long as it is not read as two teams or as a handoff in which one track produces specifications that the other executes, because then the discovery phase reappears under another name.

In a team that works with Scrum, continuous discovery does not need an event of its own. Conversations with customers feed Product Backlog refinement, what is learned informs Sprint Planning, and the Sprint Review is a natural occasion to check what has been discovered with the stakeholders.

The Scrum Guide does not prescribe any discovery practice, so the fit is a team decision that is worth making explicit. What matters is that discovery is part of the ordinary work of the sprint, with its time set aside, and that it does not become a voluntary activity that is sacrificed as soon as delivery gets tight.

14.4 Preserving the evidence

What is learned in an interview is lost if it only lives in the memory of whoever was there. A research repository, accessible to the whole team, keeps the interview snapshots (chapter 4), the quotes tagged by opportunity, and the results of the experiments, so that any decision can be traced back to what supports it.

The need grows with research democratization. According to Maze's 2026 report, 39% of respondents say that product managers in their organization already do research with users, but only 49% of organizations maintain a research repository. When more people do research and nobody keeps what is learned, each team repeats interviews that another team already did, and the conclusions depend on who remembers them.

Cadence and memory. Continuous discovery needs both things: frequent contact with customers and a place where what is learned accumulates. Without the first, the team decides with old information; without the second, every week starts from scratch.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the components of Teresa Torres's definition of continuous discovery (weekly frequency, the team that builds, small activities, and an outcome) and the role of each one.
- Organize the product trio's participation in interviews, with rotating roles for leading the conversation and taking notes.
- Design automated recruiting that keeps weekly conversations going without giving up the segment criteria.
- Integrate what is learned into Product Backlog refinement, Sprint Planning, and the Sprint Review without creating a handoff between discovering and delivering.
- Set up a research repository that makes it possible to trace a decision back to the interview snapshots and experiments that support it.

Common mistakes and antipatterns

- **Discovering in campaigns.** An intense round of interviews every six months leaves the team deciding with expired information. Protect a weekly slot.
- **Delegating the interviews.** If only a specialist interviews, the rest of the trio reads summaries and learns less. Have at least two of its members take part.
- **Turning dual-track into a handoff.** A track that hands specifications to the other reproduces the discovery phase under another name. The same people discover and deliver.
- **Learning without keeping.** Without a repository, each team repeats interviews and conclusions depend on memory. Keep the snapshots and the results where everyone can consult them.

Further reading

- [Teresa Torres — Continuous Discovery Habits](#)
- [Scrum Manager BoK — Dual-track discovery & delivery \(in Spanish\)](#)
- [Skill Arena — Product Owner](#)
- [Maze — Future of User Research Report 2026](#)

BLOCK E · DISCOVERY AS A TEAM HABIT

Chapter 15. The opportunity solution tree: from outcome to experiment · CONSOLIDATING

The opportunity solution tree connects the outcome the team is pursuing with what it learns from its customers and with the tests it decides to run.

Introduction

A team that interviews every week soon accumulates more opportunities than it can address, and every conversation seems to suggest a new feature. Without a structure that organizes what has been learned, continuous discovery degenerates into a list of requests and the team goes back to working like a feature factory.

In this chapter we will look at Teresa Torres's opportunity solution tree, its four levels, the decisions it forces, how building it has changed with AI, and where its territory ends in relation to the neighboring skills. It is the tool that joins what we saw in block A, talking to customers, with what we saw in block C, experimenting.

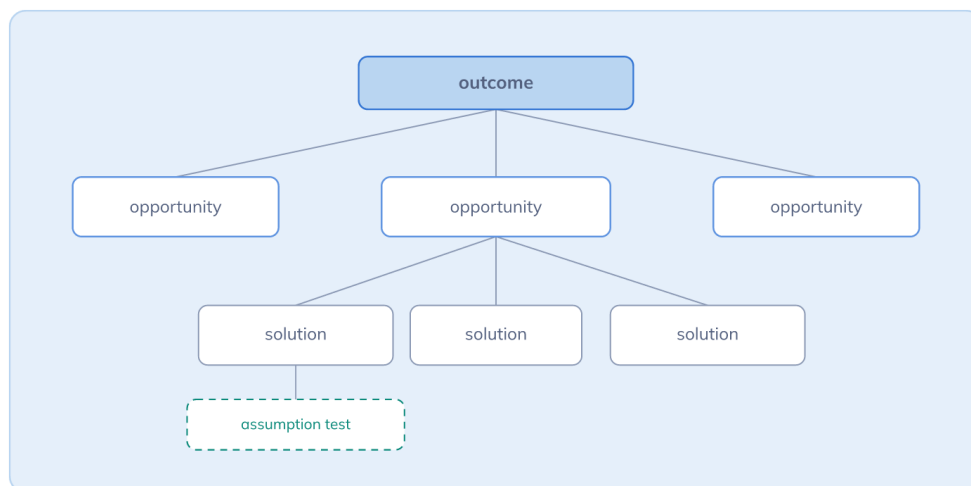


Figure 15. The opportunity solution tree: the outcome breaks down into opportunities, each opportunity into solutions, and each solution into tests of its assumptions.

15.1 The four levels

The opportunity solution tree organizes discovery from top to bottom in four levels.

- **The outcome.** At the root is the outcome the team is pursuing, expressed as a measurable change in customer behavior or in the business. It usually comes from the organization's objectives and key results.
- **The opportunities.** Below it go the customer needs, pain points, and desires that, if addressed, would move the outcome. They come from the interviews and are expressed from the customer's point of view, never as features.
- **The solutions.** For each chosen opportunity, several possible solutions are proposed that compete with one another.

- **The tests.** For each solution, the assumptions it depends on are tested, instead of building it in full to see whether it works.

Opportunities can be broken down into smaller ones, and that hierarchy is part of the value of the tree. A large, vague opportunity (“they find it hard to coordinate”) is divided until it reaches one concrete enough to ideate on (“they don't know what has changed since the last meeting”), and at that level the solutions start to become comparable.

The root deserves care. Torres distinguishes business outcomes, such as revenue or retention, from product outcomes, which the team can move directly with its work, such as more customers completing their initial setup without help. A product outcome organizes the tree better, because it is close enough to what the team decides every week for the opportunities to connect with it.

15.2 The decisions it forces

The tree is useful above all for the decisions it imposes and for the order in which it imposes them.

- **Choose an opportunity before ideating.** Comparing opportunities with one another, rather than comparing ideas, keeps the team's favorite solution from deciding through the back door which problem gets tackled.
- **Compare several solutions.** Torres recommends working with several solutions for the chosen opportunity, because with only one the team tends to confirm it rather than put it to the test.
- **Test assumptions instead of building.** Each solution is broken down into the assumptions it depends on (chapter 8), and the riskiest ones are tested first, at minimum cost.

The tree also makes it visible when to switch branches. If the tests of every solution for an opportunity fail, the problem may lie in the chosen opportunity rather than in the solutions, and the tree makes it possible to go back up a level without losing what has been learned.

15.3 The tree with AI

In 2026, tools have appeared that generate drafts of the tree from interviews. In February, Teresa Torres presented one that turns three real interviews into a first tree in minutes, first synthesizing each interview separately and then looking for patterns across them.

The same author warns against the opposite extreme, one-click trees generated by describing a market and filled with invented data. Her argument is that the point of the tree is to synthesize what is learned from real customers, so a tree with no interviews behind it is a list of ideas shaped like a tree. The AI-generated draft serves as a starting point that the trio reviews with its experience and its context, applying the synthesis rules of chapter 16.

Reviewing the draft follows a simple method. Every opportunity in the tree must be able to point to at least one interview it comes from, and those that cannot point to any are hypotheses of the team or inventions of the model, which are marked as such or removed before choosing where to go next.

The tree is alive. A tree is updated every week with what the interviews and experiments bring. If it has not changed in a month, the team has most likely stopped discovering.

15.4 Its place among the neighboring skills

The tree touches several disciplines, and it is worth knowing what belongs to each one. The outcome at the root is defined with objectives and key results, whose design has its own skill in Skill Arena, and the User stories skill uses the tree to decide where to intervene before writing stories. What comes out of the validated tests then feeds the Product Backlog.

Here we deal with the central stretch, the one that belongs to discovery: how opportunities are built from interviews, how to choose among them, and how the solutions competing for each one are ideated and tested.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Build an opportunity solution tree from real interviews, with an outcome at the root and opportunities expressed from the customer's point of view.
- Break a vague opportunity down into more concrete opportunities that can be ideated on.
- Choose an opportunity by comparing it with others before generating solutions, and justify the choice.
- Ideate several solutions for the chosen opportunity and break each one down into the assumptions that need to be tested.
- Review an AI-generated draft of the tree against the interviews it comes from, detecting opportunities with no support.

Common mistakes and antipatterns

- **Opportunities shaped like features.** “They need a dashboard” is a solution in disguise. Rephrase it from what the customer is trying to achieve.
- **A single solution per opportunity.** With no alternatives, the team confirms its favorite idea. Propose several and compare them.
- **A tree with no interviews behind it.** A tree generated by describing a market is a list of ideas shaped like a tree. Build it from what real customers said.
- **A frozen tree.** If it does not change for weeks, the team has stopped discovering. Update it with every interview and every test.

Further reading

- [Teresa Torres — Opportunity solution trees](#)
- [Teresa Torres — AI opportunity solution trees](#)
- [Skill Arena — OKRs and product strategy](#)
- [Skill Arena — User stories](#)

BLOCK F · DISCOVERING WITH ARTIFICIAL INTELLIGENCE

Chapter 16. Interviewing and synthesizing with AI: where it helps and where it erodes · CONSOLIDATING

AI speeds up the mechanical part of discovery, which is transcribing, summarizing, and grouping, and erodes the judgment part when that is delegated to it; using it sensibly means comparing its work with your own.

Introduction

In a very short time, artificial intelligence has gone from being a curiosity to being part of the daily work of anyone who researches with customers. It transcribes interviews, summarizes what was said, groups themes across conversations, and proposes draft syntheses, and there are already platforms that run the entire interview with nobody on the other side. The professional question is no longer whether to use it but where to trust it.

In this chapter we will look at what the evidence says about its adoption, how to use it in synthesis without losing what synthesis contributes, when it makes sense for it to moderate an interview, and what care it adds to the handling of data. We start from what we saw in blocks A and B, because AI changes the toolkit and leaves the two underlying disciplines intact: the interview that learns and the synthesis that reaches an insight.

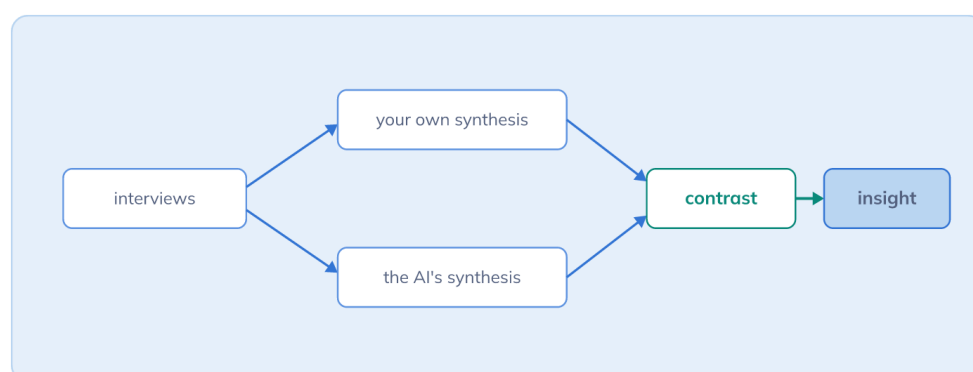


Figure 16. Compare, do not delegate: your own synthesis and the AI's are done separately and contrasted before asserting an insight.

16.1 What the evidence says about its adoption

Adoption is measured by more than one source. Maze's *Future of User Research Report 2026*, published on March 11, 2026, based on a survey of nearly five hundred research, design, and product professionals, finds that 69% use AI in at least some of their studies, 19% more than the previous year. The same report records a clear limit: 82% maintain that interpreting nuance and emotion is still human work, and 80% say the same about ethical decisions.

ProductPlan's *State of Product Management 2026* survey, conducted in late 2025 with about two hundred and fifty people, shows what it is used for. Those who use it in discovery do so mainly in the early tasks of the process:

- summarizing customer feedback (47.1%);
- drafting research summaries or briefs (40.2%);
- turning raw material into structured findings (40.2%);
- identifying themes or patterns (39.8%).

By contrast, only 14.3% use it to prioritize customer problems, and 27.9% do not use it in discovery at all. The picture that emerges is that of a tool that helps process and organize information, while the decision about which problem matters stays with the team. It is a sensible division of labor, and the rest of the chapter explains why it is worth keeping.

16.2 Synthesizing with AI without delegating judgment to it

The main risk has a name. Teresa Torres calls lazy AI interview synthesis the practice of dumping transcripts into an assistant and accepting its summary without reviewing it. The summary usually sounds convincing, because AI writes well, and that is precisely why it is easy to take it for an insight when it is often a well-written observation or a pattern without a cause.

The alternative Torres proposes is organized in three steps, and the order is part of the method:

1. synthesize first, from your own notes and transcripts, without looking at what the tool has produced;
2. then ask the AI for the same synthesis, on the same material;
3. compare the two and examine the differences, both what the AI saw and the team did not and what the team saw and the AI missed.

The comparison is where the value lies. Each method catches things the other misses, and every discrepancy forces a return to the sources, which is exactly the habit that lazy synthesis eliminates. The order matters because whoever reads the tool's summary first tends to anchor on it and to find in the notes what they already expect to find.

The transcription also interprets. The summary a transcription tool generates already selects and rephrases what was said. It is worth treating it as one more interpretation, with the same marking as your own inferences in chapter 4, and checking it against the literal transcript before citing it as evidence.

The group synthesis ritual, with its room dynamics, has its own skill in Scrum Manager, Design Thinking. What matters here is something else: using AI as a second reading of the material, which speeds up the work without taking away from the team the responsibility of deciding what it has learned.

16.3 When an AI can interview

AI-moderated interview platforms run the conversation without a human interviewer. They ask the questions, follow up according to the answers, and deliver the transcript with a summary. The Nielsen Norman Group put them to the test in a study that Maria Rosala published on January 30, 2026, with ten participants from eight countries and two platforms, Marvin and UserFlix, and its conclusions clearly separate the uses.

Where it fits

- gathering opinions about a product, especially after launch;
- screening interviews to select participants;
- teams without dedicated research staff that need a first signal;
- studies in several languages without interpreters.

Where it does not fit

- topics that require deep domain knowledge;
- exploring an unknown problem space, which is precisely the discovery interview of chapter 2;
- research in which each interview must take a different path;
- studies that require screen sharing or watching the person perform a task.

The participants' experience also counts. In the study, only three out of ten agreed that the conversation had felt natural, and five out of ten said they felt comfortable. Some felt listened to; others described the situation as artificial or felt they were losing control. For an interview whose value depends on the person confidently telling a concrete story, that finding carries weight.

When an AI interviews, obligations are added. Since August 2, 2026, Article 50 of the EU AI Act requires systems that interact with people to inform them that they are an AI, and if the tool recognizes emotions, whoever uses it must inform the people exposed to it. Chapter 5 develops these obligations and those relating to recording and transcription, which apply in the same way when the interview is run by a machine.

16.4 Data and provenance

Using an AI assistant on interview material first raises a data question. Raw notes and transcripts contain names, job titles, companies, and details that identify people, and pasting them into a tool without checking what it does with them is processing that nobody has authorized. The practical guideline, consistent with the one taught in the User stories skill, is to anonymize before analyzing and to use tools whose data handling has been checked and is covered by the consent obtained.

The other question is provenance. A finding that comes out of synthesis must be traceable to the quotes and facts that support it, and that applies equally if the AI proposed it. If the team cannot point to the interviews in which a pattern appears, it cannot claim it yet, however well written the summary that states it may be.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Place a specific use of AI in discovery among those the evidence supports (transcribing, summarizing, structuring, grouping themes) and those it still reserves for human judgment (interpreting nuance, prioritizing problems, deciding).
- Apply the compared-synthesis flow, which starts with your own synthesis, continues with the AI's synthesis of the same material, and ends with an analysis of the differences.
- Decide whether an interview can be moderated by an AI according to its purpose, distinguishing product feedback and screening from the exploration of an unknown problem.
- Treat a transcription tool's summary as an interpretation and check it against the literal transcript before using it as evidence.
- Prepare interview material for AI analysis by anonymizing identifiable data and checking how the tool handles it.

Common mistakes and antipatterns

- **Lazy AI interview synthesis.** Dumping transcripts into an assistant and accepting its summary turns a well-written observation into a false insight. Synthesize first and compare afterward.
- **Reading the tool's summary first.** The later reading of the notes stays anchored in what the AI proposed. The order of the comparison is part of the method.
- **Using an automated interviewer to explore.** AI moderation does not pursue the unexpected or change course according to the story it hears, which is what the discovery interview requires. Reserve it for product opinions, screening, or multilingual studies.
- **Pasting raw notes into an assistant.** Names, job titles, and identifiable details end up in a tool whose data handling nobody has checked. Anonymize before analyzing.

Further reading

- [Nielsen Norman Group — AI-moderated interviews](#)
- [Teresa Torres on airfocus — AI product management](#)
- [Maze — Future of User Research Report 2026](#)
- [ProductPlan — State of Product Management 2026](#)

BLOCK F · DISCOVERING WITH ARTIFICIAL INTELLIGENCE

Chapter 17. Where synthetic users sit on the scale of evidence

· EMERGING

Synthetic users are useful for preparing work with real customers, and their answers sit on the lowest rung of the scale of evidence, below any human opinion.

Introduction

Among the AI tools offered for discovery, none promises as much as synthetic users, profiles generated by a language model that answer the way a customer segment would. The promise is obvious: research in minutes, with no recruiting or waiting and as many answers as you want. So is the temptation, because the answers arrive well written, coherent, and free of contradictions.

In this chapter we will look at what they are and why they are convincing, what the evidence accumulated up to 2026 says about their validity, where they sit on the scale of the strength of evidence from chapter 12, and what legitimate use is left for them. We state the conclusion up front, because it organizes the whole journey: they are useful for preparing work with real customers and become dangerous when they replace it.

17.1 What they are and why they are convincing

A synthetic user is a profile that a language model plays on the basis of a description (age, profession, context, needs) and that answers the questions of an interview or a survey the way that person would. Some tools generate dozens of profiles at once and aggregate their answers into reports that look very much like those of real research.

They convince for the same reasons that make them dangerous. They always answer, they do so fluently, and they rarely contradict themselves, whereas real people hesitate, change their minds, go off on tangents, and sometimes cannot explain what they do. That roughness of the human is precisely the information discovery is looking for, and the synthetic profile removes it.

17.2 What the evidence says

The accumulated evidence is negative and increasingly precise. The Nielsen Norman Group already warned in 2024, in an article by Maria Rosala and Kate Moran, of two underlying problems. Synthetic users are sycophantic, because they tend to please and describe ideal behavior, such as finishing every course they start when real people drop out, and they are shallow, because they list many needs without distinguishing which ones really matter.

In August 2025, Raluca Budiu, from the same organization, reviewed the studies that compare simulated and real behavior and reached a conclusion that sums up the problem well: synthetic users capture the general trends of human behavior but fail on the magnitude of the effects and on their variability. And variability is what reveals the subgroups, the extreme cases, and the polarized opinions that change a design decision.

2026 has brought more direct academic evidence, with two studies published on arXiv in July:

- Zihan Chen, Di Zhu, and Lei Nico Zheng compared simulated and human answers in surveys on social attitudes and on values across several cultures. No model outperformed the baseline methods that do not use language models, all of them treated demographic traits as far more predictive than they are among real people, and, in segmentation tasks, they inflated the differences between segments by two to four times and led to choosing the wrong segment in half of the U.S. cases.
- Mantas Lukauskas and Viktorija Šarkauskaitė audited 37 models against a survey answered by 263 employees. They measured an acquiescence bias, the tendency to agree, of 0.84 standard deviations, and found that the models' answers resembled one another more than they resembled the human ones. They conclude that samples generated by models cannot simply replace human data.

Practitioners' opinion points in the same direction. In User Interviews' *State of Synthetic Users* survey, from May 2026, with 150 responses, 47% described themselves as skeptical, 17.3% were opposed to their use, 79% feared that stakeholders would place too much trust in AI-generated findings, and 62.7% worked in organizations with no guidance at all on them.

17.3 Their place on the scale of evidence

In chapter 12 we ordered evidence by how closely it resembles a real decision with a cost for whoever makes it. A synthetic answer has no cost for anyone, and it does not come from anyone either, because it is a model's prediction of what a person would say. That is why it sits on the lowest rung of the scale, below even the stated opinion of a real customer, which at least belongs to someone.

A simple check. Faced with a finding, it is worth asking where it comes from. If it comes from a model, it can guide the next research question, but it still says nothing about customers.

This position does not depend on models getting better. Even if a model got the direction of a behavior right, the product decision needs to know how much, in whom, and with what variation, and that is what the evidence from 2025 and 2026 shows that simulated profiles do not reproduce. It is also the reason for their emerging status: the tool is being pushed insistently, the practice is contested, and its permanence is not guaranteed.

17.4 What you can do with them

Ruling them out as evidence does not mean ruling them out entirely. The same sources that criticize them acknowledge preparatory uses, in which their answer is never taken as data about customers:

- desk research to get familiar with a new domain before talking to anyone;
- generating hypotheses that are then checked with real people;

- rehearsing an interview guide to detect questions that are confusing, leading, or invite opinions about the future;
- building draft profiles that real research will confirm or correct.

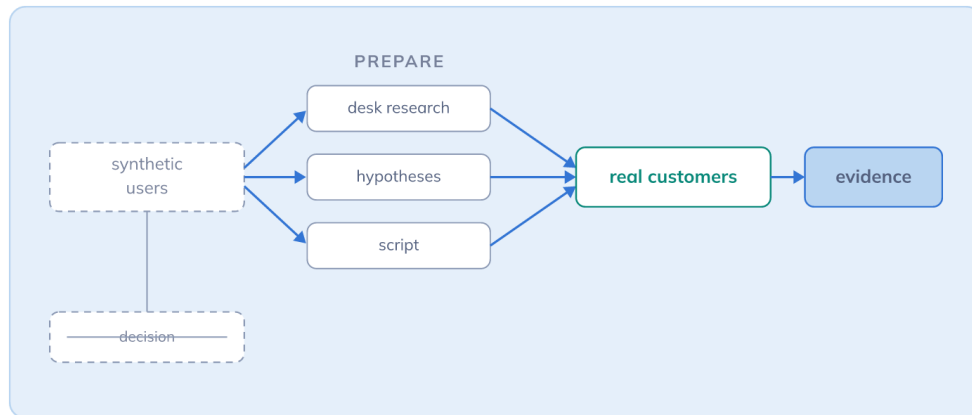


Figure 17. Prepare, don't replace: synthetic users help prepare the work with real customers, and the evidence comes only from them.

The rule that ties all these uses together is one of direction. The synthetic can push toward real people, opening questions worth asking, but it can never take their place in a decision. When a synthetic report reaches a prioritization meeting as if it were research, the problem is no longer technical: it is one of honesty about where the evidence comes from.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain what a synthetic user is and why its answers are convincing even though they do not come from any customer.
- Identify in a synthetic finding the flaws the evidence documents, such as sycophancy, shallowness, excessive weight given to demographics, and lack of variability.
- Place a synthetic answer on the scale of the strength of evidence and justify why it sits on the lowest rung.
- Distinguish preparatory uses (desk research, hypotheses, rehearsing interview guides, draft profiles) from their improper use as evidence in a decision.
- Flag the synthetic origin of a finding when it is presented in a product decision.

Common mistakes and antipatterns

- **Taking the synthetic for research.** A report with many well-written answers looks like a study, but it does not capture any customer's opinion. Call it by its name and use it to prepare.
- **Segmenting with simulated profiles.** Models exaggerate the differences between segments and can lead to choosing the wrong one. Segmentation is validated with real people.

- **Seeking confirmation in the synthetic.** A sycophantic model tends to agree with whoever asks, so it confirms almost any idea. It is confirmation bias in another disguise.
- **Hiding the provenance.** Mixing synthetic and real findings in the same report without distinguishing them leads stakeholders to trust them too much. Always state where each finding comes from.

Further reading

- [Nielsen Norman Group — Synthetic users](#)
- [Nielsen Norman Group — Evaluating AI-simulated behavior](#)
- [Chen, Zhu, and Zheng — When synthetic users fail \(arXiv, 2026\)](#)
- [Lukauskas and Šarkauskaitė — Plausible but not valid \(arXiv, 2026\)](#)

BLOCK F · DISCOVERING WITH ARTIFICIAL INTELLIGENCE

Chapter 18. Validating a product whose core is a model ·

EMERGING

When value depends on an AI model, discovery includes checking early whether that value arrives with acceptable reliability, and error analysis with real customers becomes a discovery activity.

Introduction

So far we have treated AI as a discovery tool. This chapter deals with the reverse case: products whose value depends on an AI model, such as an assistant that drafts text, a system that classifies, or an agent that acts on the user's behalf. For them, on top of the usual questions of whether the customer wants it, whether it is viable as a business, and whether we can build it, there is another that changes the order of the work: whether the system gets it right often enough, and fails acceptably, for the value to actually arrive.

We will see why that question turns feasibility into an assumption tested early, how error analysis works as a discovery activity, and where this chapter ends and the skills that deal with operating these systems begin. It is emerging ground, with practice still taking shape, and it is worth covering with due caution.

18.1 Reliability as part of the value

In a conventional product, a feature does the same thing every time it is used, and the value question can be separated from the technical one. In a model-based product, the same input can produce different outputs, and some of them will be wrong. The value the customer perceives then depends on the proportion and type of errors they tolerate in their context: a draft with the occasional mistake can save them time, while a wrong classification in a critical process can destroy their trust.

That is why the discovery question widens. It is no longer enough to know whether the customer wants the result; you need to know with what reliability they need it, which errors they find tolerable, and which make them abandon the product. Those answers only come from watching the system in the hands of real customers, with their data and in their situations.

18.2 Feasibility as an assumption tested early

In chapter 8 we saw that starting with desirability is the usual order in assumptions mapping, and that what rules is the quadrant. This is the case in which the quadrant changes the order. If the model's quality in the customer's domain is uncertain and all the value depends on it, feasibility is an important assumption with no evidence, and it is worth testing it at the start and with the real problem.

Testing it at the start does not require building the product. A prototype that connects the model with a handful of the customer's real cases, or a version in which a person reviews every output before it is delivered, makes it possible to see early how often the system gets it right and what kind of failures it makes. Chapter 10 describes these lightweight alternatives; the difference here is that what is measured includes the quality of the model's answers, as well as the customer's interest.

18.3 Error analysis as a discovery activity

Teresa Torres and Petra Wille put it this way in a conversation published in September 2025. Discovery practices inform every step of evaluating an AI product, and the most productive one is error analysis, which consists of reviewing the system's real outputs in detail, identifying its failure modes, and turning them into evals that make it possible to measure whether the system is improving. Torres has gone as far as calling evals a new discovery habit.

The parallel with the synthesis in chapter 6 is direct. Just as an insight rests on patterns that repeat across interviews, a failure mode rests on failures that repeat across cases, and the task is the same: group, name, and decide what matters. The most valuable failure modes are those that only appear with real customers, because they depend on their language, on their data, and on what they expect from the system.

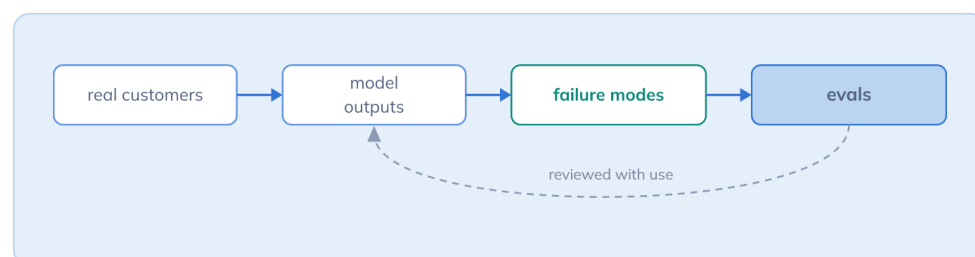


Figure 18. Error analysis as discovery: the failures that appear with real customers are grouped into failure modes, which become evals that are revised with use.

The criteria change. Torres and Wille warn about criteria drift, because what the team considers a good answer changes as it learns from real use. Evals need continuous maintenance, like the rest of the knowledge that discovery produces.

18.4 Where this chapter ends

The boundary with other skills is deliberate. What matters here is one specific stretch: validating that the product delivers value with acceptable reliability before committing the investment. How evals are designed, automated, and operated before and after launch is covered by Scrum Manager's Product Owner certification and the Harness Engineering skill, which are worth consulting when the product moves from validation to delivery.

What discovery contributes to that operation is the origin of the criteria. Evals that are not born from watching real customers measure what the team imagined, and they reproduce in the evaluation the same mistake that chapter 1 described in the product: mixing the solution one wants with the problem that exists.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Formulate, for a model-based product, the value question that includes reliability, that is, which errors the customer tolerates in their context and which make them abandon the product.
- Recognize when the model's quality is an important assumption with no evidence, and place it on the assumptions map to test it early.
- Design a lightweight test that measures the quality of the model's outputs with the customer's real cases before building the product.
- Apply error analysis to real outputs, grouping failures, naming failure modes, and turning them into evals.
- Distinguish the validation of value, which belongs to discovery, from the operation of evals, which belongs to other disciplines.

Common mistakes and antipatterns

- **Leaving the model's quality for the end.** If all the value depends on it, discovering late that it falls short forces you to throw away the work done. Test it early and with real cases.
- **Evaluating with cases invented by the team.** The criteria reflect what the team imagines and not what the customer needs. Get the cases from real use.
- **Treating evals as a closed list.** The criteria drift as learning happens. Review them at the same cadence as the rest of discovery.

- **Confusing a demo with a validation.** A demo with hand-picked cases shows what the model can do; validation shows what it does with the customer's cases, including the difficult ones.

Further reading

- [Torres and Wille — AI evals & discovery](#)
- [Teresa Torres — Assumption testing](#)
- [Skill Arena — Harness Engineering](#)
- [Skill Arena — Product Owner](#)

© 2026 Scrum Manager®. This work is published under a Creative Commons Attribution-NonCommercial 4.0 International licence (CC BY-NC 4.0). Official Scrum Manager trainers and centres are licensed under the terms of CC BY 4.0 for their training activity.